

# Structural Cheminformatics for Kinase-Centric Drug Design

Inaugural-Dissertation  
to obtain the academic degree  
Doctor rerum naturalium (Dr. rer. nat.)

submitted to the Department of Biology, Chemistry, Pharmacy  
of Freie Universität Berlin

by  
**Dominique Sydow**  
from Berlin, Germany

2022



The presented thesis was prepared from April 2018 until September 2022 under the main supervision of Prof. Dr. Andrea Volkamer at the Institute of Physiology of the Charité — Universitätsmedizin Berlin and further supervision of Prof. Dr. Gerhard Wolber at the Institute of Pharmacy at the Freie Universität Berlin.

1<sup>st</sup> reviewer: Prof. Dr. Andrea Volkamer

2<sup>nd</sup> reviewer: Prof. Dr. Gerhard Wolber

Date of defense: 7 February 2023

*Für Sonja La Grange*

## Acknowledgments

Throughout this journey into science, I was accompanied by many wonderful people whom I'd like to thank in the following.

First, I thank Andrea Volkamer for giving me the chance to explore the world of cheminformatics, structural bioinformatics, and software development in my doctoral studies. Thank you for believing in me and my place in science, and your constant mentor- and sponsorship.

Second, I thank Gerhard Wolber for opening the door for me to computational drug design during my master thesis and allowing me to work on one of my favorite projects (dynophores!). Thank you for your long-standing support, encouragement, and never-ending enthusiasm.

Special thanks to you both for supervising and reviewing my dissertation, and to Prof. Bettina Keller, Prof. Cecilia Clementi, Prof. Daniel Klinger, Dr. Annette Kiezmann, and Dr. Fabian Schumacher for agreeing to join my doctoral committee. I thank Andrea, Ferdinand, Frauke, Talia, and Gerhard for their input on this manuscript.

I would like to express my gratitude to the Volkamer Lab, more specifically, Talia Kimber, Jaime Rodríguez-Guerra, Andrea Morger, David Schaller, Corey Taylor, Yonghui Chen, and Andrea Volkamer. Thank you all for creating a friendly, supportive, and motivating atmosphere. Talia Kimber, my work wife, you inspire me every day with your positive, determined, and considerate actions. Thank you for sharing all the joy and frustration (meeTEAing calls!) over the years. Andrea Morger, I still cherish our pre-pandemic times in the office with a drawer full of food and tea. Special thanks for your support before my first conference talk. Jaime Rodríguez-Guerra, thank you for talking Python with me, I loved every second of designing APIs and reviewing code. Over the years I was lucky to share the road with amazing at-the-time master's and bachelor's students: Paula Schmiel, Eva Aßmann, Michele Wichmann, Praktik Dhakal, and Sonja Leo, it was a great pleasure working with you. Finally, conducting most of my doctoral studies working from home, I thank my fluffy canine colleague for his excellent company.

I would like to add the following special thanks: Thomas Hansen, I think it is safe to say this scientific journey started with you when you showed that protein structure to my 16-year-old self in biology class. Albert Kooistra, I enjoyed every moment of talking about kinases and KLIFS with you, thank you. Greg Landrum and the RDKit community, this welcoming and friendly space helped me to get into cheminformatics, thank you. John Chodera, thank you for bringing software best practices into my work. Christin Rakers, Marcel Bermudez, and Jérémie Mortier, I met you at the beginning of this journey and am still guided by what you taught me, thank you. Noel O'Boyle, thank you for your support and guidance throughout this year.

I am grateful for the support and patience of my friends who have shared this path (or parts of it) with me: Frauke, Talia, Rebecca, Pierre, Claudia, Fiona, Yvonne, Sandra, and Maryam — I admire you all deeply. Frauke Degenhardt, thank you for being with me at my best and worst as well as for listening to me with an open mind and giving great advice when needed. Rebecca Hunt, thank you so much for our amazing walks and precious conversations in Berlin's wilderness and beyond, always finding peace and beauty after a chaotic week.

Ich danke euch, meiner Mutter, meinen Großeltern, Jasmin, Susi, Andreas und dem Rest der Familie, herzlich für eure Unterstützung über all die Jahre, mit allen Höhen und Tiefen. Uta und Bernd, ich schätze mich glücklich, dass ihr in mein Leben treten seid; ich danke euch für euren Rat und eure Zuversicht. Sonja La Grange — meine Liebe, du bist schon lange nicht mehr hier. Ich denke an dich und widme dir diese Arbeit.

Finally, the person behind the illustrations and the emotional support for this thesis, who is always open to new adventures and still amazes me after all these years: Ferdinand Krupp, thank you for the beautiful life we have built.



Hierdurch versichere ich, dass ich meine Dissertation selbstständig verfasst und keine anderen als die von mir angegebenen Quellen und Hilfsmittel verwendet habe.

Geistiges Eigentum anderer Autoren wurde als entsprechend gekennzeichnet. Ebenso versichere ich, dass ich an keiner anderen Stelle ein Prüfungsverfahren beantragt bzw. die Dissertation in dieser oder anderer Form an keiner anderen Fakultät als Dissertation vorgelegt habe.





# Contents

<b>Abstract</b>	<b>1</b>
<b>Zusammenfassung</b>	<b>3</b>
<b>1 Introduction</b>	<b>5</b>
1.1 Protein Binding Sites . . . . .	5
1.2 Protein Binding Sites in Drug Discovery . . . . .	7
1.2.1 Advances and Challenges in Computational Target Prediction <b>Publication A</b> . . . . .	8
1.2.2 Advances and Challenges in Computational Fragment-Based Drug Design	26
1.3 Protein Kinases . . . . .	27
1.3.1 Protein Kinases as Drug Targets and Challenges . . . . .	27
1.3.2 Classification of Human Protein Kinome . . . . .	28
1.3.3 Kinase Structure . . . . .	28
1.3.4 Kinase Inhibitors . . . . .	30
1.3.5 KLIFS — a Structure-Focused Kinase Data Resource . . . . .	34
1.3.6 Kinase Bioactivity and Profiling Resources . . . . .	35
1.4 Open Science . . . . .	36
<b>2 Aim and Objectives</b>	<b>39</b>
<b>3 Methods and Results</b>	<b>41</b>
3.1 Predicting Kinome-Wide (Sub)Pocket-Based Off-Targets . . . . .	43
3.1.1 KiSSim: Predicting Off-Targets from Structural Similarities in the Ki- nome <b>Publication B</b> . . . . .	44
3.1.2 Analyzing Kinase Similarity in Small Molecule and Protein Structural Space to Explore the Limits of Multi-Target Screening <b>Publication C</b> . . . . .	90
3.1.3 Kinase Similarity Assessment Pipeline for Off-Target Prediction <b>Publication D</b> . . . . .	120
3.2 Exploring Kinome-Wide Subpocket Fragment Spaces . . . . .	133
3.2.1 KinFragLib: Exploring the Kinase Inhibitor Space Using Subpocket-Focused Fragmentation and Recombination <b>Publication E</b> . . . . .	134
3.3 FAIR Pipelines and Tools in Kinase-Centric Drug Design . . . . .	169
3.3.1 TeachOpenCADD: A Teaching Platform for Computer-Aided Drug Design Using Open Source Packages and Data <b>Publication F</b> . . . . .	170

3.3.2	TeachOpenCADD 2022: Open Source and FAIR Python Pipelines to Assist in Structural Bioinformatics and Cheminformatics Research <b>Publication G</b> . . . . .	178
3.3.3	OpenCADD-KLIFS: A Python Package to Fetch Kinase Data from the KLIFS Database <b>Publication H</b> . . . . .	188
<b>4</b>	<b>Discussion</b>	<b>195</b>
4.1	Computational Target Prediction . . . . .	195
4.1.1	Remaining Challenges of State-of-the-Art Approaches . . . . .	195
4.2	Predicting Kinome-Wide (Sub)Pocket-Based Off-Targets . . . . .	196
4.2.1	KiSSim: Enabling Kinase-Specific Encoding and Comparison . . . . .	197
4.2.2	KiSSim: Detecting Expected and Unexpected Kinase Relationships . . . . .	197
4.2.3	Assessing Kinase Similarity from Different Perspectives . . . . .	198
4.2.4	Integrating Kinase Similarity Measures as an Automated Pipeline . . . . .	199
4.2.5	Generalizing Pocket Comparison Concepts from the Kinome to Proteome . . . . .	200
4.3	Exploring Kinome-Wide Subpocket Fragment Spaces . . . . .	201
4.3.1	KinFragLib: Fragmenting Kinase Inhibitors to Explore Subpockets . . . . .	201
4.3.2	KinFragLib: Recombining Fragments for Novel Kinase Inhibitors . . . . .	202
4.3.3	Addressing Limitations of the KinFragLib Approach . . . . .	202
4.3.4	Addressing Future Applications of the KinFragLib Approach . . . . .	203
4.4	FAIR Pipelines and Tools in Kinase-Centric Drug Design . . . . .	203
4.4.1	TeachOpenCADD: Distributing a FAIR Platform for CADD Pipelines . . . . .	204
4.4.2	OpenCADD: Building a FAIR Toolkit for Structural Cheminformatics . . . . .	206
4.4.3	Advocating for Software Best Practices and FAIR Research . . . . .	207
<b>5</b>	<b>Conclusion</b>	<b>209</b>
	<b>List of Publications</b>	<b>215</b>
	<b>Appendix</b>	<b>215</b>
5.1	Further Publications . . . . .	216
5.1.1	TeachOpenCADD-KNIME: A Teaching Platform for Computer-Aided Drug Design Using KNIME Workflows <b>Publication I</b> . . . . .	216
5.1.2	Teaching Computer-Aided Drug Design Using TeachOpenCADD <b>Publication J</b> . . . . .	222
5.2	Further Projects . . . . .	247
5.2.1	Ratar: Read-Across the Targetome . . . . .	247
5.3	Project Illustrations . . . . .	254
	<b>Figures</b>	<b>261</b>
	<b>Tables</b>	<b>263</b>
	<b>Acronyms</b>	<b>267</b>
	<b>Bibliography</b>	<b>267</b>

# Abstract

Drug development is a long, expensive, and iterative process with a high failure rate, while patients wait impatiently for treatment. Kinases are one of the main drug targets studied for the last decades to combat cancer, the second leading cause of death worldwide. These efforts resulted in a plethora of structural, chemical, and pharmacological kinase data, which are collected in the KLIFS database. In this thesis, we apply ideas from structural cheminformatics to the rich KLIFS dataset, aiming to provide computational tools that speed up the complex drug discovery process. We focus on methods for target prediction and fragment-based drug design that study characteristics of kinase binding sites (also called pockets).

First, we introduce the concept of computational target prediction, which is vital in the early stages of drug discovery. This approach identifies biological entities such as proteins that may (i) modulate a disease of interest (targets or on-targets) or (ii) cause unwanted side effects due to their similarity to on-targets (off-targets). We focus on the research field of binding site comparison, which lacked a freely available and efficient tool to determine similarities between the highly conserved kinase pockets. We fill this gap with the novel method *KiSSim*, which encodes and compares spatial and physicochemical pocket properties for all kinases (kinome) that are structurally resolved. We study kinase similarities in the form of kinome-wide phylogenetic trees and detect expected and unexpected off-targets. To allow multiple perspectives on kinase similarity, we propose an automated and production-ready pipeline; user-defined kinases can be inspected complementarily based on their pocket sequence and structure (KiSSim), pocket-ligand interactions, and ligand profiles.

Second, we introduce the concept of fragment-based drug design, which is useful to identify and optimize active and promising molecules (hits and leads). This approach identifies low-molecular-weight molecules (fragments) that bind weakly to a target and are then grown into larger high-affinity drug-like molecules. With the novel method *KinFragLib*, we provide a fragment dataset for kinases (fragment library) by viewing kinase inhibitors as combinations of fragments. Kinases have a highly conserved pocket with well-defined regions (subpockets); based on the subpockets that they occupy, we fragment kinase inhibitors in experimentally resolved protein-ligand complexes. The resulting dataset is used to generate novel kinase-focused molecules that are recombinations of the previously fragmented kinase inhibitors while considering their subpockets. The KinFragLib and KiSSim methods are published as freely available Python tools.

Third, we advocate for open and reproducible research that applies FAIR principles — data and software shall be findable, accessible, interoperable, and reusable— and software best practices. In this context, we present the *TeachOpenCADD* platform that contains pipelines for computer-aided drug design. We use open source software and data to demonstrate ligand-based applications from cheminformatics and structure-based applications from structural bioinformatics. To emphasize the importance of FAIR data, we dedicate several topics to accessing life science databases such as ChEMBL, PubChem, PDB, and KLIFS. These pipelines are not only

useful to novices in the field to gain domain-specific skills but can also serve as a starting point to study research questions. Furthermore, we show an example of how to build a stand-alone tool that formalizes reoccurring project-overarching tasks: *OpenCADD-KLIFS* offers a clean and user-friendly Python API to interact with the KLIFS database and fetch different kinase data types. This tool has been used in this thesis and beyond to support kinase-focused projects.

We believe that the FAIR-based methods, tools, and pipelines presented in this thesis (i) are valuable additions to the toolbox for kinase research, (ii) provide relevant material for scientists who seek to learn, teach, or answer questions in the realm of computer-aided drug design, and (iii) contribute to making drug discovery more efficient, reproducible, and reusable.

# Zusammenfassung

Die Entwicklung von Arzneimitteln ist ein langwieriger, teurer und iterativer Prozess mit einer hohen Misserfolgsquote, während Patienten auf eine Behandlung warten. Kinasen sind eines der wichtigsten Angriffsziele für Arzneimittel (Targets), die in den letzten Jahrzehnten untersucht wurden zur Bekämpfung von Krebs, der zweithäufigsten Todesursache weltweit. Diese Bemühungen haben zu einer Fülle von strukturellen, chemischen und pharmakologischen Kinase-Daten geführt, die in der KLIFS-Datenbank zusammengetragen sind. In dieser Arbeit wenden wir Ideen aus der Strukturellen Chemieinformatik auf den reichhaltigen KLIFS-Datensatz an, mit dem Ziel, computergestützte Werkzeuge anzubieten, die den komplexen Prozess der Wirkstoffentdeckung beschleunigen können. Unser Fokus liegt dabei auf Methoden für Target-Vorhersagen und fragmentbasiertem Wirkstoffdesign, die Eigenschaften von Kinase-Bindetaschen erforschen.

Zunächst stellen wir das Konzept der computergestützten Target-Vorhersage vor, welche in den frühen Phasen der Wirkstoffentdeckung unerlässlich ist. Dieser Ansatz identifiziert biologische Entitäten wie z. B. Proteine, die (i) die Zielkrankheit modulieren (Targets oder On-Targets) oder (ii) die unerwünschte Nebenwirkungen verursachen aufgrund ihrer On-Target-Ähnlichkeit (Off-Targets). Wir konzentrieren uns auf den Forschungsbereich der Bindetaschenvergleiche, dem ein frei verfügbares und effizientes Werkzeug fehlte, um die Ähnlichkeit zwischen den hochkonservierten Kinase-Bindetaschen festzustellen. Wir schließen diese Lücke mit der neuen Methode *KiSSim*, die räumliche und physikochemische Eigenschaften der Bindetaschen aller Kinasen (Kinome), die strukturell aufgelöst sind, kodiert und vergleicht. Wir untersuchen Kinase-Ähnlichkeiten in Form von kinomweiten phylogenetischen Bäumen und erkennen zu erwartende und unerwartete Off-Targets. Um Kinase-Ähnlichkeiten aus verschiedenen Blickwinkeln zu betrachten, schlagen wir eine automatisierte und produktionsreife Pipeline vor; benutzerdefinierte Kinasen können komplementär untersucht werden auf der Grundlage ihrer Bindetaschen-Sequenzen und -Strukturen (*KiSSim*), ihrer Interaktionen zwischen Bindetasche und Ligand sowie ihrer Ligandenprofile.

Zweitens stellen wir das Konzept des fragmentbasierten Wirkstoffdesigns vor, welches nützlich ist, um aktive und vielversprechende Moleküle (Hits und Leads) zu identifizieren und optimieren. Dieser Ansatz identifiziert Moleküle mit niedrigem Molekulargewicht (Fragmente), die schwach an ein Target binden und dann erweitert werden zu größeren und stärker bindenden Molekülen. Mit der neuen Methode *KinFragLib* stellen wir einen Fragment-Datensatz für Kinasen (Fragment Library) zur Verfügung, indem wir Kinase-Inhibitoren als Kombinationen von Fragmenten betrachten. Kinasen haben eine hochkonservierte Bindetasche mit gut definierten Regionen (Subpockets); basierend auf den Subpockets, die sie besetzen, fragmentieren wir Kinase-Inhibitoren in experimentell aufgelösten Kinase-Liganden Komplexen. Der sich daraus ergebende Datensatz wird genutzt, um neue Kinase-fokussierte Moleküle zu erstellen, die aus den zuvor fragmentierten Kinase-Inhibitoren rekombiniert werden unter Berücksichtigung ihrer Subpockets. Die *KiSSim* und *KinFragLib* Methoden sind als frei verfügbare Python

Tools veröffentlicht.

Drittens setzen wir uns für offene und reproduzierbare Forschung ein, die FAIR-Prinzipien —Daten und Software sollen auffindbar (findable), zugänglich (accessible), interoperativ (interoperable) und wiederverwendbar (reusable) sein— und Best Practices in der Softwareentwicklung anwendet. In diesem Kontext stellen wir die *TeachOpenCADD*-Plattform vor, die Pipelines für computergestütztes Wirkstoffdesign enthält. Wir verwenden frei zugängliche Software und Datensätze, um ligandenbasierte Anwendungen aus der Chemieinformatik und strukturbasierte Anwendungen aus der strukturellen Bioinformatik zu demonstrieren. Um die Bedeutung von FAIR-Daten zu betonen, widmen wir mehrere Themen dem Zugang zu biowissenschaftlichen Datenbanken wie ChEMBL, PubChem, PDB und KLIFS. Diese Pipelines sind nicht nur nützlich für AnfängerInnen auf dem Gebiet, um domänenspezifische Kenntnisse zu erwerben, sondern können auch als Ausgangspunkt dienen, um Forschungsfragen zu untersuchen. Darüber hinaus zeigen wir ein Beispiel für die Entwicklung eines eigenständigen Tools, das wiederkehrende projektübergreifende Aufgaben formalisiert: *OpenCADD-KLIFS* bietet eine klare und benutzerfreundliche Python-API, um mit der KLIFS-Datenbank zu interagieren und verschiedene Kinase-Datentypen abzurufen. Dieses Tool wurde in dieser Arbeit und über diese Arbeit hinaus verwendet, um Projekte mit Kinase-Fokus zu unterstützen.

Wir denken, dass die in dieser Arbeit vorgestellten FAIR-basierten Methoden, Pipelines und Tools (i) wertvolle Ergänzungen des Werkzeugkastens für die Kinase-Forschung sind, (ii) relevantes Material für WissenschaftlerInnen bieten, die auf dem Gebiet des computergestützten Wirkstoffdesigns lernen, lehren oder Fragen beantworten wollen, und (iii) dazu beitragen, die Wirkstoffforschung effizienter, reproduzierbar und wiederverwendbar zu machen.

# Chapter 1

## Introduction

Protein binding sites are at the heart of every structure-enabled drug design campaign; understanding the physicochemical and steric characteristics of a target's binding site guides the rational design of drug candidates. In drug discovery, the term target refers to a biological entity such as a protein whose modulation by a drug might inhibit or reverse the progression of a disease of interest [1].

This introduction will set the scene for the concepts discussed in this thesis:

- Section 1.1 outlines the nature of protein binding sites and defines the terms "pocketome", "kinome", and "structural cheminformatics".
- Section 1.2 covers the importance of binding sites in the context of drug discovery, especially regarding target prediction and fragment-based drug design (FBDD).
- Section 1.3 introduces kinases as the main protein class covered in this thesis with a special focus on their binding sites.
- Section 1.4 describes the spirit of open science, which was applied to all projects described in this thesis.

### 1.1 Protein Binding Sites

*Proteins* are—from a structural point of view—a three-dimensional (3D) arrangement of an amino acid sequence, a so-called polypeptide chain (primary structure). This chain is folded into  $\alpha$ -helices and  $\beta$ -sheets, which are connected via loops (secondary structure). The organization of these motifs to each other determines the protein's structure (tertiary structure). Multiple folded polypeptide chains can form a complex (quaternary structure) [2]. These complex and dense 3D structures have a surface with irregular hollows, forming so-called clefts, cavities, or pockets, which range from being shallow to deep, solvent-exposed to buried, and small to large.

Most biological processes are mediated by the binding of molecules such as small molecules and peptides (*ligands*) to these pockets (*binding sites*) or by protein-protein interactions. *Molecular recognition* is enabled by shape and driven by physicochemical complementarity. Protein binding sites and ligands interact through a set of weak non-covalent bonds, i.e., directed (polar) hydrogen bonds and undirected hydrophobic, electrostatic, and van der Waals interactions [2]. Proteins are of flexible nature [3, 4]; interacting ligands can induce conformational change at the binding site (induced fit model [5]) or bind selectively to the most suitable conformational state of the protein's conformational ensemble (conformational selection model [6]).

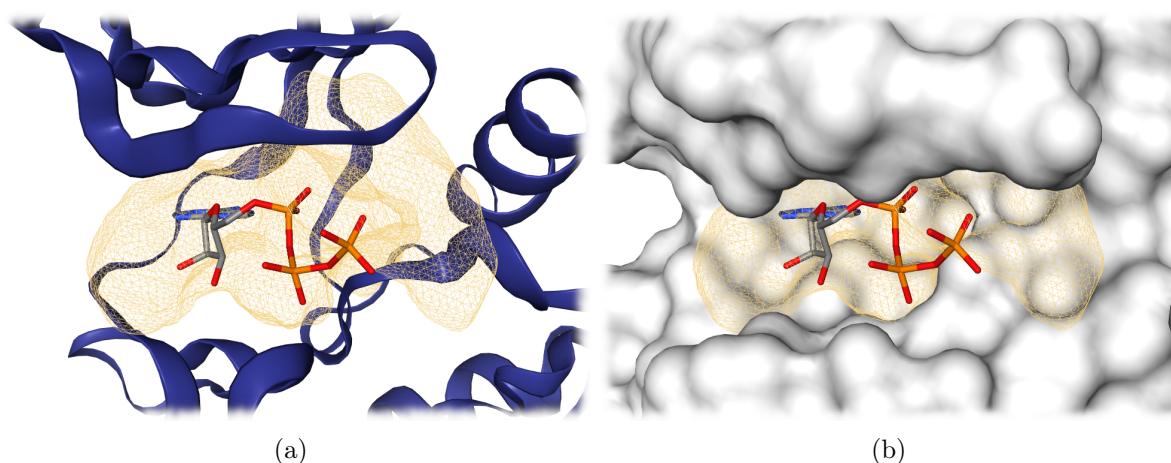


Figure 1.1: Detected binding site of kinase CDK2 overlaps correctly with co-crystallized ligand ATP: Depicted are the protein chain as a blue cartoon (a) and a surface in grey (b), and the co-crystallized ligand ATP as well in (a) and (b). The binding site volume is shown as a yellow mesh, which was predicted with DogSiteScorer on the ProteinsPlus webserver [9–11]. Example kinase is CDK2 (PDB/KLIFS IDs: 1FIN/4367 [21]).

How binding sites are defined computationally depends on whether structures contain a bound ligand or not. With a bound ligand, a common approach is to define binding sites based on distances to the ligand. Protein atoms (or residues) that are within a defined distance from any ligand atom constitute the binding site, e.g., a distance of 6.5 Å in the case of the binding site database sc-PDB [7]. Without a ligand bound to the structure, computational binding site detection methods can be invoked to compute the binding site as discussed in Volkamer et al. [8]. For example, DogSiteScorer is a grid-based geometric method that uses a Difference of Gaussian (DoG) filter from image processing to determine cavities on the protein surface [9–11]; Figure 1.1 shows the DoGSiteScorer-detected binding site of kinase CDK2, which overlaps correctly with the co-crystallized endogenous ligand adenosine triphosphate (ATP).

The ensemble of structurally resolved pockets is referred to as the *pocketome*. The pocketome can be either defined as the ensemble of protein pockets in the PDB [12–14] or as the ensemble of pockets within certain protein classes such as kinases [15] (called *kinome*), G-protein coupled receptors (GPCRs) [16], or E3 ligases [17]. Combining data on the structure of binding sites with other sources such as ligand profiling, sequences, and mutations can shed light on molecular and structural determinants for affinity to and selectivity of a ligand to a protein in different key binding regions. This integrative analysis is referred to as *structural chemogenomics* and can extrapolate information from known to unknown protein-ligand complexes [18, 19]. The methods that incorporate structural, chemical, and pharmacological information from ligands and proteins are assigned to the field of *structural cheminformatics* [20].



## 1.2 Protein Binding Sites in Drug Discovery

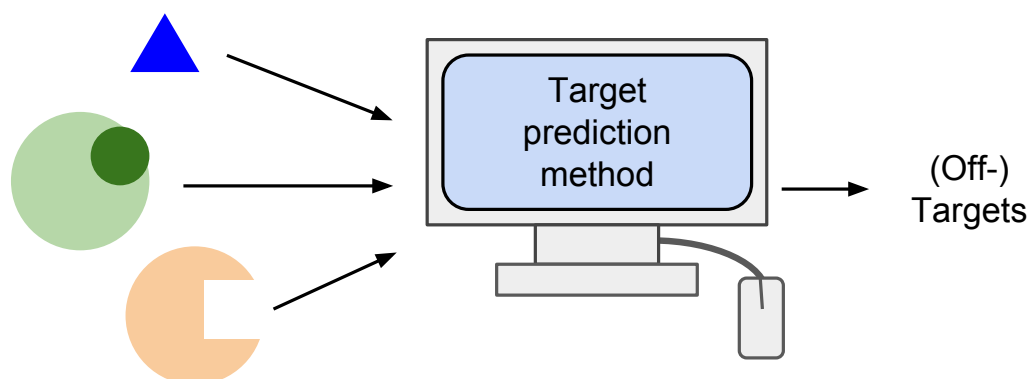
Discovering small drug-like molecules complementing the binding site of a protein of interest, a target, is a central endeavor in drug discovery campaigns in many different settings, including the following examples:

- *Target prediction* helps (i) to detect disease-driving targets, (ii) to identify the targets for a specific ligand known to act on a disease of interest, and (iii) to determine potential off-targets, i.e., unintended targets that a ligand binds to besides the intended so-called on-targets (Section 1.2.1). This step is usually performed during early drug discovery, also called pre-discovery. [22]
- *Binding site identification* involves the determination of electronic, steric, and solvation properties of a target's pocket. This helps to determine (i) the target's druggability, i.e., a target's ability to be modulated by low-molecular-weight compounds, and (ii) the ligand characteristics to aim for. [8]
- *Interaction profiling* helps to identify and characterize key amino acids of the protein that are necessary for ligand binding, e.g., based on known or similar protein-ligand structures, or mutation data. Such knowledge drives the rational design during lead identification and optimization. [23–25]
- *Fragment-based drug design*: The screening of fragment libraries —a set of small chemical molecules typically with a molecular weight less than 300 Da— determines which chemical key characteristics are essential to the binding of molecules to specific regions of the target's binding site. This strategy plays an important role in industry and academia for the discovery of novel compounds and the optimization of lead compounds (Section 1.2.2). [26]

In the following, target prediction and fragment-based drug design will be discussed in more detail from a computational point of view, in preparation for the methods that I will present later in this thesis.

### 1.2.1 Advances and Challenges in Computational Target Prediction Publication A

Computational target prediction plays an important role in early drug discovery phases when a project aims to identify targets of interest and potential off-targets. Especially in these early stages of a project where the target profile is unclear and the progression of the project uncertain, experiments are often not yet established and therefore time- and cost-sensitive. Computational methods can help to study the target of interest from ligand-based, structure-based, and hybrid angles. In this review, in collaboration with Prof. Gerard van Westen's group in Leiden, Netherlands, the method landscape is outlined with a focus on method availability and challenges in the field.



Contribution:

**Co-first author**

Conceptualization (45%)

Visualization (45%)

Writing — Original Draft (45%)

Writing — Review & Editing (45%)

Reprinted with permission from Sydow D\*, Burggraaff L\*, Szengel A, van Vlijmen HWT, IJzerman AP, van Westen GJP, Volkamer A. Advances and Challenges in Computational Target Prediction. *Journal of Chemical Information and Modeling*. 2019; 59(5):1728-1742. 10.1021/acs.jcim.8b00832 (\*contributed equally)

Copyright © 2019 American Chemical Society.

## Summary of Publication A with a focus on binding site comparison

In the fight against diseases, scientists try to identify drugs that act on one or more disease-specific targets (*on-targets*), while avoiding side effects that are caused by acting on so-called *off-targets* (or anti-targets). Target prediction is therefore a crucial step in the early stages of drug development and its outcome may decide whether a target is tractable enough to be pursued. Experiments such as activity-based proteome profiling (ABPP) and standard affinity pulldowns can shed light on on- and off-targets but they can be expensive in terms of time and cost [27]. Computational methods are a fast and cheap alternative to predicting targets and have become a default approach in early drug discovery campaigns.

The applications of computational target prediction are manifold in early drug design projects with the following aims:

- Elucidate the mode of action of a compound by identifying its potential target.
- Explore desired *polypharmacological effects* of ligands to cover disease pathways [28]; the traditional magic bullet paradigm, wherein a ligand has a high potency and selectivity towards a single target, has shifted to the understanding that a ligand affects multiple targets simultaneously [29, 30].
- Spot selectivity or toxicity problems during compound optimization, which can potentially lead to unwanted *adverse* or *side effects* [31].
- Repurpose approved drugs for different indications. Here it is investigated whether they can interact with a protein target that is part of another disease mechanism [32–34]. This process is called *drug repositioning* or *drug repurposing*.
- Select ligands that have the highest potential to be relevant *chemical probes* to characterize the biological function of a poorly understood target [35–37].

Computational target prediction methods can be roughly divided into ligand-based, structure-based, and hybrid methods [38]. The key concept to most of these approaches is the chemical similarity principle which postulates that "similar molecules have a similar biological effect" and that "similar proteins bind similar ligands" [39], respectively. The input to these methods is a ligand (*query ligand*), a protein (*query target*), or a combination of both.

**Ligand-based methods** follow the principle that similar ligands bind similar targets. Methods range from similarity searches identifying targets for a single known compound (e.g., SwissTargetPrediction [40]) to similarity ensembles identifying targets for a group of known compounds (e.g., SEA [41]) but can also involve activity prediction with classification and regression models.

**Hybrid methods** combine ligand and protein information. Proteochemometrics (PCM) uses ligand information alongside protein sequence or structure information, while network-based methods use graphs with proteins and ligands as nodes and interactions, similarities, or phenotypic effects as edges to predict drug-target interaction networks (e.g., DINIES [42]).

**Structure-based methods**, which are the focus of this thesis, follow the principle that similar targets bind similar ligands. Methods can be split into four categories: (i) binding site comparison across different proteins, where a query pocket is compared to a pocket database (e.g., ProBis [43]), (ii) interaction fingerprint comparison, where a query interaction profile is compared to an interaction profile database (e.g., TIFP [44]), (iii) reverse pharmacophore screening, where a query ligand pharmacophore is screened against a pharmacophore-based

interaction database (e.g., PharmMapper [45]), or (iv) inverse screening, where a query ligand is screened against a pocket database via docking (e.g., iRAISE [46]). These methods follow a three-step process:

1. *Binding site encoding*: Binding sites or ligand-target interactions are encoded using different descriptor techniques and stored in a target database.
2. *Target screening or comparison*: Either a query ligand is screened against the target database, or a query binding site is compared with the target database.
3. *Target ranking*: Targets are ranked based on a suitable scoring approach.

This thesis focuses on binding site comparison methods. Publication A [22] reviews developments in the field until 2019; since then, new methods have been published, such as the alignment-based PocketShape [47] and the point cloud registration method ProCare [48] (a concept borrowed from computer vision). Furthermore, advances in machine learning and deep learning have led to the development of novel binding site comparison tools such as Deeply-Tough [49] and DeepDrug3D [50].

In the following, I include a more detailed review of computational target prediction (Publication A [22]), which also outlines remaining challenges. These were addressed in the framework of this thesis with a focus on kinase research as discussed in Sections 4.1 and 4.2

## Advances and Challenges in Computational Target Prediction

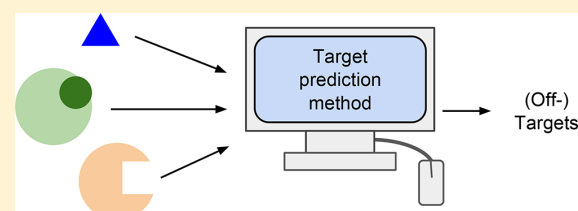
Dominique Sydow,<sup>†,||</sup> Lindsey Burggraaff,<sup>‡,||</sup> Angelika Szengel,<sup>†</sup> Herman W. T. van Vlijmen,<sup>§,‡</sup> Adriaan P. IJzerman,<sup>‡</sup> Gerard J. P. van Westen,<sup>\*,‡</sup> and Andrea Volkamer<sup>\*,†</sup>

<sup>†</sup>In silico Toxicology, Institute of Physiology, Charité – Universitätsmedizin Berlin, Charitéplatz 1, 10117 Berlin, Germany

<sup>‡</sup>Division of Drug Discovery and Safety, Leiden Academic Centre for Drug Research, Leiden University, P.O. Box 9502, 2300 RA, Leiden, The Netherlands

<sup>§</sup>Computational Chemistry, Janssen Research & Development, Turnhoutseweg 30, B-2340 Beerse, Belgium

**ABSTRACT:** Target deconvolution is a vital initial step in preclinical drug development to determine research focus and strategy. In this respect, computational target prediction is used to identify the most probable targets of an orphan ligand or the most similar targets to a protein under investigation. Applications range from the fundamental analysis of the mode-of-action over polypharmacology or adverse effect predictions to drug repositioning. Here, we provide a review on published ligand- and target-based as well as hybrid approaches for computational target prediction, together with



current limitations and future directions.

### INTRODUCTION

Target prediction is a key aspect in early preclinical drug development, pivotal to determine the clinical application and to initiate drug development campaigns. For instance, orphan compounds may be known from phenotypic screening, showing changes in cell or organism phenotypes upon compound exposure, without the underlying molecular mechanism being known.<sup>1</sup> Targets for orphan compounds can be experimentally identified with techniques based on chemical proteomics such as affinity chromatography and activity-based protein profiling (ABPP), enabling compound testing against the proteome of cell lysates or even intact cells and organisms.<sup>2–4</sup>

Since these experiments are time and cost extensive, computational alternatives to rapidly predict the primary targets have gained momentum and are commonly known as *in silico target prediction*, target identification, or target fishing.<sup>5</sup> Herein, a general distinction can be made between *ligand-based* methods, centered around small molecules, and *structure-based* methods, implementing information from protein structures.<sup>6</sup> Pivotal to most of these approaches is the chemical similarity principle stating that “similar molecules have a similar biological effect” and conversely that “similar proteins bind similar ligands”.<sup>7</sup>

One of the main applications of computational target prediction is to elucidate the *mode-of-action* of a compound by identifying its potential target. However, the traditional magic bullet paradigm, wherein a ligand has a high potency and selectivity toward a single target, has shifted to the understanding that a ligand affects multiple targets simultaneously.<sup>8,9</sup> In this context, target prediction methods can be used to explore desired *polypharmacological effects* of ligands to cover disease pathways.<sup>10</sup> Similarly, it can help to spot selectivity or

toxicity problems during compound optimization which can potentially lead to unwanted *adverse* or *side effects*.<sup>11</sup> Moreover, approved drugs, and hence clinically tested ligands, can be repurposed for different indications if they are also found to interact with a protein target that is part of another disease mechanism.<sup>12–14</sup> This process is called *drug repositioning* or *drug repurposing*. Whereas the aforementioned applications focus on predicting targets, computational target prediction methods can also be applied to select ligands that have the highest potential to be relevant *chemical probes* used for ABPP to characterize the biological function of a poorly understood target.<sup>15–17</sup>

Designed for computational biologists, medicinal chemists, and neighboring disciplines, this review aims to outline the general principle and potential of computational target prediction together with the underlying methods and their application. The article starts with ligand-based modeling, followed by hybrid approaches (using both ligand and protein data), as well as structure- and interaction-based methods (Figure 1). Finally, potential pitfalls of the different approaches are covered, and a future perspective is given.

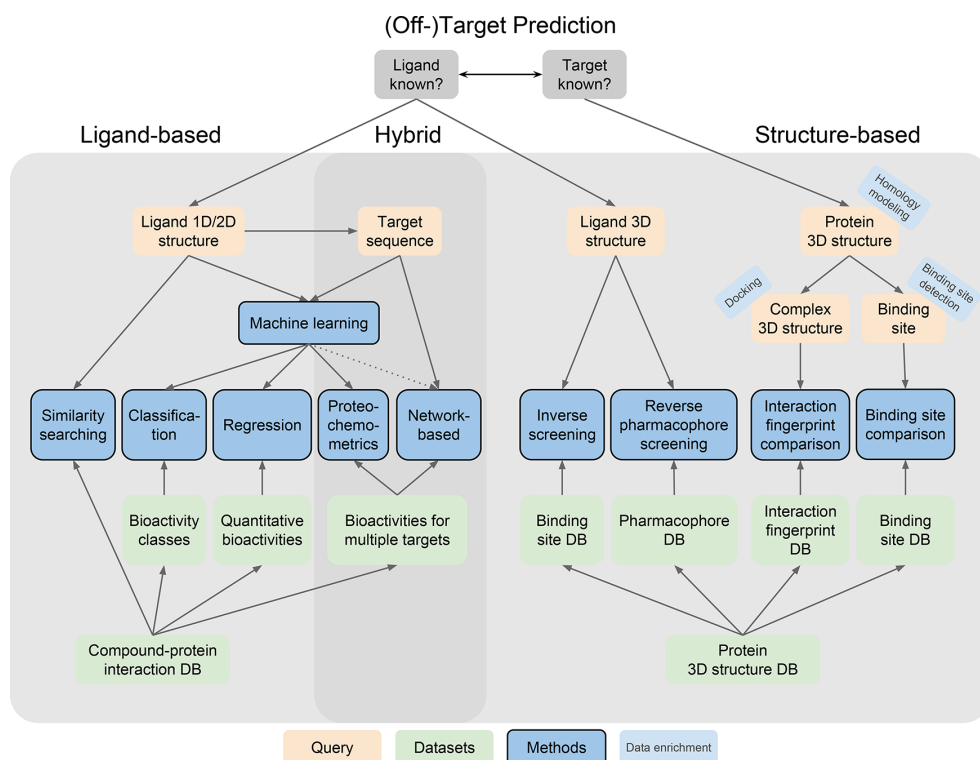
### LIGAND-BASED TARGET PREDICTION

Central to ligand-based methods is that they rely on the chemical structure of ligands and associated bioactivity of similar ligands. Ligand-based methods are often used to predict the bioactivity of novel compounds for a specific target (Figure 2). However, ligand-based methods can also be applied to predict activities for a range of targets. Generally, this can be

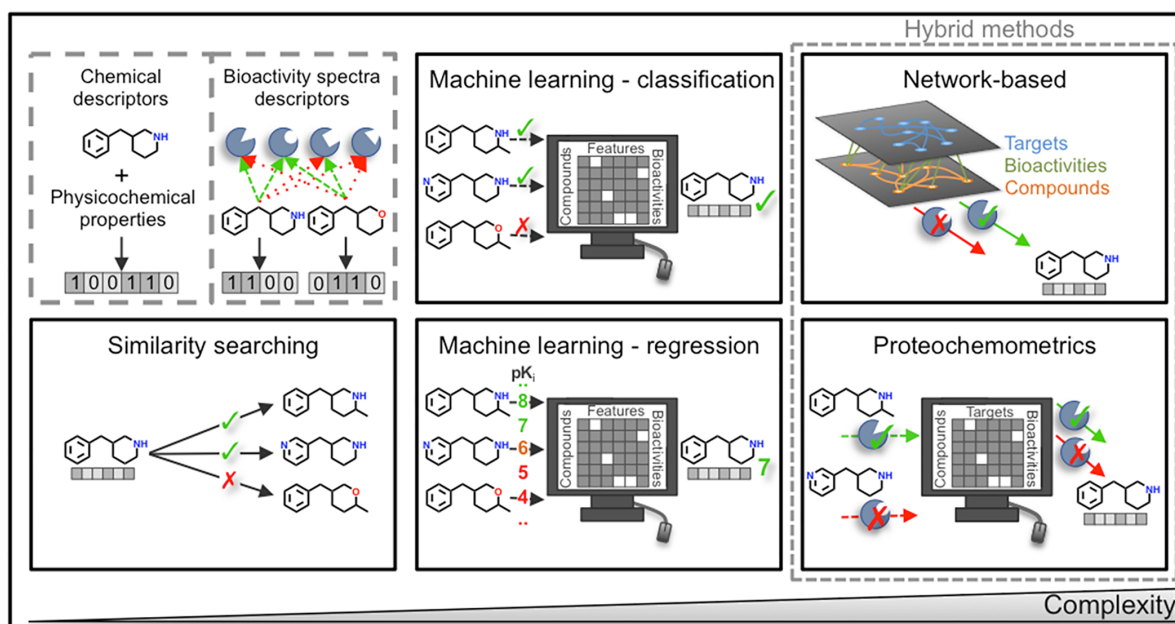
**Special Issue:** Women in Computational Chemistry

**Received:** November 21, 2018

**Published:** February 28, 2019



**Figure 1.** Overview of ligand- and structure-based as well as hybrid methods for target prediction (blue) with optional data enrichment strategies (light blue), using database (DB) or training data input (green), separated by applicability depending on available query data (orange). Necessary and potential connections are displayed with solid and dotted arrows, respectively.



**Figure 2.** Ligand-based methods for target prediction. Descriptors in ligand-based methods are shown in the dashed-lined boxes on the left. Methods increase in complexity from left to right.

Table 1. Ligand-Based and Hybrid Methods in Target Prediction<sup>a</sup>

Name	Data in model training		Training set requirements	Target ranking	Target prediction tools
	Compound	Interaction			
<b>Ligand-based models</b>					
Similarity searching	Chemical structure	–	–	Targets classified based on similarity threshold of compounds	SwissTarget-Prediction, <sup>35</sup> SuperPred, <sup>36</sup> SEA, <sup>40</sup> OCEAN, <sup>45</sup> ROCS, <sup>72</sup> FTrees <sup>73</sup>
Similarity searching	Bioactivities	–	–	Targets classified based on similarity threshold of bioactivity spectra	BASS, <sup>38</sup> BioSEA <sup>46</sup>
Machine learning: Classification	Chemical structure	Activity class	Balanced (in)active classes	Targets classified based on activity class	PIDGIN <sup>74</sup>
Machine learning: Regression	Chemical structure	Bioactivity	Normally/equally distributed bioactivities	Targets ranked based on bioactivity	–
<b>Hybrid models (ligand- and structure-based)</b>					
Protechemometrics	Chemical structure	Activity class or bioactivity	Balanced (in)active classes or normally/equally distributed bioactivities	Targets classified or ranked based on bioactivity	ChEMBL models <sup>58,65</sup>
Network-based models	Chemical structure and similarity	Activity class or bioactivity	Sufficient number of connections/bioactivities	Targets classified or ranked based on bioactivity	DINIES, <sup>68</sup> drugCIPHER <sup>69</sup>

<sup>a</sup>The table gives information on what data is used and how targets are inferred from the model output.

accomplished by ranking targets based on predicted compound activity: the target for which the highest activity is predicted is expected to be the most likely target of that query compound.

Typically, the ChEMBL database<sup>18</sup> occasionally in combination with PubChem,<sup>19</sup> e.g., in the case of the ExCAPE database,<sup>20</sup> is used as a public source for chemical structures. These databases hold experimentally validated bioactivity data for many compounds tested on a wide range of proteins.

In the following, some general compound descriptors for ligand-based methods are outlined; for specific details, the reader is referred to the review by Rognan.<sup>21</sup> Subsequently, a description of ligand-based methods ordered by increasing complexity coupled to prediction confidence is given (Table 1). The latter is expected to be higher for the more complex methods.

**Compound Descriptors.** Compounds in ligand-based models are typically described using their 2D chemical structures. Depending on the data source, an intermediate step can be the conversion from a 1D sequential textual format (e.g., SMILES<sup>22</sup>) to a 2D structure, from which more complex binary vectors such as molecular fingerprints are usually obtained.<sup>23</sup> Different fingerprints are available to describe chemical structures, e.g., atom-pair fingerprints, topological-torsion fingerprints, or circular fingerprints, where atom environments are included (e.g., ECFP).<sup>24</sup> Optionally, the 3D shape of compounds is taken into account and translated into similar molecular fingerprints. However, this requires additional information on the 3D conformation of the compounds.<sup>25,26</sup> The use of different chemical fingerprints can impact model performance and was explored by Bender et al.<sup>27</sup> Additionally, physicochemical properties, topological information, and pharmacophore features of compounds can be added as descriptors in a similar way. As a result, each compound is described by an array of numbers forming the compound descriptors. Resemblance between arrays is higher when compounds are more similar to each other.

A more complex representation of compounds, compared to chemical descriptors, are bioactivity spectra descriptors. A spectrum in its simplest form is a binary bitstring representation where each bit represents a protein. Proteins for which a given compound shows activity are marked with a “1” as opposed to those for which this is not the case (marked

with “0”). Bioactivity spectra rely on compounds being tested on a range of proteins, instead of compounds being tested on only one or a few targets. Considering compound promiscuity, it is expected that compounds display activity on a number of proteins.<sup>28</sup> Based on the bioactivity spectra, compounds that are not chemically similar but do exert a similar phenotype/bioactivity might be recognized (so-called activity cliffs<sup>29</sup>). Likewise, this bioactivity profile can form an array of numbers that can be implemented as descriptors for similarity searching or machine learning, where activities can be treated as a bioactivity fingerprint. Recently, the biological annotation of compounds has been extended to include gene expression profiles<sup>30,31</sup> and high content cellular images,<sup>32</sup> providing additional, high-dimensional descriptors that can be added to a bioactivity fingerprint in a straightforward way.

**Similarity Searching.** The simplest and fastest method for target prediction is based on molecular similarity and is often referred to as similarity search or nearest neighbor search.<sup>33</sup> Using a similarity coefficient of choice (e.g., Tanimoto) and any type of compound descriptors (e.g., ECFP), the similarity between a pair of molecules can be quickly generated. For example, finding the most similar 100 compounds for a given query compound in a PubChem-sized library (~96 million compounds) takes a few seconds using chemfp tools developed by Dalke.<sup>34</sup>

The simplest implementation for target prediction based on similarity is to rank the data set compounds based on their similarity toward the query compound and assume that the biologically tested target of the most similar compounds is also the most likely target of the query compound. Webserver tools that enable the use of this method are, e.g., SwissTargetPrediction<sup>35</sup> and SuperPred.<sup>36</sup> These tools suggest protein targets based on molecular similarity of the query compound to compounds with known bioactivity toward these targets. It should be noted however that these approaches cannot provide a direct quantification of the biological activity of the query compound on the top-ranked targets.

While similarity search is classically performed by comparing chemical descriptors, activity spectra descriptors can also be used (if enough bioactivity data is available). Early work by Kauvar et al.<sup>37</sup> characterized molecular similarity by an affinity fingerprint based on experimental screenings of molecules

against a reference panel of selected proteins. Also in BASS<sup>38</sup> (bioactivity profile similarity search), the similarity search is performed based on bioactivity spectra of chemical structures. Here, when the query has experimentally validated activities on a number of targets, additional targets can be predicted based on its bioactivity spectrum. Alternatively, gene expression profiles can be used to predict bioactivities of compounds for targets.<sup>30,39</sup> Both bioactivity spectra and gene expression profiles do not compare the molecular structure of compounds. Therefore, these methods are suited to identify different chemical structures for similar targets.

In contrast to a classical similarity search, **similarity ensemble** methods are applied to identify targets based on a group of known compounds for that target rather than a single compound. The compounds are first grouped based on interactions (e.g., bioactivity) with the same target(s). The similarity between different compound groups is subsequently calculated, and when defined as being similar, the targets that are known to interact with one compound group are identified as targets for the other compound group(s). The added benefit is that this allows the calculation of statistical measures that can score the relevance of a given retrieved target. When ensemble approaches are applied to identify targets for a query compound, the similarity is measured between this compound and the different compound groups. The targets belonging to the most similar groups are then identified as targets for the query compound. The SEA<sup>40</sup> method utilizes the similarity ensemble concept to group proteins based on ligand topology. Within SEA, the retrieved value is then compared to an expected random value (similar to the way this is implemented in BLAST<sup>41,42</sup>), and subsequently, an “E-value” is returned.<sup>43</sup> This E-value represents the extreme value and indicates the quality of the result. The (similarity) score of the selected samples is compared to what is expected when two random samples are taken into account. E-values closer to zero indicate that it is more unlikely that random samples would have equal similarity as the selected samples. The SEA method has been applied by Lounkine et al.<sup>44</sup> in a target prediction challenge. Here, side effects of 656 compounds were predicted based on compound interactions with 73 off-targets. The results were partially validated by data from hold-out databases or experimentally validated *in vitro*. Remarkably, off-targets were identified that had very low sequence similarity with the on-target (e.g., off-target serotonin transporter 5-HTT and on-target histamine H<sub>1</sub> receptor for antihistamine diphenhydramine), indicating that such a ligand-based approach can predict targets without the need of molecular biology information on protein targets. OCEAN<sup>45</sup> is a similar technique, though using different thresholds to determine compound similarities. Finally, BioSEA<sup>46</sup> also applies the same methodology; however, instead of comparing compound similarities based on chemical structure, bioactivity profiles are compared to create ensembles of compounds.

**Machine Learning.** Similarity search methods consider all features in the compound descriptors as equal. However, statistical methods can weigh the relevance of individual descriptors by connecting them to biological activity of the compounds and are often better suited to extrapolate to new compounds. Machine learning methods require a training phase, which is performed on known active and inactive compounds. Herein, a statistical model is fitted to the data to quantify how chemical descriptors relate to activity. Contrary to the similarity searching example above, this approach

returns predicted compound–protein activities rather than a number of compound structures that are similar for a query compound. When applied to a single protein target for a congeneric chemical series, these methods are named *quantitative structure–activity relationship* (QSAR) models.<sup>47</sup> Given a query compound, QSARs can predict its expected activity based on the compound descriptors. In target prediction, however, more than one protein is considered.

Machine learning can both be used for classification (e.g., is the expected affinity higher than a threshold that was defined *a priori* as active?) or for regression (e.g., what is the predicted K<sub>i</sub> value for a compound–protein interaction?). Typically, algorithms such as Random Forest,<sup>48</sup> Support Vector Machines,<sup>49</sup> and Naïve Bayes<sup>50</sup> are applied. However, with more data becoming available and to become more independent of the chosen descriptor, recent work is moving toward deep learning, a method able to directly derive features from molecular structures.<sup>51,52</sup>

An example where machine learning was applied in target prediction is the identification of novel inhibitors for the enzyme mycobacterial dihydrofolate reductase.<sup>53</sup> Here, targets were predicted for a set of query compounds using Naïve Bayesian models. The predicted compound–target interactions were validated *in vitro*, which indicates the value of such target prediction methods.

**Classification.** The most frequently used method in ligand-based target prediction is arguably classification.<sup>1,54</sup> Classification requires the setting of an activity threshold for measured interactions to separate the classes. This interaction can be measured binding affinity (e.g., pK<sub>i</sub>) but can also be efficacy or other experimental measurements (e.g., pEC<sub>50</sub>) or even a combination of multiple measurement types (e.g., pChEMBL value).<sup>55</sup> For classification models, a difference can be made between several approaches:

**Single Model Multi-Class (SMMC).** In this approach, one model is used that predicts the most probable target for a given compound, and target classes are mutually exclusive, in other words a compound cannot be active on more than one target.<sup>56</sup> Given known ligand promiscuity, the SMMC method provides an inaccurate representation of the behavior of ligands and could even be considered to be at odds with the similarity principle.

**Ensemble Model Multi-Label (EMML).** With EMML, also referred to as ensemble model multi-class, one model is used per protein, and compounds receive a prediction from each model.<sup>1,57</sup> Thus, the sum of protein models where the compound was predicted active on represents the set of potential target proteins. To build the model per protein, all compounds with an activity for the respective protein above a certain threshold are deemed the active class, and all other compounds are typically pooled in the inactive class. For the EMML approach, pooling constitutes a source of error. It might very well be that although a given compound has not been tested on the protein under consideration, it is indeed active yet pooling defines it to be inactive. Thus, potential targets for the query compound may be missed.

**Single Model Multi-Label (SMML).** Here, one model is used to predict all potential targets for a given molecule, and compounds can belong to multiple target classes (or labels).<sup>56</sup> The active class for a given protein is defined equally as is described for EMML, but all other compounds are not explicitly pooled in an inactive class, merely the ones that were tested to be inactive are considered. A caveat can be that there



are none or too few known inactive compounds for good model fitting.

When a query compound is run through a classification model, the output gives the activity class per target (e.g., active/inactive, depending on the previously described approaches and on the predetermined activity threshold). However, regression can directly predict the affinity of a compound.

**Pitfalls Defining an “Active” Class.** Typically, the activity threshold in classification models is set at 10  $\mu\text{M}$  (i.e., an affinity better than 10  $\mu\text{M}$  defines active interactions, corresponding to a  $\text{p}K_i$  of 5). This parameter carries a significant influence on effectiveness and applicability of target prediction methods. In principle, for classification, a balanced set of active and inactive compounds is desired. When the activity threshold is set at 10  $\mu\text{M}$ , this gives a skewed distribution of actives and inactives. Recently, target prediction was performed using an affinity value of  $\sim 316$  nM (corresponding to 6.5 on a logarithmic scale) as the threshold; this leads to a better distribution of active and inactive classes when using ChEMBL data.<sup>58</sup> An added benefit is that this threshold also provides a more relevant prediction of biological activity. Given that the biological error of assays is on average around  $\sim 0.5$  log units for mixed  $\text{p}K_i$  values, a model using a cutoff of  $\text{p}K_i = 6.5$  could at worst correspond to an experimental activity of a  $\text{p}K_i = 6.0$ . When a cutoff of  $\text{p}K_i = 5.0$  (10  $\mu\text{M}$ ) is used, this error would be at worst  $\text{p}K_i = 4.5$  for predicted actives.<sup>57,58</sup> However, the optimal activity threshold for balanced classification sets is dependent on the databases from which compounds and bioactivities are extracted (e.g., ExCape<sup>20</sup> contains more compounds with lower bioactivities than ChEMBL). Furthermore, the targets that are considered can be biased toward reported (in)actives (often in relation to the amount of studies focused on the target, see the [Discussion and Future Directions](#) section).

When a reasonable number of inactive compounds is available, but significantly less than the number of active compounds, some workarounds can be applied to train representative models. For instance, active compounds can be divided into smaller subsets in order to train separate models for each subset of actives with the same set of inactives (e.g., random undersampling) and, finally, recombined by ensembling. Ensembling is a technique to combine predictions from multiple models into one prediction that has shown to increase performance.<sup>58,59</sup> The downside of any ensembling method is the unavoidable increase in computational time required as predictions for multiple methods are needed.

Another workaround (which also requires increased computational time) is to construct multiple ligand-based target prediction models at different thresholds (e.g., 10  $\mu\text{M}$ , 1  $\mu\text{M}$ , 100 nM, 10 nM, and 1 nM). However, doing so decreases the available data points for the higher activity thresholds as fewer compounds are known that meet the threshold, and hence, this has a negative effect on the chemical applicability domain. In these cases, regression might allow the use of more data.

**Regression.** Contrary to classification, regression methods are able to directly train on the strength of a given ligand–protein interaction avoiding the need for a preset threshold. Trained on experimental data, regression models can make quantitative predictions (e.g.,  $K_i$  values) for compounds based on the chemical structure. These predictions can be directly translated to the interaction (e.g., affinity as a  $K_i$  value). Thus,

when regression is applied to multiple proteins (using an ensemble of models), the targets can quantitatively be ranked based on predicted compound–protein activity. In addition to predicting activity, the differences in interaction strength for different proteins can be evaluated. Using regression models, the output of a query ligand can constitute a list with ranked targets based on quantitative bioactivity predictions. The output, therefore, does not only define “active” or “inactive” targets but also the activity strength that is reflected by the predicted bioactivity values.

## ■ HYBRID METHODS FOR TARGET PREDICTION

Similarity searching and machine learning methods—which are classically built on ligand information—can also be applied in more complex systems where protein information is added. Although the underlying mechanism of the methods is the same (e.g., machine learning), the implementation can be different, in turn leading to other application possibilities. This results in alternate methods to model and analyze the data.

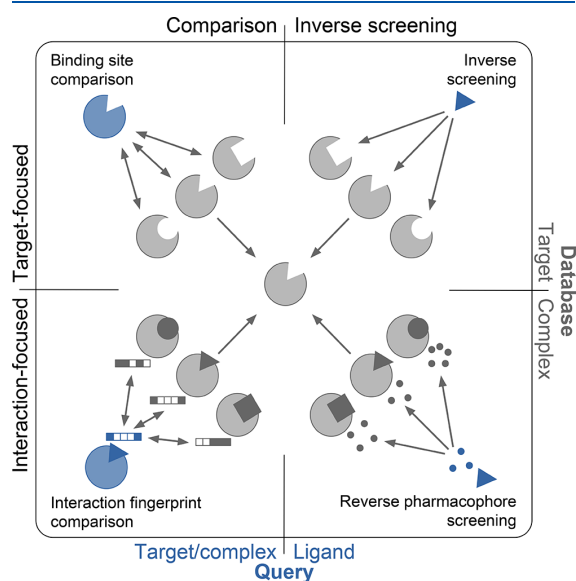
**Proteochemometrics.** With proteochemometrics (PCM), both compound and protein information are combined by addition of an explicit protein descriptor.<sup>60</sup> The most common approach is to add protein information based on knowledge derived from the protein sequence. Sequences are translated into descriptive scores (e.g., Z-scales<sup>61</sup>), reflecting the properties of the amino acid residues of the proteins.<sup>62</sup> Additionally, when structural protein information is available, this may be used to increase descriptor quality as information on binding site location can be included, making the model more accurate compared to using full sequences.<sup>63</sup> PCM can be applied to expand single target models to multiple targets: based on sequence similarity between proteins, data from one protein can be extrapolated to a related one.<sup>64</sup> Another application is increasing the amount of available data (compared to single target models) in order to increase model performance.<sup>63</sup> Several PCM models for target prediction based on ChEMBL data have been reported.<sup>58,65</sup> Such models predict the activities of a query compound for each of the incorporated targets. When these models are based on regression, the most likely target for a query compound can be derived based on the highest predicted activity for that target compared to other targets. Additionally, a quantitative activity score is given per target; therefore, it can be assessed if activity of the query compound for the highest ranked target(s) is sufficient. Noteworthy, as the combination of compound and protein descriptors defines each compound protein pair as a unique pair, even binary class PCM models behave as SMML models. A compound tested to be inactive on protein A can be distinguished from the same compound tested on protein B by the algorithm based on the protein descriptor.

**Network-Based Methods.** Protein–protein or protein–ligand interactions can be described as a large network similar to a social network. Here, nodes can be proteins, compounds, or both, with the edges being interactions, similarities, or phenotypic effects. These connections can also be weighted based on the strength of interaction (e.g.,  $\text{p}K_i$ ). Using chemical structures and similarities between connections, targets can be identified for query compounds.<sup>51</sup> This has led to the publication of several works that use network analysis tools to predict protein pharmacology.<sup>66,67</sup> Additionally, network-based target prediction tools such as DINIES<sup>68</sup> and drugCIPHER<sup>69</sup> are made available as open source tools to detect ligand–target interactions for query molecules. The

concept of network-based models is often based on similarities between chemical structures but can also include similarities between proteins. More simplistic models implement only one similarity (e.g., protein similarity), whereas more complex models can encompass similarities between protein, chemical structures, and interactions, simultaneously. Such a heterologous network was constituted using three different networks by Chen et al.<sup>70</sup> Here, a protein similarity network (based on sequence similarity) was connected to a compound similarity network by using a ligand–protein interaction network.<sup>71</sup> Therefore, in this network, protein and compound similarities can simultaneously be addressed, which is not possible with only similarity searching as described in the section regarding this topic. Targets for a given query compound can be inferred from the network based on activities (or connections) of similar ligands and their corresponding targets.

### ■ STRUCTURE-BASED TARGET PREDICTION

Methods for structure-based target prediction identify the most likely targets for a query ligand or the most similar targets for a query target, using 3D structural, i.e., steric and physicochemical, information (Figure 3). The former group of approaches



**Figure 3.** Structure-based target prediction: conceptual representation of the four main approaches, i.e. binding site comparison, inverse screening, reverse pharmacophore screening, and interaction fingerprint comparison.

focuses on docking a *query ligand* either to a set of targets (*inverse screening*) or to a set of pharmacophores inferred from ligand–target complexes (*reverse pharmacophore screening*), see Table 2. The latter group of methods compares a *query target*, either to a set of targets (*binding site comparison*) or to a set of interactions inferred from ligand–target complexes (*interaction fingerprint comparison*),<sup>5</sup> see Table 3.

Typically, the Protein Data Bank (PDB)<sup>75</sup> is used as a public source for protein structures, currently holding more than 140,000 protein structures (accessed in November 2018). Since the binding site is the key to protein function, most methods are preceded by a binding site annotation step: with

a ligand present, binding sites are extracted by a defined ligand–target residue distance cutoff, and without a co-crystallized ligand, binding site detection methods can be invoked.<sup>76</sup> A widely used resource for such annotated binding sites is the scPDB<sup>77</sup> database, containing more than 16,000 ligand-bound binding sites from the PDB and covering about 4700 proteins with 6300 ligands.

Methods for structure-based target prediction are all composed of three main steps, which are described in detail in the individual method paragraphs: (i) binding site encoding, (ii) target screening or comparison, and (iii) target ranking. First, binding sites or ligand–target interactions are encoded using different descriptor techniques and stored in a target database. Second, depending on the method, either a query ligand is screened against the target database, using different docking engines, or a query binding site is compared with the target database, using different similarity measures. Finally, targets are ranked based on a suitable scoring approach.

**Inverse Screening.** Classically, molecular docking is used to predict both the binding mode and the approximate binding free energy of a set of ligands against one target of interest. In inverse docking, also known as inverse screening or panel docking, this strategy is reversed, and one query ligand is docked to a set of target proteins in order to predict its most likely targets. Most docking tools are theoretically applicable for inverse screening, yet need adaptation with respect to inter-target instead of conventional inter-ligand ranking (Table 2).<sup>78,79</sup>

(i) **Binding Site Encoding.** Since the query compound is screened against each target in the data set, the targets need to be preprocessed accordingly. Target databases for methods using conventional docking engines simply contain structure files for binding sites (e.g., TarFisDock<sup>80</sup> and idTarget<sup>81</sup>) or for whole proteins (INVDock<sup>82</sup>), preprocessed as required for the respective docking tool. In contrast, iRAISE<sup>83</sup> prepares for an efficient comparison by encoding binding sites with triangle descriptors, which contain pharmacophoric and shape information and are stored as bitmap database, a specialized index for high-dimensional features.

(ii) **Target screening.** Most inverse screening methods use conventional docking engines, such as DOCK (TarFisDock), MEDock (idTarget), Glide (VTS<sup>84</sup>), or AutoDock Vina (VinaMPI<sup>85</sup> and IFPTarget<sup>86</sup>), in order to estimate the fit of the query compound against each protein in the target database. High computational costs are addressed by either parallel screening (VinaMPI and IFPTarget) or by search space reduction. The latter can be realized by aborting the search at the first pose reaching a threshold score based on interaction energies from reference ligand–protein complexes (INVDock) or by testing one target representative per precalculated target cluster (based on sequence identity) before screening the entire cluster (idTarget). Usually, energy-based functions, such as interaction or binding free energy functions, are used to score the resulting docking poses. In iRAISE, the query ligand is described with triangles, in the same manner as the binding sites before, and is efficiently matched based on bitmap indices, followed by respective superimposition of the ligand and binding site triangles. Finally, iRAISE docking poses are scored using a more extensive approach in the form of a scoring cascade, including a clash test, an interaction energy score, a reference score cutoff (based on the co-crystallized reference ligand), and a ligand and pocket coverage score.

Table 2. Structure-Based Target Prediction: Selected Methods for Inverse Screening and Reverse Pharmacophore Screening

Name	Encoding	Target screening		Target ranking	Av. <sup>a</sup>
		Docking engine	Scoring function		
<b>Inverse screening</b>					
INVDOCK <sup>82</sup>	Sphere-coated surface	DOCK derivative	Interaction energy	–	2
TarFisDock <sup>80</sup>	Sphere-coated surface	DOCK 4.0	Interaction energy	–	2
idTarget <sup>81</sup>	Energetic grid map	MEDock	Binding free energy (AutoDock4 score)	Z-score based on binding free energies of reference complexes	1
VTS <sup>84</sup>	Energetic grid map	Glide	Binding free energy (Glide Gscore)	Gscore comparison to Boltzmann-weighted average of reference Gscores	2
VinaMPI <sup>85</sup>	Energetic grid map	AutoDock Vina	Binding free energy (Vina score)	–	1
iRAISE <sup>83</sup>	Bitmap of binned triangles (3 pharmacophore features and cavity shape)	Index-based bitmap comparison	Scoring cascade: clash test, interaction energy and reference cutoff, ligand and pocket coverage	Gaussian-weighted score based on scores for reference complexes	1
<b>Reverse pharmacophore screening</b>					
PharmMapper <sup>90</sup>	Hash table of binned triangles (5 pharmacophore features)	Geometric hashing	Fit score (based on matching feature types and positions)	Z-score based on fit score distribution of reference complexes	1

<sup>a</sup>Av. = availability: web server, software, or code is (1) free for academic use and/or available upon request or (2) not (yet) available or unclear.

(iii) **Target Ranking.** Targets are ranked either directly based on the interaction energies of the best docking pose(s) per target (INVDOCK, TarFisDock, and VinaMPI) or based on separate functions tailor-made for inter-target ranking. In the latter approach, each target in the database is profiled beforehand either with a set of ligands using docking (iRAISE and VTS) or with one co-crystallized ligand (idTarget and IFPTarget). These reference profiles are then used to normalize the scores of docking poses of a query ligand and potential targets.

Inverse screening methods have been widely used for target prediction.<sup>78,79</sup> For example, Scafuri et al.<sup>87</sup> applied idTarget to predict potential targets of apple polyphenols, known for their chemo-preventive effect against colorectal cancer. In a bioinformatics-driven function analysis, the gene expression levels for the predicted targets were shown to be significantly altered in colorectal cancer cells, indirectly linking the investigated apple polyphenols to the predicted targets.

**Reverse Pharmacophore Screening.** Similar to inverse screening, reverse pharmacophore screening consecutively fits a query ligand in the form of a ligand-based pharmacophore into a precalculated panel of pharmacophore models, derived from protein–ligand complexes. A pharmacophore is defined as an ensemble of physicochemical and steric features that are necessary for the recognition of a ligand by a target, triggering or blocking a biological response.<sup>88</sup> Structure-based approaches derive such pharmacophores from a target complex, whereas ligand-based pharmacophores consider solely ligand properties. Several studies have conducted reverse pharmacophore screening for polypharmacology, using available standard software packages that allow for rapid pharmacophore model building and evaluation.<sup>89</sup> However, to the knowledge of the authors, the only available automated workflow for pharmacophore-based target prediction is PharmMapper.<sup>90</sup>

In PharmMapper, the interactions of selected ligand–target complexes are encoded as pharmacophore feature triplets, stored in a hash table, and deposited in a target database (i). For target screening (ii), ligand-based pharmacophores are generated for multiple conformations of the query ligand. Each conformer pharmacophore is described in form of triplets and aligned onto each pharmacophore triplet in the target database, using triangle hashing. Subsequently, targets are scored based on the overlap of feature types and positions between the

ligand and target pharmacophores. Finally, each target score is normalized by a reference score for target ranking (iii). The reference score per target reflects the score distribution of matching all ligand pharmacophores extracted from the original protein–ligand complex structures in the database against the target pharmacophore.

Reverse pharmacophore screening was often applied to search for targets of compounds in Chinese traditional medicine (CTM).<sup>79</sup> For example, Liu et al.<sup>91</sup> used PharmMapper to predict the glucocorticoid receptor, p38 mitogen-activated protein kinase, and dihydroorotate dehydrogenase as potential targets of berberine, a compound used in CTM to treat cancers including melanoma. Experimental tests confirmed the predicted targets to be potentially involved in the anti-melanoma effect of berberine.

**Binding Site Comparison.** Target comparison is based on the assumption that similar proteins—or more precisely binding sites—bind similar ligands. Various binding site comparison methods have been developed, pursuing different strategies to encode binding sites, as well as to measure and score their similarities<sup>92,93</sup> (Table 3).

(i) **Binding Site Encoding.** The structural complexity of binding sites is reduced to labeled representatives, whose spatial arrangement is encoded and stored in a database, to be compared with a query binding site encoded accordingly. Binding site representatives can be per-residue points (e.g., CavBase<sup>94</sup> or (Med-)SuMo<sup>95,96</sup>), binding site surfaces (e.g., ProBis<sup>97</sup>), or binding site volumes (e.g., Volsite/Shaper<sup>98</sup>), with labels mostly containing pharmacophoric information. The spatial arrangement of these representatives is often encoded as graphs (e.g., CavBase) and triangles/quadruplets. The latter are binned by their edge lengths and vertex labels and stored as fingerprints (e.g., FuzCav<sup>99</sup> and FLAP<sup>100</sup>), hash tables (SiteEngine<sup>101</sup>), or bitmaps (TriXP<sup>102</sup>), whereas (Med-)SuMo<sup>95,96</sup> uses a graph of adjacent triangles. Alternate methods describe binding sites as distance distributions between aforementioned per-residue points (e.g., RAP-MAD<sup>103</sup>), or with volume functions (Volsite/Shaper).

(ii) **Binding Site Similarity Measure.** Common strategies for measuring binding site similarities can be divided into alignment-based (often slower) and alignment-free methods (mostly faster), as well as accelerated alignment-based methods. The latter combine the speed of alignment-free

Table 3. Structure-Based Target Prediction: Selected Methods for Binding Site and Interaction Fingerprint Comparison<sup>a</sup>

Name	Representatives	Encoding	Label	Pattern	Comparison	Scoring	Av
<b>Binding Site Comparison</b>							
<i>Alignment-Based Methods</i>							
SiteBase <sup>104</sup>		5 atom types		<sup>5</sup> On-the-fly triangles	Geometric matching	Matching atoms	2
(Med-SuMo) <sup>95,96</sup>		Chemical groups		<sup>5</sup> Triangles as graph of adjacent triangles	Geometric matching, stepwise connection of adjacent matches	Size of connected matches	1,3
SiteEngine <sup>101</sup>		5 pharmacophoric features		<sup>5</sup> Triangles in hash table	Geometric hashing	Matching surface patches & PCs	1
CavBase <sup>94</sup>		5 pharmacophoric features		<sup>4</sup> Graph	Clique detection	Matching surface patches & PCs	3
eF-site <sup>113</sup>		Electrostatics & surface curvature		<sup>4</sup> Graph	Clique detection		1
ProBis <sup>97</sup>		5 pharmacophoric features		<sup>4</sup> Subgraphs	Clique detection (per subgraph)	Matching surface patches & residues	1
PolLMorph <sup>114</sup>		5 physicochemical features*		<sup>4</sup> Fuzzy graph/ self-organizing map	Error-tolerant graph matching	Matching vertices	2
<i>Alignment-Free Methods</i>							
Pocket-Match <sup>115</sup>		5 amino acids groups = B		<sup>6</sup> 90 distance histograms for all A-B combinations	Corresponding histogram comparison	Average matching distance bins	2
RAPMAD <sup>103</sup>		<sup>1</sup> Per-residue: $C_{\alpha}$ , $C_{\beta}$ & centroid = A		<sup>6</sup> 14 distance histograms: $P_1-P_1$ and $P_2-P_2$ per $s_i$	Corresponding histogram comparison	Jensen-Shannon divergence	2
FuzCav <sup>99</sup>		<sup>7</sup> pharmacophoric features: 7 PC subsets $s_i$		<sup>5</sup> Triangles as 4833 int fingerprint	Fingerprint comparison	Matching non-zeros	1
KRIPO <sup>116</sup>		<sup>6</sup> Per-residue: $C_{\alpha}$		<sup>5</sup> Triangles as fuzzy fingerprint(s)	Fingerprint comparison	Modified Tanimoto index	1
Pocket-FEATURE <sup>117</sup>		<sup>1</sup> Defined points relative to residue		<sup>7</sup> 480 int fingerprints per ME shell	Fingerprint comparison per ME same amino acid	Sum of bestscoring (Tanimoto index) ME pairs	2
<i>Accelerated Alignment-Based Methods</i>							
BSAlign <sup>105</sup>		5 physicochemical features*		<sup>4</sup> Reduced (red.) graph	Clique detection, red. product graph	Matching residues & RMSD	1
SiteAlign <sup>106</sup>		8 topological and chemical descriptors mapped to triangles		<sup>5</sup> 640 int fingerprint	Fingerprint comparison	Average of normalized triangle differences	1
TriXP <sup>102</sup>		3 pharmacophoric features		<sup>5</sup> Triangles in bitmap	Index-based bitmap comparison	Matching triangles	2
FLAP <sup>100</sup>		5 pharmacophoric features		<sup>5</sup> Quadruplets as 11 int fingerprints	Fingerprint comparison	Matching quadruplets	3
BioGPS <sup>118</sup>		3 pharmacophoric features		<sup>5</sup> Quadruplets as 11 int fingerprints	Fingerprint comparison	MIF volume overlap per feature	2
Volsite/Shapet <sup>98</sup>		7 pharmacophoric features		<sup>7</sup> Volume as smooth Gaussian function	Volume overlap	Matching pharmacophoric features	1
<b>Interaction Fingerprint Comparison</b>							
SIFT <sup>110</sup>	Interacting residues	7 pharmacophoric features		Per-residue: 7 bit vector, concatenated in fixed order	Fingerprint comparison	Tanimoto index	2
TIEP <sup>111</sup>	Pseudoatoms between interacting ligand-target pairs	7 pharmacophoric features		Triplets as 210 int fingerprint	Fingerprint comparison	Tanimoto index	2
SPLIF <sup>112</sup>	Interacting fragments	Atom and bond types		ECFP2 fingerprint	RMSD for all matching fingerprint bits	Matching ligand and protein atoms	2
LIFT <sup>109</sup>	Atom-by-atom ligand-target interactions	10 pharmacophoric features		Interaction fingerprint	Fingerprint comparison	Tanimoto index	2
IEPTarget <sup>86</sup>	Interacting residues	8 pharmacophoric features		Label $\times$ residue matrix	Matrix comparison between query and reference complexes	Modified Tanimoto index & energy-based score for query complex	2

<sup>a</sup>Binding sites are encoded based on <sup>1</sup>per-residue points, binding site <sup>2</sup>surfaces, and <sup>3</sup>volume, and are represented as <sup>4</sup>graph, <sup>5</sup>triangles/quadruplets (e.g. binned into fingerprints/hash tables/bitmaps), <sup>6</sup>distance distributions of atom pairs, and <sup>7</sup>volume functions. RMSD = root-mean-square deviation; MIFs = molecular interaction fields; Av. = availability; web server, software, or code is (1) free for academic use and/or available upon request, (2) not (yet) available or unclear, or (3) commercially available; \*Including pharmacophoric and additional features, e.g. buriedness.

methods with the visual interpretability of alignment-based methods. *Alignment-based methods* calculate and perform the best possible structural superimposition of two binding sites based on their encoded features, using geometric matching and hashing of two triangle sets (e.g., SiteBase<sup>104</sup> and SiteEngine, respectively) or most commonly clique detection between two graphs (e.g., CavBase). The latter approach searches the maximum complete subgraph (clique) in a product graph, which is built from a target and query graph with matching vertices and edges. Many *alignment-free methods* operate on the comparison of fingerprints (e.g., FuzCav) or of distance histograms (e.g., RAPMAD). *Accelerated alignment-based methods* use efficient data structures for rapid comparison, with subsequent binding site alignments for scoring and visual interpretation. Those methods include strategies to reduce graph complexity before clique detection (BSAlign<sup>105</sup>), to compare binding site volumes using smooth Gaussian functions (Volsite/Shaper), and to store binned 3-point pharmacophores in bitmap indices (TriXP). Moreover, properties of a binding site can be projected to a triangulated sphere positioned at its center, stored as fingerprint to be iteratively compared, and aligned to another binding site fingerprint (SiteAlign<sup>106</sup>).

(iii) *Binding Site Similarity Ranking*. Alignment-based methods score the similarity of binding sites based on the mutual overlap and/or root-mean square deviation (RMSD) of their associated encoded features. In contrast, alignment-free methods mainly calculate fingerprint similarity based on the number of matching fingerprints, if multiple fingerprints exist per binding site (e.g., FLAP), or based on the Tanimoto coefficient, if only one fingerprint per binding site (e.g., FuzCav) is calculated.

An exemplary application of binding site comparison is a study on cross-reactivity using SiteAlign by De Franchi et al.<sup>107</sup> Virtual screening of Pim-1 kinase against ATP-binding sites showed high similarity to synapsin I, a protein regulating neurotransmitter release in the synapse, suggesting a cross-reaction of protein kinase inhibitors with synapsin I. Biochemical validation revealed nanomolar affinities for pan-kinase inhibitor staurosporine and selective Pim-1 kinase inhibitor quercetagenin for synapsin I. These findings were proposed as possible explanations for the observed down-regulation of neurotransmitter release by some protein kinase inhibitors.

**Interaction Fingerprint Comparison.** Interaction fingerprints (IFPs), or protein–ligand fingerprints, are vectors that encode information on interacting ligand and target moieties, such as hydrogen bond, hydrophobic, charge, aromatic, and metal-binding interactions. IFPs are often used in combination with screening methods in order to rescore docking poses.<sup>108</sup> Only a few IFP-based pipelines have been published for target prediction so far. Note that they require a ligand placement step for IFP calculation. Thus, for IFP encoding (i), the query ligand has to be docked against the target structure(s). Generally, IFP methods either map detected interactions to ligand atoms (e.g., LIFt<sup>109</sup>), to target binding site residues (e.g., SIFt<sup>110</sup> and IFPTarget<sup>86</sup>), or define a ligand- and target-independent fixed length fingerprint (e.g., TIFP<sup>111</sup> and SPLIF<sup>112</sup>). Similar to the alignment-free fingerprint-based binding site comparison, the comparison of two IFPs is usually based on the Tanimoto coefficient (ii), and targets are rank-ordered accordingly (iii). In the following, two tools are introduced: In the first approach, interactions are mapped on

the ligand; thus, ligand IFPs are compared. In the second, information is mapped on the target residues, and subsequently, target IFPs are compared.

Cao and Wang<sup>109</sup> propose a pipeline for off-target prediction exemplified on a tubulin agent with kinase-cross activity. The tubulin agent complex structure is the starting point to generate the ligand-based interaction fingerprint (LIFt) for the query compound. Next, the query ligand is docked to a panel of kinase structures. The best-scoring pose per ligand–kinase complex is encoded as LIFt, documenting interactions per ligand atom. Finally, these predicted panel LIFts are compared (Tanimoto coefficient) to the known reference LIFt and ranked accordingly.

In contrast, IFPTarget by Li et al.<sup>86</sup> first sets up a target database, where the co-crystallized ligand is used to define the reference target IFP, documenting per-residue interactions. Next, the query ligand is docked to the same panel of targets, and the top-scoring pose for each target is used to generate the docked target IFP. Subsequently, reference and docked target IFPs are compared and ranked by a final score that integrates aforementioned energy-based docking and IFP-based scores.

The presented methods are strongly intertwined with a docking (inverse screening) procedure: Two IFPs can only be compared if they have one constant component (LIFt: same ligand in two different structures, or IFPTarget: same structure with two different ligands) because otherwise the IFP lengths and order differ. Here, the third category of ligand and protein invariant fingerprints, such as TIFP by Desaphy et al.,<sup>111</sup> could find a remedy, but has, to the knowledge of the authors, not yet been used for target prediction.

**Consideration of Target Flexibility in Structure-Based Methods.** Proteins are flexible, existing in transient conformational states, whereby only a subset may be receptive to ligand binding. Such flexibility is to some extent implicitly considered by the coarse-grained representation of binding sites in the encoding step, such as binned distances (e.g., RAPMAD and FuzCav) and fuzzified graphs (PoLiMorph<sup>114</sup>), as well as by including tolerances during the matching step. Small side-chain flexibility can be explicitly included by, e.g., representing rotatable hydrophilic interactions (TriXP) or “on-the-fly” conformational sampling of side chains (FLAP and BioGPS<sup>118</sup>). Instead of conformational sampling, different parts of the binding site can be investigated separately from each other in order to spot local similarities. Some methods therefore allow for partial shape matching (TriXP) or local examination of binding site segments (ProBis). Inverse screening methods usually treat the target structure as rigid body, while considering ligand flexibility by conformational sampling of the ligand (e.g., iRAISE and INVDOCK).

However, information on protein flexibility can be enriched by including protein ensembles in screening databases, either derived from a set of experimentally determined structures or from molecular dynamics (MD) simulations. The former approach is to some extent integrated whenever methods are built upon a database containing multiple structures per protein (e.g., scPDB-based target databases); however, so far, those structures have not been statistically evaluated as one protein ensemble. Furthermore, such PDB-derived protein ensembles can only cover protein classes with high coverage. Methods describing binding site changes based on MD simulations, as described in TRAPP<sup>119</sup> for transient pockets, are already available but have not been integrated yet into a workflow for target prediction.

## DISCUSSION AND FUTURE DIRECTIONS

Since without sufficient data computational target prediction would not be possible at all, we first discuss the beauty and peril of current data sources. We then cover challenges in target ranking and method validation as well as directions on how to overcome them.

**Data.** Usage of *in silico* techniques for target prediction has been enabled in the first place by the rapidly increasing amount of *available structural, chemical, and biological data*. In this respect, the increasing availability of open access databases for drug discovery should be appreciated, with the PDB,<sup>75</sup> ChEMBL,<sup>18</sup> PubChem,<sup>19</sup> and DrugBank<sup>120</sup> databases being arguably the most well known. While the speed of computation has increased at a phenomenal rate with transistor counts roughly doubling every two years<sup>121</sup> (slowing down in recent years<sup>122</sup>), data availability and quality still form the bottleneck.<sup>20,123</sup> Given more data, more intricate methods can be applied, which should result in higher quality predictions.<sup>21</sup> This does not only concern bioactivity data but also structural information on proteins.<sup>75</sup>

In *ligand-based methods*, the large amount of available bioactivity data is used for model training. Lack of data here typically means that there are not enough experimentally derived activities of compounds for a given target. One way to overcome this is using computational target prediction to fill in the expected bioactivities for proteins that were not experimentally tested.<sup>54,124</sup> However, even if sufficient data is available, this does not directly mean the *data quality* is adequate. It has been shown that the experimental error in bioactivity databases can be substantial.<sup>33,125</sup> In public data, experimental activities are not derived following the same standard operating procedure or are even from the same lab or assay. This leads to a relatively large experimental error in the data (on average 0.47 log units for mixed p*K<sub>i</sub>* data),<sup>33</sup> which is reflected in the prediction accuracy of the models. Data quality and bias each determine the applicability domain of a model and should therefore be addressed early on by comparing the similarity between training and screening compounds. For instance, models trained on smaller or more hydrophobic molecules may not be able to make reliable predictions for larger or more hydrophilic compounds. Furthermore, high chemical similarity within the training set leads to a *bias toward a similar group of compounds*. Therefore, a wide diversity in chemical space is more favorable than a large compound set encompassing a congeneric series of ligands. Models trained only on close analogues cannot predict activities of very dissimilar compounds reliably. In summary, in order to build reliable models, important factors to check are the amount of data and heterogeneity (as discussed here), as well as the bias toward (in)actives (see [Pitfalls Defining an “Active” Class](#) section) and toward certain targets (see [Target Ranking](#) section).

*Structure-based methods* build on the structural arrangement of binding site atoms, experimentally derived from currently mostly X-ray crystallography. Such structural arrangements are (i) less reliable with decreasing resolution and (ii) represent only a static (and maybe even artificial) conformational state. The former is usually addressed with resolution thresholds (e.g., <3 Å in case of the scPDB), whereas the latter is sometimes considered with conformational sampling (see [Consideration of Target Flexibility in Structure-Based Methods](#) section). Furthermore, using structure-based meth-

ods, only targets with available structures can be queried, introducing a *bias toward structurally known targets*. Currently, most methods rely only on the available structures in the PDB. While there are over 140,000 protein structures deposited in the PDB (accessed in November 2018), they only cover at most 30% of the human proteome and 50% of known human drug targets,<sup>126</sup> with protein classes being differently well represented. Homology modeling is a possibility to infer lacking information from determined structures of homologous proteins. Somody et al.<sup>126</sup> have shown that given a sequence identity of  $\geq 30\%$  (as generally accepted lower limit for homology modeling) the structural coverage of the modeled human proteome could approach 70% (that of known human drug targets 95%). While large scale homology models have been used, e.g., for kinome-wide druggability predictions,<sup>127</sup> they have not been widely used yet for target prediction. It should be noted that the higher the sequence identity is, the more reliable the homology models are for structural modeling purposes. Furthermore, target-focused methods such as inverse screening and binding site comparison only require 3D target structures and binding site locations, whereas interaction-focused methods require ligand–target complex information, limiting their applicability. To overcome this, such interactions can be predicted: For instance, interaction fingerprint comparison can be coupled with inverse docking, and reverse pharmacophore screening can be based on target-focused pharmacophore methods such as T<sup>2</sup>F-Pharm<sup>128</sup> that generate pharmacophores from apo-structures. However, it is important to note that such ligand- as well as structure-based models-based-on-models approaches may introduce noise to the predictions.

**Target Ranking.** Results from computational target prediction are highly dependent on the scoring function(s) used for target ranking. If two objects of the same type—for example, two small molecules or two protein binding sites—are compared, similarity of the query to the database can directly be inferred from the commonalities or mutual overlap between the objects and ranked accordingly. In contrast, if the objects to be compared are of different types, target ranking becomes more complex. For example, this is the case when the most likely targets are predicted for a small molecule based on individual machine learning models per target (ligand-based methods) or based on inverse screening against a target database (structure-based methods). While it is already challenging to predict the correct activity or binding energy of a ligand against one target, in panel predictions, the ligand is scored individually against multiple targets, requiring inter-target ranking. This is especially ambitious since the predictions are influenced by different forms of bias present in the data. Typically, some protein classes (e.g., kinases or G protein-coupled receptors) have been very well explored, whereas others have been explored less thoroughly (e.g., transporters). This means that more ligands are known for these proteins (ligand-based methods) or more structures have been elucidated (structure-based methods). Thus, the chemical or structural space is better covered, and they might score better compared to less explored chemical or structural spaces. Another form of bias influencing target ranking can be the average molecular weight of ligands for certain protein classes. For example, the molecular weight of class B GPCRs is much higher than that of other proteins such as kinases. The higher molecular weight leads to the presence of more chemical

substructures in the fingerprint vector and can increase the amount of predicted targets for these ligands.<sup>58</sup>

In an effort to reduce the effect of these biases on ligand-based prediction probability, raw probabilities can be converted to a z-score.<sup>53</sup> In this method, for all molecules in the training set, a prediction score is obtained for all proteins in the training set. Subsequently, for each protein, a mean probability and standard deviation of this probability can be derived and converted into a z-score. By applying the same z-scoring for novel compounds rather than the raw probability, the predictions are converted to a number of standard deviations over or under the mean for that particular protein. This method has been shown to be more robust than using the raw probability.<sup>58</sup> Similarly, in structure-based inverse screening, the interaction score of the ligand with each target is compared with the interaction score distribution from a set of reference ligands of the respective target complex structures, taken from X-ray structures or determined by docking.<sup>81,83,84</sup>

**Validation Strategies.** The performance of *ligand-based models* should always be estimated using external test sets to minimize overfitting (besides cross-validation). If test sets are composed randomly, this may lead to overoptimistic performance values as similar ligands may be present in both training and test sets, resulting in “easy” predictions. In order to overcome this effect, cluster splits, where the whole cluster of similar molecules is either contained in the test or training set, or temporal splits, where data from the most recent years is used for testing, can be applied.<sup>129</sup> Predictive performances of ligand-based models can be estimated by metrics such as  $R^2$  and  $Q^2$  as well as error-based metrics such as the root-mean-square error (RMSE) and mean absolute error (MAE). It is debatable what the best metric is to indicate model performance as this is dependent on the data and validation method. Generally, performance can be better estimated when multiple metrics are considered.<sup>130</sup>

Evaluating the performance of *structure-based methods* is based on diverse strategies. Binding site comparison methods, for instance, often screen a query target against a set of true (well-studied protein class with subclass classification) and decoy targets, whereas inverse screening methods often test only one or few query ligands in a set of true (known targets of the ligand) and decoy targets. Evaluation metrics are, for instance, the percentage of true targets in the top x% of the ranked hit list, the so-called enrichment factor (EF), and the area under the curve (AUC). While different sizes and compositions of benchmark data sets and the diverse use of performance metrics hamper a direct comparison between methods, efforts to unify benchmarking have been made. Since binding site comparison is a long-established approach with many published methods, proposed data sets have often been reused. Such an example is the data set compilation by Weill and Rognan,<sup>99</sup> encompassing a set of similar and dissimilar structure pairs as well as sets focused on kinases and serine endopeptidases (all scPDB-based). Also concentrating on similar and dissimilar pairs, Ehart et al.<sup>131</sup> have recently proposed a collection of new and reused data sets (ProSPECCTs) to test different performance aspects, which the authors applied to multiple binding site methods to establish guidelines for their application scope. For inverse screening methods, Schomburg et al.<sup>83</sup> proposed two data sets together with evaluation strategies: a small data set consisting of three target classes for detailed proof-of-concept and selectivity studies and a large data set with about 8000 protein

structures and over 70 drug-like ligands. In addition to the widely used EF and AUC, the authors propose performance metrics capable of measuring the early enrichments, i.e., BEDROC (Boltzman-enhanced discrimination of ROC) and NSLR (normalized sum of logarithmic ranks).

## CONCLUSION

Drug target identification is one of the most important, but also most complex, aspects of preclinical drug development. In this respect, computational target prediction is a highly valuable tool to identify the most probable targets for a compound under investigation. Such tools can guide wet lab experiments by suggesting potential targets for orphan compounds, supply tool compounds for functional analyses of poorly understood proteins, and thus help to decipher the mode-of-action of a protein under investigation. Furthermore, desired as well as undesired multitarget drug effects can be rationalized by computational (off-)target predictions, and known drugs can potentially be repositioned based on these forecasts.

Computational target prediction methods rely on the general assumption that similar molecules/structures will have similar interactions or interaction patterns. Exceptions are so-called activity cliffs, describing that small changes can cause large differences in activity.<sup>29</sup> Depending on the research question and the data available, ligand- or structure-based target prediction methods can be applied. In ligand-based methods, potential targets can either be inferred from the most similar known ligands or through elaborated machine learning models. The latter require sufficient and well annotated data in order to train proper models. Structure-based approaches compare a query protein based on their binding sites or interaction fingerprints to a panel of protein structures or screen a query compound against these panels using a docking or pharmacophore screening engine. It should be noted that usually ligand-centric methods are faster than structure-centric methods, especially when structural alignment or pose prediction is evoked. The former provides more quantitative information such as predicted bioactivities that can directly be associated with experimental values, whereas the latter can give additional information about the binding pose of ligands to potential targets. It should be noted that most methods do not consider alternate binding pockets on a single protein or the effect of protein complex formation. Although protein function or (de)activation through allosteric modulation can occur, most target prediction methods are based on the assumption that all ligands are orthosteric binders.

In our opinion, future progress needs to promote data coverage from both the ligand and protein point of view, e.g., annotation of non-biased bioactivities (reporting inactives) and deposition of novel structures or the same protein structures, but with different ligands to provide a better view on the dynamics of the ligand binding site (high-throughput crystallization). Furthermore, protein flexibility modeling and inter-target ranking are equally important matters to address. Moreover, new methods should be evaluated on standardized benchmarking data sets and performance metrics, as well as made accessible to the community in order to improve predictability, reliability, and reproducibility. Finally, holistic approaches should and will gain momentum, integrating multiple types of data, e.g., coupling chemical and structural space with information on the proteome level and pathways, linking cellular and molecular scales.

## AUTHOR INFORMATION

### Corresponding Authors

\*E-mail: [gerard@lacdr.leidenuniv.nl](mailto:gerard@lacdr.leidenuniv.nl) (G.J.P. van Westen).

\*E-mail: [andrea.volkamer@charite.de](mailto:andrea.volkamer@charite.de) (A. Volkamer).

### ORCID

Dominique Sydow: 0000-0003-4205-8705

Lindsey Burggraaff: 0000-0002-2442-0443

Herman W. T. van Vlijmen: 0000-0002-1915-3141

Adriaan P. IJzerman: 0000-0002-1182-2259

Gerard J. P. van Westen: 0000-0003-0717-1817

Andrea Volkamer: 0000-0002-3760-580X

### Author Contributions

<sup>||</sup>D. Sydow and L. Burggraaff have shared cofirst authorship.

### Notes

The authors declare no competing financial interest.

## ACKNOWLEDGMENTS

G.J.P. van Westen thanks the Dutch Scientific Council (NWO) and Stichting Technologie Wetenschappen (STW) for funding (VENI Grant 14410). A. Volkamer and D. Sydow thank the Deutsche Forschungsgemeinschaft (DFG, grant VO 2353/1-1) and the Bundesministerium für Bildung und Forschung (BMBF, grant 031A262C) for funding.

## REFERENCES

- Jenkins, J. L.; Bender, A.; Davies, J. W. In Silico Target Fishing: Predicting Biological Targets from Chemical Structure. *Drug Discovery Today: Technol.* **2006**, *3*, 413–421.
- Hart, C. P. Finding the Target After Screening the Phenotype. *Drug Discovery Today* **2005**, *10*, 513–519.
- Lee, J.; Bogyo, M. Target Deconvolution Techniques in Modern Phenotypic Profiling. *Curr. Opin. Chem. Biol.* **2013**, *17*, 118–126.
- Niphakis, M. J.; Cravatt, B. F. Enzyme Inhibitor Discovery by Activity-Based Protein Profiling. *Annu. Rev. Biochem.* **2014**, *83*, 341–377.
- Rognan, D. Structure-Based Approaches to Target Fishing and Ligand Profiling. *Mol. Inf.* **2010**, *29*, 176–187.
- Sliwoski, G.; Kothiwale, S.; Meiler, J.; Lowe, E. W. Computational Methods in Drug Discovery. *Pharmacol. Rev.* **2014**, *66*, 334–395.
- Bender, A.; Glen, R. C. Molecular Similarity: A Key Technique in Molecular Informatics. *Org. Biomol. Chem.* **2004**, *2*, 3204.
- Anighoro, A.; Bajorath, J.; Rastelli, G. Polypharmacology: Challenges and Opportunities in Drug Discovery. *J. Med. Chem.* **2014**, *57*, 7874–7887.
- Morphy, R.; Kay, C.; Rankovic, Z. From Magic Bullets to Designed Multiple Ligands. *Drug Discovery Today* **2004**, *9*, 641–651.
- AbdulHameed, M. D. M.; Chaudhury, S.; Singh, N.; Sun, H.; Wallqvist, A.; Tawa, G. J. Exploring Polypharmacology Using a ROCs-Based Target Fishing Approach. *J. Chem. Inf. Model.* **2012**, *52*, 492–505.
- Bender, A.; Scheiber, J.; Glick, M.; Davies, J.; Azzaoui, K.; Hamon, J.; Urban, L.; Whitebread, S.; Jenkins, J. Analysis of Pharmacology Data and the Prediction of Adverse Drug Reactions and Off-Target Effects from Chemical Structure. *ChemMedChem* **2007**, *2*, 861–873.
- Oprea, T. I.; et al. Drug Repurposing from an Academic Perspective. *Drug Discovery Today: Ther. Strategies* **2011**, *8*, 61–69.
- Keiser, M. J.; et al. Predicting New Molecular Targets for Known Drugs. *Nature* **2009**, *462*, 175–181.
- Ashburn, T. T.; Thor, K. B. Drug Repositioning: Identifying and Developing New Uses for Existing Drugs. *Nat. Rev. Drug Discovery* **2004**, *3*, 673–683.
- Berger, A. B.; Vitorino, P. M.; Bogyo, M. Activity-Based Protein Profiling: Applications to Biomarker Discovery, in Vivo Imaging and Drug Discovery. *Am. J. Pharmacogenomics* **2004**, *4*, 371–381.
- Schirle, M.; Bantscheff, M.; Kuster, B. Mass Spectrometry-Based Proteomics in Preclinical Drug Discovery. *Chem. Biol.* **2012**, *19*, 72–84.
- van Esbroeck, A. C. M.; et al. Activity-Based Protein Profiling Reveals Off-Target Proteins of the FAAH Inhibitor Bla 10–2474. *Science* **2017**, *356*, 1084–1087.
- Gaulton, A.; Bellis, L. J.; Bento, A. P.; Chambers, J.; Davies, M.; Hersey, A.; Light, Y.; McGlinchey, S.; Michalovich, D.; Al-Lazikani, B.; Overington, J. P. ChEMBL: A Large-Scale Bioactivity Database for Drug Discovery. *Nucleic Acids Res.* **2012**, *40*, D1100–D1107.
- Kim, S.; Thiessen, P. A.; Bolton, E. E.; Chen, J.; Fu, G.; Gindulyte, A.; Han, L.; He, J.; He, S.; Shoemaker, B. A.; Wang, J.; Yu, B.; Zhang, J.; Bryant, S. H. PubChem Substance and Compound Databases. *Nucleic Acids Res.* **2016**, *44*, D1202–D1213.
- Sun, J.; Jeliaskova, N.; Chupakhin, V.; Golib-Dzib, J.-F.; Engkvist, O.; Carlsson, L.; Wegner, J.; Ceulemans, H.; Georgiev, I.; Jeliaskov, V.; Kochev, N.; Ashby, T. J.; Chen, H. ExCAPE-DB: An Integrated Large Scale Dataset Facilitating Big Data Analysis in Chemogenomics. *J. Cheminf.* **2017**, *9*, 17.
- Rognan, D. Chemogenomic Approaches to Rational Drug Design. *Br. J. Pharmacol.* **2007**, *152*, 38–52.
- Weininger, D. SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules. *J. Chem. Inf. Model.* **1988**, *28*, 31–36.
- Cereto-Massagué, A.; Ojeda, M. J.; Valls, C.; Mulero, M.; Garcia-Vallvé, S.; Pujadas, G. Molecular Fingerprint Similarity Search in Virtual Screening. *Methods* **2015**, *71*, 58–63.
- Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742–754.
- Hawkins, P. C. D.; Stahl, G. In *Computational Methods for GPCR Drug Discovery*; Heifetz, A., Ed.; Springer: New York, 2018; pp 365–374.
- Shin, W.-H.; Zhu, X.; Bures, G. M.; Kihara, D. Three-Dimensional Compound Comparison Methods and Their Application in Drug Discovery. *Molecules* **2015**, *20*, 12841–62.
- Bender, A.; Mussa, H. Y.; Glen, R. C.; Reiling, S. Similarity Searching of Chemical Databases Using Atom Environment Descriptors (MOLPRINT 2D): Evaluation of Performance. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1708–1718.
- Hu, Y.; Bajorath, J. High-Resolution View of Compound Promiscuity. *F1000Research* **2013**, *2*, 144.
- Bajorath, J. Representation and Identification of Activity Cliffs. *Expert Opin. Drug Discovery* **2017**, *12*, 879–883.
- Subramanian, A.; et al. A Next Generation Connectivity Map: L1000 Platform and the First 1,000,000 Profiles. *Cell* **2017**, *171*, 1437–1452.
- De Wolf, H.; Cougnaud, L.; Van Hoorde, K.; De Bondt, A.; Wegner, J. K.; Ceulemans, H.; Göhlmann, H. High-Throughput Gene Expression Profiles to Define Drug Similarity and Predict Compound Activity. *Assay Drug Dev. Technol.* **2018**, *16*, 162–176.
- Simm, J.; et al. Repurposing High-Throughput Image Assays Enables Biological Activity Prediction for Drug Discovery. *Cell Chem. Biol.* **2018**, *25*, 611–618.
- Kalliokoski, T.; Kramer, C.; Vulpetti, A.; Gedeck, P. Comparability of Mixed IC50 Data - a Statistical Analysis. *PLoS One* **2013**, *8*, No. e61007.
- Dalke, A. The FPS Fingerprint Format and Chemfp Toolkit. *J. Cheminf.* **2013**, *5*, P36.
- Gfeller, D.; Grosdidier, A.; Wirth, M.; Daina, A.; Michielin, O.; Zoete, V. SwissTargetPrediction: A Web Server for Target Prediction of Bioactive Small Molecules. *Nucleic Acids Res.* **2014**, *42*, W32–W38.
- Nickel, J.; Gohlke, B.-O.; Ereman, J.; Banerjee, P.; Rong, W. W.; Goede, A.; Dunkel, M.; Preissner, R. SuperPred: Update on Drug Classification and Target Prediction. *Nucleic Acids Res.* **2014**, *42*, W26–W31.



- (37) Kauvar, L. M. Affinity Fingerprinting. *Nat. Biotechnol.* **1995**, *13*, 965–966.
- (38) Cheng, T.; Li, Q.; Wang, Y.; Bryant, S. H. Identifying Compound-Target Associations by Combining Bioactivity Profile Similarity Search and Public Databases Mining. *J. Chem. Inf. Model.* **2011**, *51*, 2440–2448.
- (39) Lamb, J.; et al. The Connectivity Map: Using Gene-Expression Signatures to Connect Small Molecules, Genes, and Disease. *Science* **2006**, *313*, 1929–1935.
- (40) Keiser, M. J.; Roth, B. L.; Armbruster, B. N.; Ernsberger, P.; Irwin, J. J.; Shoichet, B. K. Relating Protein Pharmacology by Ligand Chemistry. *Nat. Biotechnol.* **2007**, *25*, 197.
- (41) Altschul, S. F.; Gish, W.; Miller, W.; Myers, E. W.; Lipman, D. J. Basic Local Alignment Search Tool. *J. Mol. Biol.* **1990**, *215*, 403–410.
- (42) Altschul, S. F.; Madden, T. L.; Schäffer, A. A.; Zhang, J.; Zhang, Z.; Miller, W.; Lipman, D. J. Gapped BLAST and PSI-BLAST: A New Generation of Protein Database Search Programs. *Nucleic Acids Res.* **1997**, *25*, 3389–3402.
- (43) Pearson, W. R. Empirical Statistical Estimates for Sequence Similarity Searches. *J. Mol. Biol.* **1998**, *276*, 71–84.
- (44) Lounkine, E.; Keiser, M. J.; Whitebread, S.; Mikhailov, D.; Hamon, J.; Jenkins, J. L.; Lavan, P.; Weber, E.; Doak, A. K.; Côté, S.; Shoichet, B. K.; Urban, L. Large-Scale Prediction and Testing of Drug Activity on Side-Effect Targets. *Nature* **2012**, *486*, 361.
- (45) Czodrowski, P.; Bolick, W.-G. OCEAN: Optimized Cross REActivity Estimation. *J. Chem. Inf. Model.* **2016**, *56*, 2013–2023.
- (46) Cortes Cabrera, A.; Lucena-Agell, D.; Redondo-Horcajo, M.; Barasoain, I.; Diaz, J. F.; Fasching, B.; Petrone, P. M. Aggregated Compound Biological Signatures Facilitate Phenotypic Drug Discovery and Target Elucidation. *ACS Chem. Biol.* **2016**, *11*, 3024–3034.
- (47) Grover, A.; Grover, M.; Sharma, K. A Practical Overview of Quantitative Structure-Activity Relationship. *World J. Pharm. Pharm. Sci.* **2016**, *5*, 427–437.
- (48) Svetnik, V.; Liaw, A.; Tong, C.; Culberson, J. C.; Sheridan, R. P.; Feuston, B. P. Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 1947–1958.
- (49) Burbidge, R.; Trotter, M.; Buxton, B.; Holden, S. Drug Design by Machine Learning: Support Vector Machines for Pharmaceutical Data Analysis. *Comput. Chem.* **2001**, *26*, 5–14.
- (50) Xia, X.; Maliski, E. G.; Gallant, P.; Rogers, D. Classification of Kinase Inhibitors Using a Bayesian Model. *J. Med. Chem.* **2004**, *47*, 4463–4470.
- (51) Wu, Z.; Ramsundar, B.; Feinberg, E.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; Pande, V. MoleculeNet: A Benchmark for Molecular Machine Learning. *Chem. Sci.* **2018**, *9*, 513–530.
- (52) Winter, R.; Montanari, F.; Noé, F.; Clevert, D.-A. Learning Continuous and Data-Driven Molecular Descriptors by Translating Equivalent Chemical Representations. *Chemical Science* **2019**, *10*, 1692.
- (53) Mugumbate, G.; Abrahams, K. A.; Cox, J. A. G.; Papadatos, G.; van Westen, G.; Lelièvre, J.; Calus, S. T.; Loman, N. J.; Ballell, L.; Barros, D.; Overington, J. P.; Besra, G. S. Mycobacterial Dihydrofolate Reductase Inhibitors Identified Using Chemogenomic Methods and in Vitro Validation. *PLoS One* **2015**, *10*, No. e0121492.
- (54) Lagunin, A.; Stepanchikova, A.; Filimonov, D.; Poroikov, V. PASS: Prediction of Activity Spectra for Biologically Active Substances. *Bioinformatics* **2000**, *16*, 747–748.
- (55) Papadatos, G.; Gaulton, A.; Hersey, A.; Overington, J. P. Activity, Assay and Target Data Curation and Quality in the ChEMBL Database. *J. Comput.-Aided Mol. Des.* **2015**, *29*, 885–896.
- (56) Afzal, A. M.; Mussa, H. Y.; Turner, R. E.; Bender, A.; Glen, R. C. A Multi-Label Approach to Target Prediction Taking Ligand Promiscuity into Account. *J. Cheminf.* **2015**, *7*, 24.
- (57) Anger, L. T.; Wolf, A.; Schleifer, K.-J.; Schrenk, D.; Rohrer, S. G. Generalized Workflow for Generating Highly Predictive in Silico Off-Target Activity Models. *J. Chem. Inf. Model.* **2014**, *54*, 2411–2422.
- (58) Lenselink, E. B.; ten Dijke, N.; Bongers, B.; Papadatos, G.; van Vlijmen, H. W. T.; Kowalczyk, W.; IJzerman, A. P.; van Westen, G. J. P. Beyond the Hype: Deep Neural Networks Outperform Established Methods Using a ChEMBL Bioactivity Benchmark Set. *J. Cheminf.* **2017**, *9*, 45.
- (59) Zhang, Q.; Hughes-Oliver, J. M.; Ng, R. T. A Model-Based Ensembling Approach for Developing QSARs. *J. Chem. Inf. Model.* **2009**, *49*, 1857–1865.
- (60) van Westen, G. J. P.; Wegner, J. K.; IJzerman, A. P.; van Vlijmen, H. W. T.; Bender, A. Proteochemometric Modeling as a Tool to Design Selective Compounds and for Extrapolating to Novel Targets. *MedChemComm* **2011**, *2*, 16–30.
- (61) Sandberg, M.; Eriksson, L.; Jonsson, J.; Sjöström, M.; Wold, S. New Chemical Descriptors Relevant for the Design of Biologically Active Peptides. a Multivariate Characterization of 87 Amino Acids. *J. Med. Chem.* **1998**, *41*, 2481–2491.
- (62) van Westen, G. J. P.; Swier, R. F.; Cortes-Ciriano, I.; Wegner, J. K.; Overington, J. P.; IJzerman, A. P.; van Vlijmen, H. W. T.; Bender, A. Benchmarking of Protein Descriptor Sets in Proteochemometric Modeling (part 2): Modeling Performance of 13 Amino Acid Descriptor Sets. *J. Cheminf.* **2013**, *5*, 42.
- (63) van Westen, G. J. P.; van den Hoven, O. O.; van der Pijl, R.; Mulder-Krieger, T.; de Vries, H.; Wegner, J. K.; IJzerman, A. P.; van Vlijmen, H. W. T.; Bender, A. Identifying Novel Adenosine Receptor Ligands by Simultaneous Proteochemometric Modeling of Rat and Human Bioactivity Data. *J. Med. Chem.* **2012**, *55*, 7010–7020.
- (64) van Westen, G. J. P.; Wegner, J. K.; Geluykens, P.; Kwanten, L.; Vereycken, I.; Peeters, A.; IJzerman, A. P.; van Vlijmen, H. W. T.; Bender, A. Which Compound to Select in Lead Optimization? Prospectively Validated Proteochemometric Models Guide Preclinical Development. *PLoS One* **2011**, *6*, No. e27518.
- (65) Mayr, A.; Klambauer, G.; Unterthiner, T.; Steijaert, M.; Wegner, J. K.; Ceulemans, H.; Clevert, D.-A.; Hochreiter, S. Large-Scale Comparison of Machine Learning Methods for Drug Target Prediction on ChEMBL. *Chem. Sci.* **2018**, *9*, 5441–5451.
- (66) Oprea, T. I.; Nielsen, S. K.; Ursu, O.; Yang, J. J.; Taboureau, O.; Mathias, S. L.; Kouskoumvekaki, I.; Sklar, L. A.; Bologa, C. G. Associating Drugs, Targets and Clinical Outcomes into an Integrated Network Affords a New Platform for Computer-Aided Drug Repurposing. *Mol. Inf.* **2011**, *30*, 100–111.
- (67) Lo, Y.-C.; Senese, S.; Damoiseaux, R.; Torres, J. Z. 3D Chemical Similarity Networks for Structure-Based Target Prediction and Scaffold Hopping. *ACS Chem. Biol.* **2016**, *11*, 2244–2253.
- (68) Yamanishi, Y.; Kotera, M.; Moriya, Y.; Sawada, R.; Kanehisa, M.; Goto, S. DINIES: Drug-Target Interaction Network Inference Engine Based on Supervised Analysis. *Nucleic Acids Res.* **2014**, *42*, W39–W45.
- (69) Zhao, S.; Li, S. Network-Based Relating Pharmacological and Genomic Spaces for Drug Target Identification. *PLoS One* **2010**, *5*, No. e11764.
- (70) Chen, X.; Liu, M.-X.; Yan, G.-Y. Drug-Target Interaction Prediction by Random Walk on the Heterogeneous Network. *Mol. BioSyst.* **2012**, *8*, 1970–1978.
- (71) Yamanishi, Y.; Araki, M.; Gutteridge, A.; Honda, W.; Kanehisa, M. Prediction of Drug-Target Interaction Networks from the Integration of Chemical and Genomic Spaces. *Bioinformatics* **2008**, *24*, i232–i240.
- (72) OpenEye Scientific Software, ROCS. <https://www.eyesopen.com/rocs> (accessed on 2018–11–01).
- (73) Rarey, M.; Dixon, J. S. Feature Trees: A New Molecular Similarity Measure Based on Tree Matching. *J. Comput.-Aided Mol. Des.* **1998**, *12*, 471–490.
- (74) Mervin, L. H.; Afzal, A. M.; Drakakis, G.; Lewis, R.; Engkvist, O.; Bender, A. Target Prediction Utilising Negative Bioactivity Data Covering Large Chemical Space. *J. Cheminf.* **2015**, *7*, 51.
- (75) Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. The Protein Data Bank. *Nucleic Acids Res.* **2000**, *28*, 235–242.

- (76) Volkamer, A.; von Behren, M. M.; Bietz, S.; Rarey, M. *Applied Cheminformatics*; Wiley-VCH Verlag GmbH & Co. KGaA: Weinheim, Germany, 2018; pp 283–311.
- (77) Desaphy, J.; Bret, G.; Rognan, D.; Kellenberger, E. sc-PDB: A 3D-Database of Ligandable Binding Sites-10 Years On. *Nucleic Acids Res.* **2015**, *43*, D399–D404.
- (78) Xiu, X.; Huang, M.; Zou, X. Docking-Based Inverse Virtual Screening: Methods, Applications, and Challenges. *Biophys. Rep.* **2018**, *4*, 1–16.
- (79) Huang, H.; Zhang, G.; Zhou, Y.; Lin, C.; Chen, S.; Lin, Y.; Mai, S.; Huang, Z. Reverse Screening Methods to Search for the Protein Targets of Chemopreventive Compounds. *Front. Chem.* **2018**, *6*, 138.
- (80) Li, H.; Gao, Z.; Kang, L.; Zhang, H.; Yang, K.; Yu, K.; Luo, X.; Zhu, W.; Chen, K.; Shen, J.; Wang, X.; Jiang, H. TarFisDock: A Web Server for Identifying Drug Targets with Docking Approach. *Nucleic Acids Res.* **2006**, *34*, W219–W224.
- (81) Wang, J.-C.; Chu, P.-Y.; Chen, C.-M.; Lin, J.-H. idTarget: A Web Server for Identifying Protein Targets of Small Chemical Molecules with Robust Scoring Functions and a Divide-And-Conquer Docking Approach. *Nucleic Acids Res.* **2012**, *40*, W393–W399.
- (82) Chen, Y. Z.; Zhi, D. G. Ligand-Protein Inverse Docking and Its Potential Use in the Computer Search of Protein Targets of a Small Molecule. *Proteins: Struct., Funct., Genet.* **2001**, *43*, 217–226.
- (83) Schomburg, K. T.; Bietz, S.; Briem, H.; Henzler, A. M.; Urbaczek, S.; Rarey, M. Facing the Challenges of Structure-Based Target Prediction by Inverse Virtual Screening. *J. Chem. Inf. Model.* **2014**, *54*, 1676–1686.
- (84) Santiago, D. N.; Pevzner, Y.; Durand, A. A.; Tran, M.; Scheerer, R. R.; Daniel, K.; Sung, S.-S.; Lee Woodcock, H.; Guida, W. C.; Brooks, W. H. Virtual Target Screening: Validation Using Kinase Inhibitors. *J. Chem. Inf. Model.* **2012**, *52*, 2192–2203.
- (85) Ellingson, S. R.; Smith, J. C.; Baudry, J. VinaMPI: Facilitating Multiple Receptor High-Throughput Virtual Docking on High-Performance Computers. *J. Comput. Chem.* **2013**, *34*, 2212–2221.
- (86) Li, G.-B.; Yu, Z.-J.; Liu, S.; Huang, L.-Y.; Yang, L.-L.; Lohans, C. T.; Yang, S.-Y. IFPTarget: A Customized Virtual Target Identification Method Based on Protein-Ligand Interaction Fingerprinting Analyses. *J. Chem. Inf. Model.* **2017**, *57*, 1640–1651.
- (87) Scafuri, B.; Marabotti, A.; Carbone, V.; Minasi, P.; Dotolo, S.; Facchiano, A. A Theoretical Study on Predicted Protein Targets of Apple Polyphenols and Possible Mechanisms of Chemoprevention in Colorectal Cancer. *Sci. Rep.* **2016**, *6*, 32516.
- (88) Wermuth, C. G.; Ganellin, C. R.; Lindberg, P.; Mitscher, L. A. Glossary of Terms Used in Medicinal Chemistry (IUPAC Recommendations 1998). *Pure Appl. Chem.* **1998**, *70*, 1129–1143.
- (89) Schuster, D. 3D Pharmacophores As Tools for Activity Profiling. *Drug Discovery Today: Technol.* **2010**, *7*, e205–e211.
- (90) Wang, X.; Shen, Y.; Wang, S.; Li, S.; Zhang, W.; Liu, X.; Lai, L.; Pei, J.; Li, H. PharmMapper 2017 Update: A Web Server for Potential Drug Target Identification with a Comprehensive Target Pharmacophore Database. *Nucleic Acids Res.* **2017**, *45*, W356–W360.
- (91) Liu, B.; Fu, X.-Q.; Li, T.; Su, T.; Guo, H.; Zhu, P.-L.; Tse, A. K.-W.; Liu, S.-M.; Yu, Z.-L. Computational and Experimental Prediction of Molecules Involved in the Anti-Melanoma Action of Berberine. *J. Ethnopharmacol.* **2017**, *208*, 225–235.
- (92) Kellenberger, E.; Schalon, C.; Rognan, D. How to Measure the Similarity Between Protein Ligand-Binding Sites? *Curr. Comput.-Aided Drug Des.* **2008**, *4*, 209–220.
- (93) Ehrh, C.; Brinkjost, T.; Koch, O. Impact of Binding Site Comparisons on Medicinal Chemistry and Rational Molecular Design. *J. Med. Chem.* **2016**, *59*, 4121–4151.
- (94) Schmitt, S.; Kuhn, D.; Klebe, G. A New Method to Detect Related Function Among Proteins Independent of Sequence and Fold Homology. *J. Mol. Biol.* **2002**, *323*, 387–406.
- (95) Jambon, M.; Imberty, A.; Deléage, G.; Geourjon, C. A New Bioinformatic Approach to Detect Common 3D Sites in Protein Structures. *Proteins: Struct., Funct., Genet.* **2003**, *52*, 137–145.
- (96) Moriaud, F.; Richard, S. B.; Adcock, S. A.; Chanas-Martin, L.; Surgand, J.-S.; Ben Jelloul, M.; Delfaud, F. Identify Drug Repurposing Candidates by Mining the Protein Data Bank. *Briefings Bioinf.* **2011**, *12*, 336–340.
- (97) Konc, J.; Janežič, D. ProBiS Algorithm for Detection of Structurally Similar Protein Binding Sites by Local Structural Alignment. *Bioinformatics* **2010**, *26*, 1160–1168.
- (98) Desaphy, J.; Azdimousa, K.; Kellenberger, E.; Rognan, D. Comparison and Druggability Prediction of Protein-Ligand Binding Sites from Pharmacophore-Annotated Cavity Shapes. *J. Chem. Inf. Model.* **2012**, *52*, 2287–2299.
- (99) Weill, N.; Rognan, D. Alignment-Free Ultra-High-Throughput Comparison of Druggable Protein-Ligand Binding Sites. *J. Chem. Inf. Model.* **2010**, *50*, 123–135.
- (100) Sciabola, S.; Stanton, R. V.; Mills, J. E.; Flocco, M. M.; Baroni, M.; Cruciani, G.; Perruccio, F.; Mason, J. S. High-Throughput Virtual Screening of Proteins Using GRID Molecular Interaction Fields. *J. Chem. Inf. Model.* **2010**, *50*, 155–169.
- (101) Shulman-Peleg, A.; Nussinov, R.; Wolfson, H. J. Recognition of Functional Sites in Protein Structures. *J. Mol. Biol.* **2004**, *339*, 607–633.
- (102) von Behren, M. M.; Volkamer, A.; Henzler, A. M.; Schomburg, K. T.; Urbaczek, S.; Rarey, M. Fast Protein Binding Site Comparison Via an Index-Based Screening Technology. *J. Chem. Inf. Model.* **2013**, *53*, 411–422.
- (103) Krotzky, T.; Grunwald, C.; Egerland, U.; Klebe, G. Large-Scale Mining for Similar Protein Binding Pockets: With RAPMAD Retrieval on the Fly Becomes Real. *J. Chem. Inf. Model.* **2015**, *55*, 165–179.
- (104) Brakoulias, A.; Jackson, R. M. Towards a Structural Classification of Phosphate Binding Sites in Protein-Nucleotide Complexes: An Automated All-Against-All Structural Comparison Using Geometric Matching. *Proteins: Struct., Funct., Genet.* **2004**, *56*, 250–260.
- (105) Aung, Z.; Tong, J. C. BSAAlign: A Rapid Graph-Based Algorithm for Detecting Ligand-Binding Sites in Protein Structures. *Genome Inform* **2008**, *21*, 65–76.
- (106) Schalon, C.; Surgand, J.-S.; Kellenberger, E.; Rognan, D. A Simple and Fuzzy Method to Align and Compare Druggable Ligand-Binding Sites. *Proteins: Struct., Funct., Genet.* **2008**, *71*, 1755–1778.
- (107) De Franchi, E.; Schalon, C.; Messa, M.; Onofri, F.; Benfenati, F.; Rognan, D. Binding of Protein Kinase Inhibitors to Synapsin I Inferred from Pair-Wise Binding Site Similarity Measurements. *PLoS One* **2010**, *5*, No. e12214.
- (108) Salentin, S.; Haupt, V. J.; Daminelli, S.; Schroeder, M. Polypharmacology Rescored: Protein-Ligand Interaction Profiles for Remote Binding Site Similarity Assessment. *Prog. Biophys. Mol. Biol.* **2014**, *116*, 174–186.
- (109) Cao, R.; Wang, Y. Predicting Molecular Targets for Small-Molecule Drugs with a Ligand-Based Interaction Fingerprint Approach. *ChemMedChem* **2016**, *11*, 1352–1361.
- (110) Deng, Z.; Chuaqui, C.; Singh, J. Structural Interaction Fingerprint (SIFt): A Novel Method for Analyzing Three-Dimensional Protein-Ligand Binding Interactions. *J. Med. Chem.* **2004**, *47*, 337–344.
- (111) Desaphy, J.; Raimbaud, E.; Ducrot, P.; Rognan, D. Encoding Protein-Ligand Interaction Patterns in Fingerprints and Graphs. *J. Chem. Inf. Model.* **2013**, *53*, 623–637.
- (112) Da, C.; Kireev, D. Structural Protein-Ligand Interaction Fingerprints (SPLIF) for Structure-Based Virtual Screening: Method and Benchmark Study. *J. Chem. Inf. Model.* **2014**, *54*, 2555–2561.
- (113) Kinoshita, K.; Nakamura, H. Identification of Protein Biochemical Functions by Similarity Search Using the Molecular Surface Database EF-Site. *Protein Sci.* **2003**, *12*, 1589–1595.
- (114) Reisen, F.; Weisel, M.; Kriegl, J. M.; Schneider, G. Self-Organizing Fuzzy Graphs for Structure-Based Comparison of Protein Pockets. *J. Proteome Res.* **2010**, *9*, 6498–6510.
- (115) Yeturu, K.; Chandra, N. PocketMatch: A New Algorithm to Compare Binding Sites in Protein Structures. *BMC Bioinf.* **2008**, *9*, 543.

- (116) Wood, D. J.; de Vlieg, J.; Wagener, M.; Ritschel, T. Pharmacophore Fingerprint-Based Approach to Binding Site Sub-pocket Similarity and Its Application to Bioisostere Replacement. *J. Chem. Inf. Model.* **2012**, *52*, 2031–2043.
- (117) Liu, T.; Altman, R. B. Using Multiple Microenvironments to Find Similar Ligand-Binding Sites: Application to Kinase Inhibitor Binding. *PLoS Comput. Biol.* **2011**, *7*, No. e1002326.
- (118) Siragusa, L.; Cross, S.; Baroni, M.; Goracci, L.; Cruciani, G. BioGPS: Navigating Biological Space to Predict Polypharmacology, Off-Targeting, and Selectivity. *Proteins: Struct., Funct., Genet.* **2015**, *83*, 517–532.
- (119) Kokh, D. B.; Richter, S.; Henrich, S.; Czodrowski, P.; Rippmann, F.; Wade, R. C. TRAPP: A Tool for Analysis of Transient Binding Pockets in Proteins. *J. Chem. Inf. Model.* **2013**, *53*, 1235–1252.
- (120) Wishart, D. S.; Knox, C.; Guo, A. C.; Shrivastava, S.; Hassanali, M.; Stothard, P.; Chang, Z.; Woolsey, J. DrugBank: A Comprehensive Resource for in Silico Drug Discovery and Exploration. *Nucleic Acids Res.* **2006**, *34*, D668–D672.
- (121) Hilbert, M.; López, P. The World's Technological Capacity to Store, Communicate, and Compute Information. *Science* **2011**, *332*, 60–65.
- (122) Eeckhout, L. Is Moore's Law Slowing Down? What's Next? *IEEE Micro* **2017**, *37*, 4–5.
- (123) Jasial, S.; Hu, Y.; Bajorath, J. Assessing the Growth of Bioactive Compounds and Scaffolds over Time: Implications for Lead Discovery and Scaffold Hopping. *J. Chem. Inf. Model.* **2016**, *56*, 300–307.
- (124) Bender, A.; Jenkins, J. L.; Glick, M.; Deng, Z.; Nettles, J. H.; Davies, J. W. "Bayes Affinity Fingerprints" Improve Retrieval Rates in Virtual Screening and Define Orthogonal Bioactivity Space: When Are Multitarget Drugs a Feasible Concept? *J. Chem. Inf. Model.* **2006**, *46*, 2445–2456.
- (125) Tiikkainen, P.; Bellis, L.; Light, Y.; Franke, L. Estimating Error Rates in Bioactivity Databases. *J. Chem. Inf. Model.* **2013**, *53*, 2499–2505.
- (126) Somody, J. C.; MacKinnon, S. S.; Windemuth, A. Structural Coverage of the Proteome for Pharmaceutical Applications. *Drug Discovery Today* **2017**, *22*, 1792–1799.
- (127) Volkamer, A.; Eid, S.; Turk, S.; Jaeger, S.; Rippmann, F.; Fulle, S. Pocketome of Human Kinases: Prioritizing the ATP Binding Sites of (Yet) Untapped Protein Kinases for Drug Discovery. *J. Chem. Inf. Model.* **2015**, *55*, 538–549.
- (128) Mortier, J.; Dhakal, P.; Volkamer, A. Truly Target-Focused Pharmacophore Modeling: A Novel Tool for Mapping Intermolecular Surfaces. *Molecules* **2018**, *23*, 1959.
- (129) Sheridan, R. P. Time-Split Cross-Validation As a Method for Estimating the Goodness of Prospective Prediction. *J. Chem. Inf. Model.* **2013**, *53*, 783–790.
- (130) Gramatica, P.; Sangion, A. A Historical Excursus on the Statistical Validation Parameters for QSAR Models: A Clarification Concerning Metrics and Terminology. *J. Chem. Inf. Model.* **2016**, *56*, 1127–1131.
- (131) Ehrh, C.; Brinkjost, T.; Koch, O. A Benchmark Driven Guide to Binding Site Comparison: An Exhaustive Evaluation Using Tailor-Made Data Sets (ProSPECCTs). *PLoS Comput. Biol.* **2018**, *14*, No. e1006483.

### 1.2.2 Advances and Challenges in Computational Fragment-Based Drug Design

Fragment-based drug design (FBDD) has been successful in developing novel and selective compounds and is usually applied in early-stage drug discovery to identify and optimize active and promising molecules (hits and leads). Vemurafenib is an example of an FBDD-derived kinase inhibitor that is used in the clinic, which was approved by the FDA in 2011 to treat melanoma in patients who have the BRAF V600E kinase mutation [51].

Small molecules can be described as combinations of fragments, which are low-molecular-weight molecules of less than 300 Da. The small size of a fragment—typically a third of the size of a drug-like molecule—can explore the chemical space more efficiently and retrieves more protein binding information than traditional high-throughput screens (HTS) with small molecules. In more detail, FBDD has the following advantages [26]:

- Fragment libraries can be smaller in size than traditional compound libraries, which reduces the number of experiments per screen. Libraries of 1000 fragments with around 12 heavy atoms are estimated to have more complete coverage of the chemical space than a library of  $10^6 - 10^7$  lead- or drug-sized compounds.
- Progression into and through clinical trials is often at risk due to the drug candidate's molecular weight and lipophilicity properties. By definition, fragments have lower values for both properties, which offers opportunities for improved physicochemical properties during optimization.
- Fragments are weaker binders than small molecules; the dissociation constant  $K_d$  for molecules is 1 – 10  $\mu\text{M}$  while for fragments it is 100  $\mu\text{M}$  – 10 mM. Thus, fragments are more likely to bind to arbitrary targets, yielding higher hit rates, and to sample more binding patterns than traditional HTS campaigns.
- Fragments allow room for chemical novelty and their optimization can steer away from the congested intellectual property (IP) space.

Naturally, some of these advantages are accompanied by challenges, which can be met with additional experimental assessments and theoretical considerations as outlined in the following [26]:

- Fragments are weaker binders than molecules and therefore more elaborated methods are needed for their detection. False positives need to be identified early with orthogonal screening methods such as X-ray structure validation.
- Often fragment screens are run in parallel with traditional HTS campaigns, complicating a direct comparison of small fragment hits with less potency and larger hits with more potency. Solutions to this problem are measures that scale a compound's potency to its size, e.g., the ligand efficiency (LE) or the lipophilic ligand efficiency (LLE) [52].

$$\text{LE} = \text{pIC}_{50}(\text{or p}K_i) \div \text{number of heavy atoms}$$
$$\text{LLE} = \text{pIC}_{50}(\text{or p}K_i) - \text{cLogP (or LogD)}$$

- Initial weak fragment hits must be optimized by multiple orders of magnitude to reach a potency of  $K_d < 100$  nM. Planning such an optimization route is challenging and often requires integrating information about structural biology.

FBDD campaigns typically start with the screening of a fragment library to identify binders to specific regions in the target binding site. The design of such fragment libraries requires (i) the definition of the desired region of chemical space and (ii) the sampling of that region [53]. They are usually composed of chemical structures that adhere to the "*Rule of Three*" (*Ro3*), i.e., they have a molecular weight of  $< 300$  Da, fewer than three hydrogen bond donors and acceptors, fewer than three rotatable bonds, and a partition coefficient (cLogP) of  $\leq 3$ , while heavy atom counts tend to be restricted to  $< 20$  [54].

Hits from fragment library screens are optimized into larger compounds by fragment growing, merging, or linking [55]. *Fragment growing* is the most common approach in FBDD and describes the process of building sensible chemistry around (typically) a single fragment hit. The molecule is optimized in the context of structural binding site information to explore possible interaction profiles that show desired selectivity and drug-like properties. *Fragment merging* describes the merging of initially two overlapping reference molecules by substituting chemical moieties of one molecule with the core of the other. *Fragment linking* joins two molecules, that bind to different regions in the binding site, with a chemical spacer or linker.

Fragments can be generated computationally by decomposing larger compounds. The RECAP algorithm [56] automates fragmentation based on 11 distinct rules extracted from common chemical reactions, while the BRICS algorithm [57] extends the method to 19 rules including additions that incorporate more elaborate medicinal chemistry. The eMolFrag tool [58] works on top of BRICS to generate a set of (larger) "bricks" and (smaller) connecting linkers. Alternatively, the BREED algorithm [59] immediately produces recombined molecules for proteins with similar pockets such as kinases; two structures (and their co-crystallized ligands) are superimposed, and if two bonds of each ligand are close, the two ligands' overlaying fragments that are adjacent to these bonds are swapped. This produces two pocket-informed recombined molecules. In this thesis, we use the BRICS algorithm for kinase-focused fragment-based drug design and discuss the relevance of other computational fragmentation techniques in Section 3.2.

## 1.3 Protein Kinases

Most aspects of cellular life are regulated by activating and deactivating enzymes or receptors as a way of signal transduction. The most prominent mechanism involves protein phosphorylation via the enzyme classes kinases and phosphatases. Protein kinases transfer the terminal phosphate group of an ATP molecule to the hydroxyl group of a serine, threonine, or tyrosine, while protein phosphatases reverse the reaction by phosphate removal [2]. Dysregulated phosphorylation is associated with a variety of disorders including cancer, inflammation, and neurodegeneration, which makes protein kinases a frequent target of drug discovery campaigns [60, 61].

### 1.3.1 Protein Kinases as Drug Targets and Challenges

Roughly a third of all FDA-approved drugs target kinases combating a variety of serious diseases such as cancer by acting as antineoplastic agents or immunosuppressants [61]. Remarkable advances have been made over the last decades from the first approved drug imatinib in 2001 to over 70 FDA-approved kinase drugs to date. Despite the extensive research on this target family, many open challenges remain [26, 62]:

- *Bias*: Most FDA-approved drugs historically target tyrosine kinases (TK), see more details in Section 1.3.2. This leads to a large fraction of under- and unexplored kinases.

For example, as of August 2022, 6047 X-ray structures of 314 human kinases have been resolved [63] with a strong emphasis on TK and CMGC kinases such as EGFR and CDK2 (Figure 1.2a).

- *Drug resistance*: Kinases often undergo specific mutations that impair drug binding; drugs are then no longer curative but only delay tumor progression. To circumvent drug resistance, research has expanded into the development of mutation-resistant inhibitors and the identification of synergistic drug combination treatments [64].
- *IP restriction*: Due to the long-standing interest in kinases as drug targets, the chemical space of kinase inhibitors is vastly patented, making it challenging to navigate through the crowded intellectual property (IP) space [26].
- *Selectivity*: Many kinase inhibitors are promiscuous binders due to the highly conserved binding sites across the entire set of known kinases, also referred to as the kinome. Such promiscuity can cause side effects due to off-target binding or can be explored for the design of multi-target drugs (polypharmacology) [64, 65].

### 1.3.2 Classification of Human Protein Kinome

The human protein kinome consists of roughly 500 kinases\*, which are generally divided into *eukaryotic* and *atypical protein kinases* as well as *pseudokinases*: eukaryotic protein kinases share a similar sequence and structure, whereas atypical protein kinases have biochemical kinase activity but lack sequence similarity to the typical kinase domain. In contrast, so-called pseudokinases have a kinase-like domain without conserved catalytic residues and are therefore predicted to be inactive [62].

Eukaryotic protein kinases —the focus of this thesis— can be classified based on their sequence identity into eight main kinase groups: AGC, CAMK, CK1, CMGC, STE, TK, TKL, and "Other" (Table 1.1) [67, 68]. Tyrosine-specific protein kinases (TPK) belong to the TK kinases and are subdivided into receptor and non-receptor kinases. In contrast, serine-/threonine-specific protein kinases (STPK) are more heterogeneous and are divided into the six kinase groups AGC, CAMK, CK1, CMGC, STE, and TKL. The "Other" group contains additional diverse protein kinases, that do not fit into the previous groups. The classification of the human kinome alongside its phylogenetic tree representation was published in 2002 by Manning et al. [67] and is the basis of the web-based tool KinMap that allows users to visualize human kinome data such as profiling or structural data [69] interactively (see examples in Figure 1.2).

### 1.3.3 Kinase Structure

Protein kinase structures consist of two domains, the N-terminal  $\beta$ -sheet-rich and the C-terminal  $\alpha$ -helix-rich lobes; the N- and C-lobes are connected via the hinge region (Figure 1.4a). The majority of kinase inhibitors target the catalytic cleft between these lobes, which contains a highly conserved ATP binding site.

Based on over 1200 kinase-ligand crystal structures, van Linden et al. [74] have defined the binding site to comprise 85 residues and 19 well-defined regions and motifs, covering a front cleft and a back cleft connected by a so-called gate area. This nomenclature, including a pocket residue numbering from 1 – 85, is applied to all available kinase structures and published in the KLIFS database [63]; see more details on KLIFS in the section "KLIFS pocket definition and

---

\*This number varies depending on the data resource, see an overview in [66].

Short name	Description	Specificity
AGC	PKA, PKG, and PKC containing families	STPK
CAMK	Calcium/calmodulin-dependent protein kinase	STPK
CK1	Casein kinase 1	STPK
CMGC	CDK, MAPK, GSK3, and CLK containing families	STPK
STE	Homologues of yeast sterile 7, sterile 11, sterile 20 kinases	STPK
TK	Receptor and non-receptor tyrosine kinases	TPK
TKL	Tyrosine kinase-like	STPK
"Other"	Protein kinases that do not fit any of the major groups	-

Table 1.1: Overview of eukaryotic kinase groups and their specificity towards either tyrosine- (TPK) or serine/threonine-specific kinases (STPK).

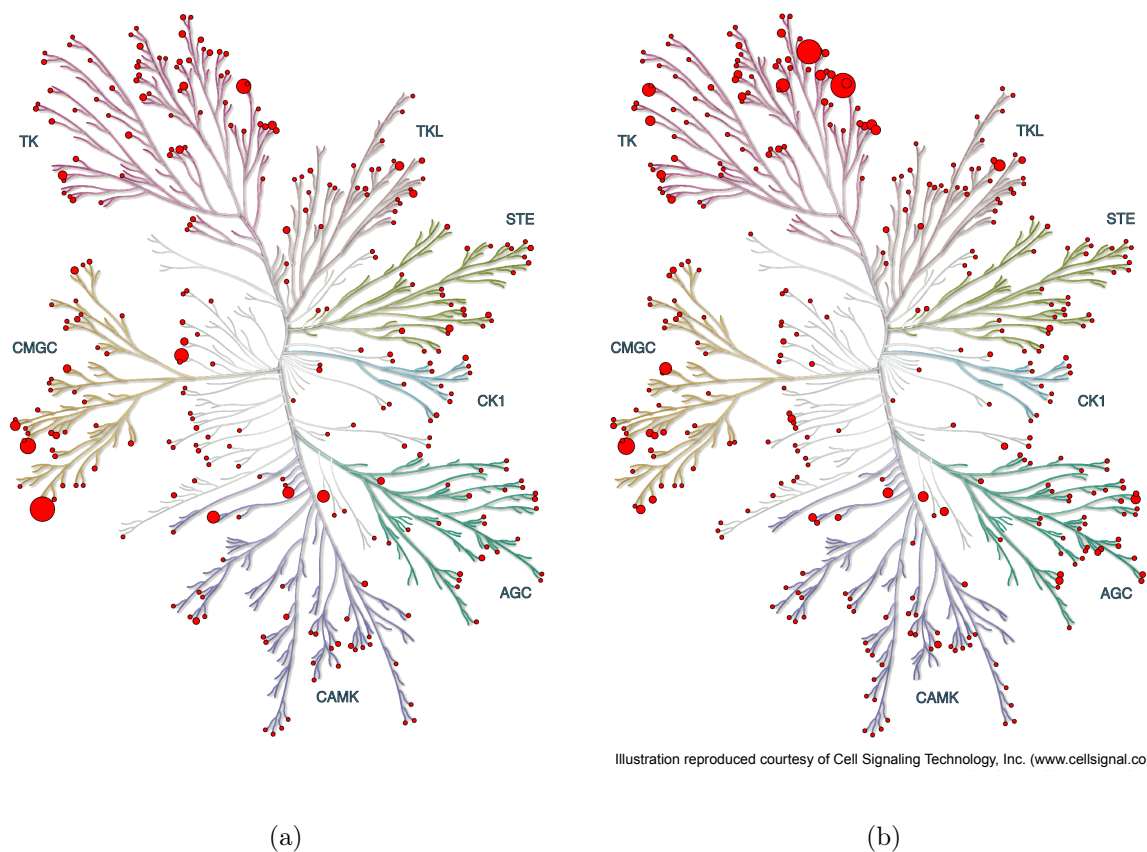


Figure 1.2: Exploration of kinase data coverage across the kinome: (a) structural landscape in the PDB [70] with a minimum of 1 and maximum of 432 structures as of 2022-09-09, and (b) bioactivity landscape in ChEMBL29 [71, 72] with a minimum of 1 and maximum of 5637 bioactivities. The KinMap-based [69] tree figures can be reproduced with [73]; atypical kinases were excluded.

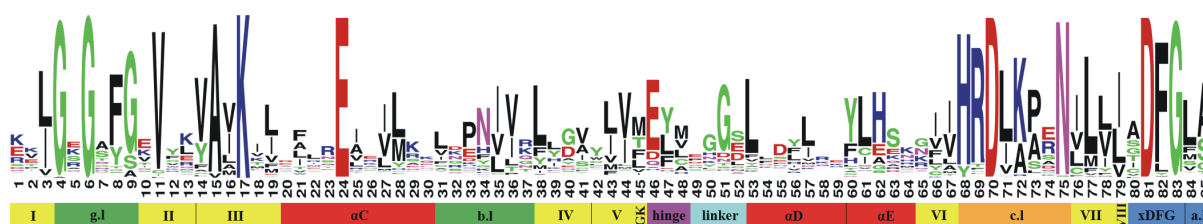


Figure 1.3: Kinase binding site residues and regions as defined by KLIFS. Shown are the residues 1 – 85, categorized into different subregions along the amino acid sequence:  $\beta$ -sheets I-VIII (yellow),  $\alpha$ -helices  $\alpha$ C/D/E (red), G-rich loop g.l and  $\alpha$ C-to-IV-connecting loop b.l (green), linker and hinge region (cyan and magenta), gate keeper (GK), catalytic loop c.l (orange), as well as xDFG motif and activation loop a.l (blue). This plot shows, in the form of a sequence logo [75], the conservation of the 85 kinase pocket residues across over 1200 kinase structures as defined in the initial KLIFS publication by van Linden et al. [74], from where this figure is taken.

alignment". In the following, key regions and residues are highlighted including their respective KLIFS residue numbering in brackets (1D and 3D views in Figures 1.3 and 1.4, respectively).

The *front cleft* accommodates the full ATP and contains the *hinge region* (46 – 48, magenta), *linker* (49 – 52, cyan), *glycine-rich loop* (4 – 9, green), and *catalytic loop* (68 – 75, orange). The *hinge region* forms key hydrogen bonds with ATP's adenine group as well as most kinase inhibitors (Figure 1.4b). The *glycine-rich* loop stabilizes ligand binding and the *catalytic loop* contains the aspartate D70, which functions as a base acceptor for the proton transfer during phosphorylation.

The *gate area* contains the *DFG motif* (81 – 83, blue), the conserved lysine K17 (17), and the gatekeeper residue (45), which is often used to address inhibitor selectivity and precedes the hinge region. The DFG motif can undergo a significant conformational change induced by flips between aspartate D81 and phenylalanine F82. These DFG-in and DFG-out conformations drive the active and inactive states of the kinase. In the DFG-in state, D81 binds  $Mg^{2+}$  ions that coordinate the phosphates of ATP to position them for phosphate transfer. In the DFG-out state, the flip opens a hydrophobic region in the back cleft targeted by inhibitors stabilizing the inactive state [74, 76]. Examples of DFG-in and DFG-out structures are shown for ATP and imatinib, respectively, in Figures 1.4b and 1.4d.

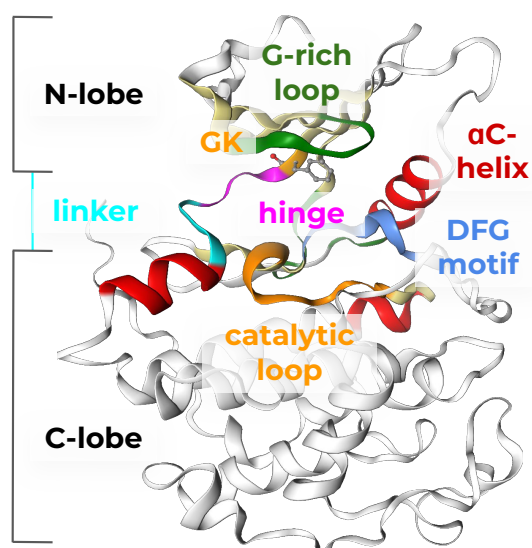
The *back cleft* contains amongst others the  $\alpha$ C-helix (20 – 30, red) with the conserved glutamine E24 (24), which forms a conserved K17-E24 salt bridge in the  $\alpha$ C-in conformation as opposed to no salt bridge in the  $\alpha$ C-out conformation. K17 and E24 in the  $\alpha$ C-in state interact with the phosphates of ATP to anchor and orient the ATP. Examples of  $\alpha$ C-in and  $\alpha$ C-out structures are shown for ATP and gefitinib, respectively, in Figures 1.4b and 1.4c.

### 1.3.4 Kinase Inhibitors

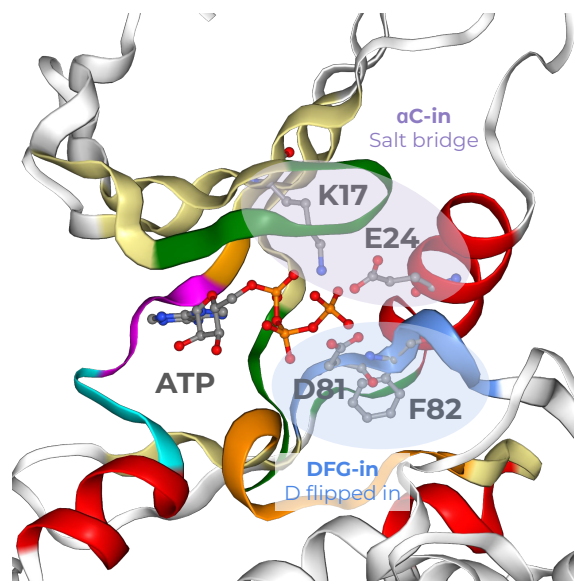
According to the PKIDB [83] database, over 300 kinase inhibitors are currently in clinical trials with the following percentages of inhibitors in the different phases [84, 85]: 1.3% in phase 0 to explore if and how the new drugs may work, 14.1% in phase I to check if the treatment is safe, 37.2% in phase II to check if the treatment works, 21.4% in phase III to determine if the new drug is better than already available drugs, and 26.0% in phase IV to follow up the drugs' effect after it has been approved (Figure 1.5a).

The first drug imatinib (Gleevec) was approved in 2001 to act on BCR-Abl and treat

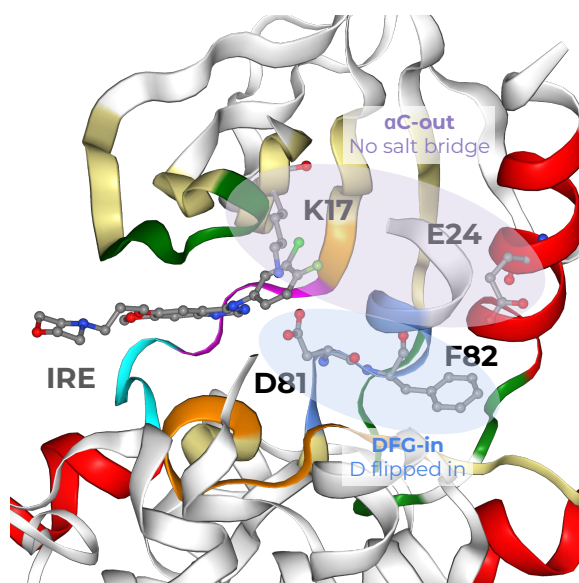




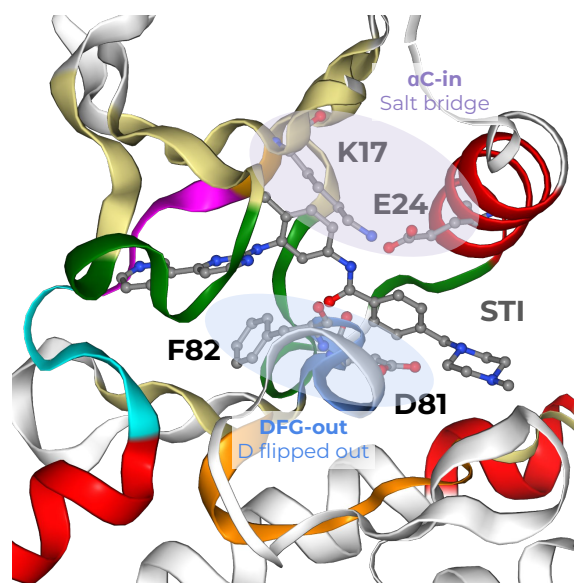
(a) Important structural regions and motifs in kinases; GK - gate keeper residue. Example kinase is CDK2 (PDB/KLIFS IDs: 1FIN/4367 [21]).



(b) CDK2 bound to ATP constitutes a DFG-in and  $\alpha$ C-in conformation with the K17-E24 salt bridge (PDB/KLIFS IDs: 1FIN/4367 [21]).



(c) EGFR bound to gefitinib (IRE) constitutes a DFG-in and  $\alpha$ C-out conformation (PDB/KLIFS IDs: 4I22/823 [77]).



(d) ABL1 bound to imatinib (STI) constitutes a DFG-out and  $\alpha$ C-in conformation (PDB/KLIFS IDs: 2HYY/1092 [78]).

Figure 1.4: Structural kinase regions, motifs, and conformations: (a) full structure view, (b) endogenous ligand ATP in DFG-in/ $\alpha$ C-in, (c) gefitinib-bound DFG-in/ $\alpha$ C-out, and (d) imatinib-bound DFG-out/ $\alpha$ C-in conformations. Figures were generated with this Jupyter Notebook [79] using the NGLviewer [80, 81] and OpenCADD [82]. The coloring scheme for the kinase regions corresponds to Figure 1.3. The interaction patterns are shown in 2D in Figure 1.6.

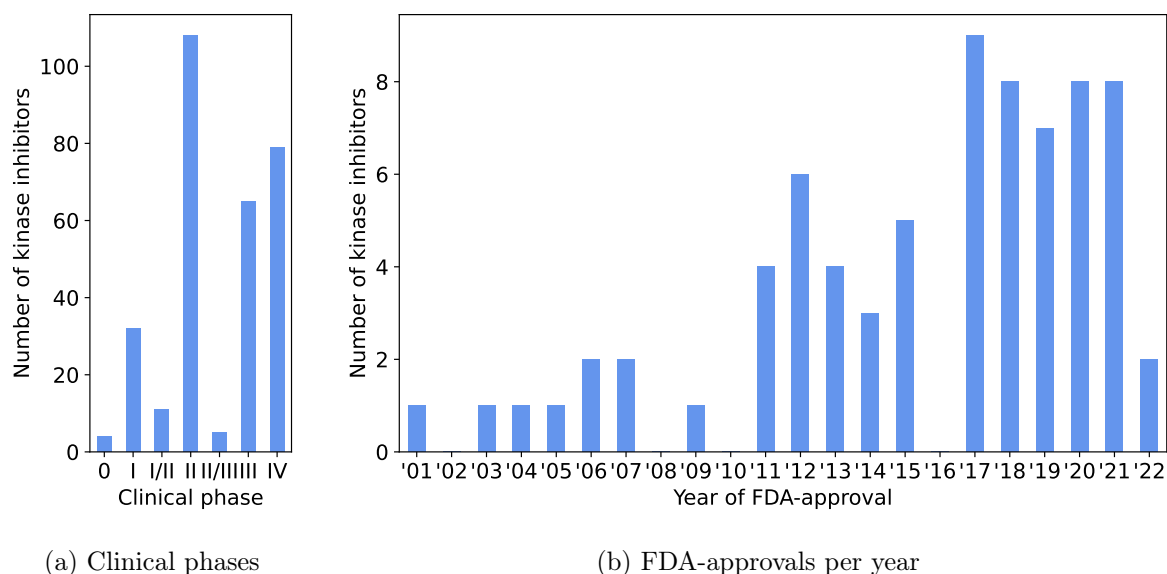
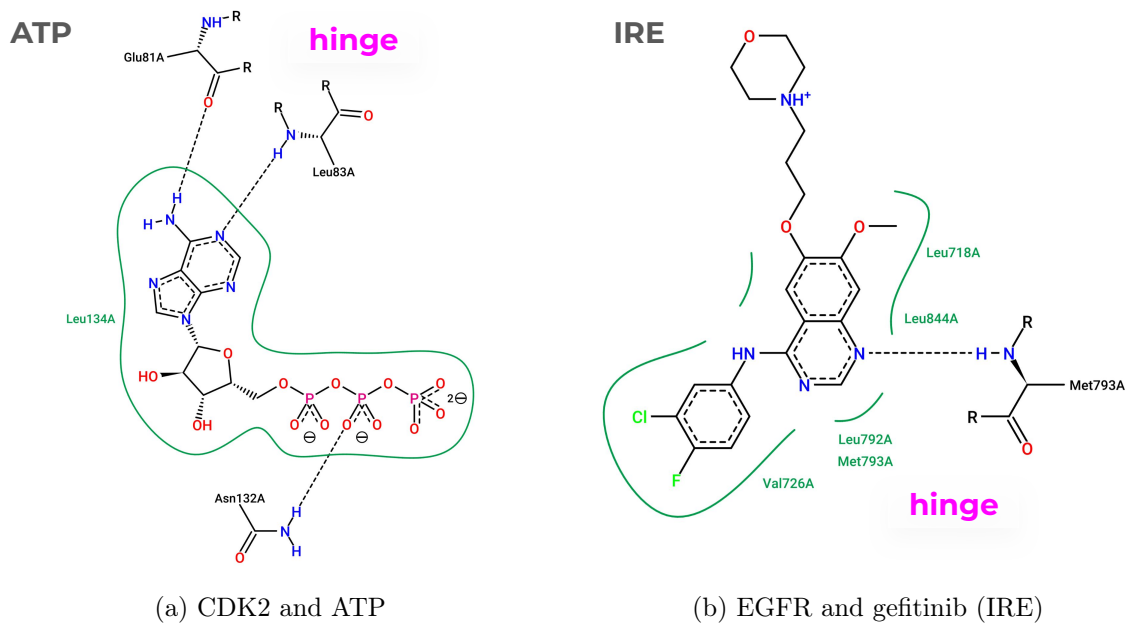


Figure 1.5: Number of kinase inhibitors that are (a) in clinical phases and (b) FDA-approved every year since the first approval of imatinib in 2001; based on data from the PKIDB [83] as of 2022-07-18. Note that the y-axes have different scales.

leukemia, followed by gefitinib (Iressa) and erlotinib (Tarceva) in 2003 and 2004 to act on EGFR and treat non-small cell lung cancer. Underlining the popularity of kinases as drug targets, the number of kinase inhibitors has doubled since 2016/17. To date, 71 kinase inhibitors are FDA-approved including the latest approvals in 2022, i.e., abrocitinib and pacritinib (Figure 1.5b). The majority of FDA-approved kinase inhibitors are active against more than one type of cancer, while only a few of them have non-oncological indications [86].

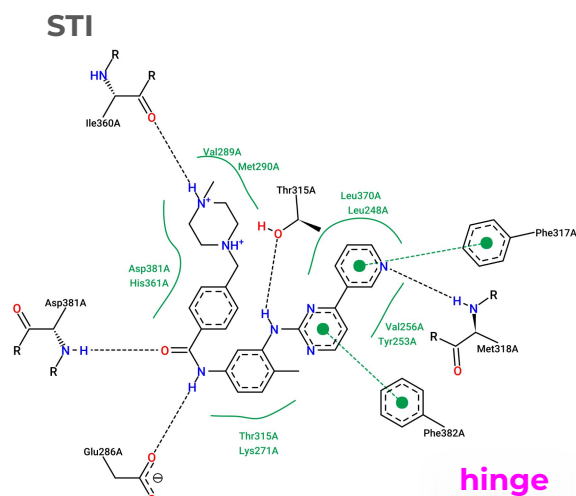
Most of the FDA-approved drugs bind in the ATP-binding pocket and intermediate surroundings and are classified by their binding modes with respect to properties such as orthosteric/allosteric binding, DFG-in/out kinase conformations, and reversible/irreversible binding [87]:

- *Type I and II* inhibitors accommodate the ATP-binding front pocket and form hydrogen bonds with the hinge region.
  - *Type I and I/2* inhibitors bind to the active and inactive DFG-in conformation, e.g., gefitinib (Figure 1.6b) and erlotinib, respectively.
  - *Type II* inhibitors stabilize the inactive DFG-out conformation, e.g., imatinib (Figure 1.6c).
- *Type III and IV* inhibitors are allosteric inhibitors that bind next to the ATP binding site and outside of the catalytic cleft, e.g., trametinib and everolimus, respectively.
- *Type V* inhibitors are bivalent binders, i.e., they bind two different kinase regions simultaneously. To date, no FDA-approved inhibitor has this binding mode.
- *Type VI* inhibitors are covalent binders, e.g., afatinib, whereas type I-V inhibitors bind reversibly.



(a) CDK2 and ATP

(b) EGFR and gefitinib (IRE)



(c) ABL1 and imatinib (STI)

Figure 1.6: Binding modes of ATP, gefitinib (IRE), and imatinib (STI) in complex with CDK2, EGFR, and ABL1, generated with PoseView on the ProteinsPlus webserver [10, 11]. Structural complexes (PDB/KLIFS IDs) used to illustrate the binding modes are (a) CDK2-ATP (1FIN/4367) [21], (b) EGFR-IRE (4I22/823) [77], and (c) ABL1-STI (2HYY/1092) [78].

### 1.3.5 KLIFS — a Structure-Focused Kinase Data Resource

The focus on the protein kinase family in drug discovery has resulted in a plethora of freely available databases, resources, and tools to explore bioactivity/profiling data, structures, sequences, and disease associations, which are thoroughly reviewed in [18, 62]. In the following, we will concentrate on the structure-focused KLIFS database [63].

KLIFS is a kinase database that extracts protein kinase-focused information on structures from the PDB [70]. To date (2022-08-06), KLIFS collects annotations and provides analyses for 6047 kinase PDB structures, which cover 314 kinases, 12898 monomeric structures, and 3788 unique ligands. KLIFS contains the following kinase, structure, and ligand annotations:

- *Kinases* in KLIFS are named according to the gene symbols defined by the HUGO Gene Nomenclature Committee (HGNC) [88]. The kinases are annotated with their kinase family, kinase group, species (to date, human and mouse), and cross-references to the UniProt [89] and IUPHAR/BPS Guide to PHARMACOLOGY (GtoPdb) [90] databases.
- *Ligands* bound to kinase structures are analyzed regarding the interactions they form with the 85 pocket residues, see more details in the section "KLIFS interaction fingerprint (IFP)". Furthermore, KLIFS defines three main pockets (front cleft, gate area, and back cleft) and twelve subpockets, as well as records which of those are occupied by structure-bound ligands. Structure-bound ligands are annotated with bioactivities from ChEMBL [71] and information about clinical trials from the PKIDB [83] if available.
- *Structures* representing ligand-bound or -unbound kinases are fetched from the PDB and processed as follows: (i) All multi-chain structures are split into monomers and aligned to each other with a special focus on a pre-defined binding site of 85 residues, see more details in the section "KLIFS pocket definition and alignment". (ii) This alignment enables the lookup of binding site residues such as the hinge region residues in any of the monomeric kinases in KLIFS. (iii) The monomeric structures in KLIFS are annotated with their originating PDB ID, chain, and alternate model (if multiple coordinates exist of PDB structure's atoms). (iv) Their quality is documented with the structure's resolution and KLIFS quality score, which ranges from 0 (bad) to 10 (flawless) accounting for the structural alignment, resolution, as well as missing residues and atoms. (v) The structures' conformations are described by the state of the DFG motif, the  $\alpha$ C-helix, the salt bridge between K17 and E24, the activation loop, and the G-rich loop. (vi) Each structure entry is assigned to a KLIFS ID, as are kinases and ligands.

The KLIFS data can be accessed in many ways; how to best interact with the KLIFS database depends on the amount of data to be accessed and on the user's coding experience:

- Manually using the website's interface at <https://klifs.net> [63], which is to be preferred when exploring a smaller set of structures.
- Automated with KLIFS KNIME nodes [20, 91], which is useful to process large datasets without the need to code.
- Programmatically using KLIFS' REST API and OpenAPI specifications [63] to perform larger scale queries or to integrate different queries into programmatic workflows.
- Programmatically using OpenCADD-KLIFS [92], a Python wrapper around KLIFS' REST API, to facilitate sending KLIFS requests and streamline the responses into a table format, so-called Pandas DataFrames [93], see Section 3.3.3 of this thesis.

## KLIFS pocket definition and alignment

The core of the KLIFS database consists of the kinase pocket definition and alignment procedure introduced by van Linden et al. [74] in 2012. A master alignment was created of all the kinase domains and subsequently optimized with a focus on known conserved patterns ("kinase domain sequence alignment"). The KLIFS pocket was determined based on 1252 unique ligand-kinase monomers extracted from the at the time 1734 kinase PDB structures. In this dataset, the defined 85 pocket residues interact with any bound kinase inhibitor within the catalytic front cleft, gate area, and/or back cleft (Figure 1.3) and cover the binding modes of type I, I<sup>1/2</sup>, II, and III inhibitors.

To allow for a structural alignment, the KLIFS authors defined a specifically constructed kinase template set. This set contains 24 structures of kinases representing all eight eukaryotic protein kinase groups with three structures per group (Table S2 in [74]). These template structures were sequence-aligned using the "kinase domain sequence alignment" and then structure-aligned by the superimposition of selected residues based on the sequence-alignment (the "superpose" selection). All kinase structures in KLIFS are aligned (based on sequence and structure) to the three template structures of their respective kinase groups as described before.

This procedure makes it possible to easily and instantly look up any pocket residue of interest across the full structurally covered kinome and opens the door for many applications in the field of structural cheminformatics, such as exploring interaction patterns across the kinome based on interaction fingerprints.

## KLIFS interaction fingerprint (IFP)

Interaction fingerprints (IFPs) convert the binding mode of a ligand in a binding site, i.e., the protein-ligand interactions that are present in a structurally resolved complex into a machine-readable bit string. This can, for example, be used to identify important (e.g., frequent or rare) interactions or interaction patterns for ligand design, off-target prediction, or selectivity studies.

KLIFS annotated kinase-ligand interactions are based on the FingerPrintLib developed by Marcou and Rognan [94], which encodes the presence (1) or absence (0) of seven different interaction types between each of the 85 pocket residues and the ligand: Hydrophobic, face-to-face and face-to-edge aromatic, hydrogen bond donor and acceptor, as well as positive and negative ionic interactions. This results in an  $85 \cdot 7 = 595$ -bit-long IFP per pocket-ligand pair, where each bit  $i$  represents the same residue and interaction type in every IFP across the kinome (Figure 1.7).

### 1.3.6 Kinase Bioactivity and Profiling Resources

The longstanding research focus on protein kinases in academia and the pharmaceutical industry resulted in a wealth of not only structural—as discussed in detail in the previous section—but also bioactivity data that is either deposited in databases or published alongside research articles.

ChEMBL is one of the primary resources for bioactivity data and holds roughly 20 million bioactivity values on over two million compounds and 15000 targets; the latest version ChEMBL31 was released in July 2022. Kinase-focused subsets of the ChEMBL dataset are provided, for example, (i) on the KLIFS database fetching bioactivity values for all deposited structurally resolved kinase-ligand complexes and (ii) on the KinoData platform fetching all bioactivity values associated with kinases (latest release covers ChEMBL30 [96]). As with other data types, the coverage of bioactivity data is highly unbalanced among the human kinases,

1							2							3							85						
HYD	F-F	F-E	DON	ACC	ION+	ION-	HYD	F-F	F-E	DON	ACC	ION+	ION-	HYD	F-F	F-E	DON	ACC	ION+	ION-	HYD	F-F	F-E	DON	ACC	ION+	ION-
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0

Figure 1.7: Schematic depiction of a KLIFS interaction fingerprint (IFP): Seven interaction types are detected between each of a kinase structure’s 85 pocket residues and a co-crystallized ligand. Interaction types include hydrophobic contacts (HYD), face-to-face aromatic interactions (F-F), face-to-edge aromatic interactions (F-E), protein hydrogen bond donors (DON), protein hydrogen bond acceptors (ACC), protein cationic interactions (ION+), and protein anionic interactions (ION-). The bits 0 and 1 stand for the absence and presence of a specific interaction at a specific amino acid residue of the 85 pocket sequence. Figure is taken from [95] (CC-BY 4.0 license), which was adapted from [74].

depending on how much research has been spent on certain kinases (Figure 1.2b).

However, a major challenge for using ChEMBL and other public databases is the inherent data heterogeneity rooted in bioactivity measurements originating from various experimental setups. Hence, kinase-specific chemogenomics datasets —profiling multiple kinases with a set of compounds— are used to assess polypharmacology effects of compounds: Kinase profiles by Karaman et al. [97] and Davis et al. [98] cover 38 and 72 kinase inhibitors across a panel of 317 and 442 kinases, respectively, and are build-in datasets on the KinMap webserver [69]. Efforts to generate a comprehensive kinase chemogenomics set (KCGS) by Drewry et al. [99] resulted in the PKIS2 dataset assaying 645 compounds on 392 kinases. Alternatively, studies have combined multiple smaller profiling datasets. The KIBA dataset by Tang et al. [100] combines three selectivity profiles, including the Davis et al. [98] dataset, and covers 52498 compounds on 467 targets, while the Moret et al. [101] dataset combines six kinase inhibitor libraries to allow the user to generate selectivity profiles.

## 1.4 Open Science

Open science aims to increase the reuse of research and ensures that scientific data are accessible to all. The key to achieving this goal is adhering to FAIR principles, which were designed in a workshop held in 2014 in Leiden, Netherlands, by stakeholders from academia, industry, funding agencies, and scholarly publishers [102]. The FAIR principles are summarized on the GO FAIR website [103] as follows:

- F. Findable:** The first step in (re)using data is to find it. Metadata and data need to be easy to find for both humans and computers.
- A. Accessible:** Once the user finds the required data, one needs to know how it can be accessed, possibly including authentication and authorization.
- I. Interoperable:** The data usually needs to be integrated with other data. In addition, the data needs to interoperate with applications or workflows for analysis, storage, and processing. This requires standardized vocabulary, unique identifiers, and good data models.

- R. Reusable:** The ultimate goal of FAIR is to optimize the reuse of data. To achieve this, metadata and data should be well-described so that they can be replicated and/or combined in different settings.

FAIRness is illustrated by Wilkinson et al. [102] with the example of the UniProt [89] resource, which archives protein sequences and annotations: UniProt has a stable URL linking to data and metadata (F) that is human- (HTML) and machine-readable (text and RDF) (A). The RDF-formatted response uses a shared vocabulary and ontology (I) and UniProt entries are interlinked with more than 150 different databases in the RDF representation (R).

These principles are not only applicable to data but also to algorithms, tools, and workflows that lead to that data. To improve the sharing and reuse of research software, the FAIR for Research Software (FAIR4RS) initiative has applied the FAIR principles from data to research software. Many of the principles can be directly applied by treating software and data as similar digital research objects. However, specific characteristics of software—such as its executability, composite nature, and continuous evolution and versioning—make it necessary to revise and extend the principles as summarized in [104]:

- F. Findable:** Software and its associated metadata are easy for both humans and machines to find.
- F1.** Software is assigned a globally unique and persistent identifier.
  - F2.** Software is described with rich metadata.
  - F3.** Metadata clearly and explicitly includes the identifier of the software they describe.
  - F4.** Metadata is FAIR, searchable, and indexable.
- A. Accessible:** Software and its metadata are retrievable via standardized protocols.
- A1.** Software is retrievable by its identifier using a standardized communications protocol.
  - A2.** Metadata is accessible, even when the software is no longer available.
- I. Interoperable:** Software interoperates with other software by exchanging data and/or metadata, and/or through interaction via application programming interfaces (APIs), described through domain-relevant standards.
- I1.** Software reads, writes, and exchanges data in a way that meets domain-relevant community standards.
  - I2.** Software includes qualified references to other objects.
- R. Reusable:** Software is both usable (can be executed) and reusable (can be understood, modified, built upon, or incorporated into other software).
- R1.** Software is described with a plurality of accurate and relevant attributes.
  - R2.** Software includes qualified references to other software.
  - R3.** Software meets domain-relevant community standards.

Lamprecht et al. [105] outlined how software can adhere to these principles: Software can be findable (F) when registered with an identifier such as issued by Zenodo [106] and following metadata standards such as PEP566 for Python [107]. It can be accessible (A) via HTTP/S on

code repositories such as GitHub [108], GitLab [109], and BitBucket [110]. It can be interoperable (I) by including versioning, dependencies, and interfaces such as OpenAPI [111], as well as by packaging or using software containers such as Docker [112] for portability across operating systems. Finally, it can be reusable (R) when assigned a license, provenance, and requirements, while following code standards such as PEP8 for Python [113].

Chemistry has a long history of developing software and algorithms to tackle chemical problems. While in the beginning, most cheminformatics software was freely available, only little effort was spent to make it usable [114]. In 2005, several open chemistry and cheminformatics projects such as Open Babel [115] and The Chemistry Development Kit (CDK) [116–119] joined forces to enhance interoperability. This movement was named "Blue Obelisk" and covers the areas Open Data, Open Standards, and Open Source (ODOSOS) with the aim that knowledge can be freely used, modified, and redistributed: [114, 120]

- *Open Data in Chemistry*: One can obtain all scientific data in the public domain and reuse it for whatever purpose.
- *Open Standards in Chemistry*: One can find visible community mechanisms for protocols and communicating information.
- *Open Source in Chemistry*: One can use other people's code without further permission, including changing it for one's own use and distributing it again.

The Blue Obelisk movement discussed in 2011 that open source software is valuable to not only academia but also industry because it allows for independent validation of source code data and computational procedures. O'Boyle et al. [120] stated at the same time that most Blue Obelisk projects are not remunerated and contributors do much of the work in their spare time.

Since then, many organizations and initiatives have been established to finance open source software projects, such as the Open Molecular Software Foundation (OMSF) [121], Google Summer of Code (GSoC) [122], Chan Zuckerberg Initiative (CZI) [123], and Quansight Labs [124]. The Chemistry Consortium in the National Research Data Infrastructure (NFDI4Chem) [125] builds an open and FAIR infrastructure for research data management in chemistry in Germany. The Molecular Sciences Software Institute (MolSSI) has been founded to promote open science and software best practices and is a great example of providing software expertise, infrastructure, education, and training. Workshops such as the CICAG "Open Source Tools for Chemistry Workshops" give a stage for developers to show their open source tools to potential users at the interface of chemistry, biology, and informatics.

The software projects shown in this thesis are built on top of a rich and amazing landscape of (i) open source toolkits for data science such as NumPy [126], Pandas [93], Scikit-learn [127], Matplotlib [128], Seaborn [129], JupyterLab [130], and for life sciences such as the RDKit [131], Biotite [132], PyPDB [133], ChEMBL webresource client [134], and more, as well as (ii) open datasets such as the PDB [70], KLIFS [63], ChEMBL [71], PubChem [135] databases and ProteinPlus [136] webservices. Our thanks go to all the contributors and maintainers of these open resources.



## Chapter 2

# Aim and Objectives

Drug discovery is a complex process that takes approximately 13.5 years and costs around US\$ 1.8 billion to bring a new molecular entity (NME) on the market as a new drug, while only 8% of NMEs transition successfully from the pre-clinical stage to approval [137]. Phrased in more drastic terms, drug development is expensive, lengthy, and highly prone to failure, while patients wait for treatment. Computer-aided drug design (CADD) is today a standard discipline in pharmaceutical research and development from target identification to lead optimization to help shorten the timelines, reduce costs, and improve the success rates of new drugs [138, 139]. This thesis aims to contribute to this endeavor with a focus on kinases.

Kinases belong to the most studied target classes due to their role in cancer, which is the world’s second-largest health problem [140]. Decades of research have yielded large amounts of structural, chemical, and pharmacological data. In this thesis, these rich datasets build the basis for the development of structural cheminformatics tools with an emphasis on kinase-focused computational target prediction and fragment-based drug design while applying FAIR principles to support open science.

Computational target prediction plays a major role in the target identification phase of drug discovery campaigns. Such prediction methods are applied to explore potential targets, polypharmacology, off-target effects, drug repurposing, and potential chemical probes. To achieve this, they encode and compare information based on ligands, protein sequences, protein-ligand interactions, or protein structures. Within the latter structure-based category, binding site comparison is a target prediction approach that assumes that similar binding sites bind similar ligands. While many binding site comparison methods have been published since the early 2000s as reviewed in **Publication A** [22] (Section 1.2.1), the toolbox for kinase research was still missing a kinase-focused and open-sourced tool (i) that can detect and rationalize pocket similarities across the structurally covered kinome and (ii) that is embedded in an automated pipeline with alternative similarity measures.

**Section 3.1** presents three publications that help to close this gap in the field of *kinome-wide (off-)target prediction*. In **Publication B** [141] (Section 3.1.1), we introduced the novel KiSSim method, which encodes and compares the structural kinome provided in the public KLIFS database [63]. KiSSim captures the spatial and physicochemical characteristics of the conserved kinase binding sites based on KLIFS’ residue-by-residue pocket alignment. We show that this method can detect unexpected kinase inhibitor off-targets and explain structural differences in kinase profiles. In **Publication C** [142] (Section 3.1.2), we present a study in collaboration with the Kolb Lab in Marburg, Germany. Our collaborators were trying to design multi-target kinase inhibitors with specific on- and off-target profiles based on docking screens but observed

that their target profiles deemed more difficult than anticipated. We investigate if the observed difficulties result from unexpected high kinase similarities between on- and off-targets by applying different similarity measures based on the pocket sequences and structures, pocket-ligand interaction profiles, and ligand promiscuity. In **Publication D** [95] (Section 3.1.3), we propose a pipeline based on our findings in Publications B and C. We streamline similarity measures as discussed in Publication C while replacing the therein used kinase-unspecific structure-based method with our novel kinase-specific method KiSSim. This automated similarity analysis allows analyzing user-defined sets of kinases of interest from multiple views.

**Section 3.2** treats kinase pockets from another perspective to advance *kinase-focused computational fragment-based drug design* (FBDD). FBDD searches for the right combinations of relevant fragments to build novel and potent molecules. In **Publication E** [143] (Section 3.2.1), we introduce the KinFragLib project to guide this search by extracting the known chemical space of relevant kinase subpockets. We utilize the public data on the kinase pocketome focusing on the binding poses that kinase inhibitors occupy in experimentally resolved kinase-ligand complexes. We treat kinase inhibitors as combinations of fragments that occupy kinase-typical subpockets. These subpocket-specific fragment libraries are subsequently used to (i) explore the chemical subpocket space of kinases and (ii) recombine kinase fragments guided by their subpocket connections to generate novel kinase-focused molecules.

**Section 3.3** stresses the importance of *FAIR pipelines and toolkits* for computer-aided drug design to allow for an efficient and reproducible drug hunting process. With the example of kinases, we provide Python-based solutions for common tasks in ligand- and structure-based drug design published as part of the TeachOpenCADD platform in **Publications F and G** [144, 145] (Sections 3.3.1 and 3.3.2). Furthermore, we present the Python tool OpenCADD-KLIFS in **Publication H** [92] (Section 3.3.3), which builds the backbone for acquiring structural kinase data from the KLIFS database, which we use throughout all kinase-focused projects discussed in this thesis.

We seek to make all presented methods, tools, pipelines, and datasets publicly available and to follow FAIR principles and software best practices. In summary, the central questions that this thesis aims to answer are:

- **Predicting kinome-wide (sub)pocket-based off-targets:** How can we build an open-sourced and kinase-specific pocket fingerprint that can explain and predict unexpected kinase inhibitor off-targets? How can we incorporate this measure with other similarity measures for production-ready usage in drug discovery projects?
- **Exploring kinome-wide subpocket fragment spaces:** How can we build and explore subpocket-focused fragment libraries based on public kinase-ligand structures? Can we generate novel kinase-focused molecules by recombining these fragments guided by their original subpocket connections?
- **FAIR pipelines and tools in kinase-centric drug design:** How can we contribute to the scientific community not only with novel scientific insights but also with open-sourced scientific infrastructure?

We will answer these questions in Chapter 3 and discuss our findings in Chapter 4. Finally, we will conclude with the impact of this thesis on the scientific community in Chapter 5.

# Chapter 3

## Methods and Results

This doctoral thesis consists of ten publications (Figure 3.1): One review on computational target prediction was presented in the Introduction (Publication A, Section 1.2.1) and seven articles are summarized in this chapter (Publications B–H, Section 3). One article and one book chapter are part of the appendix (Publications I–J, Appendix 5.1). This chapter contains the following sections:

- Section 3.1: Predicting kinome-wide (sub)pocket-based off-targets (Publications B–D)
- Section 3.2: Exploring kinome-wide subpocket fragment spaces (Publication E)
- Section 3.3: FAIR pipelines and tools in kinase-centric drug design (Publications F–H)

Each section is preceded by an illustration by Ferdinand Krupp in Figures 3.2–3.4.

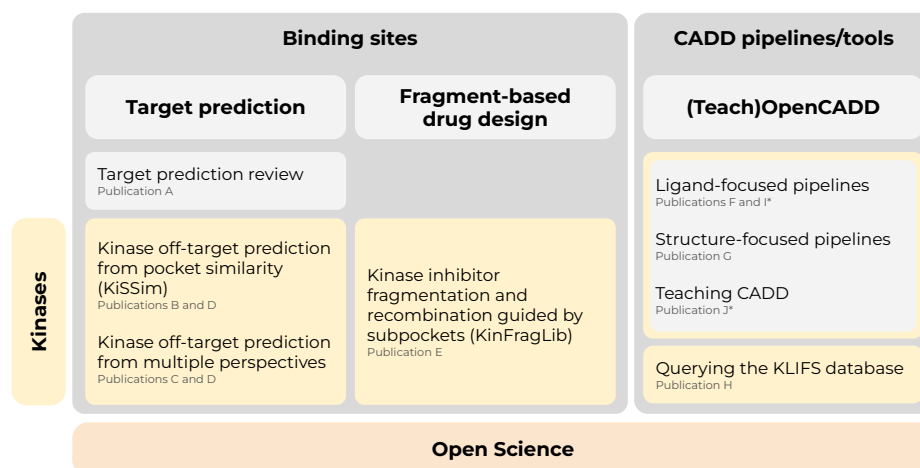


Figure 3.1: Schematic overview of the ten publications that are included in this doctoral thesis based on the following criteria: Publications related to the study of binding sites and kinases as well as dedicated to open science and pipelines or tools for computer-aided drug design (CADD). Refer to the List of Publications to find the full references to Publications A–J (the symbol \* indicates publications listed in this thesis' appendix).



### 3.1 Predicting Kinome-Wide (Sub)Pocket-Based Off-Targets

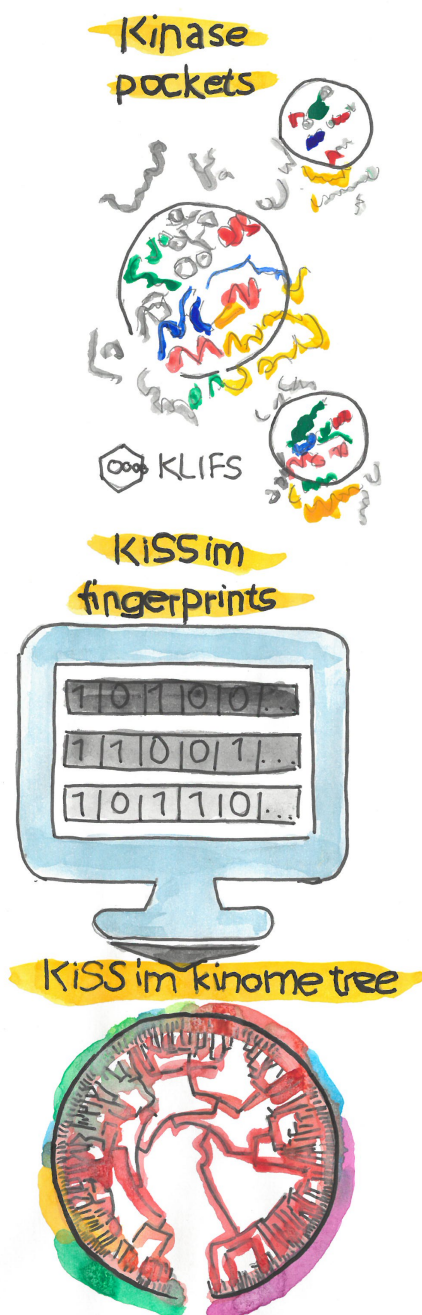



Figure 3.2: Predicting kinome-wide (sub)pocket-based off-targets as illustrated by Ferdinand Krupp, adapted from Sydow et al. [141].


### 3.1.1 KiSSim: Predicting Off-Targets from Structural Similarities in the Kinome

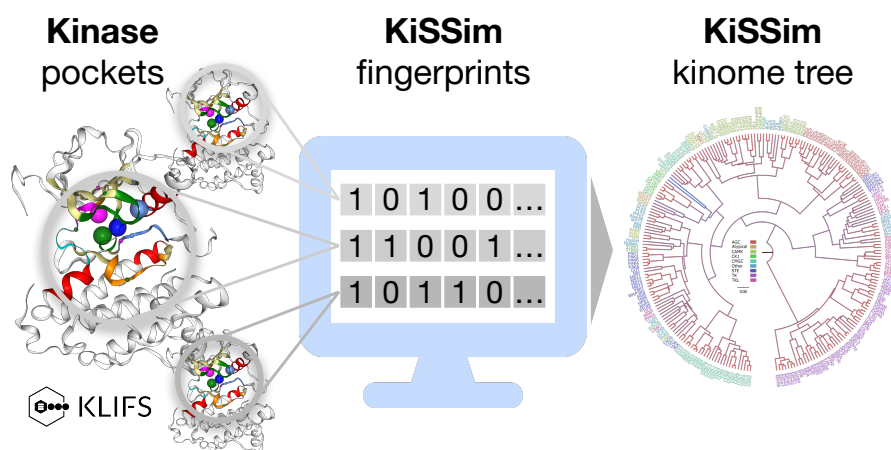
#### Publication B

In this study, we present the novel kinase-focused and subpocket-based fingerprint KiSSim that encodes physicochemical and spatial properties of kinase binding sites as defined by the KLIFS database [63]. The pre-aligned KLIFS pockets enable a direct and computationally inexpensive bit-by-bit comparison. We show how the kinome-wide KiSSim comparison can be used to (i) build phylogenetic trees to study kinase relationships, (ii) explain kinase inhibitor off-targets that are reported in kinase profiling datasets, (iii) evaluate KiSSim's performance to other similarity measures, and (iv) rationalize (dis)similarities in 3D.

 <https://github.com/volkamerlab/kissim>

 [https://github.com/volkamerlab/kissim\\_app](https://github.com/volkamerlab/kissim_app)

 <https://kissim.readthedocs.io/en/latest>



Contribution:

#### First author

Conceptualization (50%)

Data Curation (95%)

Formal Analysis (90%)

Investigation (90%)

Methodology (50%)

Software (100%)

Validation (90%)

Visualization (90%)

Writing — Original Draft (90%)

Writing — Review & Editing (85%)

Reprinted with permission from Sydow D, Aßmann E, Kooistra AJ, Rippmann F, Volkamer A. KiSSim: Predicting Off-Targets from Structural Similarities in the Kinome. *Journal of Chemical Information and Modeling*. **2022**; 62(10):2600-2616. 10.1021/acs.jcim.2c00050.

Copyright © 2022 American Chemical Society.

## KiSSim: Predicting Off-Targets from Structural Similarities in the Kinome

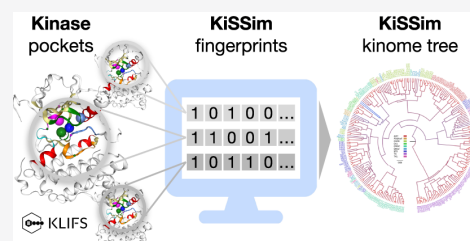
Dominique Sydow, Eva Aßmann, Albert J. Kooistra, Friedrich Rippmann, and Andrea Volkamer\*

 Cite This: *J. Chem. Inf. Model.* 2022, 62, 2600–2616 Read Online

ACCESS |

 Metrics & More Article Recommendations Supporting Information

**ABSTRACT:** Protein kinases are among the most important drug targets because their dysregulation can cause cancer, inflammatory and degenerative diseases, and many more. Developing selective inhibitors is challenging due to the highly conserved binding sites across the roughly 500 human kinases. Thus, detecting subtle similarities on a structural level can help explain and predict off-targets among the kinase family. Here, we present the kinase-focused, subpocket-enhanced KiSSim fingerprint (Kinase Structural Similarity). The fingerprint builds on the KLIFS pocket definition, composed of 85 residues aligned across all available protein kinase structures, which enables residue-by-residue comparison without a computationally expensive alignment. The residues' physicochemical and spatial properties are encoded within their structural context including key subpockets at the hinge region, the DFG motif, and the front pocket. Since structure was found to contain information complementary to sequence, we used the fingerprint to calculate all-against-all similarities within the structurally covered kinome. We could identify off-targets that are unexpected if solely considering the sequence-based kinome tree grouping; for example, Erlotinib's known kinase off-targets SLK and LOK show high similarities to the key target EGFR (TK group), although belonging to the STE group. KiSSim reflects profiling data better or at least as well as other approaches such as KLIFS pocket sequence identity, KLIFS interaction fingerprints (IFPs), or SiteAlign. To rationalize observed (dis)similarities, the fingerprint values can be visualized in 3D by coloring structures with residue and feature resolution. We believe that the KiSSim fingerprint is a valuable addition to the kinase research toolbox to guide off-target and polypharmacology prediction. The method is distributed as an open-source Python package on GitHub and as a conda package: <https://github.com/volkamerlab/kissim>.



### INTRODUCTION

Protein kinases are involved in most aspects of cell life due to their role in signal transduction. Their dysregulation can cause severe diseases such as cancer, inflammation, and neurodegeneration,<sup>1</sup> which makes them a frequent target of drug discovery campaigns. In 2015, 30% of FDA-approved small molecules targeted kinases.<sup>2</sup> The roughly 500 kinases in the human genome share a highly conserved binding site, causing serious challenges for selective drug design for a single kinase or a well-defined set of kinases (polypharmacology) and avoiding binding to undesired off-targets.<sup>3,4</sup>

Protein kinases bind adenosine triphosphate (ATP) to catalyze the transfer of its phosphate group to serine, threonine, or tyrosine residues of themselves or other proteins. ATP and most other ligands bind to the front cleft of the kinase pocket that is situated between the two kinase domains, the C- and N-terminal lobes. These domains are connected via a hinge region, which is forming important hydrogen bonds to ATP as well as most studied ligands. The gate area contains the conserved DFG (aspartate-phenylalanine-glycine) motif, whose phenylalanine flips in and out of the front pocket, opening and closing a hydrophobic region in the back cleft, i.e., constituting the DFG-in and DFG-out conformation, respectively. The back cleft also comprises the  $\alpha$ C-helix with a

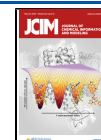
conserved glutamine residue, which forms a salt bridge with a conserved lysine residue in the gate area. Such a conformation is called  $\alpha$ C-in as opposed to  $\alpha$ C-out.<sup>5</sup>

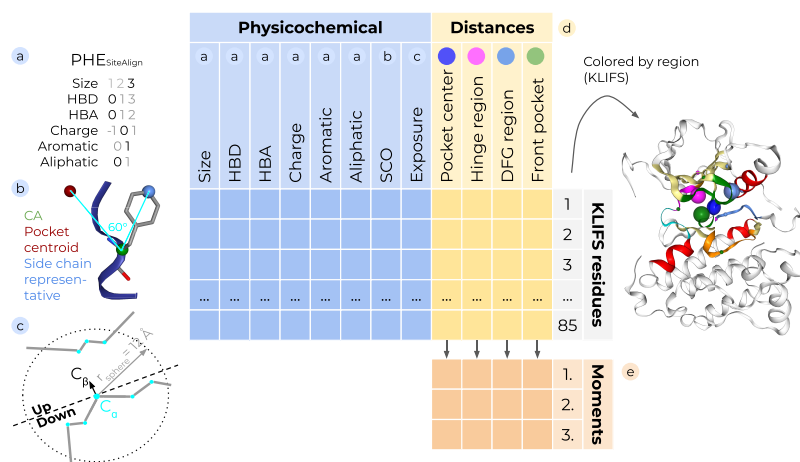
Researchers have studied kinase similarity between the full—or parts of the—kinome from many different angles. Manning et al.<sup>6</sup> used a multiple sequence alignment (MSA) to cluster the kinome into eight main groups of eukaryotic protein kinases (ACG, CAMK, CK1, CMGC, STE, TK, TKL, and Other) and the atypical protein kinase families. Recently, Modi and Dunbrack<sup>7</sup> assigned some kinases, which were left unassigned in the Other category, based on a structurally validated MSA.

While sequence comparison—and thus, evolutionary similarity—can explain many observations from kinase profiling experiments, other more distantly related off-targets remain undetected. For example, profiling Erlotinib against 48 kinases

Received: January 16, 2022

Published: May 10, 2022





**Figure 1.** The KiSSim fingerprint encodes physicochemical and spatial properties of kinase pockets. The fingerprint builds on the KLIFS<sup>14</sup> pocket definition, i.e., 85 residues aligned across all available protein kinase structures, which enables residue-by-residue comparison without a computationally expensive alignment. Each residue is encoded physicochemically and spatially. Physicochemical properties include the following features per residue (example: phenylalanine/PHE): (a) Pharmacophoric features and size categories are taken from the SiteAlign<sup>18</sup> binding site comparison methodology. (b) Side chain orientation (SCO) is adapted from SiteAlign and defined as inward-facing, intermediate, or outward-facing depending on the vertex angle between the pocket centroid, the residue's side chain representative (Table S3), and the CA atom. (c) Solvent exposure is defined as high, intermediate, or low depending on the ratio of CA atoms in the upper half of a sphere cut in half by a normal plane spanned by the residue's CA-CB vector. The implementation is based on BioPython's HSExposure.<sup>22,23</sup> Spatial properties are defined as follows: (d) Each residue's distance to the pocket center and important kinase subpockets, i.e., the hinge region, DFG region, and the front pocket. On the right, example locations are shown in the 3D representation of kinase EGFR (PDB ID: 2TTO; KLIFS structure ID: 783). (e) The distance distributions per pocket center and subpocket are furthermore described by their first three moments, i.e., the mean, standard deviation, and skewness. The figure is adapted from <https://github.com/volkamerlab/kissim/>.

revealed high affinity against the on-target EGFR (TK group) but also the non-TK off-targets SLK, LOK, and GAK,<sup>8</sup> or the chemical probe SGC-STK17B-1 binds both DRAK2 and CaMMK,<sup>9</sup> although they are dissimilar when judged solely by their sequence.<sup>6</sup> Focusing on the kinase pocket instead of the whole sequence already helps: The 50 most similar kinases to EGFR are only TK kinases when ranked by full-length sequence while listing non-TK kinases when considering the pocket sequence only.<sup>10</sup> The KinCore phylogenetic tree produced by a kinome-wide structure-guided MSA<sup>7,11</sup> overall confirms the assignment from Manning et al.<sup>6</sup> but provides higher precision, e.g., regarding previously unassigned kinases. Schmidt et al.<sup>12</sup> have recently investigated the similarities between a panel of nine kinases—EGFR, ErbB2, PIK3CA, KDR, BRAF, CDK2, LCK, MET, and p38a—based on different pocket encodings, including the pocket sequence identity, pocket structure similarity, interaction fingerprint similarity, and ligand promiscuity. Individual kinase relationships differed according to these different perspectives, while some trends could be observed such as the atypical kinase PIK3CA being an outlier among the otherwise typical kinases in this panel.

In an attempt to facilitate computer-aided kinase similarity studies, we here aim to add another perspective. Binding site comparison methods employed so far can be applied to any binding site regardless of the protein class. Kuhn et al.<sup>13</sup> have applied such a method, Cavbase, to the structurally resolved kinome and could detect expected and unexpected kinase relationships. Since kinases are highly conserved and have been aligned and annotated across the full structurally covered kinome, a binding site comparison method tailored to kinases may provide an extended perspective on kinase similarities. We

make use of data in the KLIFS<sup>14</sup> database, a rich resource for kinase research that extracts protein kinase-focused information on structures from the PDB,<sup>15</sup> on inhibitors in clinical trials from the PKIDB,<sup>16</sup> on bioactivities from ChEMBL,<sup>17</sup> and much more. All kinase structures from the PDB are split into single chains and models and aligned with respect to the sequence and structure across the fully structurally covered kinome. The KLIFS authors defined the kinase pocket as a set of 85 residues that interact with cocrystallized ligands in the initial KLIFS dataset of more than 1200 structures.<sup>5</sup> Thanks to this structural alignment, it is possible to look up all 85 residues in any kinase structure, given that the residue is structurally resolved and not in a gap position. This pocket alignment is the basis for the here introduced KiSSim fingerprint.

The kinase-focused and subpocket-enhanced KiSSim (Kinase Structural Similarity) fingerprint builds on the KLIFS<sup>14</sup> pocket, whose alignment allows a computationally inexpensive residue-by-residue comparison. The residues' physicochemical and spatial properties are encoded within their structural context including important kinase subpockets—the hinge region, DFG region, and front pocket—building on features from previously published methods such as SiteAlign,<sup>18</sup> KinFragLib,<sup>19</sup> and Ultrafast Shape Recognition (USR).<sup>20</sup> We used the fingerprint to calculate all-against-all similarities within the structurally covered kinome and to generate a KiSSim-based kinome tree. Detected similarities can be used to predict off-targets or guide polypharmacology studies and to rationalize profiling observations on a structural level. We distribute the method as an open-source Python package at <https://github.com/volkamerlab/kissim> and as a conda package, alongside the data and analysis notebooks at <https://>



github.com/volkamerlab/kissim\_app to support FAIR<sup>21</sup> science.

## METHODS AND DATA

In the following, we outline the KiSSim methodology and implementation, the datasets used, and the method's evaluation. All data, fingerprints, and analyses are available at [https://github.com/volkamerlab/kissim\\_app](https://github.com/volkamerlab/kissim_app).

**KiSSim Methodology.** The KiSSim methodology consists of three steps: the encoding of a set of kinase binding sites as KiSSim fingerprints (Figure 1), the all-against-all comparison of these structures using their fingerprints, and—since one kinase can be represented by multiple structures—the mapping of multiple structure/fingerprint pairs to one kinase pair.

**Encoding: From a Structure to a Fingerprint.** The KiSSim fingerprint encodes the 85 KLIFS pocket residues in the form of physicochemical and spatial properties as illustrated in Figure 1. We summarize the encoding procedure in the following; for a detailed description, please refer to the Supplementary Methods section.

**Physicochemical Properties.** Physicochemical properties are encoded by eight features in the form of categorical values. Pharmacophoric and size features are taken from the SiteAlign categories for standard amino acids.<sup>18</sup> They encode the size based on the number of heavy atoms, the number of hydrogen bond donors (HBD) and hydrogen bond acceptors (HBA), the charge (negative, neutral, or positive), and aromatic and aliphatic properties (present or not present) of a residue (Table S1). The side chain orientation (inward-facing, intermediate, or outward-facing) is based on the vertex angle from the residue's CA atom (vertex) to the pocket center and to the residue's outermost side chain atom, the side chain representative (Table S3). The solvent exposure of a residue (high, intermediate, or low) is based on the ratio of CA atoms in the upper half of a sphere that is placed around the residue's CA atom (radius, 12 Å) and cut in half by a normal plane spanned by the residue's CA-CB vector, as implemented in BioPython's HSExposure module.<sup>22,23</sup>

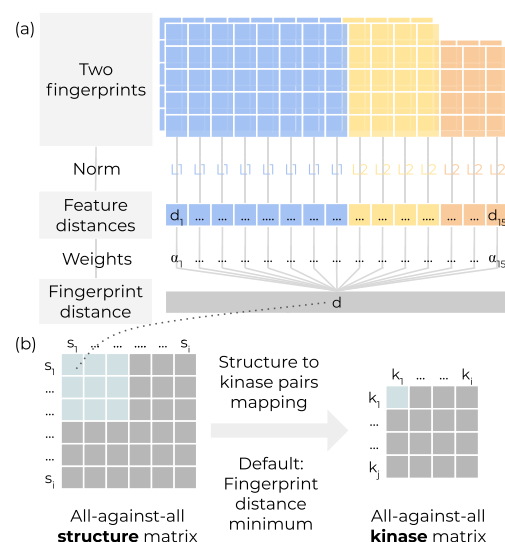
**Spatial Properties.** Spatial properties are described by discrete values, i.e., distances and moments. Spatial distances are calculated from each residue's CA atom to the pocket's geometric center and to prominent subpocket centers. The pocket center is the centroid of all pocket CA atoms. The selected subpocket centers include functionally well-characterized kinase regions such as the hinge region, DFG region, and front pocket. Each subpocket center is calculated based on the centroid of three anchor residues' CA atoms (Table S4), following the idea described in the KinFragLib methodology.<sup>19</sup> We added the code to calculate the subpocket centers to the structural cheminformatics library OpenCADD (module `opencadd.structure.pocket`)<sup>24</sup> to allow for easy access in other projects. Spatial moments describe each of the four distributions of distances to the pocket center, hinge region, DFG region, and front pocket. In KiSSim, the first three moments are used: the mean, the standard deviation, and the cube root of the skewness. This procedure is inspired and adapted from the ligand-based Ultrafast Shape Recognition (USR)<sup>20</sup> method. The comparison of distance distributions via moments is possible given the conserved nature of kinase binding sites; note that it is untested yet if such a procedure would suffice for less similar binding sites.

**Fingerprint Length.** The final full-length fingerprint encompasses eight discrete physicochemical features (8

features  $\times$  85 residues), four continuous spatial distance features (4 features  $\times$  85 residues), and three continuous spatial moment features (3 moments  $\times$  4 distributions), resulting in a 1032-bit vector. Optionally, a subset of residues can be selected to generate a subset fingerprint emphasizing certain residues. We offer a subset of residues that is based on frequently interacting cocrystallized ligands<sup>25</sup> and key residues identified by Martin and Mukherjee<sup>26</sup> (see Table S5), but users can also inject their own list of residues.

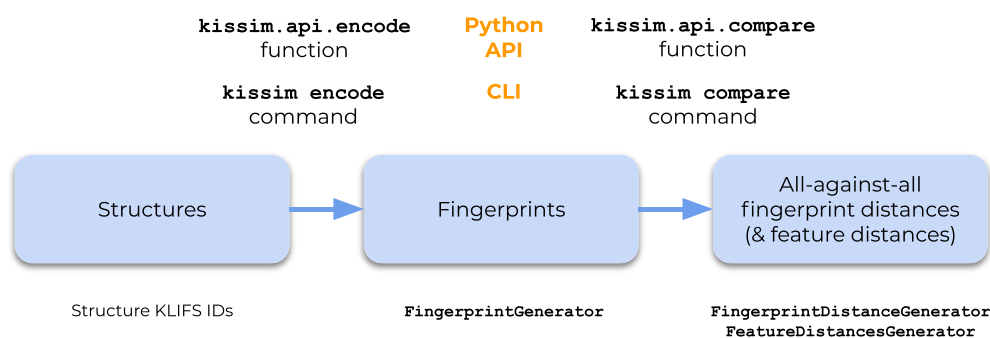
**Normalization.** Fingerprints are normalized to values between 0 and 1 by applying a min-max normalization. For discrete features, the minimum and maximum categorical values are used. For continuous features, the minimum and maximum values for each spatial feature are set to the minimum and maximum values observed across all structures; distance extrema are defined for each residue position individually, while moment extrema are defined for the first, second, and third moments individually.<sup>27</sup>

**Pairwise Structure Comparison.** Two kinase pocket structures—encoded as two fingerprints—can be compared in two steps (Figure 2). First, we calculate for each feature the



**Figure 2.** Structures—encoded as KiSSim fingerprints—are compared pairwise and mapped to kinase pairs. (a) The discrete physicochemical features (blue) are compared using the scaled L1 norm, while the continuous spatial features (yellow/orange) are compared using the scaled L2 norm, resulting in a feature distance vector composed of one distance per feature. Custom weighting of these features results in the final fingerprint distance. By default, the features are weighted equally. (b) Two kinases of interest may have multiple structures each. Thus, multiple structure/fingerprint pairs can represent the same kinase pair. By default, we select the minimum (fingerprint) distance value among all structure/fingerprint pairs to represent the (kinase) distance between a kinase pair.

distance between the corresponding two feature vectors across the 85 residue entries, producing a feature distance vector of length 15 (i.e., aggregating over the columns in Figure 2a). For example, the two fingerprints' 85-bit size feature vectors—representing the size of each of the 85 pocket residues—will be reduced to a single-size feature distance. The distance between discrete features is defined as the scaled L1 norm



**Figure 3.** The `kissim` library's Python API and CLI. Structures from the KLIFS database can be encoded as fingerprints using the `FingerprintGenerator` class (details in Figure 1) and compared using the `FeatureDistancesGenerator` and `FingerprintDistanceGenerator` classes (details in Figure 2). The package offers the wrappers `encode` and `compare` for quick and easy access from within a Python script (Python API) or from the command line (CLI). Please also refer to the `kissim` library's documentation at <https://kissim.readthedocs.io>.

$\|\mathbf{x}\|_1 = \frac{1}{n} \sum_{i=1}^n |x_i|$  (scaled Manhattan distance), whereas the distance between continuous features is defined as the scaled L2 norm  $\|\mathbf{x}\|_2 = \frac{1}{n} \sqrt{\sum_{i=1}^n x_i^2}$  (scaled Euclidean distance), where  $\mathbf{x}$  is a vector of length  $n$ .<sup>28</sup> Second, we calculate the weighted sum of the 15-bit feature distance vector with feature-level weights  $\alpha_{1..15}$  to produce the final fingerprint distance. By default, the 15 features are equally weighted with a weight of  $\frac{1}{15}$  each.

Summarizing both steps, the fingerprint distance  $d(\mathbf{f}_i, \mathbf{f}_j)$  between two fingerprints  $\mathbf{f}_i$  and  $\mathbf{f}_j$  is defined in eq 1. The different KiSSim features are denoted as  $m$ : 1 = size; 2 = HBD; 3 = HBA; 4 = charge; 5 = aromatic; 6 = aliphatic; 7 = side chain orientation; 8 = solvent exposure; 9 = distance to pocket center; 10 = distance to hinge region; 11 = distance to DFG region; 12 = distance to front pocket; 13 = first moment; 14 = second moment; 15 = third moment.

$$d(\mathbf{f}_i, \mathbf{f}_j) = \sum_{m=1}^8 \alpha_m \frac{\|\mathbf{f}_i^m - \mathbf{f}_j^m\|_1}{85} + \sum_{m=9}^{12} \alpha_m \frac{\|\mathbf{f}_i^m - \mathbf{f}_j^m\|_2}{85} + \sum_{m=13}^{15} \alpha_m \frac{\|\mathbf{f}_i^m - \mathbf{f}_j^m\|_2}{4} \quad (1)$$

**Kinome-Wide Comparison.** The kinome-wide comparison is based on an all-against-all comparison of all available structures. Note that a kinase can be represented by multiple structures (see the **KLIFS Data** section); thus, a kinase pair can be represented by multiple structure pairs with multiple distance values. Our final goal is to assign one distance value to each kinase pair as a measure of the similarity between these two kinases (Figure 2b). The structural coverage of kinases is highly imbalanced: Some kinases are represented by one structure only, and others like EGFR or CDK2 are represented by more than 100 structures. We select the structure pair with the lowest distance as a representative for the kinase pair, hence always picking the two closest structures in the dataset. For example, if a dataset consists of 10 structures representing three kinases, the  $10 \times 10$  all-against-all structure distance matrix will be reduced to a  $3 \times 3$  all-against-all kinase distance matrix, consisting of the lowest distance values only after mapping structure pairs to kinase pairs.

**Fingerprint and Similarity Visualization in 3D.** Fingerprint features can be visualized in 3D using the `NGLviewer`<sup>29,30</sup> and

`IPyWidgets`<sup>31</sup> for the following applications: (a) Fingerprint features of a structure can be visualized in 3D by coloring the residues by different feature values. (b) The difference between two structures can be highlighted to spot positions of high or low similarity between two structures. The differences are shown for each feature type individually. (c) The standard deviation of spatial features between all structures available for one kinase can be mapped onto an example structure in 3D to show regions of high or low variability between different kinase conformations.

**KiSSim Tree.** The kinase distance matrix produced as described in the **Kinome-Wide Comparison** section is submitted to a hierarchical clustering as implemented in `SciPy`<sup>32</sup> using as a metric the Euclidean distance and as a linkage Ward's criterion. We generate a phylogenetic tree in the Newick format based on this KiSSim kinase clustering. The tree branches are labeled with the mean of all distances belonging to that branch; the tree leaves are annotated with the kinase names and their assigned Manning kinase groups. We visualize the tree in an automatized way using `BioPython's` `Phylo`<sup>22,33</sup> module to be used in Jupyter Notebooks and in a manual way using the freely available `FigTree`<sup>34</sup> software to produce publication-ready circular trees.

**KiSSim Implementation.** The `kissim` library is implemented as an open-source Python package, which is available on GitHub at <https://github.com/volkamerlab/kissim> and as a conda package at conda-forge.<sup>35,36</sup> Structures are retrieved via the `OpenCADD-KLIFS` module<sup>24</sup> and are encoded as fingerprints using the `FingerprintGenerator` class; fingerprints can be compared using the `FingerprintDistanceGenerator` class. We also offer quick access `encode` and `compare` functionalities as Python API and as a command-line interface (CLI) (see Figure 3). Last, the `kissim.encoding.tree` module offers an automatized all-against-all clustering and phylogenetic tree generation, while the 3D visualization of fingerprints and pairwise comparisons is implemented in the `kissim.viewer` module.

Structural data is read and processed with `BioPython`<sup>22</sup> and `BioPandas`;<sup>37</sup> computation is performed with `NumPy`,<sup>38</sup> `Pandas`,<sup>39</sup> `SciPy`,<sup>32</sup> and `Scikit-learn`.<sup>40</sup> The code for operations that are of use outside of the KiSSim project has been added to the `OpenCADD` library:<sup>24</sup> KLIFS queries are implemented in the `OpenCADD-KLIFS` module and subpocket centers can be defined and visualized with the `OpenCADD-pocket` module.

All code is written in Python 3<sup>41</sup> following the PEP8 style guide. We document the code following NumPy docstrings<sup>42</sup> as well as format and lint the code and notebooks with black,<sup>43</sup> black-nb,<sup>44</sup> flake8,<sup>45</sup> and flake8-nb.<sup>46</sup> A detailed documentation is hosted on ReadTheDocs<sup>47</sup> at <https://kissim.readthedocs.io> using sphinx.<sup>48</sup> We test the kissim code using pytest<sup>49</sup> with a code coverage of over 90%, measured with CodeCov.<sup>50</sup> Notebooks are checked with nbval<sup>51</sup> and continuous integration is deployed with GitHub Actions<sup>52</sup> on a weekly basis.

**Data.** We are using the following sources of external data: KLIFS kinase structures<sup>14</sup> and the profiling datasets by Karaman et al.<sup>8</sup> and Davis et al.<sup>53</sup> filtered and processed as described in the following. All prepared datasets described here are accessible via the src.data module at [https://github.com/volkamerlab/kissim\\_app](https://github.com/volkamerlab/kissim_app).

**KLIFS Data.** We downloaded the human structural kinase dataset from the KLIFS database version 3.2<sup>14</sup> on 2021-09-02. This dataset contained 11,806 human monomeric structures, i.e., PDB entries split into monomeric structures if consisting of multiple chains and alternate models. We filtered the dataset for human kinases with a resolution  $\leq 3$  Å and a KLIFS quality score  $\geq 6$ . The KLIFS quality score ranges from 0 (bad) to 10 (flawless) and describes the quality of the structural alignment and resolution regarding missing residues and atoms. In addition, we excluded structures with more than three pocket mutations or with more than eight missing pocket residues. To reduce computational costs, we selected the best structure per kinase in each PDB entry (kinase–PDB pair); the best structure per kinase–PDB pair is defined as the structure with the least missing pocket residues, the least missing pocket atoms, the lowest alternate model identifier, and the lowest chain identifier (in that order). Structures were excluded if they are flagged as problematic structures in KLIFS and if they could not be encoded as KiSSim fingerprints. We produced three final datasets of structures for KiSSim fingerprint generation and all-against-all comparison: structures in any DFG conformation, DFG-in conformation only, and DFG-out conformation only. Table S6 lists the number of structures remaining after each filtering step.

**Bioactivity Profiling Data.** To compare predicted and measured on- and off-targets, we use two kinase bioactivity datasets available through KinMap:<sup>56</sup> The Karaman et al.<sup>8</sup> and Davis et al.<sup>53</sup> datasets on KinMap contain inhibition profiles ( $K_d$  values) for 38 and 72 kinase inhibitors across 317 and 442 kinases, respectively. The lower the  $K_d$  value, the higher the binding affinity, which is used as a proxy for activity. We pooled data from both datasets by taking the union of all kinase–ligand pairs. If kinase–ligand pairs have bioactivity values in both datasets, we proceeded as follows: If both measurements  $K_{d,1}$  and  $K_{d,2}$  are (a) below or equal to or (b) above or equal to the chosen activity cutoff of  $K_d^{\text{cutoff}} = 100$  nM, we kept the lower  $K_d$ , i.e., the more active measurement. If one of the measurements is above and the other is below that cutoff, we kept the lower  $K_d$  if the difference is  $|K_{d,1} - K_{d,2}| \leq 100$  nM; otherwise, the measurements were discarded. That way, we keep the measurement with the lowest  $K_d$  if both measurements agree on the ligand's activity, including a tolerance zone around our defined activity cutoff, and we remove measurements if they disagree considerably. This approach results in a  $353 \times 80$  kinase–ligand matrix with 7619 measurements, named the Karaman–Davis dataset from here on.

**Evaluation.** We evaluate our KiSSim results by comparison to profiling data as well as alternative similarity measures based on KLIFS pocket sequences, KLIFS pocket interaction fingerprints (IFPs), and SiteAlign.<sup>18</sup> All prepared datasets and evaluation strategies described here are accessible via the src.data and src.evaluation modules at [https://github.com/volkamerlab/kissim\\_app](https://github.com/volkamerlab/kissim_app).

**KiSSim Evaluation Using Profiling Data.** To evaluate how well KiSSim detects kinase similarities, we need to define an experimental reference point for kinase similarities. We use profiling data as a surrogate for this, since we assume that kinases that are targeted by the same ligand share similar binding sites. To this end, we use the profiling Karaman–Davis dataset, which describes the activity of ligands against a panel of kinases. We assign each ligand  $l_i$  in the profiling dataset to their reported key target(s)  $k_j(l_i)$  in the PKIDB,<sup>16</sup> ranging from one target to multiple targets, e.g., Erlotinib is assigned to EGFR only, while Imatinib binds to ABL1, KIT, RET, TRKA, FMS, and PDGFRa. These examples result in the following kinase–ligand pairs: EGFR–Erlotinib, ABL1–Imatinib, KIT–Imatinib, RET–Imatinib, TRKA–Imatinib, FMS–Imatinib, and PDGFRa–Imatinib. Note that we only included (a) kinases whose name could be mapped to the KinMap kinase names and (b) ligands that are listed in the PKIDB. Furthermore, only kinase–ligand pairs (a) whose kinase was tested active against the ligand ( $K_d \leq 100$  nM) and (b) that share at least 10 kinases between the Karaman–Davis and KiSSim datasets, of which at least three have measured ligand activities of  $K_d \leq 100$  nM, were included. For example, the EGFR–Erlotinib pair shares Erlotinib profiling measurements and EGFR KiSSim distances for 50 kinases, of which four are defined as active using the aforementioned  $K_d$  cutoff. Each remaining kinase–ligand pair is evaluated as demonstrated here for the EGFR–Erlotinib pair ( $l_1 = \text{Erlotinib}$  and  $k_1 = \text{EGFR}$ ):

1. We define the kinases in both lists as active or inactive based on the chosen activity threshold of  $K_d = 100$  nM.
2. We rank all kinases by their KiSSim distance to EGFR. These are our KiSSim-based kinase similarities.
3. We calculate ROC curves to demonstrate how well the profiling data is predicted by our KiSSim-based kinase similarities.

Some kinase activities measured in the profiling dataset are rather unexpected from a sequence-based similarity point of view. For the EGFR–Erlotinib example, we use the KinMap server to plot the profiling-based and KiSSim-based ranked kinases onto the kinome tree by Manning et al.<sup>6</sup> For example, we highlight kinases with measured activities against Erlotinib as well as the 50 most similar kinases to EGFR as detected by KiSSim. All kinases that are part of the KiSSim dataset are shown as well to define which data points are available for similarity predictions.

**KiSSim Comparison to Other Methods.** We outline here the preparation of all-against-all kinase distance matrices based on different similarity measures to be compared to the KiSSim kinase distance matrix (see the KiSSim Dataset section): KLIFS pocket sequence, KLIFS pocket–ligand interaction fingerprint (IFP), and SiteAlign's pocket structure. All distance matrices underwent a min-max normalization<sup>57</sup> and can be loaded via src.data.distances at [https://github.com/volkamerlab/kissim\\_app](https://github.com/volkamerlab/kissim_app).

**KLIFS Pocket Sequence.** We performed an all-against-all comparison of the sequence identity within the KLIFS pocket of 85 residues. The sequence identity is defined as the number of identical pocket residues divided by all 85 pocket residues; gap positions are treated as identical if both structures show a gap. If two sequences are identical, the sequence identity is 1; if two sequences do not have a single residue in common, the sequence identity is 0. To make these values comparable with the kinase distance matrices, we define distance = 1 – identity.

**KLIFS Pocket IFP.** We performed an all-against-all comparison of the KLIFS IFP describing interactions between cocrystallized ligands and the KLIFS pocket. For each pocket residue, seven potential protein–ligand interaction types were defined as described by Marcou and Rognan.<sup>58</sup> The presence or absence of a certain type of interaction is noted as 1 or 0 in the bit-string. This results in an  $85 \cdot 7 = 595$ -bit-long IFP per pocket–ligand pair. The Jaccard distance is used to compare the IFPs. If multiple IFP pairs describe the same kinase pair, we selected the minimum distance as the representative measure for the kinase pair, following the same procedure as described for the KiSSim methodology.

**SiteAlign.** We performed an all-against-all comparison using the pocket comparison method SiteAlign<sup>18</sup> (version 4.0). In this approach, properties of a binding site are projected to a triangulated sphere positioned at the pocket center, stored as a fingerprint to be compared and aligned to another binding site fingerprint iteratively. Since we used the existing KLIFS alignment, a few SiteAlign parameters were adapted to reduce runtime: we decreased the number of alignment steps in SiteAlign from 3 to 1 and the translational steps from 5 to 3 and reduced the rotational and translational intensity from  $2\pi$  to  $\frac{1}{4}\pi$  and from 4 to 1, respectively. Comparison of the SiteAlign performance for > 4000 structure pairs with the default and adjusted settings showed that the adjusted settings resulted in lower distances (average decrease of 6%) while matching a higher number of triangles (average increase of 15%). Pocket residues with modifications (e.g., phosphorylated threonines) were excluded to avoid segmentation faults.

## RESULTS AND DISCUSSION

We present here the generated KiSSim dataset and the resulting KiSSim-based kinome tree. Furthermore, we evaluate the KiSSim results in comparison to profiling data (KiSSim evaluation using the Bioactivity Profiling Data section) and other pocket encoding methods (see the KiSSim Comparison to Other Methods section).

**KiSSim Dataset.** KLIFS structures are filtered as described in detail in the KLIFS Data section (Table S6), then encoded and compared as described in the KiSSim Methodology section. When considering structures in DFG-in conformations only, 4112 fingerprints representing 257 kinases result in a  $4112 \times 4112$  structure distance matrix and—after mapping structure-to-kinase pairs as described in the Kinome-Wide Comparison section—in a  $257 \times 257$  kinase distance matrix (Table 1).

**Fingerprint Feature Value Distribution.** The KiSSim fingerprint encodes the 85 KLIFS pocket residues in the form of physicochemical and spatial properties. Physicochemical properties include pharmacophoric and size features, side chain orientation, and solvent exposure; spatial properties include each residue's distance to the pocket center as well as to three subpockets and the first three moments of the

Table 1. KiSSim Dataset<sup>a</sup>

	all	DFG-in	DFG-out
number of structures	4,681	4,112	406
number of kinases	279	257	71
number of structure pairs	10,953,540	8,452,216	82,215
number of kinase pairs	38,781	32,896	2485

<sup>a</sup>Number of structures and kinases as well as number of structure and kinase pairs encoded and compared with the KiSSim methodology. The number of structure/kinase pairs does not contain self-comparisons. See notebooks for more details.<sup>54,55</sup>

resulting distance distributions (Figure 1). We investigate here the fingerprint feature value distribution across all KiSSim fingerprints.

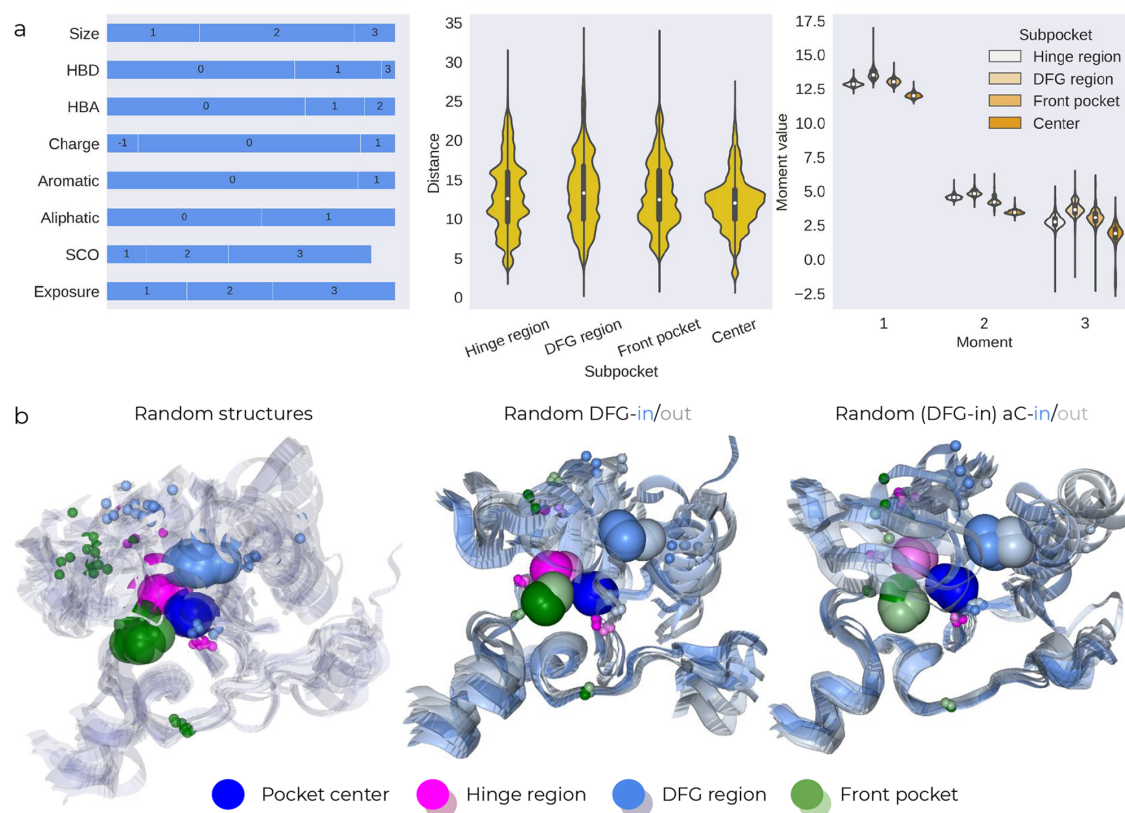
The value distributions for pharmacophoric and size features differ depending on the feature type (Figure 4a) and the residue position (Figures S2 and S3). For example, the amino acid size is more evenly distributed than the aromatic or charge feature, since most amino acids are neither aromatic nor charged (Figure 4a, left). Since the five pharmacophoric and residue size features encode—in an abstracted manner—the pocket sequence, features are more robust at more conserved pocket positions than at other positions; examples are the conserved salt bridge between residues 17 and 24 or the DFG residues 81–83 (Figure S2).

Spatial distances range between 2 and 33 Å (Figure 4a, middle); however, depending on the residue position, the values cover only a subset of this range. For example, the hinge region residues 46–48 are close to the hinge region center while further away from the DFG region center (Figure S3). Distances from subpocket centers to regions such as the G-rich loop (residues 4–9), the  $\alpha$ C-helix (residues 20–30), and the DFG motif vary more than, for example, to the hinge region, which agrees with knowledge on more flexible vs more stable regions in the kinase pocket. The spatial moment features describe the distance distributions between the pocket residues to the subpocket centers. They show lower variability for the mean and standard deviation but high variability for the skewness (Figure 4a, right).

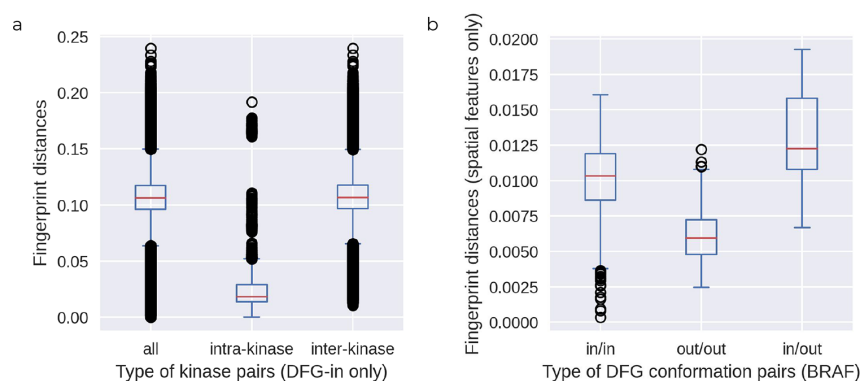
The spatial features are based on the KiSSim subpockets as described in the Encoding: From a Structure to a Fingerprint section. These subpockets are calculated for each structure individually; however, they show robustness over the structural kinome. The subpocket centers occupy the same space across the aligned KLIFS structures, while the front pocket and DFG region center show higher variability than the hinge region and pocket center (Figure 4b), as to be expected. Therefore, the subpocket definition procedure seems to be robust enough to span comparable subpocket centers while fine-grained enough to encode structural differences.

In conclusion, the feature space encoded in the KiSSim fingerprint, on the one hand, reflects sequence-related similarities between kinases on a generalized level through the defined physicochemical properties and, on the other hand, incorporates information on flexible and stable regions through the defined spatial properties.

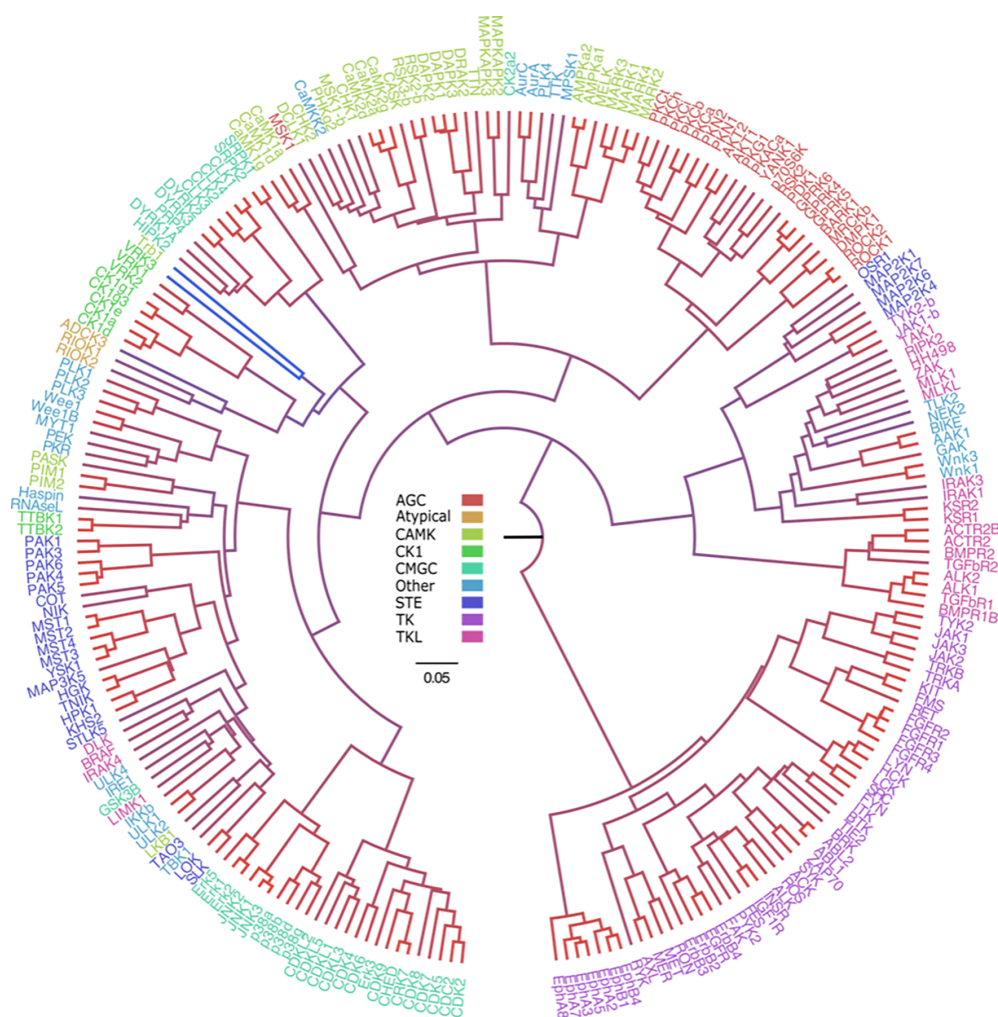
**Fingerprint Distances to Compare Structures.** Moving on from the structure encoding (fingerprints) to the structure comparison (fingerprint distances), we aimed to explore if the KiSSim fingerprint can be used to discriminate between kinases and between DFG-in and DFG-out conformations.



**Figure 4.** Fingerprint feature and subpocket distributions. (a) Distribution of all over 400,000 feature values aggregated from all structures and all pocket residues. Categorical physicochemical features (in blue) include size, hydrogen bond donor count (HBD), hydrogen bond acceptor count (HBA), charge, aromatic, aliphatic, side chain orientation (SCO), and solvent exposure. Distance features (in yellow) include distances to the subpocket centers for the hinge region, DFG region, and front pocket as well as the pocket centroid. Moment features (in orange) include the first three moments, i.e., mean, standard deviation, and scaled skewness, for each structure's distance distribution. (b) The subpocket centers are shown in 3D for example structures (left), highlighted by DFG conformations (middle) and  $\alpha$ C-helix conformations for example DFG-in structures (right). See notebooks for more details.<sup>59–61</sup> Note that we show here unnormalized fingerprints; for the downstream fingerprint comparison, the fingerprints are normalized to values between 0 and 1 first.



**Figure 5.** KiSSim fingerprint can distinguish between kinases and DFG conformations. (a) We compare fingerprint distances (based on all fingerprint bits) for structure pairs representing any kinase (all), the same kinase (intra-kinase), or different kinases (inter-kinase); here, we use only DFG-in conformations. The dataset includes about 8.4 million pairwise structure distances, of which about 200,000 and 8.2 million are intra-kinase and inter-kinase pairs, respectively. (b) We compare fingerprint distances (based on spatial distance fingerprint bits only) for structure pairs representing the BRAF kinase in different DFG conformations. The dataset includes 28 DFG-in and 21 DFG-out structures, resulting in 378 DFG-in/in, 210 DFG-out/out, and 588 DFG-in/out pairwise structure distances. The box-and-whisker plot extends from the Q1 to Q3 quartile values of the data; the whiskers extend no more than  $1.5 \cdot \text{IQR}$  with  $\text{IQR} = Q_3 - Q_1$ . See notebooks for more details.<sup>62,63</sup>



**Figure 6.** KiSSim-based kinome tree based on 257 structurally resolved kinases in the DFG-in conformation. Tree nodes are colored from red to blue, showing small to large distances (0.01–0.20), describing high to low similarities; tree leaves represent kinases colored by the kinase group. The tree is based on a clustering of the kinase distance matrix using the Euclidean distance as a metric and Ward's criterion as linkage. The clusters are converted to the Newick format and visualized using FigTree.<sup>34</sup> See the notebook for more details.<sup>65</sup>

First, we measured the discriminating power between kinases by comparing KiSSim fingerprint distances between DFG-in structures of the same kinase and of different kinases, i.e., intra-kinase and inter-kinase distances, respectively. With a median of 0.02 compared to 0.11, the intra-kinase distances (about 200,000) are significantly lower than the inter-kinase distances (about 8.2 million) as shown in Figure 5a, indicating that the fingerprint can discriminate between kinases. Note that the distances between structure pairs describing the same kinase pair can vary a lot (Figure S4); for the all-against-all comparison, we consider only the most similar structure pair per kinase pair.

Second, we measured KiSSim's discriminating power between DFG conformations by comparing fingerprint distances between structure pairs in DFG-in/in, DFG-out/out, and DFG-in/out conformations. For this analysis, we used the distances based on only the spatial fingerprint features to exclude the eight physicochemical features and therefore to

focus on conformational information. While the distributions for the three categories are similar when considering all kinases, they differ when split by kinase as shown exemplarily for the BRAF kinase in Figure 5b, indicating that the fingerprint can discriminate between DFG conformations. We conducted this analysis for other kinases with sufficient structural coverage for DFG-in and DFG-out conformations and observed the same for CDK8, EphA2, MET, and p38a (see details in the notebook<sup>62</sup>).

Before we use the KiSSim fingerprints for an all-against-all comparison, we confirmed two important properties: First, the KiSSim fingerprint distances for structures describing the same kinase are significantly lower than for structures describing different kinases (here based on DFG-in structures only). Second, the fingerprint distances for structures in the same DFG conformation are lower than for DFG-in/out structure pairs (here based on spatial features only).

**KiSSim-Based Kinome Tree.** A structure is known to be more conserved than a sequence,<sup>64</sup> and previous studies have shown that including structural information adds orthogonal information to shed light on unexpected similarities between kinases and off-target effects.<sup>7,12</sup> To help detect such relationships between more distantly related kinases, we generated KiSSim kinome trees based on the DFG-in conformations, as described in detail in the **KiSSim Tree** section, to investigate all-against-all relationships between kinases compared to the sequence-based kinome tree by Manning et al.<sup>6</sup> (Figure 6). Note that we can base the comparison on structurally resolved kinases only, i.e., 257 out of the roughly 500 human kinases.

The KiSSim-based kinome tree (structure-based) shows a large overlap with most kinase groups as annotated by Manning et al.<sup>6</sup> (sequence-based). We will summarize the KiSSim clusters and highlight differences in comparison to Manning et al.'s kinase groups AGC, CAMK, CK1, CMGC, STE, TKL, TK, the atypical group, and the unassigned kinases (Other).

Kinases from the TK group build a single large cluster with two outliers, i.e., the pseudokinases TYK2-b and JAK1-b. Known highly similar kinases, which form (sub)families in the Manning tree, are grouped together, e.g., the families Erb (EGFR, Erb2, Erb3, and Erb4), Eph (EphB[1,4] and EphA[1,2,3,5,7,8]), JakA (JAK1, JAK2, JAK2, and TYK2), and JakB (JAK1-b and TYK2-b).

Kinases from the CAMK group mainly cluster together. In addition, the following kinases from other kinase groups are included in our CAMK-like cluster: (a) CaMKK2 (Other), (b) MSK1 (AGC), (c) CK2a2 (CMGC), and (d) AurA, AurC, PLK4, TTK, and MPSK1 (Other). This is partly in agreement with the findings by Modi and Dunbrack<sup>7</sup> who have reassigned 10 kinases from Manning's Other group to the CAMK group, of which seven are part of the KiSSim dataset (AurA, AurC, CaMKK2, PLK1, PLK2, PLK3, and PLK4) and three are not (AurB, CaMKK1, and PLK5). The KiSSim-based similarity of CaMKK2 to CAMK kinases is further supported by profiling data for the chemical probe SGC-STK17B-1, which targets both CaMKK2 and DRAK2 (part of the CAMK group).<sup>9</sup> Note that the following kinases belong to the CAMK group but are found outside of our CAMK-like cluster: (a) Trb1, (b) LKB1, and (c) PASK, PIM1, and PIM2.

Kinases from the STE group are assigned mostly to a single cluster that is, however, shared with kinases from many other kinase groups. The STE kinases MAP2K[1,4,6,7] and OSR1 are separated from the other STE kinases.

Kinases from the CMGC group are clustered in two subgroups: kinases from the CDK, CDKL, and MAPK families build one cluster, while kinases from the DYRK, SRPK, and CLK families build another. The CK2a2 kinase (CK2 family) is an outlier.

Kinases from the TKL group are mainly clustered together with kinases from the Other group, but some are separated from the rest (DLK, BRAK, IRAK2, and LIMK1). Kinases from the CK1 group build one group except for TTBK1 and TTBK2. Kinases from the AGC group cluster together as well; MSK1 is the only outlier that is found closer to the CAMK kinases. Last, only three atypical kinases are included in the KiSSim dataset (ADCK3, RIOK1, and RIOK2) and build their own cluster, neighboring to the CK1 kinases.

Overall, the KiSSim dataset retrieves the sequence-based kinome tree by Manning et al.<sup>6</sup> including sub-branches as

discussed for the kinases assigned to the TK and CMGC groups. This is not surprising because we do encode the sequence in an abstracted manner in the physicochemical KiSSim fingerprint bits. However, some kinases show deviating relationships, of which some can be rationalized such as the CaMKK2 and DRAK2 relationship shown also in profiling data. Thus, the addition of structural information in the KiSSim fingerprint allows us to cluster more distantly related kinases. This aspect of the KiSSim tree is of interest because it predicts novel information on kinase similarities.

**KiSSim Evaluation Using Profiling Data.** As discussed, the KiSSim tree shows expected and unexpected kinase (dis)similarities. To evaluate the specificity and sensitivity of our method, we use profiling data as a surrogate for (real) expected kinase (dis)similarities: if a ligand targets a set of kinases with high activity, these kinases have similar binding sites and are therefore treated as similar kinases.

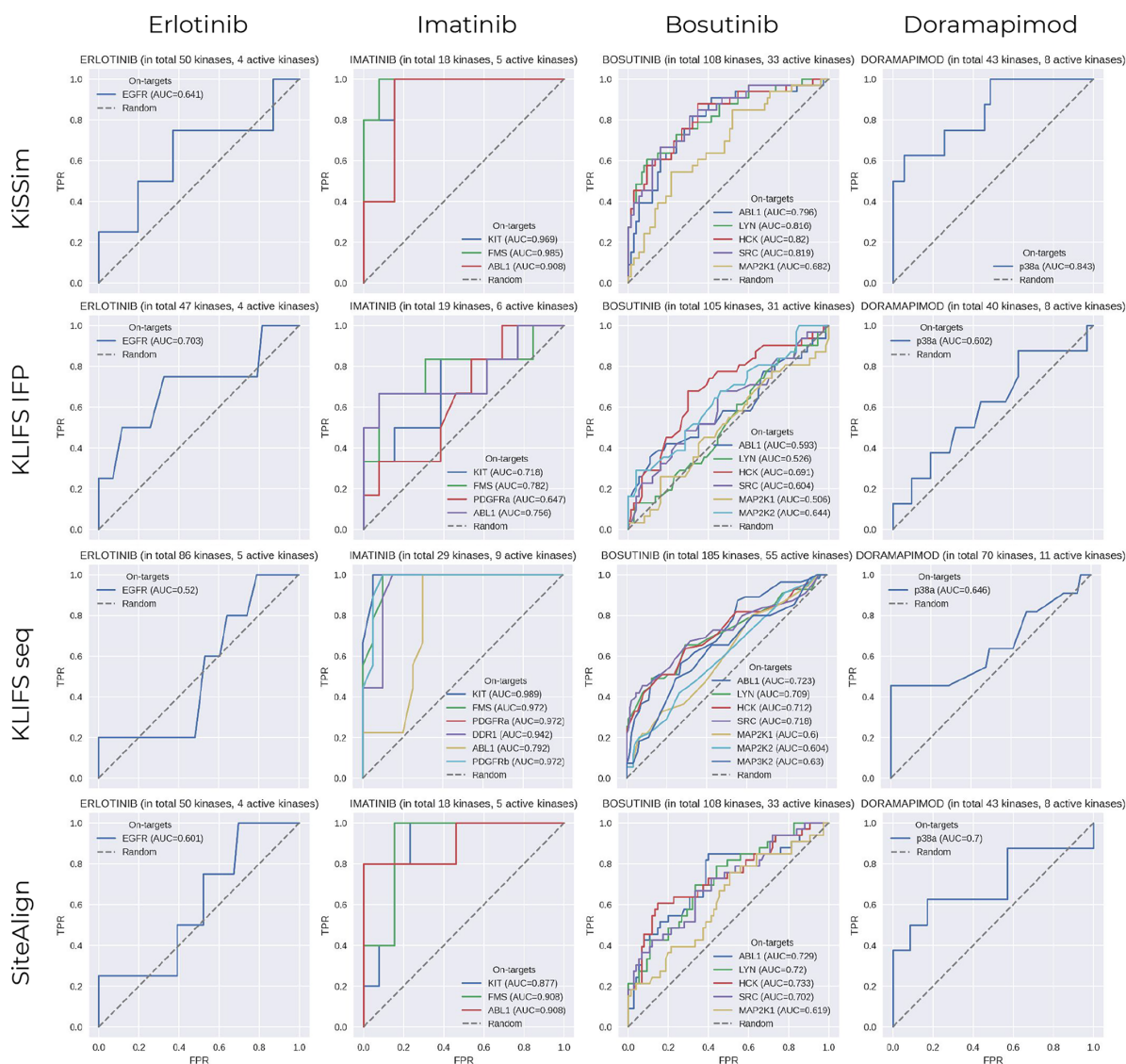
To this end, we pooled the Karaman et al.<sup>8</sup> and Davis et al.<sup>53</sup> datasets and filtered for inhibitors and their targets as listed in the PKIDB.<sup>16</sup> The dataset preparation is described in detail in the **Bioactivity Profiling Data** section. We show the KiSSim method's performance in the form of ROC curves for each inhibitor's listed targets.

For example, Imatinib has three reported on-targets (assigned in PKIDB) and two off-targets (based on activity data in the Karaman–Davis dataset); KiSSim's performance is evaluated by looking up these five Imatinib targets in KiSSim's most similar kinases with respect to the on-targets (1) ABL1, (2) KIT, and (3) FMS, producing three ROC curves (Figure 7, first row, second plot). Details are described in the **KiSSim Evaluation Using Profiling Data** section. In total, we analyzed KiSSim's performance across 48 kinase–ligand pairs involving 21 ligands; the AUCs range from 0.49 to 1.0 with a mean of  $0.75 \pm 0.12$ . In the following, we discuss a few examples in Figure 7 (first row); please refer to the full set of ligands in Figure S8.

The Erlotinib profiling and KiSSim datasets share 50 kinases, of which 4 show high activity ( $K_d \leq 100$  nM), i.e., the on-target EGFR (TK,  $K_d = 19.0$  nM) and the off-targets SLK (STE,  $K_d = 3.10$  nM), LOK (STE,  $K_d = 0.67$  nM), and GAK (Other,  $K_d = 0.67$  nM). The top 20 KiSSim ranks for EGFR are dominated by TK kinases but include the STE kinases LOK and SLK on ranks 11 and 20 out of the 50 shared kinases, respectively; the GAK kinase is not detected by KiSSim, being found on rank 44 only (AUC = 0.641). The EGFR–GAK fingerprint pair shows many differences in their physicochemical bits, which stem from their relatively high pocket sequence dissimilarity (Figure 8). The fingerprint differences for the EGFR–GAK pair are visualized in 3D in Figure 9 for selected fingerprint features with high differences such as the HBA, aliphatic, and hinge region features. Such a comparison of fingerprint values in 3D can provide insights into the rational design of selective inhibitors.

The Imatinib profiling and KiSSim datasets share 18 kinases, of which 5 TK kinases show high activity, i.e., the key target ABL1 as well as ABL2, LCK, KIT, and FMS. Compared to ABL1, all active kinases are ranked within KiSSim's top 7 most similar kinases (AUC = 0.908).

The Bosutinib profiling and KiSSim datasets share 108 kinases, of which 33 show high activity, mainly from the TK and STE groups. Compared to ABL1, which is one of the key targets, the TK kinases are found first in the top 35, followed by the STE kinases in the top 61 (AUC = 0.796).



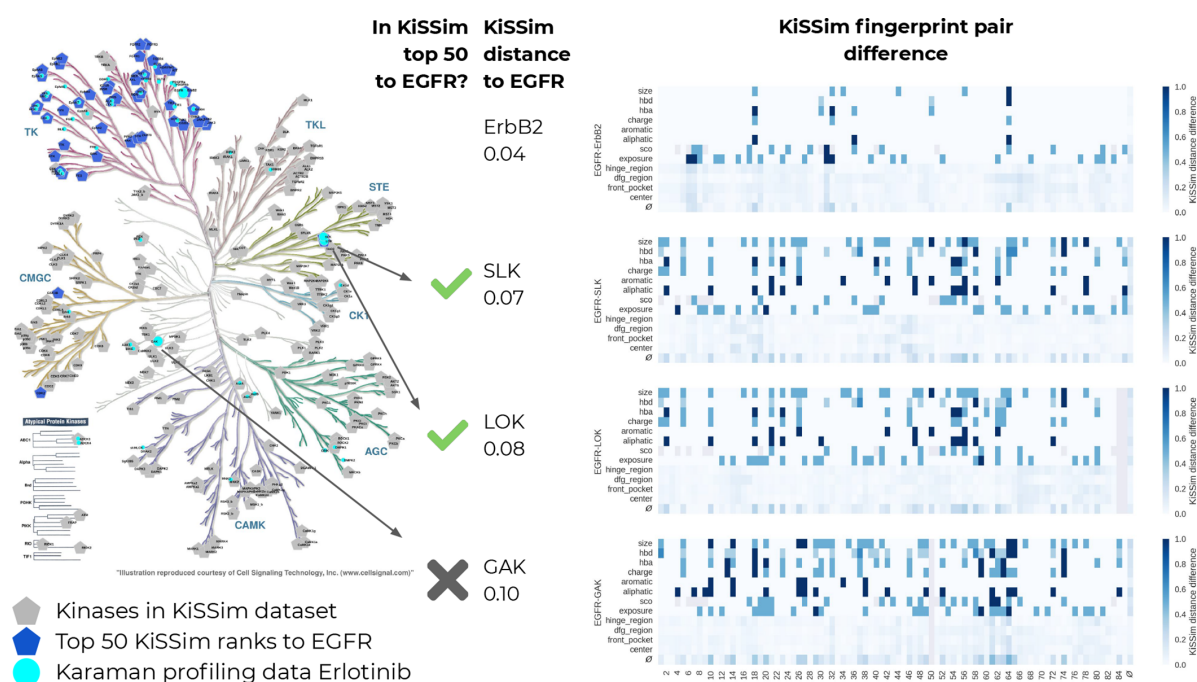
**Figure 7.** Performance of KiSSim and other similarity measures against profiling data. ROC curves comparing predicted and profiling-based kinase similarities (FPR = false positive rate; TPR = true positive rate). Predicted similarities against a selected kinase  $k$  are based on the KiSSim similarities (KiSSim), the KLIFS pocket IFP similarity (KLIFS IFP), the KLIFS pocket sequence identity (KLIFS seq), and the SiteAlign pocket structure similarity (SiteAlign). Profiling-based kinase similarities define kinases as similar if they are targeted by the same ligand with  $K_d \leq 100$  nM, including the ligand's on-target(s) as reported in the PKIDB. The kinases, for which the ligand shows lower activities with  $K_d > 100$  nM, are treated as dissimilar to the ligand's on-target(s). Find more details in the [Bioactivity Profiling Data](#) section. The first rank is always occupied by the kinase  $k$ . We show here only a selection of kinase–ligand pairs; please refer to [Figures S8–S11](#) to inspect the full datasets. See notebooks for more details.<sup>66–70</sup>

The Doramapimod profiling and KiSSim datasets share 43 kinases, of which 8 show high activity, including the on-target p38a and four additional CMGC kinases (p38b, p38d, p38g, and JNK2), two STE kinases (HGK and LOK), and the TK kinase TIE2. Compared to p38a, the CMGC kinases cover the top 7 KiSSim ranks, followed by the STE kinases and TIE2 in the top 25 (AUC = 0.845).

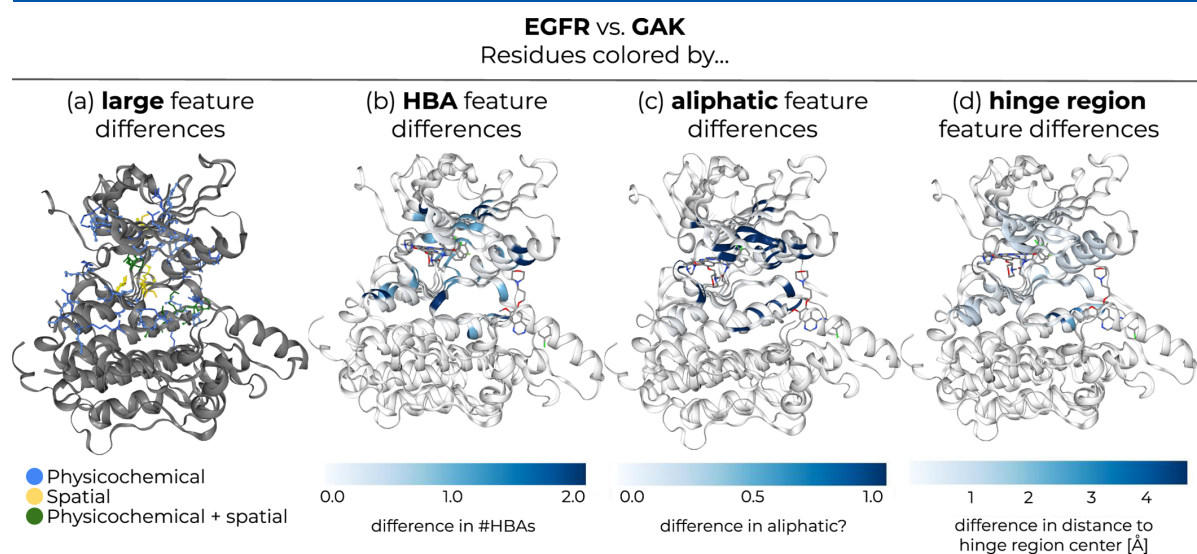
Using profiling data as an estimate for binding site similarity comes with three challenges: First, some ligands are more promiscuous than others because of their chemical structures.

Profiling data for a selective ligand is not easily comparable to data from a less selective ligand and therefore does not necessarily reflect the degree of binding site similarity. In this complex problem, KiSSim can only answer part of the question: KiSSim highlights potential off-targets based on (target-focused) pocket similarities but does not imply that any inhibitor binding to a target will also bind to the closely related target. On the contrary, it helps to identify those targets that one should take into account to possibly prevent off-target





**Figure 8.** KiSSim similarities between EGFR and Erlotinib's off-targets SLK, LOK, and GAK. (Left) The KinMap<sup>56</sup> tree shows the Karaman profiling data for Erlotinib (cyan), the top 50 most similar kinases to Erlotinib's on-target EGFR (blue), and all kinases that are covered by the KiSSim dataset (gray). (Right) KiSSim fingerprint pair differences between EGFR and selected kinases: ErbB2 (as an example for highly similar kinases) as well as SLK, LOK, and GAK (unexpected off-targets for Erlotinib). Similarities between EGFR and SLK/LOK are detected by KiSSim (top 50 of all 279 kinases covered in KiSSim), while GAK stays undetected due to higher differences in the overall KiSSim fingerprints. See the notebook for more details.<sup>75</sup>



**Figure 9.** 3D visualization of KiSSim fingerprint differences between EGFR and GAK (EGFR and GAK structure KLIFS IDs: 12159<sup>76</sup> and 10329,<sup>77</sup> respectively). (a) Highlighted residues with at least one large difference in their physicochemical bits ( $\Delta d_{\text{normalized}} = 0.6$ , blue), spatial bits ( $\Delta d_{\text{normalized}} = 0.2$ , yellow), or both (green). Colored residues by their differences in their (b) HBA, (c) aliphatic, and (d) hinge region features, ranging from no difference (white) to the highest difference (blue). See the notebook for more details.<sup>75</sup> The 3D visualization is part of the kissim Python library using the NGLviewer (usable in e.g. Jupyter notebooks).

effects during the design process and to drive selectivity optimization.

Second, the prediction tasks evaluated with the ROC curves may vary in difficulty based on data availability: (a) Generally,

only a few data points are available for this analysis. (b) Erlotinib-based vs Imatinib-based evaluations stem from predictions across different kinase groups vs within the TK group only. (c) Erlotinib-based vs Bosutinib-based evaluations are based on a dataset with a share of active kinases of 1 out of 10 and 1 out of 3, respectively.

Third, the evaluation results will vary based on the experimental dataset used. Besides the pooled Karaman–Davis dataset discussed here, we also evaluated KiSSim based on the Moret et al. dataset (Figure S12),<sup>71</sup> resulting in similar mean AUC values, i.e.,  $0.75 \pm 0.12$  (Karaman–Davis) and  $0.74 \pm 0.14$  (Moret et al.), while predictions for some ligands perform better, e.g., 0.641 and 0.836 (Erlotinib–EGFR; ratios of tested/active kinases are 50/4 and 16/5), or worse, e.g., 0.908 and 0.4 (Imatinib–ABL1; ratios of tested/active kinases are 18/5 and 6/5). See the notebook<sup>72</sup> for more details.

**Applying KiSSim to Residue Subsets.** We applied the KiSSim methodology to different residue subsets: (a) residues involved in binding individual ligands, (b) residues that show frequent interactions in kinase–ligand complexes, and (c) residues that have been identified by Martin and Mukherjee<sup>26</sup> as “privileged” kinase pocket residues.

(a) We performed the same profiling-based evaluations for subset KiSSim fingerprints, solely including residues that interact with the respective ligand were included. Ligand-interacting residues were selected from X-ray kinase structures based on the KLIFS IFP, i.e., 12, 57, 26, and 13 structures have cumulatively 21, 31, 27, and 35 interacting residues with Erlotinib, Imatinib, Bosutinib, and Doramapimod, respectively. In the case of Erlotinib and Bosutinib, the performance improves when including only the ligand-interacting residues—LOK, SLK, and the previously KiSSim-undetected GAK are all in the top 20 most kinase similarities compared to EGFR—while the performance decreases slightly in the case of Imatinib and Doramapimod (see the notebook<sup>73</sup> for more details). Thus, depending on the user’s research question such as predicting off-target for one or multiple ligands of interest, known interaction profiles can be used to guide the selection of residues for the KiSSim fingerprint.

(b) We provide subsets of the 85 KLIFS residues based on  $\geq 1\%$  interaction frequency across the unique kinase–ligand combinations in our KiSSim (calculated based on the available KLIFS interaction fingerprints), i.e., 51, 56, or 65 residues if taking into account only DFG-in, only DFG-in, or all structures as listed in Figure S5. The resulting KiSSim kinome tree in Figure S5 is overall similar to the clustering in Figure 6.

(c) We apply the KiSSim methodology to the residue subset published by Martin and Mukherjee<sup>26</sup> comprising 16 residues, which could all be mapped to the KLIFS residue numbering (see the notebook for more details<sup>74</sup>). The resulting KiSSim kinome tree in Figure S6 overall clusters kinase groups together. Using the residue subset seems to be more suitable than the full residue set (Figure 6) to find the high proximity between EGFR and SLK/LOK, while the full residue set seems to be more suited than the subset to find the relationship between CaMKK2 and the CAMK kinases as discussed before.

**Comparison of KiSSim to Other Methods.** In the next step, we investigated all-against-all comparisons based on the KiSSim fingerprints, the KLIFS pocket sequence, KLIFS ligand–pocket interaction fingerprints (IFP), and the SiteAlign scores. The data preparation steps are described in detail in the [KiSSim Comparison to Other Methods](#) section.

The KiSSim fingerprint contains physicochemical bits, which generalize the pocket sequence, and spatial bits, which consider the individual atom/residue positions in the underlying kinase conformations. First, we use the KLIFS pocket sequence (KLIFS seq) to probe if the KiSSim fingerprint’s generalized sequence and spatial information improve predictions compared to sequence information only. Second, we use the KLIFS pocket IFP (KLIFS IFP) to probe if the KiSSim fingerprint, which does not contain any information about interactions, improves kinase similarity predictions compared to interaction-based fingerprints. The advantage of IFPs is that they emphasize important residues and interactions as seen based on one or more ligands; the disadvantage is that not all possibly relevant interactions have been seen yet. Note that combining the IFP information with KiSSim—using only interacting residues in the KiSSim fingerprint—can improve the KiSSim performance as discussed in the [KiSSim Evaluation Using Profiling Data](#) section. Third, we use kinase similarities calculated with the SiteAlign methodology (SiteAlign), from which we adapted some of the physicochemical KiSSim features, to confirm that the KiSSim fingerprint adds relevant kinase-focused information.

**Correlation.** We compared the pairwise kinase distances between the four different method setups (Figure S13). We observed a rather strong correlation between the KiSSim distances and (a) the KLIFS pocket sequence distances ( $r = 0.77$ ), reflecting the sequence-generalizing physicochemical features in the KiSSim fingerprint, and (b) the SiteAlign distances ( $r = 0.73$ ), reflecting the partly shared physicochemical features in KiSSim and SiteAlign (pharmacophoric and size features). In contrast, the correlation between KiSSim and KLIFS IFP distances is low ( $r = 0.39$ ), possibly reflecting the lack of information on ligand–kinase interaction patterns.

**Performance.** We performed the same profiling analysis, which we discussed for KiSSim (mean AUC,  $0.75 \pm 0.12$ ) in the [KiSSim Evaluation Using Profiling Data](#) section, for the KLIFS seq (mean AUC,  $0.78 \pm 0.15$ ), KLIFS IFP (mean AUC,  $0.63 \pm 0.12$ ), and SiteAlign (mean AUC,  $0.71 \pm 0.12$ ) datasets (see Figure 7).

The KiSSim approach performs slightly worse compared to the KLIFS pocket sequence comparison in the case of ligands like Imatinib, whose reported on-targets all belong to the TK group, but shows better performance for Erlotinib, Bosutinib, and Doramapimod, which have known kinase targets belonging to different kinase groups. Hence, while the sequence-based approach picks up kinase group assignments as to be expected, KiSSim picks up more distant and less obvious off-targets.

The KLIFS pocket IFP comparison performs similarly to the KiSSim comparison in the case of Erlotinib; however, it performs worse for the other three ligands. In contrast to the KiSSim approach, pocket similarities can only be detected by the IFP approach if the respective kinases have been cocrystallized with ligands that form similar interaction patterns. Such an IFP-based comparison probably can be more successful for a defined kinase set with a high coverage of cocrystallized ligands in contrast to a kinome-wide comparison as performed here.

The SiteAlign methodology projects topological and chemical properties onto a sphere that sits in the center of a protein pocket. The spheres are aligned based on these projections, and a similarity score is calculated between the aligned fingerprints. Finding the right alignment is a time-

consuming step; hence, we offered SiteAlign already the KLIFS-aligned structures as a starting point and reduced the iterations as described in the [KiSSim Comparison to Other Methods](#) section. KiSSim outperforms the SiteAlign results in most cases, however, often not considerably much.

**Runtime.** The runtime for the methods discussed here differs considerably: Generating the KLIFS seq dataset takes about a second (based on about 500 kinases), while the KLIFS IFP dataset is ready within half a minute (based on about 8800 IFPs); both procedures build on the processed and curated KLIFS datasets, i.e., both the pocket sequences and the pocket interaction fingerprints are ready for use. Generating the KiSSim kinase matrix takes about 24 h, while the all-against-all comparison with SiteAlign is ready after >20000 h using the optimized SiteAlign settings (both based on over 4000 structures and a single-core/thread execution). Parallelization is built in for the KiSSim approach to speed up the calculation.

Taking all these findings together, the KiSSim methodology compares well with established methods while often improving predictions between kinase pairs without an obvious relationship based on the sequence. The pocket sequence and IFP-based methods are much faster than the structure-based methods KiSSim and SiteAlign; however, the overall kinase similarity assessment benefits from the added structural pocket information. KiSSim's setup and runtime are more convenient than those of the SiteAlign method; however, KiSSim does rely on the KLIFS 85-residue pocket alignment.

## CONCLUSIONS

We presented here the KiSSim (Kinase Structural Similarity) fingerprint as a novel structure-enabled pocket encoding tailored to kinase pockets. The fingerprint encodes physicochemical and spatial properties of the 85 KLIFS residues, which are aligned across the structurally covered kinome. On the one hand, the majority of physicochemical bits—size, HBD, HBA, charge, aromatic, and aliphatic, which are adapted from the SiteAlign method—encode the pocket sequence in a generalized, pharmacophoric way. On the other hand, the side chain orientation, solvent exposure, and the spatial bits—the distances to the pocket center and key subpocket centers and the distance distributions' moments—account for the structural conformation. Across all fingerprints, we saw that the fingerprint captures the physicochemical property variability (e.g., most residues are uncharged, whereas HBD/HBA features vary) and the conserved residue positions (e.g., distances to the DFG region are more widely spread than to the hinge region).

We used the fingerprint to calculate all-against-all distances—small distances refer to high similarity, and large distances refer to low similarity—within the structurally covered kinome: the DFG-in and DFG-out datasets consist of 4112 and 406 structures, representing 257 and 71 kinases, respectively. We found that the fingerprint can distinguish between intra- and inter-kinase similarities and between DFG-in and DFG-out structures.

Some kinases are represented by multiple structures; hence, some kinase pairs are represented by multiple structure pairs. The distribution of structure distances for one kinase pair can be broad; we selected per kinase pair the closest structure pair that is experimentally observed. We clustered the resulting kinase distance matrix to produce a KiSSim-based kinome tree. While the tree reproduced large parts of the sequence-based Manning tree, some relationships could be observed that are

unexpected from a sequence perspective only. For example, we found similarities between CaMKK2 (STE) and DRAK2 (CAMK), which are targeted by the same chemical probe SGC-STK17B-1;<sup>9</sup> we also could confirm the reassignment of AurA, AurC, PLK4, and CaMMK2 from the Other group to the CAMK group as proposed by Modi and Dunbrack.<sup>7</sup>

Besides the averaged tree view, we also investigated the top-ranked kinases given a query kinase to show that KiSSim can partially explain profiling data. While some ligand profiles are reflected completely in the KiSSim dataset (e.g., Imatinib), other ligand profiles are covered partially (e.g., Erlotinib's off-targets LOK and SLK are detected, while GAK is not).

In comparison with other similarity measures—focusing on the pocket sequence (KLIFS seq), interaction profiles (KLIFS IFP), or topological and chemical pocket properties (SiteAlign)—KiSSim performs equally or slightly better in most cases. The sequence- and IFP-based measures are easy and fast to compute thanks to the preprocessed kinase pockets available at KLIFS; we recommend including these datasets in any case when investigating kinase similarities. SiteAlign is a powerful tool to compare pockets across all protein classes; if interested only in kinases, KiSSim is a kinase-focused and faster alternative with slightly better results in most of the investigated cases.

As for all structure-based methods, the imbalanced dataset of kinase structures is a challenge. Some kinases are structurally well represented (e.g., EGFR or CDK2), while others have only few structures available. Also, unfortunately, still roughly half of the human kinome has no structural information available at all. The recent breakthrough of AlphaFold2<sup>78</sup> could help here; predicted structures for almost all human kinases are available now on the AlphaFold DB.<sup>79</sup> Modi and Dunbrack<sup>80</sup> have already classified the structures' conformations and found most structures in the DFG-in conformation. An AlphaFold-enhanced KiSSim tree may further increase the usefulness of the KiSSim methodology for kinome-wide similarity studies. Furthermore, the KiSSim fingerprint can be applied in machine learning, e.g., to extract the most important features in the kinase pocket.

KiSSim is a target-focused methodology and is applied primarily in the context of off-target prediction. The method can flag targets with similar binding sites beyond the traditional sequence identity and similarity measures, which are usually applied during the target traceability phases of drug design campaigns. Beyond this purpose, KiSSim can help highlight target-based structural differences between a set of targets. During hit optimization—once the ligand binding pose is assessed, e.g., via docking studies—the target's binding site is usually inspected intensively for potential ligand modifications based on unoccupied subpockets or promising interactions. At this stage, KiSSim can be used to explore such opportunities in comparison to other off- or on-targets. KiSSim will not be able to accurately predict kinase inhibitor selectivity, but it can serve as an idea generator. For a set of two or more kinases, for which selectivity shall be achieved, differences in KiSSim fingerprints can be visualized in 3D (see example in [Figure 9](#)) to explore opportunities to modify or extend compounds in kinase drug discovery projects. Such an analysis would probably be jointly performed between medicinal chemists and computational chemists; the 3D visualization for KiSSim is executable as of now from Jupyter notebooks using the NGLviewer; thus, technical support might be needed from a computational chemist.

We believe that the KiSSim fingerprint is a valuable tool for kinase research to explain and predict off-targets and polypharmacology. Since the code is open-sourced and available as a Python package, the KiSSim fingerprint can easily be integrated in other larger-scale workflows.

## ■ ASSOCIATED CONTENT

### Supporting Information

The Supporting Information is available free of charge at <https://pubs.acs.org/doi/10.1021/acs.jcim.2c00050>.

Detailed description of the KiSSim fingerprint; Table S1: SiteAlign categories for standard amino acids (one-letter code); Table S2: list of nonstandard amino acids and their parent standard amino acids used in KiSSim; Table S3: list of standard amino acids and their side chain representatives as defined for the KiSSim side chain orientation feature calculation; Table S4: KLIFS residue IDs used to calculate the CA atoms' centroid defining the KiSSim pocket center and subpocket centers; Table S5: subsets of KLIFS pocket residues; Table S6: KiSSim dataset: filtering steps and final number of structures and kinases as well as final number of structure and kinase pairs; Figure S1: number of structures with missing residues; Figure S2: physicochemical feature distribution per residue position across all fingerprint pairs; Figure S3: spatial distance feature distribution per residue position across all fingerprint pairs; Figure S4: structure pair distances for the most frequent kinases; Figure S5: KiSSim-based kinome tree focused on a subset of residues, i.e., residues frequently interacting with ligands as seen in DFG-in structures; Figure S6: KiSSim-based kinome tree focused on a subset of residues, i.e., pocket residues as identified by Martin and Mukherjee;<sup>26</sup> Figure S7: coloring scheme for kinase groups as used in the KiSSim kinome trees; Figure S8: KiSSim performance against profiling data; Figure S9: KLIFS IFP performance against profiling data; Figure S10: KLIFS sequence performance against profiling data; Figure S11: SiteAlign performance against profiling data; Figure S12: KiSSim performance against Moret et al. profiling data; Figure S13: comparison of distance values for pairwise kinase structure comparisons using KiSSim, KLIFS pocket sequence, KLIFS pocket IFP, and SiteAlign (PDF)

## ■ AUTHOR INFORMATION

### Corresponding Author

**Andrea Volkamer** – *In Silico Toxicology and Structural Bioinformatics, Institute of Physiology, Charité–Universitätsmedizin Berlin, corporate member of Freie Universität Berlin and Humboldt-Universität zu Berlin, 13353 Berlin, Germany; [orcid.org/0000-0002-3760-580X](https://orcid.org/0000-0002-3760-580X); Email: [andrea.volkamer@charite.de](mailto:andrea.volkamer@charite.de)*

### Authors

**Dominique Sydow** – *In Silico Toxicology and Structural Bioinformatics, Institute of Physiology, Charité–Universitätsmedizin Berlin, corporate member of Freie Universität Berlin and Humboldt-Universität zu Berlin, 13353 Berlin, Germany; [orcid.org/0000-0003-4205-8705](https://orcid.org/0000-0003-4205-8705)*

**Eva Aßmann** – *In Silico Toxicology and Structural Bioinformatics, Institute of Physiology,*

*Charité–Universitätsmedizin Berlin, corporate member of Freie Universität Berlin and Humboldt-Universität zu Berlin, 13353 Berlin, Germany; [orcid.org/0000-0002-7249-069X](https://orcid.org/0000-0002-7249-069X)*

**Albert J. Kooistra** – *Department of Drug Design and Pharmacology, University of Copenhagen, 2100 Copenhagen, Denmark; [orcid.org/0000-0001-5514-6021](https://orcid.org/0000-0001-5514-6021)*

**Friedrich Rippmann** – *Computational Chemistry & Biologics, Merck Healthcare KGaA, 64293 Darmstadt, Germany; [orcid.org/0000-0002-4604-9251](https://orcid.org/0000-0002-4604-9251)*

Complete contact information is available at: <https://pubs.acs.org/doi/10.1021/acs.jcim.2c00050>

## Author Contributions

Conceptualization, D.S., A.V.; Data Curation, D.S., A.K.; Formal Analysis, D.S., E.A., A.K., A.V.; Funding Acquisition, A.V.; Investigation, D.S., E.A., A.K., A.V.; Methodology, D.S., E.A., A.V.; Project Administration, D.S., A.V.; Resources, A.V.; Software, D.S.; Supervision, A.K., F.R., A.V.; Validation, D.S., A.V.; Visualization, D.S.; Writing: Original Draft, D.S., A.V.; Writing: Review and Editing, D.S., E.A., A.K., F.R., A.V.

## Notes

The authors declare no competing financial interest.

KiSSim library (kissim): <https://github.com/volkamerlab/kissim> and <https://kissim.readthedocs.io>; KiSSim datasets: <https://doi.org/10.5281/zenodo.5774521>; KiSSim application and analyses (kissim\_app): [https://github.com/volkamerlab/kissim\\_app](https://github.com/volkamerlab/kissim_app).

## ■ ACKNOWLEDGMENTS

D.S. thanks Talia B. Kimber for insightful and motivating discussions about the more mathematical aspects of this project. D.S. thanks Jaime Rodríguez-Guerra for bringing best software practices into the lab and for helpful and enthusiastic Python conversations. A.V. and D.S. gratefully acknowledge funding from the Deutsche Forschungsgemeinschaft (grant VO 2353/1-1). A.V. acknowledges support from the Bundesministerium für Bildung und Forschung (grant 031A262C). A.V., D.S., and E.A. thank the HPC service of ZEDAT, Freie Universität Berlin,<sup>81</sup> for cluster time and support.

## ■ LIST OF ABBREVIATIONS

KiSSim, Kinase Structural Similarity; ATP, adenosine triphosphate; IFP, interaction fingerprint; DFG, asparagine-phenylalanine-glycine; MSA, multiple sequence alignment; SCO, side chain orientation; HBD, hydrogen bond donors (here: number of HBD); HBA, hydrogen bond acceptors (here: number of HBA); ROC, receiver operating characteristic; AUC, area under the curve

## ■ REFERENCES

- (1) Cohen, P.; Alessi, D. R. Kinase Drug Discovery - What's Next in the Field? *ACS Chem. Biol.* **2013**, *8*, 96–104.
- (2) Santos, R.; Ursu, O.; Gaulton, A.; Bento, A. P.; Donadi, R. S.; Bologa, C. G.; Karlsson, A.; Al-Lazikani, B.; Hersey, A.; Oprea, T. I.; Overington, J. P. A Comprehensive Map of Molecular Drug Targets. *Nat. Rev. Drug Discov.* **2017**, *16*, 19–34.
- (3) Cohen, P.; Cross, D.; Jänne, P. A. Kinase Drug Discovery 20 Years after Imatinib: Progress and Future Directions. *Nat. Rev. Drug Discov.* **2021**, *20*, 551–569.
- (4) Morphy, R. Selectively Nonselective Kinase Inhibition: Striking the Right Balance. *J. Med. Chem.* **2009**, *53*, 1413–1437.

- (5) van Linden, O. P. J.; Kooistra, A. J.; Leurs, R.; de Esch, I. J. P.; de Graaf, C. KLIFS: A Knowledge-Based Structural Database to Navigate Kinase-Ligand Interaction Space. *J. Med. Chem.* **2014**, *57*, 249–277.
- (6) Manning, G.; Whyte, D. B.; Martinez, R.; Hunter, T.; Sudarsanam, S. The Protein Kinase Complement of the Human Genome. *Science* **2002**, *298*, 1912–1934.
- (7) Modi, V.; Dunbrack, R. L., Jr. A Structurally-Validated Multiple Sequence Alignment of 497 Human Protein Kinase Domains. *Sci. Reports* **2019**, *9*, 1–16.
- (8) Karaman, M. W.; Herrgard, S.; Treiber, D. K.; Gallant, P.; Atteridge, C. E.; Campbell, B. T.; Chan, K. W.; Ciceri, P.; Davis, M. I.; Edeen, P. T.; Faraoni, R.; Floyd, M.; Hunt, J. P.; Lockhart, D. J.; Milanov, Z. V.; Morrison, M. J.; Pallares, G.; Patel, H. K.; Pritchard, S.; Wodicka, L. M.; Zarrinkar, P. P. A Quantitative Analysis of Kinase Inhibitor Selectivity. *Nat. Biotechnol.* **2008**, *26*, 127–132.
- (9) Structural Genomics Consortium, SGC-STK17B-1: A Chemical Probe for STK17B/DRAK2 Kinase. <https://www.thesgc.org/chemical-probes/SGC-STK17B-1>, [accessed 2021-08-16].
- (10) Kooistra, A. J.; Volkamer, A. Kinase-Centric Computational Drug Development. *Annu. Rep. Med. Chem.* **2017**, *50*, 197–236.
- (11) KinCore, Phylogeny of Human Protein Kinase Domains. <https://dunbrack3.fccc.edu/kincore/phylogeny>, [accessed 2021-08-11].
- (12) Schmidt, D.; Scharf, M. M.; Sydow, D.; Aßmann, E.; Martí-Solano, M.; Keul, M.; Volkamer, A.; Kolb, P. Analyzing Kinase Similarity in Small Molecule and Protein Structural Space to Explore the Limits of Multi-Target Screening. *Molecules* **2021**, *26*, 629.
- (13) Kuhn, D.; Weskamp, N.; Hüllermeier, E.; Klebe, G. Functional Classification of Protein Kinase Binding Sites Using Cavbase. *ChemMedChem* **2007**, *2*, 1432–1447.
- (14) Kanev, G. K.; de Graaf, C.; Westerman, B. A.; de Esch, I. J. P.; Kooistra, A. J. KLIFS: An Overhaul after the First 5 Years of Supporting Kinase Research. *Nucleic Acids Res.* **2021**, *49*, D562–D569.
- (15) Berman, H. M.; Kleywegt, G. J.; Nakamura, H.; Markley, J. L. The Protein Data Bank at 40: Reflecting on the Past to Prepare for the Future. *Structure* **2012**, *20*, 391–396.
- (16) Carles, F.; Bourg, S.; Meyer, C.; Bonnet, P. PKIDB: A Curated, Annotated and Updated Database of Protein Kinase Inhibitors in Clinical Trials. *Molecules* **2018**, *23*, 908.
- (17) Gaulton, A.; Hersey, A.; Nowotka, M.; Bento, A. P.; Chambers, J.; Mendez, D.; Motow, P.; Atkinson, F.; Bellis, L. J.; Cibrián-Uhalte, E.; Davies, M.; Dedman, N.; Karlsson, A.; Magariños, M. P.; Overington, J. P.; Papadatos, G.; Smit, I.; Leach, A. R. The ChEMBL Database in 2017. *Nucleic Acids Res.* **2017**, *45*, D945–D954.
- (18) Schalon, C.; Surgand, J.-S.; Kellenberger, E.; Rognan, D. A Simple and Fuzzy Method to Align and Compare Druggable Ligand-Binding Sites. *Proteins Struct., Funct., Bioinf.* **2008**, *71*, 1755–1778.
- (19) Sydow, D.; Schmiel, P.; Mortier, J.; Volkamer, A. KinFragLib: Exploring the Kinase Inhibitor Space Using Subpocket-Focused Fragmentation and Recombination. *J. Chem. Inf. Model.* **2020**, *60*, 6081–6094.
- (20) Ballester, P. J.; Richards, W. G. Ultrafast Shape Recognition to Search Compound Databases for Similar Molecular Shapes. *J. Comput. Chem.* **2007**, *28*, 1711–1723.
- (21) Wilkinson, M. D.; Dumontier, M.; Aalbersberg, I. J.; Appleton, G.; Axton, M.; Baak, A.; Blomberg, N.; Boiten, J.-W.; da Silva Santos, L. B.; Bourne, P. E.; Bouwman, J.; Brookes, A. J.; Clark, T.; Crosas, M.; Dillo, I.; Dumon, O.; Edmunds, S.; Evelo, C. T.; Finkers, R.; Gonzalez-Beltran, A.; Gray, A. J.; Groth, P.; Goble, C.; Grethe, J. S.; Heringa, J.; 't Hoen, P. A.; Hooft, R.; Kuhn, T.; Kok, R.; Kok, J.; Lusher, S. J.; Martone, M. E.; Mons, A.; Packer, A. L.; Persson, B.; Rocca-Serra, P.; Roos, M.; van Schaik, R.; Sansone, S.-A.; Schultes, E.; Sengstag, T.; Slater, T.; Strawn, G.; Swertz, M. A.; Thompson, M.; van der Lei, J.; van Mulligen, E.; Velterop, J.; Waagmeester, A.; Wittenburg, P.; Wolstencroft, K.; Zhao, J.; Mons, B. The FAIR Guiding Principles For Scientific Data Management and Stewardship. *Scientific Data* **2016**, *3*, 160018.
- (22) Cock, P. J. A.; Antao, T.; Chang, J. T.; Chapman, B. A.; Cox, C. J.; Dalke, A.; Friedberg, I.; Hamelryck, T.; Kauff, F.; Wilczynski, B.; de Hoon, M. J. L. Biopython: Freely Available Python Tools for Computational Molecular Biology and Bioinformatics. *Bioinformatics* **2009**, *25*, 1422–1423.
- (23) Hamelryck, T. An Amino Acid Has Two Sides: A New 2D Measure Provides a Different View of Solvent Exposure. *Proteins Struct., Funct., Bioinf.* **2005**, *59*, 38–48.
- (24) Volkamer Lab, OpenCADD. <https://github.com/volkamerlab/opencadd>, [accessed 2021-11-27].
- (25) KiSSim, KLIFS Pocket Residue Subsets for DFG-in and DFG-out Conformations. [https://github.com/volkamerlab/kissim/blob/main/kissim/data/klifs\\_pocket\\_residue\\_subset.json](https://github.com/volkamerlab/kissim/blob/main/kissim/data/klifs_pocket_residue_subset.json), [accessed 2021-11-11].
- (26) Martin, E.; Mukherjee, P. Kinase-Kernel Models: Accurate In silico Screening of 4 Million Compounds Across the Entire Human Kinome. *J. Chem. Inf. Model.* **2012**, *52*, 156–170.
- (27) Volkamer Lab, Extrema Used for Min-Max Normalization of KiSSim's Spatial Features. <https://github.com/volkamerlab/kissim/tree/v1.1.0/kissim/data>, Version 1.1.0 [accessed 2022-04-24].
- (28) Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press, 2016; Chapter 2, pp 37–38, <http://www.deeplearningbook.org>.
- (29) Rose, A. S.; Hildebrand, P. W. NGL Viewer: A Web Application for Molecular Visualization. *Nucleic Acids Res.* **2015**, *43*, W576–W579.
- (30) Nguyen, H.; Case, D. A.; Rose, A. S. NGLView - Interactive Molecular Graphics for Jupyter Notebooks. *Bioinformatics* **2017**, *34*, 1241–1242.
- (31) IPyWidgets, *IPyWidgets Documentation*. <https://ipywidgets.readthedocs.io/en/latest/>, [accessed 2021-10-05].
- (32) Virtanen, P.; Gommers, R.; Oliphant, T. E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; van der Walt, S. J.; Brett, M.; Wilson, J.; Millman, K. J.; Mayorov, N.; Nelson, A. R. J.; Jones, E.; Kern, R.; Larson, E.; Carey, C. J.; Polat, İ.; Feng, Y.; Moore, E. W.; VanderPlas, J.; Laxalde, D.; Perktold, J.; Cimrman, R.; Henriksen, I.; Quintero, E. A.; Harris, C. R.; Archibald, A. M.; Ribeiro, A. H.; Pedregosa, F.; van Mulbregt, P.; SciPy 1.0 Contributors; Vijaykumar, A.; Bardelli, A. P.; Rothberg, A.; Hilboll, A.; Kloeckner, A.; Scopatz, A.; Lee, A.; Rokem, A.; Woods, C. N.; Fulton, C.; Masson, C.; Häggström, C.; Fitzgerald, C.; Nicholson, D. A.; Hagen, D. R.; Pasechnik, D. V.; Olivetti, E.; Martin, E.; Wieser, E.; Silva, F.; Lenders, F.; Wilhelm, F.; Young, G.; Price, G. A.; Ingold, G. L.; Allen, G. E.; Lee, G. R.; Audren, H.; Probst, I.; Dietrich, J. P.; Silterra, J.; Webber, J. T.; Slavič, J.; Nothman, J.; Buchner, J.; Kulick, J.; Schönberger, J. L.; de Miranda Cardoso, J. V.; Reimer, J.; Harrington, J.; Rodríguez, J. L. C.; Nunez-Iglesias, J.; Kuczynski, J.; Tritz, K.; Thoma, M.; Newville, M.; Kümmerer, M.; Bolingbroke, M.; Tarte, M.; Pak, M.; Smith, N. J.; Nowaczyk, N.; Shebanov, N.; Pavlyk, O.; Brodtkorb, P. A.; Lee, P.; McGibbon, R. T.; Feldbauer, R.; Lewis, S.; Tygiar, S.; Sievert, S.; Vigna, S.; Peterson, S.; More, S.; Pudlik, T.; Oshima, T.; Pingel, T. J.; Robitaille, T. P.; Spura, T.; Jones, T. R.; Cera, T.; Leslie, T.; Zito, T.; Krauss, T.; Upadhyay, U.; Halchenko, Y. O.; Vázquez-Baeza, Y. SciPy 1.0 Contributors, SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat. Methods* **2020**, *17*, 261–272.
- (33) BioPython, *Bio.Phylo package*. <https://biopython.org/docs/latest/api/Bio.Phylo.html>, [accessed 2021-08-16].
- (34) FigTree, *FigTree*. <http://tree.bio.ed.ac.uk/software/figtree/>, [accessed 2021-08-16].
- (35) Anaconda Software Distribution, *Anaconda Documentation*. <https://docs.anaconda.com/>, [accessed 2021-07-30].
- (36) Conda-Forge Community, *The Conda-Forge Project: Community-Based Software Distribution Built on the Conda Package Format and Ecosystem*. 2015.
- (37) Raschka, S. BioPandas: Working with Molecular Structures in Pandas DataFrames. *J. Open Source Software* **2017**, *2*, 279.
- (38) Harris, C. R.; Millman, K. J.; van der Walt, S. J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N. J.; Kern, R.; Picus, M.; Hoyer, S.; van Kerkwijk, M. H.; Brett, M.; Haldane, A.; del Río, J. F.; Wiebe, M.; Peterson, P.; Gérard-Marchant, P.; Sheppard, K.; Reddy, T.; Weckesser, W.; Abbasi, H.; Gohlke, C.;

- Oliphant, T. E. Array Programming with NumPy. *Nature* **2020**, *585*, 357–362.
- (39) The Pandas Development Team, *pandas-dev/pandas: Pandas*. 2020.
- (40) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, P.; Prettenhofer, M. a.; Weiss, R.; Dubourg, V.; Vanderplas, A.; Passos, J. a.; Cournapeau, D.; Brucher, M.; Perrot, E. M. a. *Duchessnay Scikit-learn: Machine Learning in Python. J Mach. Learn. Res.* **2011**, *12*, 2825–2830.
- (41) Van Rossum, G.; Drake, F. L. *Python 3 Reference Manual*; CreateSpace: Scotts Valley, CA, 2009.
- (42) numpdoc, *numpdoc*. <https://numpdoc.readthedocs.io/en/latest/format.html>, [accessed 2021–11–27].
- (43) Python Software Foundation, *Black: The Uncompromising Python Code Formatter*. <https://github.com/psf/black>, [accessed 2021–10–06].
- (44) Black-nb, *Black-nb: The Uncompromising Code Formatter, for Jupyter Notebooks*. <https://github.com/tomcatling/black-nb>, [accessed 2021–10–06].
- (45) flake8, *flake8*. <https://flake8.pycqa.org/>, [accessed 2021–10–06].
- (46) flake8-nb, *flake8-nb*. <https://flake8-nb.readthedocs.io/>, [accessed 2021–10–06].
- (47) Read the Docs, *Read the Docs*. <https://readthedocs.org/>, [accessed 2021–07–31].
- (48) sphinx, *sphinx - Python Documentation Generator*. <https://www.sphinx-doc.org/>, [accessed 2021–10–06].
- (49) pytest, *pytest*. <https://docs.pytest.org/>, [accessed 2021–10–06].
- (50) CodeCov, *CodeCov*. <https://docs.codecov.com/docs>, [accessed 2021–11–27].
- (51) nbval, *nbval*. <https://nbval.readthedocs.io/en/latest/>, [accessed 2021–10–06].
- (52) GitHub, *GitHub Actions*. <https://docs.github.com/en/actions>, [accessed 2021–10–06].
- (53) Davis, M. I.; Hunt, J. P.; Herrgard, S.; Ciceri, P.; Wodicka, L. M.; Pallares, G.; Hocker, M.; Treiber, D. K.; Zarrinkar, P. P. Comprehensive Analysis of Kinase Inhibitor Selectivity. *Nat. Biotechnol.* **2011**, *29*, 1046–1051.
- (54) Volkamer Lab, *KiSSim notebook: KLIFS Data Preparation and Exploration*. [https://github.com/volkamerlab/kissim\\_app/blob/v1.1.0/notebooks/002\\_structures/001\\_prepare\\_dataset.ipynb](https://github.com/volkamerlab/kissim_app/blob/v1.1.0/notebooks/002_structures/001_prepare_dataset.ipynb), Version 1.1.0 [accessed 2022-04-24].
- (55) Volkamer Lab, *KiSSim Notebook: Loading KiSSim Results*. [https://github.com/volkamerlab/kissim\\_app/blob/v1.1.0/notebooks/001\\_quick\\_start/001\\_quick\\_start\\_kissim.ipynb](https://github.com/volkamerlab/kissim_app/blob/v1.1.0/notebooks/001_quick_start/001_quick_start_kissim.ipynb), Version 1.1.0 [accessed 2022-04-24].
- (56) Eid, S.; Turk, S.; Volkamer, A.; Rippmann, F.; Fulle, S. KinMap: A Web-Based Tool for Interactive Navigation Through Human Kinome Data. *BMC Bioinformatics* **2017**, *18*, 16.
- (57) Sebastian Raschka, *About Min-Max Scaling*. [https://sebastianraschka.com/Articles/2014\\_about\\_feature\\_scaling.html#about-min-max-scaling](https://sebastianraschka.com/Articles/2014_about_feature_scaling.html#about-min-max-scaling), [accessed 2021–11–27].
- (58) Marcou, G.; Rognan, D. Optimizing Fragment and Scaffold Docking by Use of Molecular Interaction Fingerprints. *J. Chem. Inf. Model.* **2007**, *47*, 195–207.
- (59) Volkamer Lab, *KiSSim Notebook: Feature Distributions*. [https://github.com/volkamerlab/kissim\\_app/blob/v1.1.0/notebooks/004\\_fingerprints/003\\_feature\\_distributions.ipynb](https://github.com/volkamerlab/kissim_app/blob/v1.1.0/notebooks/004_fingerprints/003_feature_distributions.ipynb), Version 1.1.0 [accessed 2022-04-24].
- (60) Volkamer Lab, *KiSSim Notebook: Subpocket Center Robustness*. [https://github.com/volkamerlab/kissim\\_app/blob/v1.1.0/notebooks/003\\_subpockets/002\\_subpocket\\_robustness.ipynb](https://github.com/volkamerlab/kissim_app/blob/v1.1.0/notebooks/003_subpockets/002_subpocket_robustness.ipynb), Version 1.1.0 [accessed 2022-04-24].
- (61) Volkamer Lab, *KiSSim Notebook: Influence of Conformations on Subpockets*. [https://github.com/volkamerlab/kissim\\_app/blob/v1.1.0/notebooks/003\\_subpockets/003\\_subpocket\\_vs\\_conformations.ipynb](https://github.com/volkamerlab/kissim_app/blob/v1.1.0/notebooks/003_subpockets/003_subpocket_vs_conformations.ipynb), Version 1.1.0 [accessed 2022-04-24].
- (62) Volkamer Lab, *KiSSim Notebook: Can Fingerprint Distances Discriminate DFG Conformations?* [https://github.com/volkamerlab/kissim\\_app/blob/v1.1.0/notebooks/005\\_comparison/004\\_fingerprint\\_distances\\_vs\\_dfg.ipynb](https://github.com/volkamerlab/kissim_app/blob/v1.1.0/notebooks/005_comparison/004_fingerprint_distances_vs_dfg.ipynb), Version 1.1.0 [accessed 2022-04-24].
- (63) Volkamer Lab, *KiSSim Notebook: Fingerprint Distances Between Structures for the Same Kinase*. [https://github.com/volkamerlab/kissim\\_app/blob/v1.1.0/notebooks/005\\_comparison/005\\_structure\\_kinase\\_mapping.ipynb](https://github.com/volkamerlab/kissim_app/blob/v1.1.0/notebooks/005_comparison/005_structure_kinase_mapping.ipynb), Version 1.1.0 [accessed 2022-04-24].
- (64) Illergård, K.; Ardell, D. H.; Elofsson, A. Structure is Three to Ten Times More Conserved Than Sequence - A Study of Structural Response in Protein Cores. *Proteins Struct., Funct., Bioinf.* **2009**, *77*, 499–508.
- (65) Volkamer Lab, *KiSSim Notebook: KiSSim-Based Kinome Tree*. [https://github.com/volkamerlab/kissim\\_app/blob/v1.1.0/notebooks/005\\_comparison/006\\_kissim\\_kinome\\_tree.ipynb](https://github.com/volkamerlab/kissim_app/blob/v1.1.0/notebooks/005_comparison/006_kissim_kinome_tree.ipynb), Version 1.1.0 [accessed 2022-04-24].
- (66) Volkamer Lab, *KiSSim Notebook: Predict Ligand Profiling Using KiSSim (Pooled Karaman and Davis Dataset)*. [https://github.com/volkamerlab/kissim\\_app/blob/v1.1.0/notebooks/006\\_evaluation/004\\_profiling\\_karaman\\_davis.ipynb](https://github.com/volkamerlab/kissim_app/blob/v1.1.0/notebooks/006_evaluation/004_profiling_karaman_davis.ipynb), Version 1.1.0 [accessed 2022-04-24].
- (67) Volkamer Lab, *KiSSim Notebook: Predict Ligand Profiling Using IFPs (Pooled Karaman and Davis Dataset)*. [https://github.com/volkamerlab/kissim\\_app/blob/v1.1.0/notebooks/006\\_evaluation/011\\_profiling\\_karaman\\_davis\\_ifp.ipynb](https://github.com/volkamerlab/kissim_app/blob/v1.1.0/notebooks/006_evaluation/011_profiling_karaman_davis_ifp.ipynb), Version 1.1.0 [accessed 2022-04-24].
- (68) Volkamer Lab, *KiSSim Notebook: Predict Ligand Profiling Using Sequence (Pooled Karaman and Davis Dataset)*. [https://github.com/volkamerlab/kissim\\_app/blob/v1.1.0/notebooks/006\\_evaluation/012\\_profiling\\_karaman\\_davis\\_seq.ipynb](https://github.com/volkamerlab/kissim_app/blob/v1.1.0/notebooks/006_evaluation/012_profiling_karaman_davis_seq.ipynb), Version 1.1.0 [accessed 2022-04-24].
- (69) Volkamer Lab, *KiSSim Notebook: Predict Ligand Profiling Using SiteAlign (Pooled Karaman and Davis Dataset)*. [https://github.com/volkamerlab/kissim\\_app/blob/v1.1.0/notebooks/006\\_evaluation/013\\_profiling\\_karaman\\_davis\\_sitealign.ipynb](https://github.com/volkamerlab/kissim_app/blob/v1.1.0/notebooks/006_evaluation/013_profiling_karaman_davis_sitealign.ipynb), Version 1.1.0 [accessed 2022-04-24].
- (70) Volkamer Lab, *KiSSim Notebook: Compare AUC Values Between KiSSim and Other Methods*. [https://github.com/volkamerlab/kissim\\_app/blob/v1.1.0/notebooks/006\\_evaluation/014\\_comparative\\_analyses\\_auc.ipynb](https://github.com/volkamerlab/kissim_app/blob/v1.1.0/notebooks/006_evaluation/014_comparative_analyses_auc.ipynb), Version 1.1.0 [accessed 2022-04-24].
- (71) Moret, N.; Clark, N. A.; Hafner, M.; Wang, Y.; Lounkine, E.; Medvedovic, M.; Wang, J.; Gray, N.; Jenkins, J.; Sorger, P. K. Cheminformatics Tools for Analyzing and Designing Optimized Small-Molecule Collections and Libraries. *Cell Chem. Biol.* **2019**, *26*, 765–777.e3.
- (72) Volkamer Lab, *KiSSim Notebook: Predict Ligand Profiling Using KiSSim (Moret Dataset)*. [https://github.com/volkamerlab/kissim\\_app/blob/v1.1.0/notebooks/006\\_evaluation/016\\_profiling\\_moret.ipynb](https://github.com/volkamerlab/kissim_app/blob/v1.1.0/notebooks/006_evaluation/016_profiling_moret.ipynb), Version 1.1.0 [accessed 2022-04-24].
- (73) Volkamer Lab, *KiSSim Notebook: KiSSim Matrix Only Based on Ligand-Interacting Residues*. [https://github.com/volkamerlab/kissim\\_app/blob/v1.1.0/notebooks/006\\_evaluation/015\\_subset\\_kissim\\_fingerprints.ipynb](https://github.com/volkamerlab/kissim_app/blob/v1.1.0/notebooks/006_evaluation/015_subset_kissim_fingerprints.ipynb), Version 1.1.0 [accessed 2022-04-24].
- (74) Volkamer Lab, *KiSSim Notebook: Pocket subsets from literature*. [https://github.com/volkamerlab/kissim\\_app/blob/v1.1.0/notebooks/004\\_fingerprints/006\\_literature\\_pocket\\_subsets.ipynb](https://github.com/volkamerlab/kissim_app/blob/v1.1.0/notebooks/004_fingerprints/006_literature_pocket_subsets.ipynb), Version 1.1.0 [accessed 2022-04-24].
- (75) Volkamer Lab, *KiSSim Notebook: Fingerprint Bit Differences*. [https://github.com/volkamerlab/kissim\\_app/blob/v1.1.0/notebooks/005\\_comparison/007\\_fingerprint\\_diffs\\_3d.ipynb](https://github.com/volkamerlab/kissim_app/blob/v1.1.0/notebooks/005_comparison/007_fingerprint_diffs_3d.ipynb), Version 1.1.0 [accessed 2022-04-24].
- (76) KLIFS, *6JRK - Chain A | Epidermal Growth Factor Receptor*. [https://klifs.net/details.php?structure\\_id=12159](https://klifs.net/details.php?structure_id=12159), [accessed 2022-04-09].
- (77) KLIFS, *5Y80 - Chain A (Model A) | Cyclin G Associated Kinase*. [https://klifs.net/details.php?structure\\_id=10329](https://klifs.net/details.php?structure_id=10329), [accessed 2022-04-09].

(78) Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Židek, A.; Potapenko, A.; Bridgland, A.; Meyer, C.; Kohl, S. A. A.; Ballard, A. J.; Cowie, A.; Romera-Paredes, B.; Nikolov, S.; Jain, R.; Adler, J.; Back, T.; Petersen, S.; Reiman, D.; Clancy, E.; Zielinski, M.; Steinegger, M.; Pacholska, M.; Berghammer, T.; Bodenstein, S.; Silver, D.; Vinyals, O.; Senior, A. W.; Kavukcuoglu, K.; Kohli, P.; Hassabis, D. Highly Accurate Protein Structure Prediction with AlphaFold. *Nature* **2021**, *596*, 583–589.

(79) Tunyasuvunakool, K.; Adler, J.; Wu, Z.; Green, T.; Zielinski, M.; Židek, A.; Bridgland, A.; Cowie, A.; Meyer, C.; Laydon, A.; Velankar, S.; Kleywegt, G. J.; Bateman, A.; Evans, R.; Pritzel, A.; Figurnov, M.; Ronneberger, O.; Bates, R.; Kohl, S. A. A.; Potapenko, A.; Ballard, A. J.; Romera-Paredes, B.; Nikolov, S.; Jain, R.; Clancy, E.; Reiman, D.; Petersen, S.; Senior, A. W.; Kavukcuoglu, K.; Birney, E.; Kohli, P.; Jumper, J.; Hassabis, D. Highly Accurate Protein Structure Prediction for the Human Proteome. *Nature* **2021**, *596*, 590–596.

(80) Modi, V.; Dunbrack, R. L. Kincore: A Web Resource for Structural Classification of Protein Kinases and Their Inhibitors. *Nucleic Acids Res.* **2021**, *50*, D654.

(81) Bennett, L.; Melchers, B.; Proppe, B. *Curta: A General-Purpose High-Performance Computer at ZEDAT*, Freie Universität Berlin. 2021.

## Recommended by ACS

### Modeling MEK4 Kinase Inhibitors through Perturbed Electrostatic Potential Charges

Rama K. Mishra, Karl A. Scheidt, *et al.*

SEPTEMBER 30, 2019  
JOURNAL OF CHEMICAL INFORMATION AND MODELING

READ 

### graphDelta: MPNN Scoring Function for the Affinity Prediction of Protein–Ligand Complexes

Dmitry S. Karlov, Petr Popov, *et al.*

MARCH 09, 2020  
ACS OMEGA

READ 

### In Silico Design and Analysis of a Kinase-Focused Combinatorial Library Considering Diversity and Quality

Yan Yang, Haichun Liu, *et al.*

DECEMBER 30, 2019  
JOURNAL OF CHEMICAL INFORMATION AND MODELING

READ 

### Machine Learning Models Based on Molecular Fingerprints and an Extreme Gradient Boosting Method Lead to the Discovery of JAK2 Inhibitors

Minjian Yang, Xiaojian Wang, *et al.*

NOVEMBER 20, 2019  
JOURNAL OF CHEMICAL INFORMATION AND MODELING

READ 

Get More Suggestions >

## Supporting Information

# KiSSim: Predicting off-targets from structural similarities in the kinome

Dominique Sydow,<sup>†</sup> Eva ABmann,<sup>†</sup> Albert J. Kooistra,<sup>‡</sup> Friedrich Rippmann,<sup>¶</sup>  
and Andrea Volkamer<sup>\*,†,§</sup>

<sup>†</sup>*In Silico Toxicology and Structural Bioinformatics, Institute of Physiology, Charité –  
Universitätsmedizin Berlin, corporate member of Freie Universität Berlin and  
Humboldt-Universität zu Berlin, Augustenburger Platz 1, 13353 Berlin, Germany*

<sup>‡</sup>*Department of Drug Design and Pharmacology, University of Copenhagen,  
Universitetsparken 2, 2100 Copenhagen, Denmark*

<sup>¶</sup>*Computational Chemistry & Biologics, Merck Healthcare KGaA, Frankfurter Str. 250,  
64293 Darmstadt, Germany*

<sup>§</sup>*Corresponding author: Andrea Volkamer; andrea.volkamer@charite.de*

E-mail: andrea.volkamer@charite.de



## Supplementary methods

### KiSSim fingerprint

The KiSSim fingerprint encodes the 85 KLIFS pocket residues in the form of physicochemical and spatial properties (Figure 1). *Physicochemical properties* include pharmacophoric and size features, side chain orientation, and solvent exposure. *Spatial properties* include each residue’s distance to the pocket center as well as three prominent kinase subpockets and the first three moments of the resulting distance distributions.

*Pharmacophoric and size features* (Figure 1 a) are taken from the SiteAlign categories for standard amino acids:<sup>1</sup> The size of residues with less than 4, 4–6, or more than 6 heavy atoms is defined as 1, 2, or 3, respectively. The number of hydrogen bond donors (HBD) and hydrogen bond acceptors (HBA) range from 0–3 and 0–2, respectively. The charge is set to  $-1$ ,  $0$ , or  $1$  in the case of negative, neutral, or positive residues, respectively. Aromatic and aliphatic properties are set to 1 if present or 0 if not present. The feature values for standard amino acids are listed in TableS1. Non-standard residues are mapped to their parent residues listed in the kinase sequence if possible (TableS2), otherwise the feature is set to NaN.

*Side chain orientation* (Figure 1 b) is adapted from the SiteAlign definitions. In KiSSim, this feature is based on the vertex angle  $\alpha$  from the residue’s CA atom (vertex) to the pocket centroid (based on all 85 pocket CA atoms) and to the residue’s side chain representative. The latter is defined for each standard amino acid individually and refers to the out-most atom in the side chain (Table S3). Non-standard amino acids are handled as described before. Side chain orientation is defined as inward-facing (1), intermediate (2), and outward-facing (3) if  $0 \leq \alpha \leq 45^\circ$ ,  $45 < \alpha \leq 90^\circ$ , and  $90 < \alpha \leq 180^\circ$ , respectively.

*Solvent exposure* (Figure 1 c) is based on the HSExposure<sup>2</sup> functionality in BioPython.<sup>?</sup> The CA-CB vector of a residue spans a normal plane, which cuts a sphere in half that sits around the residue’s CA atom with a radius of  $12\text{\AA}$ . The ratio  $\rho$  is calculated between the

number of CA atoms in the upper half and all CA atoms in the sphere. Solvent exposure is defined as high (1), intermediate (2), and low (3) if  $0.0 \leq \rho \leq 0.45$ ,  $0.45 < \rho \leq 0.55$ , and  $0.55 < \rho \leq 1.0$ , respectively. If the residue’s CA atom is missing, the feature is set to NaN. If a residue’s CB atom is missing, HSExposure calculates a pseudo-CB atom inferred from neighboring atoms as described in.<sup>2</sup> If this approach fails, the feature is set to NaN.

*Spatial distances* (Figure 1 d) are calculated from each residue’s CA atom to the pocket center and to prominent subpocket centers. The *pocket center* is the centroid of all — structurally resolved — pocket CA atoms. Prominent subpocket centers include the hinge region, DFG region, and front pocket. Each *subpocket center* is calculated based on the centroid of three anchor residues’ CA atoms, following the idea described in the KinFragLib methodology.<sup>3</sup> We selected anchor residues manually by fine-tuning the resulting subpocket center to be situated in front of the *hinge region*, the *DFG region* or below the *front loop* (Table S4). Pocket residue positions with high gap rates in sequence or structures were not considered (Figure S1). If an anchor residue’s CA atom is missing in one of the structures, the centroid of both neighboring CA atoms is used instead. If only one neighboring CA atom is present, this atom is used instead. If no neighboring CA atom is available, the feature is set to NaN. The subpocket center calculation is implemented in the structural cheminformatics library OpenCADD (module `opencadd.structure.pocket`).<sup>4</sup>

*Spatial moments* (Figure 1 e) describe each of the four distributions of distances to the pocket center, and three subpocket centers of the hinge region, DFG region, and front pocket. In KiSSim, the first three moments are used: the mean, the standard deviation, and the cube root of the skewness. This procedure is adapted from the Ultrafast Shape Recognition (USR)<sup>5</sup> method.

## Supplementary tables

Table S1: **SiteAlign features.** SiteAlign<sup>1</sup> categories for standard amino acids (one-letter code) including size, hydrogen bond donor (HBD), hydrogen bond acceptor (HBA), charge, aromatic, and aliphatic features.

Feature name	Feature value	Amino acids
Size	1	A C G P S T V
	2	D E H I K L M N Q
	3	F R W Y
HBD	0	A D E F G I L M P V
	1	C H K N Q S T W Y
	3	R
HBA	0	A C F G I K L M P R V W
	1	H N Q S T Y
	2	D E
Charge	-1	D E
	0	A C F G H I L M N P Q S T V W Y
	1	K R
Aromatic	0	A C D E G I K L M N P Q R S T V
	1	F H W Y
Aliphatic	0	D E F G H K N Q R S W Y
	1	A C I L M P T V

Table S2: **Non-standard amino acid conversion.** List of non-standard amino acids and their parent standard amino acids used in KiSSim.

Non-standard amino acid	Parent standard amino acid
CAF	CYS
CME	CYS
CSS	CYS
OCY	CYS
KCX	LYS
MSE	MET
PHD	ASP
PTR	TYR

Table S3: **Side chain representative atoms.** List of standard amino acids and their side chain representatives as defined for the KiSSim side chain orientation feature calculation. \*pCB = pseudo-CB calculated with BioPython<sup>2</sup>

Amino acid (three-letter code)	Amino acid (one-letter code)	Atom PDB nam
ALA	A	CB
ARG	R	CG
ASN	N	CG
ASP	D	CG
CYS	C	SG
GLN	Q	CD
GLU	E	CD
GLY	G	pCB*
HIS	H	CE1
ILE	I	CD1
LEU	L	CG
LYS	K	NZ
MET	M	CE
PHE	F	CZ
PRO	P	CB
SER	S	OG
THR	T	CB
TRP	W	CE2
TYR	Y	OH
VAL	V	CB

Table S4: **KiSSim subpocket anchor residues.** KLIFS residue IDs used to calculate the CA atoms' centroid defining the KiSSim pocket center and subpocket centers.

Center name	Anchor residue KLIFS IDs
Pocket center	1–85
Hinge region	16, 47, 80
DFG region	19, 24, 81
Front pocket	10, 48, 72

Table S5: **Subsets of KLIFS pocket residues** based on (a)  $\geq 1\%$  interaction frequency across the unique kinase-ligand combinations in KLIFS version 3.2 with the build update from 2021-09-02 and (b) based on pocket residues identified by Martin and Mukherjee<sup>6</sup>.

Subset criterion	KLIFS pocket residues
DFG-in	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16, 17, 19, 20, 21, 23, 24, 25, 27, 28, 31, 35, 36, 37, 38, 41, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 59, 60, 61, 64, 66, 67, 68, 69, 70, 72, 74, 75, 76, 77, 79, 80, 81, 82, 83, 84, 85
DFG-out	3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 15, 16, 17, 19, 20, 21, 23, 24, 25, 27, 28, 31, 35, 36, 37, 38, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 54, 55, 60, 61, 64, 66, 67, 68, 69, 70, 74, 75, 77, 79, 80, 81, 82, 83, 84, 85
DFG-all	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16, 17, 19, 20, 21, 23, 24, 25, 27, 28, 31, 35, 36, 37, 38, 41, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 59, 60, 61, 64, 66, 67, 68, 69, 70, 72, 74, 75, 76, 77, 79, 80, 81, 82, 83, 84, 85
Martin and Mukherjee <sup>6</sup>	5, 8, 28, 35, 38, 44, 45, 46, 48, 51, 52, 66, 67, 77, 80, 84

Table S6: **KiSSim dataset.** Upper half: Filtering steps performed on the human dataset from KLIFS version 3.2<sup>7</sup> downloaded on 2021-09-02 to generate the KiSSim dataset. Lower half: Number of structures and kinases as well as number of structure and kinase pairs encoded and compared with the KiSSim methodology; number of structure/kinase pairs does not contain self-comparisons. See notebooks for more details.<sup>8,9</sup>

	Number of structures		
	all	DFG-in	DFG-out
Select species: human	11806		
Select KLIFS structures without flag	11650		
Select resolution: $\leq 3$	10690		
Select quality score: $\geq 6$	10236		
Select mutated pocket residues: $\leq 3$	10155		
Select missing pocket residues: $\leq 8$	10150		
Select conformation	10150	8982	786
Select best structure per PDB and kinase pair	4690	4120	407
Encode structures as fingerprints	4681	4112	406
Number of structures	4681	4112	406
Number of kinases	279	257	71
Number of structure pairs	10953540	8452216	82215
Number of kinase pairs	38781	32896	2485

## Supplementary figures

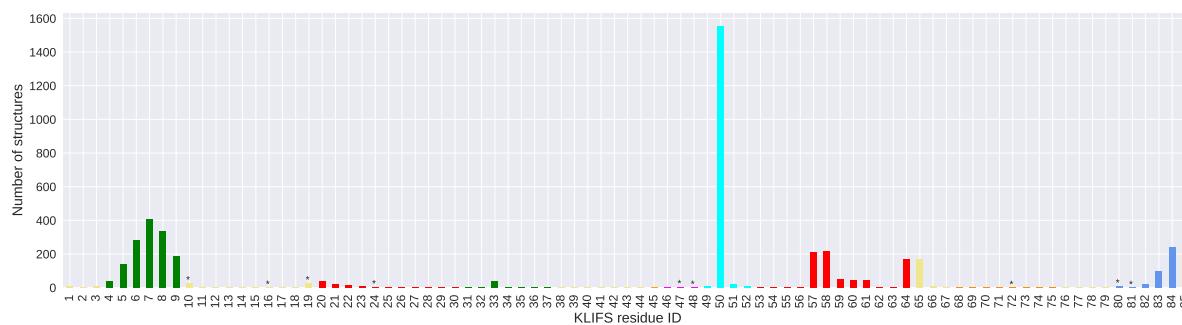


Figure S1: **Number of structures with missing residues.** Missing residues are shown for the KLIFS pocket (residues 1-85) and colored by KLIFS region (loops in green and orange, linker region in cyan, hinge region in magenta,  $\alpha$ -helices in red,  $\beta$ -sheets in yellow, and DFG region in blue). Residues selected as anchor residues to calculate the KiSSim subpockets are marked with \*. See notebook for more details.<sup>10</sup>

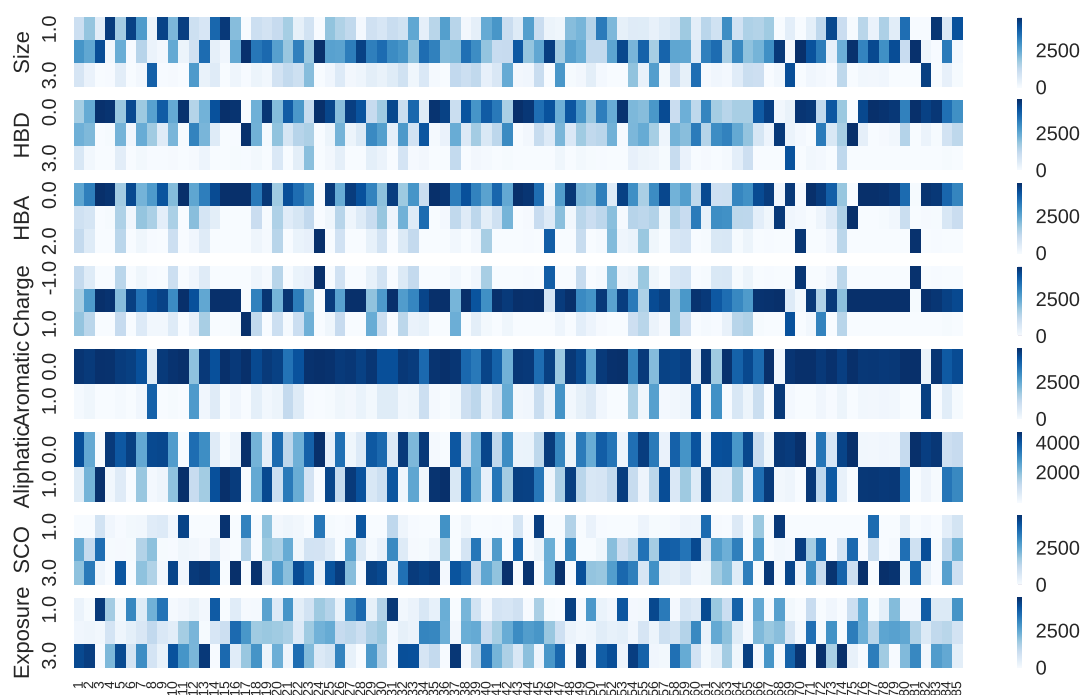


Figure S2: **Physicochemical feature distribution per residue position** across all fingerprint pairs: Across all fingerprint distances, feature value per residue position for the size, number of hydrogen bond donors (HBD), number of hydrogen bond acceptors (HBA), charge, aromatic, aliphatic, side chain orientation (SCO), and solvent exposure. See notebook for more details.<sup>11</sup>



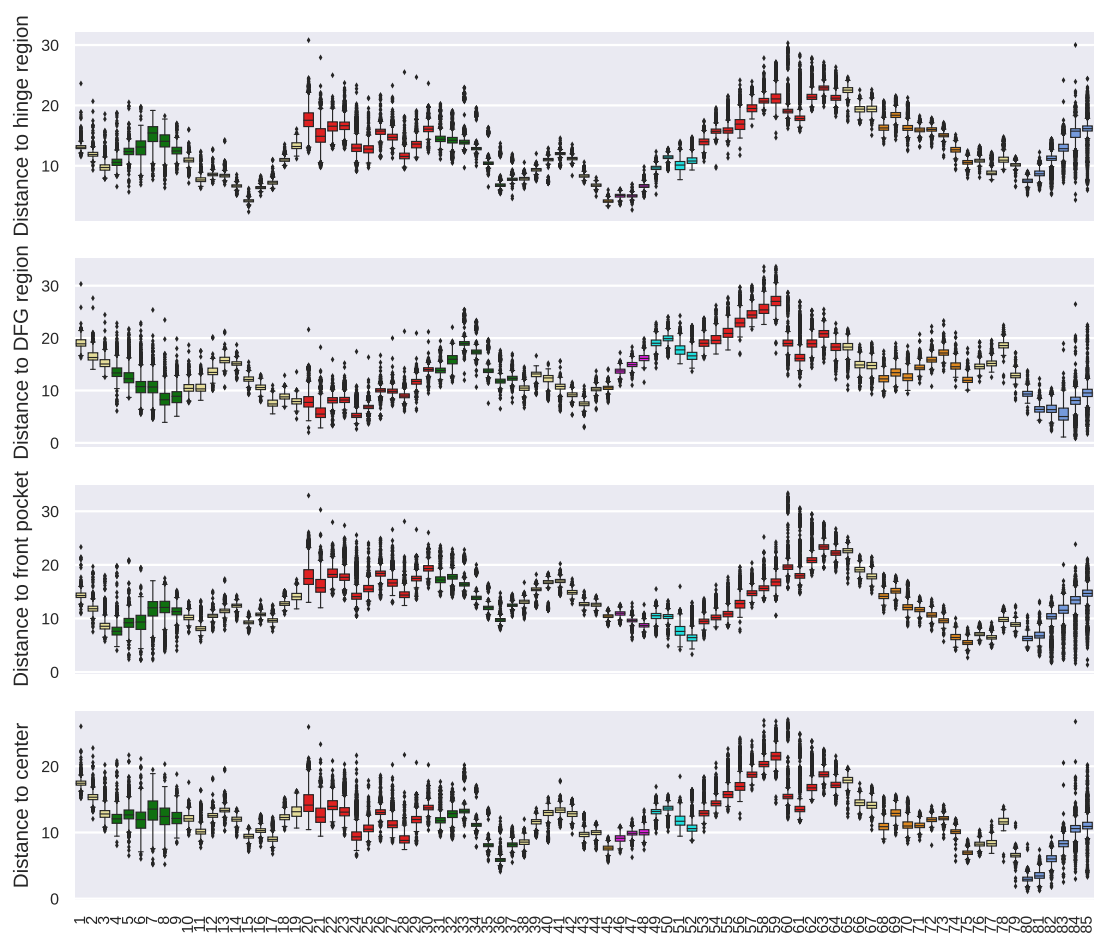


Figure S3: **Spatial distances feature distribution per residue position** across all fingerprint pairs: Distances per residue position to the hinge region, DFG region, front pocket, and pocket center. See notebook for more details.<sup>11</sup>

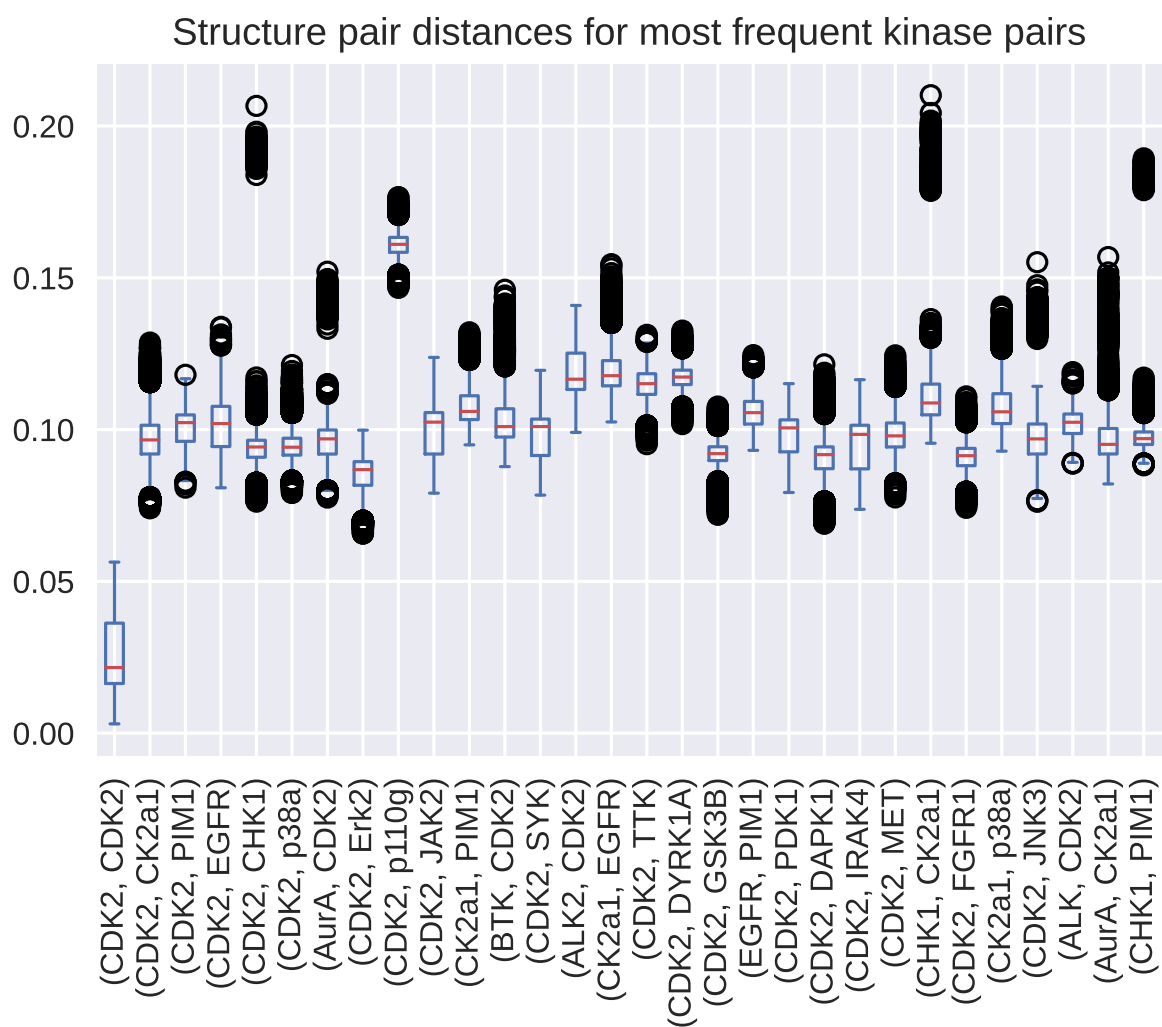


Figure S4: **Structure pair distances for the most frequent kinases.** A kinase pair can be described by varying structure pair distances; the number of distances per structure pair ranges from 83845 (CDK2-CDK2) to 19435 (CHK1-PIM1). Sanity check: Structure pairs describing the same kinase pairs (e.g. CDK2-CDK2) show lower structure distances than structure pairs describing different kinase pairs. See notebook for more details.<sup>12</sup>



Figure S5: **KiSSim-based kinome tree** focused on a subset of residues, i.e. residues frequently interacting with ligands as seen in DFG-in structures (see Table S5). The tree is based on 257 structurally resolved kinases in the DFG-in conformation. Tree nodes are colored from red to blue showing small to large distances (0.008–0.102), describing high to low similarities; tree leaves represent kinases colored by kinase group (see S7). The tree is based on a clustering of the kinase distance matrix using as metric the Euclidean distance and as linkage Ward's criterion. The clusters are converted to the Newick format and visualized using FigTree.<sup>13</sup>



Figure S6: **KiSSim-based kinome tree** focused on a subset of residues, i.e. pocket residues as identified by Martin and Mukherjee<sup>6</sup>). The tree is based on 257 structurally resolved kinases in the DFG-in conformation. See Figure S5's caption for more details on the tree parameters; tree nodes are colored from red to blue showing small to large distances (0.003–0.118), describing high to low similarities.

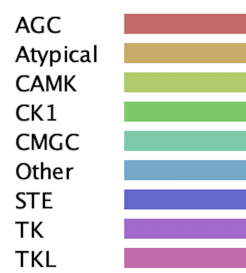


Figure S7: **Coloring scheme for kinase groups** as used in the KiSSim kinome trees.

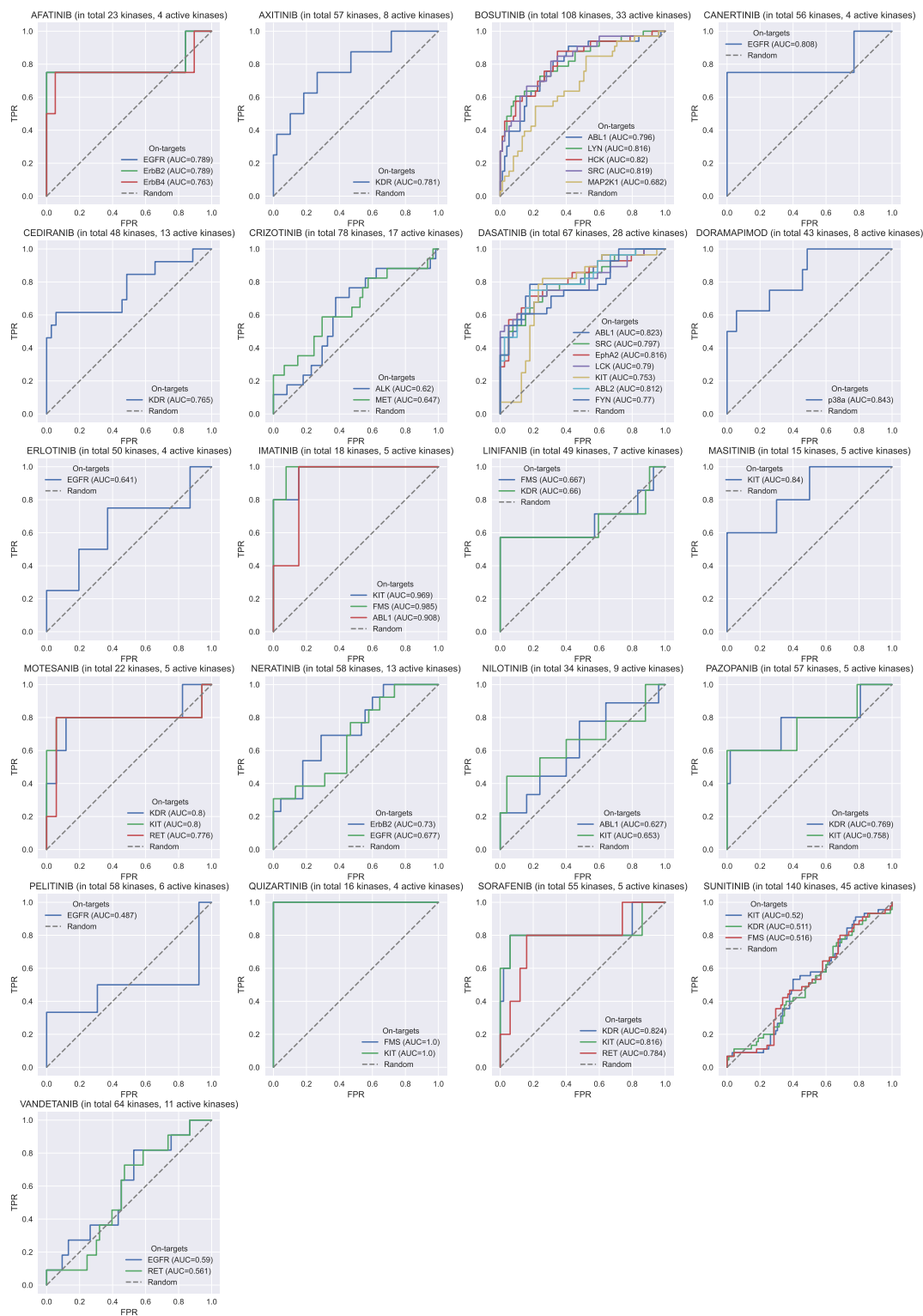


Figure S8: (Continued on the following page.)

Figure S8: **KiSSim performance against Karaman-Davis profiling data.** ROC curves comparing predicted and profiling-based kinase similarities (FPR = False positive rate; TPR = True positive rate). *Predicted similarities* against a selected kinase  $k$  are based on the KiSSim methodology. *Profiling-based kinase similarities* define kinases as similar if they are targeted by the same ligand with  $K_d \leq 100$  nM, including the ligand's on-target(s) as reported in the PKIDB. The kinases, for which the ligand shows lower activities with  $K_d > 100$  nM, are treated as dissimilar to the ligand's on-target(s). Find more details in the Bioactivity profiling data section; we pooled profiling data from Karaman et al.<sup>14</sup> and Davis et al.<sup>15</sup> for this analysis. The first rank is always occupied by the kinase  $k$ . See notebook for more details.<sup>16</sup>

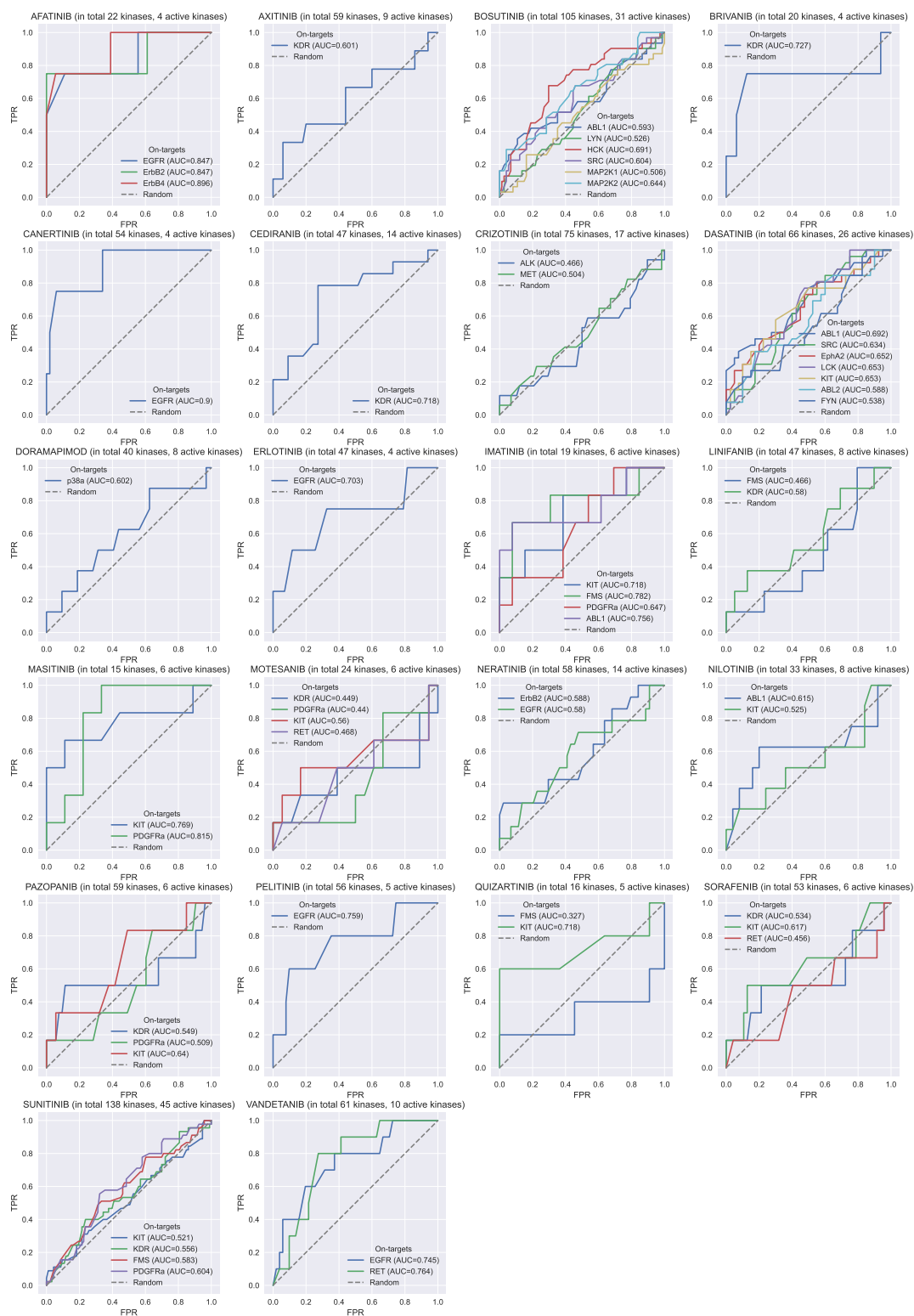


Figure S9: (Continued on the following page.)



Figure S9: **KLIFS IFP performance against Karaman-Davis profiling data.** ROC curves comparing predicted and profiling-based kinase similarities (FPR = False positive rate; TPR = True positive rate). *Predicted similarities* against a selected kinase  $k$  are based on the KLIFS IFP. *Profiling-based kinase similarities* are defined as described in Figure S8's caption. The first rank is always occupied by the kinase  $k$ . See notebook for more details.<sup>17</sup>

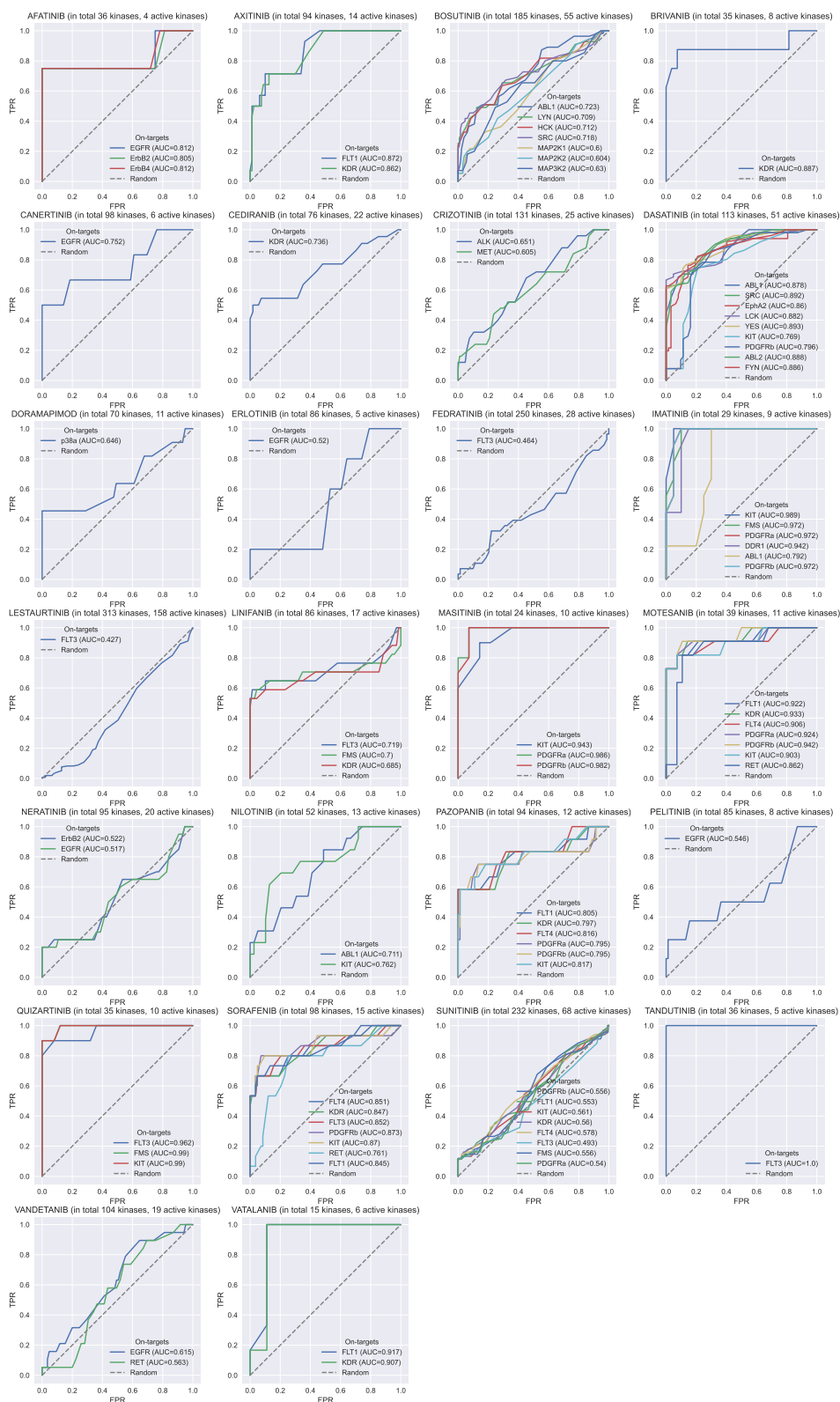


Figure S10: (Continued on the following page.)

Figure S10: **KLIFS sequence performance against Karaman-Davis profiling data.** ROC curves comparing predicted and profiling-based kinase similarities (FPR = False positive rate; TPR = True positive rate). *Predicted similarities* against a selected kinase  $k$  are based on the KLIFS sequence identity. *Profiling-based kinase similarities* are defined as described in Figure S8's caption. The first rank is always occupied by the kinase  $k$ . See notebook for more details.<sup>18</sup>

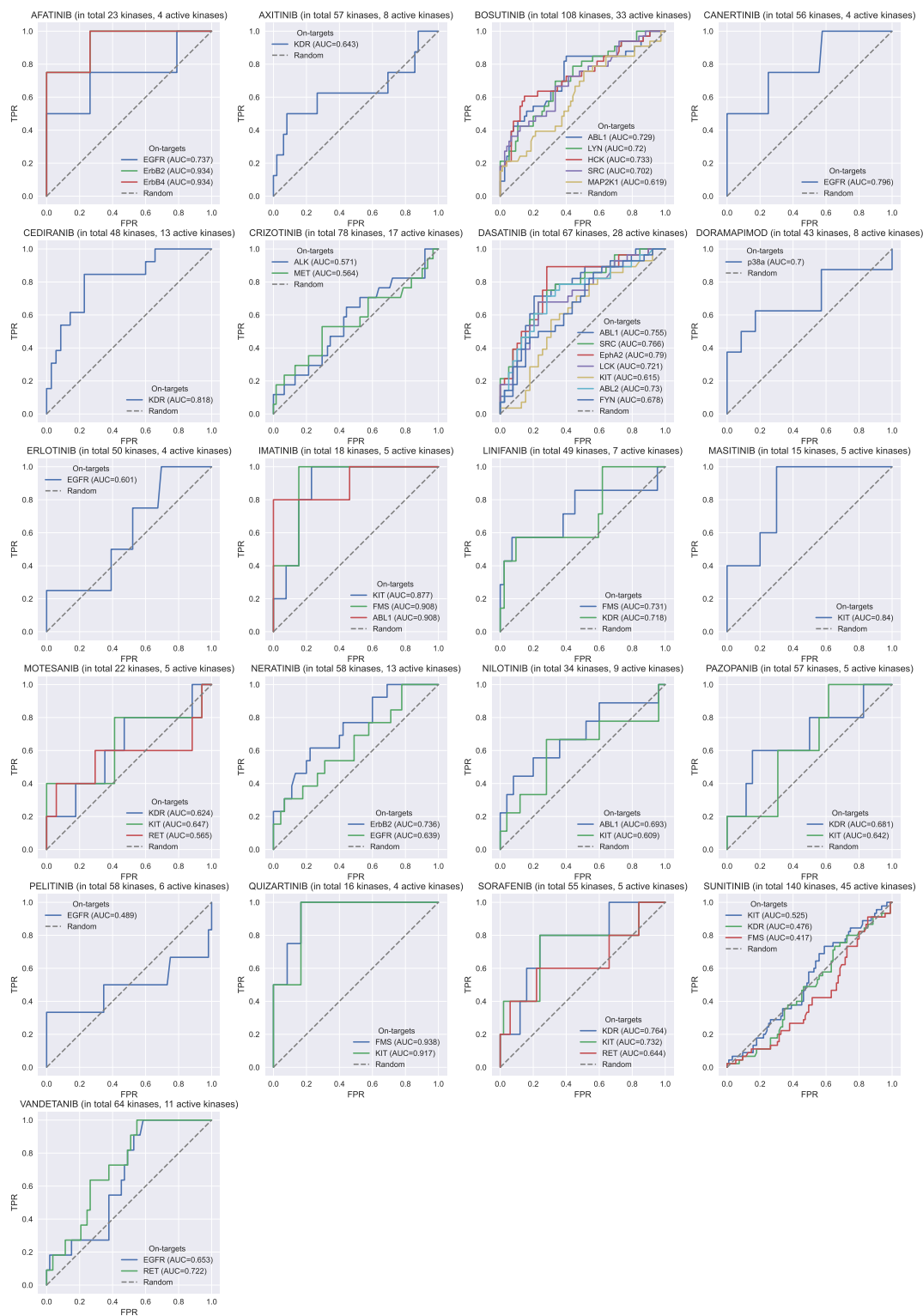


Figure S11: (Continued on the following page.)

Figure S11: **SiteAlign performance against Karaman-Davis profiling data.** ROC curves comparing predicted and profiling-based kinase similarities (FPR = False positive rate; TPR = True positive rate). *Predicted similarities* against a selected kinase  $k$  are based on the SiteAlign methodology. *Profiling-based kinase similarities* are defined as described in Figure S8's caption. The first rank is always occupied by the kinase  $k$ . See notebook for more details.<sup>19</sup>

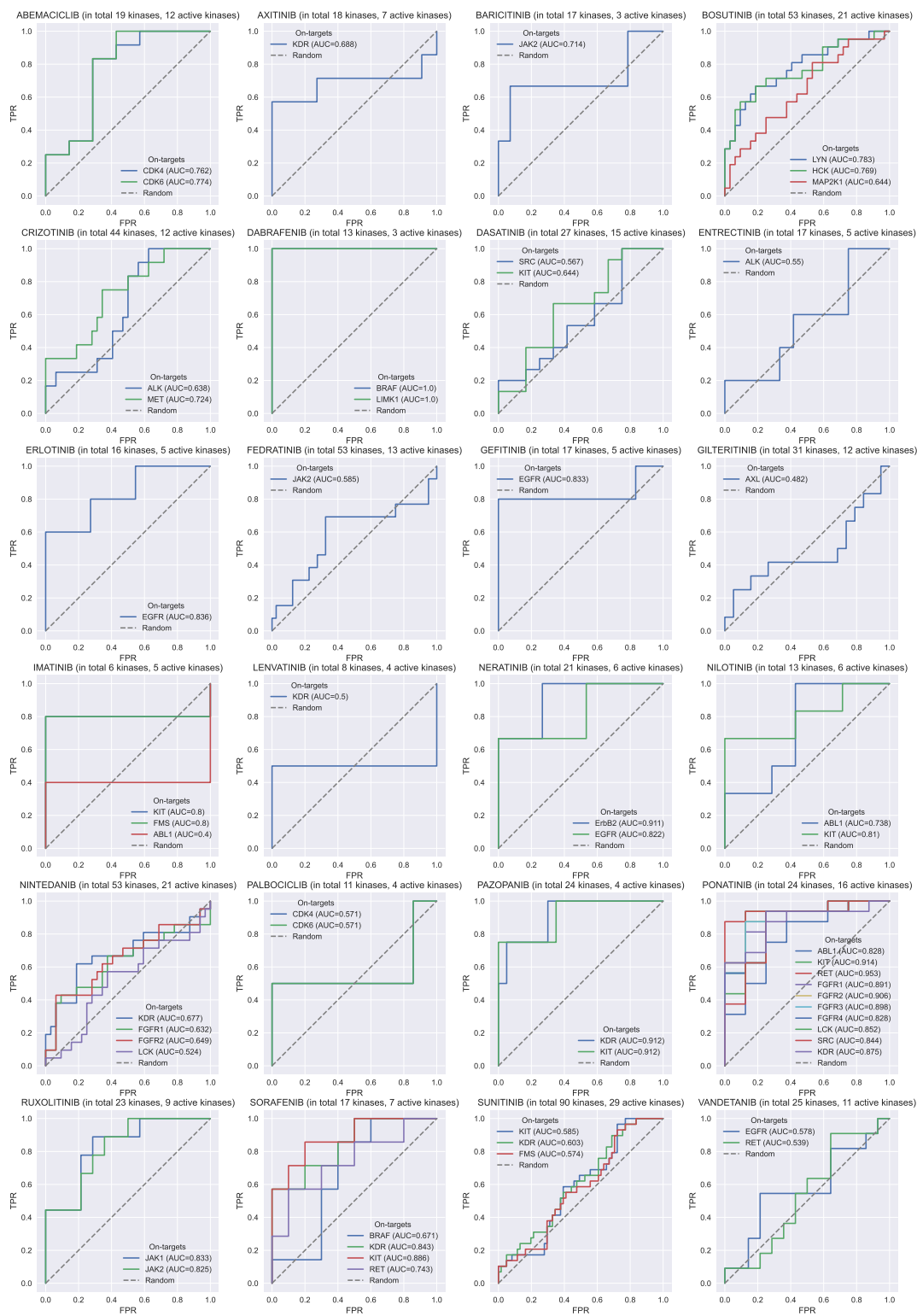


Figure S12: (Continued on the following page.)

Figure S12: **KiSSim performance against Moret profiling data.** ROC curves comparing predicted and profiling-based kinase similarities (FPR = False positive rate; TPR = True positive rate). *Predicted similarities* against a selected kinase  $k$  are based on the KiSSim methodology. *Profiling-based kinase similarities* are defined as described in Figure S8's caption but using the profiling data by Moret et al.<sup>20</sup>. The first rank is always occupied by the kinase  $k$ . See notebook for more details.<sup>21</sup>

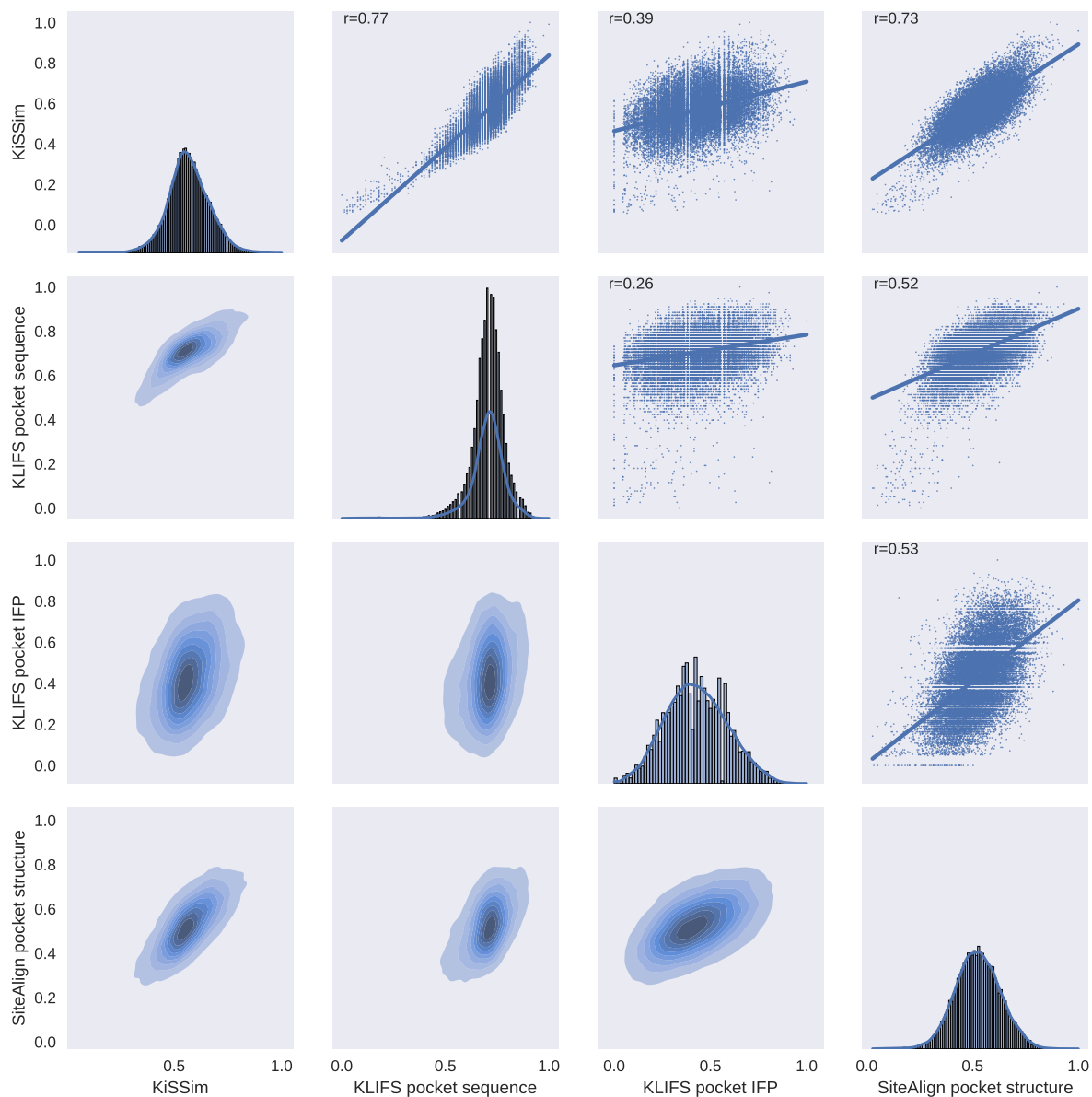


Figure S13: Comparison of different distance values for pairwise kinase structure comparisons, including all conformations. Diagonal: Distributions of pairwise kinase distances calculated based on the KiSSim method, KLIFS pocket sequence identity, KLIFS pocket IFP similarity, and SiteAlign pocket structure similarity. Lower triangular matrix: Bivariate distributions of pairwise kinase distances, shown as isocontours with dark blue indicating high densities and light blue indicating low densities. Upper triangular matrix: Scatter plots of pairwise kinase distances with fitted regression lines (dark lines) and 95% CI intervals of regression (light blue shades). See notebook for more details.<sup>22</sup>



## References

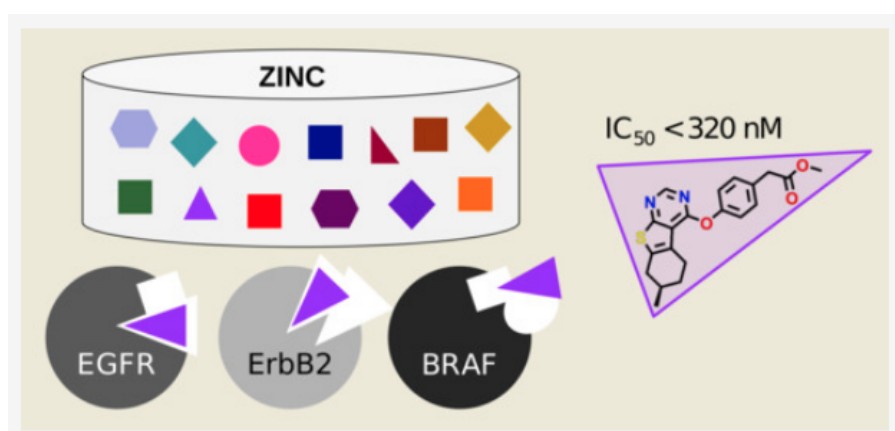
- (1) Schalon, C.; Surgand, J.-S.; Kellenberger, E.; Rognan, D. A Simple and Fuzzy Method to Align and Compare Druggable Ligand-Binding Sites. *Proteins Struct. Funct. Bioinforma.* **2008**, *71*, 1755–1778.
- (2) Hamelryck, T. An Amino Acid Has Two Sides: A New 2D Measure Provides a Different View of Solvent Exposure. *Proteins Struct. Funct. Bioinforma.* **2005**, *59*, 38–48.
- (3) Sydow, D.; Schmiel, P.; Mortier, J.; Volkamer, A. KinFragLib: Exploring the Kinase Inhibitor Space Using Subpocket-Focused Fragmentation and Recombination. *J. Chem. Inf. Model.* **2020**, *60*, 6081–6094.
- (4) Volkamer Lab, OpenCADD. <https://github.com/volkamerlab/opencadd>, [accessed 2021-11-27].
- (5) Ballester, P. J.; Richards, W. G. Ultrafast Shape Recognition to Search Compound Databases for Similar Molecular Shapes. *J. Comput. Chem.* **2007**, *28*, 1711–1723.
- (6) Martin, E.; Mukherjee, P. Kinase-Kernel Models: Accurate In silico Screening of 4 Million Compounds Across the Entire Human Kinome. *J. Chem. Inf. Model.* **2012**, *52*, 156–170.
- (7) Kanev, G. K.; de Graaf, C.; Westerman, B. A.; de Esch, I. J. P.; Kooistra, A. J. KLIFS: An Overhaul after the First 5 Years of Supporting Kinase Research. *Nucleic Acids Res.* **2021**, *49*, D562–D569.
- (8) Volkamer Lab, KiSSim notebook: KLIFS Data Preparation and Exploration. [https://github.com/volkamerlab/kissim\\_app/blob/v1.1.0/notebooks/002\\_structures/001\\_prepare\\_dataset.ipynb](https://github.com/volkamerlab/kissim_app/blob/v1.1.0/notebooks/002_structures/001_prepare_dataset.ipynb), Version 1.1.0 [accessed 2022-04-24].
- (9) Volkamer Lab, KiSSim Notebook: Loading KiSSim Results. <https://github.com/>

- volkamerlab/kissim\_app/blob/v1.1.0/notebooks/001\_quick\_start/001\_quick\_start\_kissim.ipynb, Version 1.1.0 [accessed 2022-04-24].
- (10) Volkamer Lab, KiSSim Notebook: KLIFS Data Exploration. [https://github.com/volkamerlab/kissim\\_app/blob/v1.1.0/notebooks/002\\_structures/002\\_explore\\_dataset.ipynb](https://github.com/volkamerlab/kissim_app/blob/v1.1.0/notebooks/002_structures/002_explore_dataset.ipynb), Version 1.1.0 [accessed 2022-04-24].
- (11) Volkamer Lab, KiSSim Notebook: Per-Residue Feature Values. [https://github.com/volkamerlab/kissim\\_app/blob/v1.1.0/notebooks/004\\_fingerprints/004\\_feature\\_distribution\\_per\\_residue.ipynb](https://github.com/volkamerlab/kissim_app/blob/v1.1.0/notebooks/004_fingerprints/004_feature_distribution_per_residue.ipynb), Version 1.1.0 [accessed 2022-04-24].
- (12) Volkamer Lab, KiSSim Notebook: Fingerprint Distances Between Structures for the Same Kinase. [https://github.com/volkamerlab/kissim\\_app/blob/v1.1.0/notebooks/005\\_comparison/005\\_structure\\_kinase\\_mapping.ipynb](https://github.com/volkamerlab/kissim_app/blob/v1.1.0/notebooks/005_comparison/005_structure_kinase_mapping.ipynb), Version 1.1.0 [accessed 2022-04-24].
- (13) FigTree, FigTree. <http://tree.bio.ed.ac.uk/software/figtree/>, [accessed 2021-08-16].
- (14) Karaman, M. W.; Herrgard, S.; Treiber, D. K.; Gallant, P.; Atteridge, C. E.; Campbell, B. T.; Chan, K. W.; Ciceri, P.; Davis, M. I.; Edeen, P. T.; Faraoni, R.; Floyd, M.; Hunt, J. P.; Lockhart, D. J.; Milanov, Z. V.; Morrison, M. J.; Pallares, G.; Patel, H. K.; Pritchard, S.; Wodicka, L. M.; Zarrinkar, P. P. A Quantitative Analysis of Kinase Inhibitor Selectivity. *Nat. Biotechnol.* **2008**, *26*, 127–132.
- (15) Davis, M. I.; Hunt, J. P.; Herrgard, S.; Ciceri, P.; Wodicka, L. M.; Pallares, G.; Hocker, M.; Treiber, D. K.; Zarrinkar, P. P. Comprehensive Analysis of Kinase Inhibitor Selectivity. *Nat. Biotechnol.* **2011**, *29*, 1046–1051.
- (16) Volkamer Lab, KiSSim Notebook: Predict Ligand Profiling Using KiSSim (Pooled Karaman and Davis Dataset). [https://github.com/volkamerlab/kissim\\_app/](https://github.com/volkamerlab/kissim_app/)

- blob/v1.1.0/notebooks/006\_evaluation/004\_profiling\_karaman\_davis.ipynb, Version 1.1.0 [accessed 2022-04-24].
- (17) Volkamer Lab, KiSSim Notebook: Predict Ligand Profiling Using IFPs (Pooled Karaman and Davis Dataset). [https://github.com/volkamerlab/kissim\\_app/blob/v1.1.0/notebooks/006\\_evaluation/011\\_profiling\\_karaman\\_davis\\_\\_ifp.ipynb](https://github.com/volkamerlab/kissim_app/blob/v1.1.0/notebooks/006_evaluation/011_profiling_karaman_davis__ifp.ipynb), Version 1.1.0 [accessed 2022-04-24].
- (18) Volkamer Lab, KiSSim Notebook: Predict Ligand Profiling Using Sequence (Pooled Karaman and Davis Dataset). [https://github.com/volkamerlab/kissim\\_app/blob/v1.1.0/notebooks/006\\_evaluation/012\\_profiling\\_karaman\\_davis\\_\\_seq.ipynb](https://github.com/volkamerlab/kissim_app/blob/v1.1.0/notebooks/006_evaluation/012_profiling_karaman_davis__seq.ipynb), Version 1.1.0 [accessed 2022-04-24].
- (19) Volkamer Lab, KiSSim Notebook: Predict Ligand Profiling Using SiteAlign (Pooled Karaman and Davis Dataset). [https://github.com/volkamerlab/kissim\\_app/blob/v1.1.0/notebooks/006\\_evaluation/013\\_profiling\\_karaman\\_davis\\_\\_sitealign.ipynb](https://github.com/volkamerlab/kissim_app/blob/v1.1.0/notebooks/006_evaluation/013_profiling_karaman_davis__sitealign.ipynb), Version 1.1.0 [accessed 2022-04-24].
- (20) Moret, N.; Clark, N. A.; Hafner, M.; Wang, Y.; Lounkine, E.; Medvedovic, M.; Wang, J.; Gray, N.; Jenkins, J.; Sorger, P. K. Cheminformatics Tools for Analyzing and Designing Optimized Small-Molecule Collections and Libraries. *Cell Chem. Biol.* **2019**, *26*, 765–777.e3.
- (21) Volkamer Lab, KiSSim Notebook: Predict Ligand Profiling Using KiSSim (Moret Dataset). [https://github.com/volkamerlab/kissim\\_app/blob/v1.1.0/notebooks/006\\_evaluation/016\\_profiling\\_moret.ipynb](https://github.com/volkamerlab/kissim_app/blob/v1.1.0/notebooks/006_evaluation/016_profiling_moret.ipynb), Version 1.1.0 [accessed 2022-04-24].
- (22) Volkamer Lab, KiSSim Notebook: Compare Different Similarity Methods. [https://github.com/volkamerlab/kissim\\_app/blob/v1.1.0/notebooks/006\\_evaluation/010\\_comparative\\_analyses.ipynb](https://github.com/volkamerlab/kissim_app/blob/v1.1.0/notebooks/006_evaluation/010_comparative_analyses.ipynb), Version 1.1.0 [accessed 2022-04-24].

### 3.1.2 Analyzing Kinase Similarity in Small Molecule and Protein Structural Space to Explore the Limits of Multi-Target Screening Publication C

This study is a collaboration with members of Peter Kolb's group in Marburg, Germany. They encountered difficulties to find selective hits for intended kinase profiles defining multiple on- and off-targets. In the past, their docking-based virtual screening approach showed more promising results; thus, our group investigated if the kinase profiles of interest represent difficult profiles in terms of kinase similarity. Investigated measures included the similarity across pocket sequences, interaction fingerprints, pocket structure, and ligand profiles.



Contribution:

#### Middle author

Conceptualization (10%)

Data Curation (20%)

Formal Analysis (20%)

Investigation (20%)

Methodology (20%)

Software (20%)

Validation (20%)

Visualization (20%)

Writing — Original Draft (10%)






Writing — Review & Editing (20%)

Reprinted from Schmidt D, Scharf MM, Sydow D, Aßmann E, Martí-Solano M, Keul M, Volkamer A, Kolb P. Analyzing Kinase Similarity in Small Molecule and Protein Structural Space to Explore the Limits of Multi-Target Screening. *Molecules*. **2021**; 26(3):629. 10.3390/molecules26030629.

Open access article licensed under a CC BY 4.0 license.

## Article

# Analyzing Kinase Similarity in Small Molecule and Protein Structural Space to Explore the Limits of Multi-Target Screening

Denis Schmidt <sup>1,†</sup>, Magdalena M. Scharf <sup>2</sup>, Dominique Sydow <sup>3</sup>, Eva Aßmann <sup>3</sup>, Maria Martí-Solano <sup>2,‡</sup>, Marina Keul <sup>4</sup>, Andrea Volkamer <sup>3,\*</sup> and Peter Kolb <sup>2,\*</sup>

<sup>1</sup> Institut für Pharmazeutische und Medizinische Chemie, Heinrich-Heine-Universität Düsseldorf, 40225 Düsseldorf, Germany; denis.schmidt@uni-duesseldorf.de

<sup>2</sup> Pharmaceutical Chemistry, Philipps-University Marburg, Marbacher Weg 6, 35037 Marburg, Germany; magdalena.scharf@pharmazie.uni-marburg.de (M.M.S.); mariamartisolano@gmail.com (M.M.-S.)

<sup>3</sup> In Silico Toxicology and Structural Bioinformatics, Institute of Physiology, Charité—Universitätsmedizin Berlin, Charitéplatz 1, 10117 Berlin, Germany; dominique.sydow@charite.de (D.S.); eva.assmann@mail.de (E.A.)

<sup>4</sup> Chemical Biology, Technical University Dortmund, Otto-Hahn-Str. 4a, 44227 Dortmund, Germany; marina.keul@tu-dortmund.de

\* Correspondence: andrea.volkamer@charite.de (A.V.); peter.kolb@uni-marburg.de (P.K.)

† Current address: Computational Chemistry, Medicinal Chemistry, Boehringer Ingelheim Pharma GmbH & Co. KG, 88397 Biberach, Germany.

‡ Current address: MRC Laboratory of Molecular Biology, Cambridge CB2 0QH, UK.



**Citation:** Schmidt, D.; Scharf, M.M.; Martí-Solano, M.; Aßmann, E.; Sydow, D.; Keul, M.; Volkamer, A.; Kolb, P. Analyzing Kinase Similarity in Small Molecule and Protein Structural Space to Explore the Limits of Multi-Target Screening. *Molecules* **2021**, *26*, 629. <https://doi.org/10.3390/molecules26030629>

Academic Editors: Pilar Cossio, Claudio Cavasotto and Mattia Sturlese  
Received: 30 November 2020  
Accepted: 19 January 2021  
Published: 26 January 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Abstract:** While selective inhibition is one of the key assets for a small molecule drug, many diseases can only be tackled by simultaneous inhibition of several proteins. An example where achieving selectivity is especially challenging are ligands targeting human kinases. This difficulty arises from the high structural conservation of the kinase ATP binding sites, the area targeted by most inhibitors. We investigated the possibility to identify novel small molecule ligands with pre-defined binding profiles for a series of kinase targets and anti-targets by in silico docking. The candidate ligands originating from these calculations were assayed to determine their experimental binding profiles. Compared to previous studies, the acquired hit rates were low in this specific setup, which aimed at not only selecting multi-target kinase ligands, but also designing out binding to anti-targets. Specifically, only a single profiled substance could be verified as a sub-micromolar, dual-specific EGFR/Erbb2 ligand that indeed avoided its selected anti-target BRAF. We subsequently re-analyzed our target choice and in silico strategy based on these findings, with a particular emphasis on the hit rates that can be expected from a given target combination. To that end, we supplemented the structure-based docking calculations with bioinformatic considerations of binding pocket sequence and structure similarity as well as ligand-centric comparisons of kinases. Taken together, our results provide a multi-faceted picture of how pocket space can determine the success of docking in multi-target drug discovery efforts.

**Keywords:** multi-target ligands; docking; chemoinformatics; bioinformatics; kinases; binding site comparison; anti-target

## 1. Introduction

Small-molecule modulators of protein function are the most frequent type of molecules in use for the treatment of diseases due to their favorable pharmacokinetic properties [1]. Such ligands bind to cavities on protein surfaces—the binding sites—and compete with substrates or native ligands, or they alter the protein conformation. For such a molecule to become an efficacious drug, it has to possess adequate affinity for its protein target, solubility, membrane permeability and stability. Furthermore, its overall binding profile has to be compatible with its intended mode of action. On the one hand, unintended binding to proteins other than the primary target can cause side effects. On the other hand, several

diseases require the simultaneous modulation of multiple proteins in order to be treated successfully [2–4]. While the binding profile of a ligand can certainly be engineered through medicinal chemistry, starting out from a scaffold or a molecule that already displays the desired affinities towards several target proteins can only be advantageous. Successful approaches to identify dual-selective compounds by means of docking have been published before [5–7]. In this work, we were interested to see whether this docking-based approach can be broadened beyond what has previously been done by applying it to more than two proteins and by also including anti-targets.

Our target family of choice for this study were kinases, which are established drug targets to combat cancer and inflammatory diseases [8]. They play a major role in signal transduction by phosphorylating other proteins and are frequently mutated in tumors [9,10]. The human kinome consists of over 540 protein kinases that were clustered by Manning et al. [11] into eight major groups, e.g., tyrosine kinases (TKs), based on overall sequence similarity. The interest in this protein family has resulted in the generation of a wealth of freely available compound, bioactivity and structural data, which can be used for computer-aided analysis and guidance in drug design [12]. Such data have also successfully been applied to develop predictive models [13]. As of July 2020, there are 4864 X-ray structures of human kinases available in the PDB [14] (number obtained from KLIFS, an open-source database for kinase–ligand interaction fingerprints and structures, [15–17]) and 53 small molecule kinase drugs (only counting ‘-ribs’) have made it to FDA approval [18]. Most of the approved drugs bind to the ATP-binding pocket and its immediate surroundings, which include important regions like the hinge region (forming key hydrogen bonds), the DFG motif, the  $\alpha$ C-helix and the glycine-rich (G-rich) loop. They either block the active state of the kinase or lock the protein in an inactive state. In the active state, the DFG motif’s phenylalanine (F) is pointing into the hydrophobic pocket, while the aspartate (D) coordinates a magnesium ion for ATP binding (DFG-in conformation). Additionally, conserved residues in the  $\alpha$ C-helix and  $\beta$ 3-sheet form a salt bridge ( $\alpha$ C-in conformation) and the G-rich loop stabilizes ATP. Different descriptors have been defined to classify activity states based on these structural properties [15,16,19,20]. Since kinases share a similar fold—especially in the active site—most kinase inhibitors suffer from promiscuous binding. Sunitinib, for example, was found to bind to more than 50% of a panel of 290 kinases [21]. Similarly, dasatinib binds to a broad spectrum of TKs with high affinity [21]. Such promiscuous binding has been related to several of the side effects of current kinase inhibitors. Taken together, these facts clearly demonstrate the need for methods which are able to filter out compounds binding to kinases considered as anti-targets in order to facilitate the design of more selective kinase inhibitors.

In this study, we investigated the possibility to design and identify ligands with a defined polypharmacology through structure-based approaches. To that end, we docked identical molecule sets against multiple kinase targets to identify novel kinase inhibitors with defined rationally-selected profiles. Importantly, the resulting hits were not only selected for their ability to simultaneously bind to multiple kinase targets but also specifically filtered to avoid an established kinase anti-target. We also used available kinase-focused data to analyze different facets of kinase similarity in an attempt to evaluate the likelihood with which certain kinase combinations can be targeted simultaneously or individually. We evaluated the similarity of the binding sites based on the correlation of the docking ranks of the individual molecules, i.e., we considered binding sites to be similar when they were predicted to bind the same compounds in a similar docking rank order. Moreover, we assessed the congruence of this ranking with other ligand-centric as well as protein sequence- and structure-based measures. We evaluated our docking calculations by predicting selective as well as multi-target ligands (with defined targets and anti-targets) for three triplets of kinases and tested these predicted ligands experimentally. In this way, we identified and validated a sub-micromolar dual inhibitor of EGFR and ErbB2, with no activity against BRAF as the anti-target.

The results of our study allow us to reflect on the similarity boundaries determining the suitability of structure-based drug design (SBDD) to successfully address a specific multi-target combination. In particular, they show the necessity for ever-larger libraries that hold diverse molecules, in order to increase the likelihood of identifying ligands tailored towards predefined selectivity profiles.

## 2. Results and Discussion

Herein, the selected kinase profiles are rationalized first and the virtual screening results against these panels are discussed. Then, the experimental results for the selected compounds are presented. Finally, the similarity between the kinases of the studied profiles is analyzed with respect to different ligand- and protein-centric measures.

### 2.1. Kinase Profiles

We focused our analysis on a target panel comprising kinases with medical relevance as well as a typical anti-target, known to be associated with frequent side effects of kinase inhibitors. All kinases in this set have been thoroughly characterized in the literature and are summarized in Table 1.

**Table 1.** List of kinases used in this study.

Kinase <sup>a</sup>	Synonyms	UniProt ID	Group	Family
EGFR	ErbB1	P00533	TK	EGFR
ErbB2	Her2	P04626	TK	EGFR
PI3K	PI3KCA, p110 $\alpha$	P42336	Atypical	PIK
VEGFR2	KDR	P35968	TK	VEGFR
BRAF	-	P15056	TKL	RAF
CDK2	-	P24941	CMGC	CDK
LCK	-	P06239	TK	Src
MET	-	P08581	TK	MET
p38 $\alpha$	MAPK14	Q16539	CMGC	MAPK

<sup>a</sup> EGFR, epidermal growth factor receptor; ErbB2, Erythroblastic leukemia viral oncogene homolog 2; PI3K, phosphatidylinositol-3-kinase; VEGFR2, vascular endothelial growth factor receptor 2; BRAF, rapidly accelerated fibrosarcoma isoform B; CDK2, cyclic-dependent kinase 2; LCK, lymphocyte-specific protein tyrosine kinase; MET, mesenchymal-epithelial transition factor; p38 $\alpha$ , p38 mitogen activated protein kinase  $\alpha$ .

The Erythroblastic leukemia viral oncogene homolog (ErbB) subclass of Receptor Tyrosine Kinases (RTKs) consists of four members named from ErbB1 (better known as epidermal growth factor receptor [EGFR]) to ErbB4 and they bind the EGF family of peptides with their extracellular region [22]. The ErbB family is involved in the regulation of a multitude of signaling pathways associated with cell development. It is thus not surprising that aberrant ErbB signaling occurs in many cancers. Of note, patients with altered EGFR and ErbB2 expression suffer from a more aggressive disease. Especially breast cancer over-expressing ErbB2 is associated with poor patient prognosis [23]. Unfortunately, therapy is often effective only for a short time and tumors will escape inhibition by activating pathways downstream of ErbB receptors via other kinases. This has been demonstrated for the phosphatidylinositol-3-kinase (PI3K) pathway, which is directly or indirectly activated by most ErbBs [24]. After initial downregulation of PI3K activity upon inhibition of ErbBs, this pathway often recovers. Combination therapies are used to circumvent this problem, albeit with limited success. There is also evidence that tumor cells escape the negative effects of EGFR inhibition by upregulating tumor angiogenesis-promoting growth factors. A study used two antibodies against EGFR and VEGFR2 (vascular endothelial growth factor receptor 2), respectively, to treat gastric cancer grown in nude mice [25]. The combination resulted in significantly greater inhibition of tumor growth.

Based on these experimental observations, we aggregated the investigated kinases in “profiles” (Table 2). Profile 1 combined EGFR and ErbB2 as targets (indicated by a ‘+’) and BRAF (from rapidly accelerated fibrosarcoma isoform B) as a (general) anti-target

(designated by a ‘-’). Out of similar considerations, Profile 2 consisted of EGFR and PI3K as targets and BRAF as anti-target. This profile is expected to be more challenging as PI3K is an atypical kinase and thus less similar to EGFR than for example ErbB2 used in Profile 1. Profile 3, comprised of EGFR and VEGFR2 as targets and BRAF as anti-target, was contrasted with the hit rate that we found with a standard docking against the single target VEGFR2 (Profile 4).

**Table 2.** Definitions of kinase profiles and the numbers of screening compounds selected for each profile.

ID	Kinase Profile <sup>a</sup>	No. of Tested Compounds
1	+EGFR+ErbB2-BRAF	18 <sup>b,c</sup>
2	+EGFR+PI3K-BRAF	9 <sup>b</sup>
3	+EGFR+VEGFR2-BRAF	8 <sup>c</sup>
4	+VEGFR2	4

<sup>a</sup> + and - indicate targets and anti-targets, respectively. <sup>b</sup> Three compounds are identical between Profiles 1 and 2 but were independently selected from the docking calculations against both profiles. <sup>c</sup> One compound is identical between Profiles 1 and 3 but was independently selected from the docking calculations against both profiles.

To broaden the comparison and obtain an estimate for the promiscuity of each compound, the kinases CDK2 (cyclic-dependent kinase 2), LCK (lymphocyte-specific protein tyrosine kinase), MET (mesenchymal-epithelial transition factor) and p38 $\alpha$  (p38 mitogen activated protein kinase  $\alpha$ ) were included in the experimental assay panel and the structure-based bioinformatics comparison as commonly used anti-targets.

## 2.2. Virtual Screening against Kinase Profiles

Following our previous approach to identify ligands with tailored selectivity profiles by virtual screening [6], the aim of this study was to evaluate the possibility to add anti-targets to a kinase profile. We hence modified our previous approach to incorporate profiles with more than two kinases, multiple structures per kinase and the selection of targets and anti-targets (Equation (1) in Section “Data and Methods”).

Starting from the EGFR/ErbB2 pair, we included BRAF as a promiscuous anti-target, resulting in Profile 1 (see Section 2.4.1 for a discussion of promiscuity values). We therefore prioritized molecules with high rank (i.e., favorable docking scores) in EGFR and ErbB2 as well as low rank (i.e., unfavorable docking interactions) in BRAF. The ZINC lead-like and ZINC drug-like subsets, containing 4.6 and 10.6 million molecules, respectively, were docked into each of the selected structures of these kinases (cf. “Data and Methods”). After docking the smaller lead-like subset to EGFR, ErbB2 and BRAF, the kinases comprising Profile 1, we identified a high mutual overlap in terms of well-ranked compounds between these three kinases (6982 common compounds in the top-ranked 25,000 compounds for EGFR and ErbB2, 4732 for ErbB2/BRAF and 4675 for EGFR/BRAF, respectively, each number representing the maximum over all pairwise comparisons of all docking runs of the lead-like ZINC subset into the different structures of these kinases). Thus, many promising poses in EGFR/ErbB2 were invalidated by a high-rank in the anti-target BRAF. Therefore, we deemed the docking of the larger drug-like subset necessary to obtain a sufficient number of poses with reasonable binding modes to select from after re-ranking. The re-ranking procedure was devised to prioritize molecules matching the requested profile, i.e., molecules with favorable docking rank in all targets but unfavorable docking ranks in all anti-target structures (see “Data and Methods” for details). Finally, we selected 18 molecules (see Table 2 and Table S1) based on visual inspection for this profile (see “Data and Methods” for more detail) from the re-ranked lists of both molecule sets and evaluated these experimentally.

Similarly, for Profile 2, using EGFR and PI3K as targets and again BRAF as an anti-target (Table 2), we docked both the ZINC lead-like as well as the drug-like subsets. Again, we deemed the drug-like subset to be necessary due to the large overlap of the top-scoring



lead-like molecules of the targets with the ones ranked favorably in the anti-target (4683, 4675 and 6591 for EGFR/PI3K, EGFR/BRAF, and PI3K/BRAF, respectively). For this profile, we selected nine molecules (Table 2 and Table S1).

The parallel docking calculations for Profiles 3 and 4 (Table 2) yielded eight and four candidate ligands, respectively (Table 2 and Table S1). For Profile 3, the number of common molecules in the top 25,000 was 4610 and 5544 for VEGFR2/EGFR and VEGFR2/BRAF, respectively. As above, the overlap between EGFR and BRAF was 4675.

### 2.3. Experimental Validation

In total, 24 compounds selected from Profiles 1 and 2 (Table 2 and Table S1) were tested in the DiscoverX assay against kinases EGFR, ErbB2, BRAF, VEGFR2, LCK, CDK2, MET, p38 $\alpha$  and PI3K (Table S2), as well as in an additional confirmatory assay by Eurofins against EGFR, ErbB2, BRAF and PI3K (Table S3). Only one of the 24 compounds, DS39984, showed measurable binding to the desired kinases (Profile 1, Table 3 and Tables S1–S3), while binding to neither Profile 1's anti-target BRAF nor any of the other tested kinases (VEGFR2, CDK2, LCK, MET, p38 $\alpha$  and PI3K). This compound DS39984 emerged from the screening campaign against Profile 1 (+EGFR+ErbB2–BRAF) and was picked from the drug-like subset of the ZINC database. We further validated the binding of this ligand and determined binding curves in an independent assay with IC<sub>50</sub> values of 324 and 220 nM (note that both enantiomers were docked—with the R-enantiomer more favorably ranked, but the racemate was tested) against EGFR and ErbB2insYVMA (a variant of ErbB2 with an insertion of four residues distant from the binding pocket), respectively (Table 3, Rauh Lab).

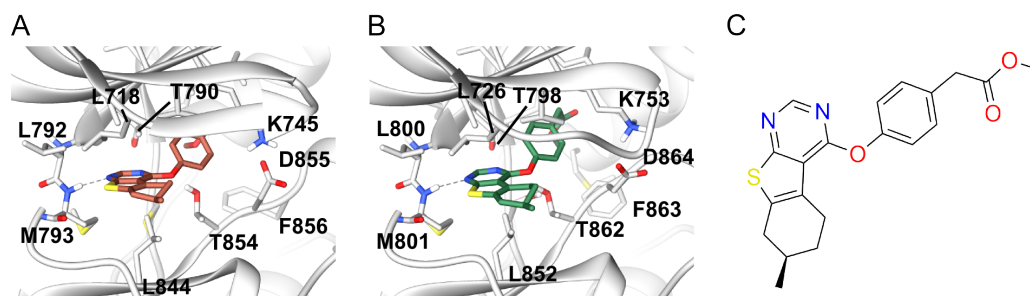
**Table 3.** Assay results for identified hit molecules.

Compound	P <sup>a</sup>	Research Lab	Unit	EGFR	ErbB2	ErbB2insYVMA	BRAF
DS39984	1	DiscoverX	% ctrl. activity at 10 $\mu$ M	17	21	n.d. <sup>c</sup>	– <sup>d</sup>
DS39984	1	Eurofins	% inhib. at 20 $\mu$ M $\pm$ s.d.	59 $\pm$ 3.2	– <sup>b</sup>	n.d. <sup>c</sup>	– <sup>b</sup>
DS39984	1	Rauh Lab	IC <sub>50</sub> $\pm$ s.d.	324 $\pm$ 50 nM	n.d. <sup>c</sup>	220 $\pm$ 3 nM	n.d. <sup>c</sup>
K001MM011	4	DiscoverX	% ctrl. activity at 10 $\mu$ M	1.4	53	n.d. <sup>c</sup>	– <sup>d</sup>

<sup>a</sup> Profile as per Table 2; <sup>b</sup> Below 50% cutoff for hit as recommended by Eurofins; <sup>c</sup> not determined; <sup>d</sup> percent control activity  $\geq$  99%, i.e., no measurable inhibition.

As shown in the predicted binding modes in EGFR and ErbB2 (Figure 1), DS39984 adopts a similar binding orientation in both proteins, with the pyrimidine portion forming a hydrogen bond to the hinge region. The methylester moiety is oriented more towards the back of the binding pocket, where both kinases feature rather voluminous cavities. This predicted binding mode to the hinge region is consistent with the sensitivity of DS39984 towards the T790M mutation: Affinity for the EGFR L858R/T790M double mutant is abolished (IC<sub>50</sub> > 10  $\mu$ M), whereas the affinity for the EGFR L858R mutant is 2351  $\pm$  397 nM. In contrast, in both BRAF structures used herein, the predicted poses are flipped and have their methylester moiety pointing towards the solvent (Figure S1). A similar hinge binding interaction as in EGFR and ErbB2 is only present in one of the two poses (in the docking to BRAF structure 1UWH). This occurs despite the fact that in the 1UWH crystal structure the deep back pocket is open due to the crystallized ligand. Thus, in principle, a binding mode of DS39984 similar to the ones predicted in EGFR and ErbB2 is not per se excluded in BRAF due to steric reasons.

Note that DS39984 is not present in ChEMBL and has low similarity to known kinase ligands in ChEMBL (no ligand with Tanimoto similarity >0.7 as implemented in the ChEMBL web interface as of 18 October 2020). Furthermore, none of the additionally tested kinases (LCK, CDK2, MET and p38 $\alpha$ ) were inhibited by the molecule, which underlines, together with absence of BRAF inhibition, the potential of DS39984 as a novel, selective nanomolar EGFR and ErbB2 inhibitor.



**Figure 1.** Docking poses of the R-enantiomer of compound DS39984 bound to (A) EGFR (PDB 3POZ, DFG-in) and (B) ErbB2 (PDB 3PP0, DFG-in); and (C) 2D representation of DS39984. The protein structure is shown as cartoon, colored in grey, the compound as sticks. Interacting binding site residues are represented as sticks and labeled. Hydrogen bonds between protein and ligand are indicated by dark gray dashed lines.

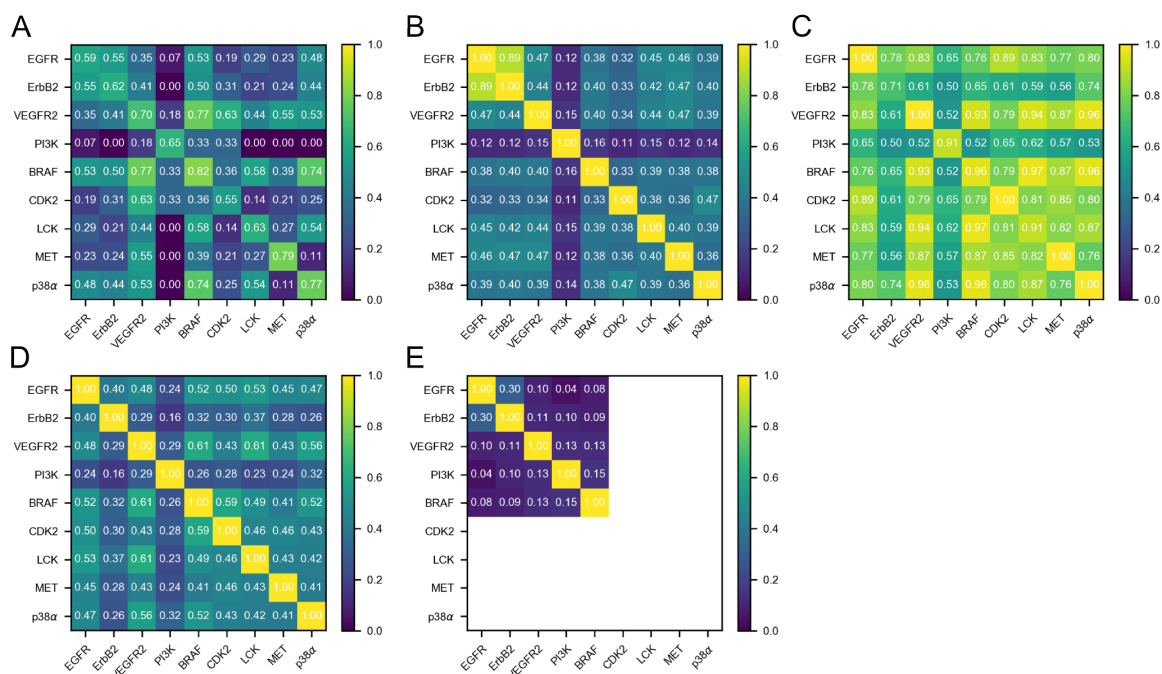
Eight compounds were selected for Profile 3 (+EGFR+VEGFR2–BRAF, Table 2 and Table S1) and tested in the DiscoverX assay against EGFR, VEGFR2, BRAF and ErbB2. However, none of the compounds exhibited a relevant effect against any of these kinases. To crudely estimate the ligandability of VEGFR2, we docked against this target individually (Profile 4). However, we did not observe many poses that passed our visual inspection (see “Data and Methods” for details) and were able to select only four compounds from the docking to VEGFR2. These were tested in the same assay. Again, none of these compounds showed an effect on VEGFR2 activity. While the number of tested compounds is certainly too small to draw clear conclusions, the fact that only few compounds could be considered in the first place and that those few were inactive might indicate that VEGFR2 is more challenging with respect to the identification of ligands by docking than for example EGFR and ErbB2. One explanation for this could be associated with the fact that the vast majority of VEGFR2 structures show DFG-out(like) conformations (ratio DFG-in/out(like) structures in the PDB: 5/34 for VEGFR2 compared to 168/22 for EGFR, as of KLIFS 25 November 2020). Note that several FDA-approved kinase inhibitors bind to DFG-out(like) VEGFR2 conformations, e.g., axitinib, sunitinib and sorafenib [26]. In contrast, we used DFG-in conformations of VEGFR2 for docking in order to maximize comparability with the other kinase structures used.

Unexpectedly, however, we found that one of these four compounds selected for VEGFR2 inhibition, K001MM011, actually inhibited EGFR and, to a lesser extent, ErbB2 (Tables 3 and Table S2). While K001MM011 was picked from the docking to VEGFR2 only, we retrospectively inspected the ranking of this compound in the docking to EGFR and ErbB2. In EGFR, K001MM011 was found to be ranked within the best 10,000 compounds (rank 9527) of the lead-like subset in PDB 3POZ, while, in ErbB2, K001MM011 was ranked not as highly (best rank: 123,665 in PDB 3PP0).

In light of these experimental results and the comparative scarcity of ligands with the intended profiles, we decided to better investigate the kinases involved, with a view towards the possibility to predict the sensibility of a particular target combination.

#### 2.4. Kinase Similarities

Designing kinase inhibitors with intended dual target activity that avoid binding to one or several specific anti-targets is a non-trivial task, as evidenced by the docking part of our study. To better understand how difficult it may be to design such inhibitors rationally, five different measures of inter-kinase similarity—each contributing a different level of granularity and a different viewpoint—were investigated (Figure 2). Such an analysis potentially enables a priori estimations of the success of these endeavors for a given target/anti-target profile.



**Figure 2.** Heat maps of pairwise kinase similarities for the different measures used in this work: (A) ligand profile similarity (LigProfSim); (B) pocket sequence similarity (PocSeqSim); (C) interaction fingerprint similarity (IFPSim); (D) pocket structure similarity (PocStrucSim); and (E) docking rank similarity (DockRankSim) based on the lead-like subset of ZINC. Note that docking was only performed for the five kinases of Profiles 1–4, thus the remaining entries remain empty (white) in the matrix.

#### 2.4.1. Ligand Profile Similarity (LigProfSim)

A first glance at the ChEMBL kinase ligand subsets revealed that none of the investigated kinases seems to be overly selective in terms of the ligands it recognizes, which is in accordance with previous kinome-wide profiling studies [21,27]. Given that the promiscuity values (Table 4, diagonal of Figure 2A and Table S4) range from 0.55 for CDK2 to 0.82 for BRAF, all nine kinases bind more than half of the compounds tested against them at an affinity cut-off of 500 nM. Accordingly, BRAF is the most promiscuous kinase in the set, justifying its use as a general kinase anti-target in this study.

Second, considering LigProfSim, it becomes evident that EGFR, ErbB2 and BRAF are more similar to each other than the remaining kinases (top-left quarter of Figure 2A), which renders finding a compound for Profile 1 (Table 2) a difficult task. With LigProfSim values of 0.53 and 0.55, EGFR is more similar to ErbB2 and BRAF, respectively, than to any other kinase in the set (Table S4). The same holds true for ErbB2, while BRAF has also higher similarities to other kinases in the set. In contrast, with a mean similarity value of 0.18, PI3K has the lowest mean LigProfSim similarity to all nine kinases. This is not unexpected, given that PI3K is the only atypical kinase in the set, but it underlines how challenging the definition of Profile 2 is. Note that, while 4150 compounds were tested against PI3K (with 2706 being active), PI3K has fewer than five common actives with most kinases, except for EGFR (13 common actives of 180 compounds tested against both targets) and VEGFR2 (32 of 175) (see Tables 5 and Tables S5 and S6). While all kinases were assayed against at least 1500 compounds, a few other kinase pairs—not including PI3K—exist that have only a low number of tested compounds in common, e.g., CDK2/BRAF (14), CDK2/p38α (8) or ErbB2/p38α (9, see Table S5), which makes thorough comparison difficult. Finally, with a value of 0.35, EGFR and VEGFR2 do not show high similarity from this ligand-centric

perspective, while, as mentioned above, VEGFR2 and BRAF show considerably higher similarity (0.77). These numbers indicate that Profile 3 is very difficult.

**Table 4.** Kinase promiscuity measures calculated as the ratio of ligands active on a specific kinase (column 2). In Columns 3–6, mean values and standard deviations (s.d.) of ligand profile similarity (LigProfSim), pocket sequence similarity (PocSeqSim), interaction fingerprint similarity (IFPSim) and pocket structure similarity (PocStrucSim) per kinase are given. Note: Two kinases having a similar mean value for a particular similarity measurement does not imply that they are similar to each other (especially when large s.d. values are associated with the measure; see Figure 2 for pairwise kinase comparisons).

Kinase	Promiscuity <sup>a</sup>	Mean ( $\pm$ s.d.) <sup>b</sup>			
		LigProfSim	PocSeqSim	IFPSim	PocStrucSim
EGFR	0.59	0.37 ( $\pm$ 0.18)	0.50 ( $\pm$ 0.28)	0.81 ( $\pm$ 0.10)	0.50 ( $\pm$ 0.21)
ErbB2	0.62	0.37 ( $\pm$ 0.19)	0.50 ( $\pm$ 0.28)	0.64 ( $\pm$ 0.09)	0.38 ( $\pm$ 0.24)
PI3K	0.65	0.18 ( $\pm$ 0.23)	0.23 ( $\pm$ 0.29)	0.61 ( $\pm$ 0.13)	0.33 ( $\pm$ 0.26)
BRAF	0.82	0.56 ( $\pm$ 0.18)	0.42 ( $\pm$ 0.23)	0.82 ( $\pm$ 0.16)	0.52 ( $\pm$ 0.21)
CDK2	0.55	0.33 ( $\pm$ 0.17)	0.40 ( $\pm$ 0.24)	0.80 ( $\pm$ 0.12)	0.49 ( $\pm$ 0.21)
LCK	0.63	0.34 ( $\pm$ 0.22)	0.45 ( $\pm$ 0.23)	0.82 ( $\pm$ 0.13)	0.50 ( $\pm$ 0.21)
MET	0.79	0.31 ( $\pm$ 0.24)	0.45 ( $\pm$ 0.23)	0.79 ( $\pm$ 0.14)	0.46 ( $\pm$ 0.22)
p38 $\alpha$	0.77	0.43 ( $\pm$ 0.27)	0.44 ( $\pm$ 0.23)	0.82 ( $\pm$ 0.15)	0.47 ( $\pm$ 0.21)
VEGFR2	0.70	0.51 ( $\pm$ 0.18)	0.46 ( $\pm$ 0.23)	0.83 ( $\pm$ 0.16)	0.50 ( $\pm$ 0.23)

<sup>a</sup> Kinase promiscuity based on ligand affinity data from ChEMBL, measured as ratio of active compounds over tested compounds (activity threshold IC<sub>50</sub> = 500 nM, cf. Table 5); <sup>b</sup> mean  $\pm$  standard deviation (s.d.) values for LigProfSim, PocSeqSim, IFPSim and PocStrucSim of the respective kinase to all nine kinases (including itself) in the set.

**Table 5.** Dataset composition for the similarity analysis. Listed are the numbers of compounds active and tested against each kinase used for the ligand profile similarity (LigProfSim), as well as number of structures used for the interaction fingerprint similarity (IFPSim) and the pocket structure similarity (PocStrucSim) calculations.

Kinase (Family/Group)	# Compounds		# Structures	
	Actives	Tested	IFPSim	PocStrucSim
EGFR (TK/EGFR)	3382	5702	150	15
ErbB2 (TK/EGFR)	1048	1690	2	2
PI3K (Atypical/PIK)	2706	4150	26	2
VEGFR2 (TK/VEGFR)	5197	7426	41	13
BRAF (TKL/RAF)	2968	3625	69	25
CDK2 (CMGC/CDK)	837	1520	377	43
LCK (TK/Src)	976	1552	34	29
MET (TK(MET))	2248	2851	70	11
p38 $\alpha$ (CMGC/MAPK)	2753	3581	196	36
Total	22,115	32,097	965	176

#### 2.4.2. Pocket Sequence Similarity (PocSeqSim)

Classically, kinases are clustered based on their full sequence similarity, such as in the well-known phylogenetic human kinome tree by Manning et al. [11]. The kinome tree is often considered when checking for relationships among kinases, cross-reactivity and anti-targets. Arguably, EGFR and ErbB2 are the most closely related kinases in the set, both belonging to the TK branch and the EGFR family, followed by similarity to VEGFR2 (TK branch, VEGFR family). BRAF is less closely related (tyrosine-kinase-like [TKL] branch, RAF family). Finally, PI3K belongs to the atypical kinases and is only distantly related. Full kinase details are listed in Table 1.

Here, we refined this sequence-based view of similarity to only consider the 85 residues forming the binding site in each kinase (PocSeqSim). Also in this “pocket sequence” space, the two EGFR family members EGFR and ErbB2 show the highest similarity of 0.89 (Figure 2B, numbers in Table S7). All other kinase pairs have similarity values below 0.48, thus less than 50% identical pocket residues. VEGFR2, MET and LCK, three other kinases from the TK class, have PocSeqSim between 0.42 and 0.47 to EGFR and Erb2; BRAF (TKL), p38 $\alpha$  and CDK2 (both from the CMGC family) have values in the range of 0.32 to 0.40. Again, PI3K shows the lowest similarity to all other eight kinases. This indicates that, first, the pocket sequence similarities follow a similar trend as the whole-sequence similarities and, second, that—due to the close relationship of EGFR and ErbB2—other less similar kinases of the TK branch such as VEGFR2, MET and LCK, but also BRAF (TKL), p38 $\alpha$  and CDK2 (both from the CMGC family), could be easier-to-satisfy anti-targets of +EGFR+ErbB2 ligands (Figure 2B).

#### 2.4.3. Interaction Fingerprint Similarity (IFPSim)

To take the interplay between the ligand and the protein into account, interaction fingerprint similarities (IFPSim) were investigated. Note that, for each kinase pair, *all* available X-ray structures were compared and that only the similarity between the highest-scoring pair is reported (Figure 2C, numbers in Table S8). In the IFPSim matrix, the diagonal describes the best match among all pairwise IFP comparisons between different structures from the same kinase. Interestingly, ErbB2 has a self-similarity of only 0.71. This could be a consequence of the relatively low structural coverage of this kinase. In fact, ErbB2 is only represented by two structures, whereas, for EGFR, 150 structures are available (Table 5).

With mean similarity values between 0.61 (lowest for PI3K) and 0.83 (highest for VEGFR2), the IFPSim values are generally higher than the LigProfSim and PocSeqSim values described above (Table 4). EGFR has a high mean similarity to all kinases of 0.81, whereas ErbB2 has a lower mean value of 0.64; note again the low structural coverage of ErbB2. While ErbB2 is most similar to EGFR (0.78) with respect to IFPSim (Figure 2C), it is less similar to BRAF (0.65), which would favor the development of a Profile 1 (+EGFR+ErbB2–BRAF) inhibitor. Interestingly, PI3K shows one of the highest similarities to EGFR (0.65), while it is less similar to BRAF (0.52), which, in contrast to other similarity measures, would support the feasibility of designing +EGFR+PI3K–BRAF compounds (Profile 2). In the case of VEGFR2, although similarity to EGFR is high (0.83), we observe an even higher similarity to BRAF (0.93), giving another indication of how difficult it may be to design-out this anti-target. On the other hand, the comparatively high similarity of VEGFR2 to EGFR might give an indication of why our Profile 4 compound actually inhibited EGFR.

#### 2.4.4. Pocket Structure Similarity (PocStrucSim)

Similarities with respect to structural and physicochemical properties of the binding sites were analyzed using the CavBase fast cavity graph comparison algorithm [28,29] (Figure 2D, numbers in Table S9). Note that binding sites were automatically detected using LigSite and thus may vary in precision throughout the different structures, even within the same kinase. Pairwise kinase similarities range from 0.16 (PI3K/ErbB2) to 0.61 (BRAF/VEGFR2 and LCK/VEGFR2) and are—with a mean value of 0.46 over all kinase pairs—generally lower than the IFPSim values described above (Table 4). Interestingly, EGFR and ErbB2 share only moderate similarity in this measure (0.40), while EGFR is more similar to all other kinases (including BRAF; 0.52), except PI3K (0.24). However, it should be noted that the structural coverage for ErbB2 and PI3K is much lower than for the other kinases, with only two structures each (Table 5). Note that EGFR is most similar to the anti-target BRAF (0.52). Thus, according to PocStrucSim, it appears difficult to develop ligands against all multi-target profiles (1–3, Table 2).

#### 2.4.5. Docking Rank Similarity (DockRankSim)

Finally, we leveraged the results of our docking experiments to derive a complementary similarity measure based on the rank correlation of the docked lead-like compounds (Figure 2E). DockRankSim values were calculated using only the top-scoring 25,000 lead-like molecules for each structure (about 0.5% of the ZINC lead-like subset at that time), since control calculations taking into account the entirety of docked molecule sets showed poor discrimination between different kinases. This lack of discrimination is likely due to the fact that the majority of molecules in the lead-like set are not kinase inhibitor-like. Therefore, the docking rank order of molecules past a certain threshold is noisy, i.e., all of them are more or less equally unlikely to bind. However, they will still receive different ranks based on small scoring differences, and these different ranks will lead to rather different—yet meaningless—correlations between the rankings. Only the five kinases that were included in the four docking profiles (Table 2) were considered, i.e., no values for CDK2, LCK, MET and p38 $\alpha$  were determined.

EGFR and ErbB2 have by far the highest mutual similarity of 0.3 within this set of kinases and a DockRankSim below 0.12 to all other kinases. While their higher mutual DockRankSim is not surprising given the close relationship between EGFR and ErbB2, it is encouraging that the docking results capture this.

Interestingly, the second highest DockRankSim observed is between PI3K and BRAF (0.15), followed by BRAF and VEGFR2 (0.13) as well as PI3K and VEGFR2 (0.13). This is surprising as PI3K, as atypical kinase, shares a rather low similarity to the remaining kinases using most other measures employed in this study (Figure 2A–D). The remaining DockRankSim values are around 0.1, which seems to be the center of the distribution. The smallest DockRankSim was observed between EGFR and PI3K (0.04), an indication that Profile 2 (+EGFR+PI3K–BRAFF) inhibitor design might be a challenge, at least computationally.

#### 2.4.6. Comparison of Similarity Analyses

To shed light on the ease of identifying inhibitors for the respective profiles and the possibility to predict the likelihood that multi-target design endeavors will be successful, five different protein similarity measures were calculated (Figure 2A–E). While the individual relationships between the nine kinases studied differ according to the five measures (which might also be due to missing data or noise in the data, as discussed above), several trends can be observed.

The similarity scores of the PocStrucSim and the IFPSim comparisons are distributed more evenly and clearly correlate with each other ( $R = 0.78$ ,  $p < 0.001$ , Figure S2). In addition, the pocket structure- and sequence-based comparisons follow a similar trend (PocStrucSim vs. PocSeqSim  $R = 0.73$ ,  $p < 0.001$ ). All other pairwise comparisons are less correlated, showing values in the range of  $R = [0.55, 0.59]$  with  $p < 0.001$  (Figure S2). While several measurements appeared to be correlated, differences between them are not surprising since the measures capture diverse views and thus complementary information of similarity. Nonetheless, it should be noted that the calculated values highly depend on the amount of available data. The conformational space of a kinase might be underrepresented if few kinase structures are available, which affects the structure-related measurements. Furthermore, since ChEMBL only provides a very sparse kinase-compound matrix of experimental measurements, the basis of compounds considered per kinase pair may differ strongly, affecting the LigProfSim values (as well as the promiscuity as defined here).

Besides PocStrucSim, all other measures imply a high similarity between EGFR and ErbB2, which is in favor of +EGFR+ErbB2 inhibitor design. Furthermore, LigProfSim, PocStrucSim and PocSeqSim suggest BRAF as a relevant and frequent anti-target, while this is less clear-cut for the IFPSim and DockRankSim measures. This fact renders design for all three profiles a challenging task. Furthermore, while PI3K is very dissimilar to EGFR from a sequence point of view (cf. Manning tree annotation), it showed higher similarity based on other measures such as IFPSim, which is encouraging for Profile 2 (+EGFR+PI3K–BRAFF) design. In this sense, the fact that our docking results did not yield

compounds with such a profile would suggest that similarity to the anti-target (in this case, BRAF) larger than to the intended target could be a key factor complicating the detection of the desired compounds.

Overall, our analyses suggest that ligand-, sequence- and structure-based approaches complement each other and can thus yield consistent insights into kinase similarities. It therefore seems advisable to carry out all of these analyses before a (virtual) screening campaign in order to take appropriate steps, e.g., adaptation of the molecule library to be screened, early on. Our ranking comparisons also suggest that similarity between one of the targets and the anti-target that is higher than the similarity between the two intended targets can be used as a prognostic indicator for difficult multi-target profiles.

### 3. Data and Methods

#### 3.1. Docking-Based Virtual Screening

Kinase crystal structures that were suitable for docking in general, as well as for the herein discussed purpose in particular, were carefully selected from the Protein Data Bank [14]. Structures were prioritized based on their resolution and the number of missing heavy atoms, with a focus on residues in and around the binding site. Furthermore, structures for target pairs were selected such that the structures for the two kinases involved were as similar as possible. The rationale behind this aim was to maximize the possibility to identify inhibitors binding to both structures. This structural similarity included the overall state of the kinase structure, as determined by the conformation of the DFG and  $\alpha$ C motifs, as well as visual comparisons of the binding site residues. Structures with similar side-chain conformations of equivalent amino acids were preferred, as far as such structures existed and the equivalence of amino acids could be rationally established, i.e., for homologous amino acids in EGFR/ErbB2 structure pairs, whereas this was not applicable to, e.g., EGFR/PI3K structure pairs due to their higher dissimilarity. Finally, the crystal structures (PDB IDs given in parentheses) for EGFR (1XKK [30], 3POZ [31]), ErbB2 (3PP0 [31], 3RCD [32]), BRAF (1UWH [33], 3PPK [34]), PI3K (4JPS [35]) and VEGFR2 (2P2H, 3WZD [36]) were downloaded from the PDB (a summary of structural details is presented in Table 6).

**Table 6.** Kinase structures used in docking experiments.

Kinase	PDB	DFG <sup>a</sup>	$\alpha$ C <sup>b</sup>
EGFR	1XKK	in	out
EGFR	3POZ	in	out
ErbB2	3PP0	in	in
ErbB2	3RCD	in	out-like
BRAF	1UWH	out	out
BRAF	3PPK	in	in
PI3K	4JPS	in	in
VEGFR2	2P2H	in	out
VEGFR2	3WZD	in	out

<sup>a</sup> Orientation of the conserved DFG motif (in/out), annotation from KLIFS [12]; <sup>b</sup> conformation of the  $\alpha$ C-helix (in/out), annotation from KLIFS [12].

The structures were prepared following the protocol in Kolb et al. [37]. Briefly, the first protein chain was used in case several were crystallized. Hydrogens were placed and minimized using the CHARMM (version 31b2) HBUILD command. The ZINC12 [38] lead-like and drug-like subsets (as of July 2015), containing 4.6 and 10.6 M molecules, respectively, were docked into the prepared receptor structures using DOCK 3.6 [39–43] as described in Schmidt et al. [6]. For EGFR, for which a ligand/decoy set is available from DUD-E [44], the prepared structures were additionally validated by their ability to enrich ligands over decoys. AUC values were found to be 0.87 (1XKK) and 0.85 (3POZ), which compares favorably to the value of 0.84 as published by DUD-E [44].

Based on these docking results, compounds were re-scored according to the different selectivity profiles of interest. In our previous work, we introduced a selectivity score for protein pairs, i.e., two docking runs, both being considered as target. Compounds were penalized for unfavorable (i.e., high) ranks in each docking run as well as a high rank difference between these two docking calculations (i.e., good/bad performance in docking A/B; Equation (1) in Schmidt et al. [6]).

Here, this procedure was extended to be applicable to more than two proteins, multiple structures per protein and the proper incorporation of anti-targets. Specifically, the docking calculations for multiple structures of the same kinase (e.g., 1XKK and 3POZ for EGFR) were aggregated by using only the best (i.e., numerically smallest) rank in any of the structures. Second, anti-targets were incorporated by inverting the docking rank order, based on the idea that a good docking performance is disfavored in anti-targets. Third, the equation was extended to multiple proteins by using the average rank (note that ranks for anti-targets were inverted beforehand) in all protein docking calculations of the respective profile (e.g., EGFR, ErbB2 and BRAF) and the rank difference between the highest and lowest docking rank in all proteins. Finally, in contrast to our previous procedure [6], logarithmic ranks were used to focus on the top-scoring molecules, based on the notion that the docking scores (and hence docking ranks) become less discriminating beyond the first few percent of the docked database for very large (and diverse) ligand sets, such as the ones used herein. Altogether, the score  $S$  of a molecule for the profile comprising kinases 1 to  $N$  was defined as follows:

$$S_{1,\dots,N} = \frac{1}{N} \sum_{k=1}^N P_k + \frac{(\max\{P_1, \dots, P_k, \dots, P_N\} - \min\{P_1, \dots, P_k, \dots, P_N\})}{2} \quad (1)$$

with

$$P_k = \log\left(\min_s R_{k,s}\right)$$

$$R_{k,s} = \frac{r_{k,s}}{m_{k,s}}$$

if kinase  $k$  was defined as target or

$$P_k = \log\left(\max_s R_{k,s}\right)$$

$$R_{k,s} = 1 - \frac{r_{k,s}}{m_{k,s}}$$

if kinase  $k$  was defined as anti-target. Here,  $P_k$  denotes the rank of a compound in kinase  $k$  aggregated over all structures  $s$  of this kinase.  $R_{k,s}$  denotes the scaled docking rank of the compound, calculated from the nominal docking rank  $r_{k,s}$  of this compound and the total number of molecules  $m_{k,s}$  that were docked into the  $s$ th structure of the  $k$ th kinase.

The poses of molecules receiving top ranks after applying this rescoring were visually inspected in their respective protein structure. This inspection is necessary in order to remove compounds which are ranked favorably for the wrong reasons, i.e., because of deficiencies in present-day force fields. Examples are unsatisfied hydrogen bond donors; burial of polar protein residues through apolar ligand moieties; charge mismatches; and ligand conformations with high strain.

### 3.2. Experimental Testing

#### 3.2.1. DiscoverX KINOMEScan

Ligand binding experiments for the molecules selected from Profiles 1 and 2 towards nine kinases (EGFR, ErbB2, LCK, CDK2, BRAF, MET, p38 $\alpha$ , PI3K and VEGFR2) and for molecules selected from Profiles 3 and 4 towards four kinases (EGFR, ErbB2, BRAF and VEGFR2) were carried out by DiscoverX using the supplied protocol as described in the



Supplementary Materials. Briefly, ligand affinity was measured by competition with a resin-bound standard ligand and washed-off kinase concentration was determined via qPCR.

Summarizing, binding of a compound to a kinase was tested in comparison to a control compound (see Table S2). Lower values generally indicate a higher affinity of the compound to the protein with values below 35% being considered as significant binding according to the information of DiscoverX.

### 3.2.2. Eurofins In Vitro Assay

Kinase inhibition assays for EGFR, ErbB2, PI3K and BRAF were carried out by Eurofins Cerep following the protocols of Weber et al. [45] (EGFR), Quian et al. [46] (ErbB2), Sinnaon et al. [47] (PI3K) and Kupcho et al. [48] (BRAF). Briefly, except for PI3K, compounds were incubated with the respective kinase, ATP, and a substrate analog, and the effect of each compound on phosphorylation was measured. In the case of PI3K, the displacement of biotinylated PIP3 from a PIP3-binding complex by unlabelled PIP3 (produced from PIP2 by PI3K) was measured by Homogeneous Time Resolved Fluorescence (HTRF).

Finally, inhibition of the respective kinases is calculated as the percentage inhibition of control activity. According to Eurofins, values above 50% inhibition represent significant inhibition and values between 25% and 50% weak inhibitory effects (Table S3).

### 3.2.3. IC<sub>50</sub> Determination

IC<sub>50</sub> determinations for EGFR, its mutants and ErbB2-insYVMA (Carna Biosciences, lot13CBS-0005K for EGFR-wt; Carna, lot13CBS-0537B for EGFR-L858R; Carna, lot12CBS-0765B for EGFR-L858R/T790M; and ProQinase, lot1525-0000-1/003 for ErbB2-insYVMA) were performed with the HTRF KinEASE-TK assay from Cisbio according to the manufacturer's instructions. Briefly, the amount of kinase in each reaction well was set to 0.60 ng EGFR-wt (0.67 nM), 0.10 ng EGFR-L858R (0.11 nM), 0.07 ng EGFR-T790M/L858R (0.08 nM), or 0.01 ng ErbB2-insYVMA (0.01 nM). An artificial substrate peptide (TK-substrate from Cisbio) was phosphorylated by EGFR or ErbB2. After completion of the reaction (reaction times: 25 min for EGFR-wt, 15 min for L858R, 20 min for L858R/T790M, and 40 min for ErbB2-insYVMA), the reaction was stopped by addition of buffer containing EDTA as well as an anti-phosphotyrosine antibody labeled with europium cryptate and streptavidin labeled with the fluorophore XL665. FRET between europium cryptate and XL665 was measured after an additional hour of incubation to quantify the phosphorylation of the substrate peptide. ATP concentrations were set at their respective  $K_m$ -values (9.5  $\mu$ M for EGFR-wt, 9  $\mu$ M for L858R, 4  $\mu$ M for L858R/T790M and 6  $\mu$ M for ErbB2-insYVMA) while a substrate concentration of 1  $\mu$ M, 225 nM, 200 nM and 1  $\mu$ M was used. Kinase and inhibitor were preincubated for 30 min before the reaction was started by addition of ATP and substrate peptide. An EnVision multimode plate reader (Perkin Elmer) was used to measure the fluorescence of the samples at 620 nm (Eu<sup>3+</sup>-labeled antibody) and 665 nm (XL665-labeled streptavidin) 50  $\mu$ s after excitation at 320 nm. The quotient of both intensities for reactions made with eight different inhibitor concentrations was then analyzed using the Quattro Software Suite for IC<sub>50</sub>-determination. Each reaction was performed in duplicate, and at least three independent determinations of each IC<sub>50</sub> were made.

## 3.3. Kinase Similarity Measures

The nine protein kinases investigated in this study were compared with five measures: their ligand binding profiles (LigProfSim), pocket sequence (PocSeqSim), interaction fingerprint (IFPSim) and structural information (PocStrucSim), as well as docking ranks (DockRankSim).

### 3.3.1. Ligand Profile Similarity (LigProfSim)

To compare kinases from a ligand point of view, their similarity with respect to binding the same ligands was investigated. The kinase subset of ChEMBL v.27 [49] was used as the profiling dataset, assembled from <https://github.com/openkinome/kinodata/releases/>

[tag/\\_pub\\_ligprofsim](#) (accessed September 2020). Only compounds measured in binding assays yielding a standard activity value as  $IC_{50}$  were taken into account. If the same compound was measured several times in the same assay (against the same kinase), only the lowest  $IC_{50}$  value was kept (most active). Compounds were considered active against a kinase if their  $IC_{50}$  value was below 500 nM, otherwise inactive. For each of the nine kinases studied here, the total number of measured compounds and the number of active compounds was determined (Table 5).

The pairwise ligand profile similarity (LigProfSim) between two kinases was calculated as the ratio of compounds active on both kinases divided by the total number of compounds tested on both kinases (Figure 2A, absolute values in Tables S4–S6). Note that, for the individual kinases, this “self-similarity” yields the fraction of active compounds with respect to all compounds tested, which can also be interpreted as a simple measure for promiscuity (Table 4).

### 3.3.2. Pocket Sequence Similarity (PocSeqSim)

Pocket sequences and binding site definitions were taken from the KLIFS database [15–17]. Based on the analysis of known kinase–ligand crystal structures, van Linden et al. [15] defined the ATP-binding pocket of kinases by 85 residues which cover most interactions with known inhibitors (front and back-cleft binders). These residues include known motifs such as the DFG motif, the hinge region and the  $\alpha$ C-helix.

To compare kinase binding sites based on sequences, the master multiple sequence alignment (MSA) of the 85 binding pocket residues for all human kinases available from KLIFS was used and the nine kinases investigated in this work were extracted. Pocket sequence similarity (PocSeqSim)—in this case residue identity—between two kinases was computed by comparing the residues at each of the 85 positions. Thus, the PocSeqSim for two binding site sequences equals the ratio of identical residues within the fixed length MSA of 85 positions. The score ranges from 0 to 1, where 0 indicates no identical residues and 1 indicates complete identity (Table S7).

### 3.3.3. Interaction Fingerprint Similarity (IFPSim)

All DFG-in and DFG-out structures for the nine human kinases under investigation, namely EGFR, ErbB2, PI3K, MET, CDK2, BRAF, p38 $\alpha$ , LCK and VEGFR2, were fetched from the KLIFS database with <https://github.com/volkamerlab/opencadd>, which uses the KLIFS Swagger API [17]. This query yielded 2091 structures (as of 27 July 2020). Only structures with orthosteric ligands were kept (1817 structures). For many kinases, several PDB structures are available and many structures contain more than one chain (and occasionally also alternative models), which are provided as separate entries in KLIFS. Whenever one structure was represented by more than one chain/alternative model entry, only the entry with the highest KLIFS quality score [16] was selected (if two had the same quality, the first one was kept arbitrarily). The quality score describes the alignment and structure quality ranging from 0 (bad) to 10 (flawless). This yielded a filtered set of 965 kinase structures (numbers per kinase in Table 5). For every structure, KLIFS provides information on the kinase–ligand interaction stored in an Interaction FingerPrint (IFP). The IFP encodes seven different interaction types (hydrophobic contact, aromatic face-to-face, aromatic edge-to-face, H-bond donor–acceptor, H-bond acceptor–donor, ionic positive–negative and ionic negative–positive) that can potentially be formed between each of the 85 pocket residues and the respective ligand in a bit string as either present (1) or absent (0) [15,16]. The Tanimoto similarity between every IFP pair of the 965 structures was calculated, resulting in multiple structure-pair comparisons for each kinase pair. Finally, a reduced matrix of size  $9 \times 9$  was produced in which for each kinase pair only the highest IFP similarity (IFPSim) score among all structure-pair scores was stored (Table S8).

### 3.3.4. Pocket Structure Similarity (PocStrucSim)

For the particular set of kinases investigated here, a set of 183 different PDB structures was compiled manually using the KLIFS dataset and a set of structures that has initially been considered for the docking screens. The manual selection was focused on choosing those kinase structures that featured similar binding sites to EGFR/ErbB2 and high structural quality (such as high resolution and few missing residues), also considering the correlation coefficient of the docking ranks. Furthermore, DFG-in and DFG-out structures were included to allow for diversity. After downloading the structures from the PDB, the files were processed with the API-RP package in the CSD Enterprise suite 2018 by CCDC, detecting all cavities using LigSite [50,51]. The predicted set of 909 cavities for 181 structures was further reduced by filtering for cavities containing at least one orthosteric ligand, resulting in 248 cavities from 176 different structures. It should be noted that some of these cavities emerged from different chains of the same structure and, therefore, contained the same ligand. Although the number of structures was decreased during this process, we made sure that at least two different structures for each kinase were still present in the final cavity set (Table 5). Furthermore, the set contained cavities for each of the structures used during the docking calculations, except for the structure with PDB ID 4JPS (PI3K), for which LigSite was not able to detect the correct cavity.

Each of the remaining cavities was then compared to all other cavities using the fast graph comparison method by CCDC [29]. In brief, the binding pocket is described by a graph model based on a set of pseudocenters with assigned surface patches containing information about the properties of the surrounding amino acids. In addition to the original CavBase implementation, the new method includes convexity and concavity measures in the pseudocenters as shape representation. Finally, two binding pockets were compared using a clique detection algorithm which was improved from the original CavBase algorithm [28,29]. Last, as for the IFPSim measure, the maximum similarity over all structure comparisons per kinase pair is reported.

### 3.3.5. Docking Rank Similarity (DockRankSim)

The docking rank similarity was calculated based on the notion that similar structures enrich similar ligands in the docking process. The similarity between two docking runs, each targeting a certain structure, was quantified by calculating the Spearman rank correlation of the common molecule set of the top-scoring molecules of both dockings. More precisely, to calculate the DockRankSim between two dockings, the top-ranked 25,000 molecules in both dockings were taken and the molecules common to both sets identified. For the calculation of the DockRankSim, only the dockings of the ZINC lead-like subset were considered. For this intersection, the ranks of the molecules were renumbered and the Spearman rank correlation was calculated. We restricted the calculation of the rank correlation to the top-scoring molecules, as we found this to lead to more discriminating DockRankSim values (data for full set not shown). A cutoff of 25,000 was identified to yield relevant results. However, it must be noted that this cutoff was not systematically optimized to yield the largest possible spread in DockRankSim values. The values calculated in this way describe how similar the compound ranking between two docking runs, i.e., two protein structures, is. To compare kinases instead of structures, we used the maximum observed DockRankSim of all pairwise structure comparisons between the respective two kinases.

## 4. Conclusions

In this study, we investigated parallel docking to disease-relevant kinase profiles, combining two targets and one anti-target. The choice of the initial profile was guided by biology: dual inhibitors of EGFR and ErbB2 are regarded as an advantageous treatment option for several carcinomas, whereas BRAF is a common undesired anti-target.

While being biologically meaningful, this profile is also a challenging test case of the precision of docking calculations, given the mutual similarity of the ATP binding site of the

three kinases. Nonetheless, we were able to identify one ligand with the desired profile, namely compound DS39984 against Profile 1, with  $IC_{50}$  values on the targets below 324 nM. This is very close to the expectation value assuming a hit rate of approximately 10–25% ( $0.25 \times 0.25 \times 0.90 = 0.056$ ) and a selection of 18 molecules from the docking calculations.

We then compared this with another profile combination, +EGFR+PI3K–BRAF (Profile 2), and at the same time investigated whether the likelihood for success (i.e., finding a ligand that fulfils the profile) can be predicted based on data derived from the protein structures. The profile +EGFR+PI3K–BRAF turned out to be hard to find a ligand for, and this was also reflected in the kinase similarity metrics (Figure 2). Finally, we tested a profile including EGFR and VEGFR2 as targets, due to the interest in them for cancer treatment, and tried again to design out binding to BRAF. As in the case of +EGFR+PI3K–BRAF, the higher similarity of VEGFR2 to BRAF (compared to EGFR) in most measures can be a hint why this docking did not yield the desired results. An alternative option, which would agree with the lack of positive results in the single docking performed for the target VEGFR2, is to select alternative starting structures, if available, or a different ligand database to further explore this profile.

Based on our findings and the further investigations into different similarity measures of kinases, several conclusions about the factors that determine the likelihood of successful predictions in multi-target settings can be drawn. First, for the present set of kinases, the various measures we calculated in this work largely agree with respect to which kinases are more similar to each other. This is important, because it means that, for a first estimate, one can go with a measure that can be computed in a fast and computationally inexpensive way and already get a largely correct view of the relationship of the targets involved. It also means that the ligand-centric and protein-centric views of ligand–protein interactions match to quite some degree.

Second, we only managed to pick few compounds from the docking runs, because few potential hits with plausible binding modes were identified in the top ranks of the combined scoring. Naturally, this means that the results for several of the profiles need to be interpreted with caution, as the numbers of data points are small. However, even if we had picked more compounds from lower ranks, the vast majority of them would likely have been inactive, as docking in general is able to prioritize ligands over nonbinders [52].

Third, the docking rank correlation of the top-ranked poses is very low (Figure 2E), which indicates that there exists only a limited number of substances in chemical databases for a given kinase profile. This lends additional support to docking strategies using (ultra-)large libraries of virtual compounds, as having access to larger and more diverse fractions of chemical space is certainly beneficial [52,53]. It has to be noted, however, that a certain amount of the rank correlation difference might also stem from the use of rigid protein structures in docking.

In conclusion, while docking to identify ligands gets progressively harder with more and more elaborate profiles composed of targets and anti-targets, one can try to estimate the chances of success already from protein-structure-, protein-sequence- and ligand-space-based methods. This is encouraging in the sense that protein and ligand space show a certain amount of congruence, i.e., that kinases that are close in structure or sequence space also recognize similar ligands, and supports the ongoing efforts to computationally expand chemical space to search for kinase inhibitors with tailored binding profiles.

**Supplementary Materials:** The following are available online. Figure S1: Docking poses of ligand DS39984 bound to BRAF structures, Figure S2: Comparison of different similarity measures for pairwise kinase structure comparisons, Table S1: IDs and 2D depictions of all compounds tested in the different kinase assays as well as the docking profile they were selected from, Table S2: Experimental % control values from the DiscoverX kinase assay, Table S3: Experimental results from the Eurofins assay, Table S4: LigProfSim: Pairwise similarity matrix, Table S5: LigProfSim counts: Number of ChEMBL compounds commonly tested in each kinase pair, Table S6: LigProfSim common actives: Number of ChEMBL compounds commonly active in each kinase pair, Table S7: PocSeqSim:

Pairwise similarity matrix, Table S8: IFPSim: Pairwise similarity matrix, Table S9: PocStrucSim: Pairwise similarity matrix.

**Author Contributions:** Conceptualization, A.V. and P.K.; Formal analysis, D.S. (Denis Schmidt), M.M.S., D.S. (Dominique Sydow), E.A., M.M.-S. and M.K.; Funding acquisition, A.V. and P.K.; Investigation, D.S. (Denis Schmidt), M.M.S., D.S. (Dominique Sydow), M.M.-S., M.K. and P.K.; Methodology, D.S. (Denis Schmidt); Project administration, A.V. and P.K.; Supervision, A.V. and P.K.; Visualization, D.S. (Denis Schmidt) and D.S. (Dominique Sydow); Writing—original draft, D.S. (Denis Schmidt), A.V., M.K. and P.K.; and Writing—review and editing, D.S. (Denis Schmidt), M.M.S., D.S. (Dominique Sydow), E.A., M.M.-S., A.V. and P.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by German Research Foundation (DFG) grant number KO4095/1-1 (to P.K.) and VO2353/1-1 (to A.V. and D.S. [Dominique Sydow]). The APC was funded by the German Research Foundation (DFG) and the Open Access Publication Fund of Charité—Universitätsmedizin Berlin.

**Data Availability Statement:** Raw data files of docking and similarity calculations are available from the corresponding authors upon reasonable request.

**Acknowledgments:** We thank Daniel Rauh for providing support to M.K. for the in vitro tests. We also thank Jaime Rodríguez-Guerra for his great support in assembling the kinase profiling data freely available at [https://github.com/openkinome/kinodata/releases/tag/\\_pub\\_ligprofsim](https://github.com/openkinome/kinodata/releases/tag/_pub_ligprofsim).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Santos, R.; Ursu, O.; Gaulton, A.; Bento, A.P.; Donadi, R.S.; Bologa, C.G.; Karlsson, A.; Al-Lazikani, B.; Hersey, A.; Oprea, T.I.; et al. A comprehensive map of molecular drug targets. *Nat. Rev. Drug Discov.* **2016**, *16*, 19–34. [[CrossRef](#)] [[PubMed](#)]
2. Roth, B.L.; Sheffler, D.J.; Kroeze, W.K. Magic shotguns versus magic bullets: Selectively non-selective drugs for mood disorders and schizophrenia. *Nat. Rev. Drug Discov.* **2004**, *3*, 353–359. [[CrossRef](#)] [[PubMed](#)]
3. Garuti, L.; Roberti, M.; Bottegoni, G. Multi-Kinase Inhibitors. *Curr. Med. Chem.* **2015**, *22*, 695–712. [[CrossRef](#)] [[PubMed](#)]
4. Gentile, C.; Martorana, A.; Lauria, A.; Bonsignore, R. Kinase Inhibitors in Multitargeted Cancer Therapy. *Curr. Med. Chem.* **2017**, *24*, 1671–1686. [[CrossRef](#)] [[PubMed](#)]
5. Moser, D.; Wisniewska, J.M.; Hahn, S.; Achenbach, J.; Buscató, E.; Klingler, F.M.; Hofmann, B.; Steinhilber, D.; Proschak, E. Dual-Target Virtual Screening by Pharmacophore Elucidation and Molecular Shape Filtering. *ACS Med. Chem. Lett.* **2012**, *3*, 155–158. [[CrossRef](#)]
6. Schmidt, D.; Bernat, V.; Brox, R.; Tschammer, N.; Kolb, P. Identifying Modulators of CXC Receptors 3 and 4 with Tailored Selectivity Using Multi-Target Docking. *ACS Chem. Biol.* **2015**, *10*, 715–724. [[CrossRef](#)]
7. Jaiteh, M.; Zeifman, A.; Saarinen, M.; Svenningsson, P.; Bréa, J.; Loza, M.I.; Carlsson, J. Docking Screens for Dual Inhibitors of Disparate Drug Targets for Parkinson's Disease. *J. Med. Chem.* **2018**, *61*, 5269–5278. [[CrossRef](#)]
8. Klebl, B.; Müller, G.; Hamacher, M.; Mannhold, R.; Kubinyi, H.; Folkers, G. *Protein Kinases as Drug Targets*, 49th ed.; Methods and Principles in Medicinal Chemistry; Wiley-VCH Verlag GmbH & Co. KGaA: Weinheim, Germany, 2011. [[CrossRef](#)]
9. Lin, J.; Gan, C.M.; Zhang, X.; Jones, S.; Sjöblom, T.; Wood, L.D.; Parsons, D.W.; Papadopoulos, N.; Kinzler, K.W.; Vogelstein, B.; et al. A multidimensional analysis of genes mutated in breast and colorectal cancers. *Genome Res.* **2007**, *17*, 1304–1318. [[CrossRef](#)]
10. Wood, L.D.; Parsons, D.W.; Jones, S.; Lin, J.; Sjöblom, T.; Leary, R.J.; Shen, D.; Boca, S.M.; Barber, T.; Ptak, J.; et al. The genomic landscapes of human breast and colorectal cancers. *Science* **2007**, *318*, 1108–1113. [[CrossRef](#)]
11. Manning, G.; Whyte, D.; Martinez, R.; Hunter, T.; Sudarsanam, S. The protein kinase complement of the human genome. *Science* **2002**, *298*, 1912–1934. [[CrossRef](#)]
12. Kooistra, A.J.; Volkamer, A. Kinase-Centric Computational Drug Development. In *Annual Reports in Medicinal Chemistry*; Academic Press: Cambridge, MA, USA, 2017; Volume 50, pp. 263–299. [[CrossRef](#)]
13. Sorgenfrei, F.A.; Fulle, S.; Merget, B. Kinome-Wide Profiling Prediction of Small Molecules. *ChemMedChem* **2018**, *13*, 495–499. [[CrossRef](#)] [[PubMed](#)]
14. Berman, H.M. The Protein Data Bank. *Nucleic Acids Res.* **2000**, *28*, 235–242. [[CrossRef](#)] [[PubMed](#)]
15. van Linden, O.P.J.; Kooistra, A.J.; Leurs, R.; de Esch, I.J.P.; de Graaf, C. KLIFS: A knowledge-based structural database to navigate kinase-ligand interaction space. *J. Med. Chem.* **2014**, *57*, 249–277. [[CrossRef](#)] [[PubMed](#)]
16. Kooistra, A.J.; Kanev, G.K.; van Linden, O.P.; Leurs, R.; de Esch, I.J.; de Graaf, C. KLIFS: A structural kinase-ligand interaction database. *Nucleic Acids Res.* **2016**, *44*, D365–D371. [[CrossRef](#)]
17. Kanev, G.K.; de Graaf, C.; Westerman, B.A.; de Esch, I.J.P.; Kooistra, A.J. KLIFS: An overhaul after the first 5 years of supporting kinase research. *Nucleic Acids Res.* **2020**, *49*, D562–D569. [[CrossRef](#)] [[PubMed](#)]

18. Roskoski, R. FDA Approved Kinase Inhibitors ('-nibs'). Available online: <http://www.brimr.org/PKI/PKIs.htm> (accessed on 3 July 2020).
19. Ung, P.M.U.; Rahman, R.; Schlessinger, A. Redefining the Protein Kinase Conformational Space with Machine Learning. *Cell Chem. Biol.* **2018**, *25*, 916–924.e2. [[CrossRef](#)]
20. Modi, V.; Dunbrack, R.L. Defining a new nomenclature for the structures of active and inactive kinases. *Proc. Natl. Acad. Sci. USA* **2019**, *116*, 6818–6827. [[CrossRef](#)]
21. Karaman, M.W.; Herrgard, S.; Treiber, D.K.; Gallant, P.; Atteridge, C.E.; Campbell, B.T.; Chan, K.W.; Ciceri, P.; Davis, M.I.; Edeen, P.T.; et al. A quantitative analysis of kinase inhibitor selectivity. *Nat. Biotechnol.* **2008**, *26*, 127–132. [[CrossRef](#)]
22. Holbro, T.; Hynes, N.E. ErbB receptors: Directing Key Signaling Networks Throughout Life. *Annu. Rev. Pharmac. Toxic.* **2004**, *44*, 195–217. [[CrossRef](#)]
23. Seshadri, R.; Firgaira, F.A.; Horsfall, D.J.; McCaul, K.; Setlur, V.; Kitchen, P. Clinical significance of HER-2/neu oncogene amplification in primary breast cancer. The South Australian Breast Cancer Study Group. *J. Clin. Oncol.* **1993**, *11*, 1936–1942. [[CrossRef](#)]
24. Klein, S.; Levitzki, A. Targeting the EGFR and the PKB pathway in cancer. *Curr. Op. Cell Biol.* **2009**, *21*, 185–193. [[CrossRef](#)] [[PubMed](#)]
25. Jung, Y.D.; Mansfield, P.F.; Akagi, M.; Takeda, A.; Liu, W.; Bucana, C.D.; Hicklin, D.J.; Ellis, L.M. Effects of combination anti-vascular endothelial growth factor receptor and anti-epidermal growth factor receptor therapies on the growth of gastric cancer in a nude mouse model. *Eur. J. Cancer* **2002**, *38*, 1133–1140. doi:10.1016/S0959-8049(02)00013-8. [[CrossRef](#)]
26. McTigue, M.; Murray, B.W.; Chen, J.H.; Deng, Y.L.; Solowiej, J.; Kania, R.S. Molecular conformations, interactions, and properties associated with drug efficiency and clinical performance among VEGFR TK inhibitors. *Proc. Natl. Acad. Sci. USA* **2012**, *109*, 18281–18289. [[CrossRef](#)] [[PubMed](#)]
27. Davis, I.M.; Hunt, P.J.; Herrgard, S.; Ciceri, P.; Wodicka, M.L.; Pallares, G.; Hocker, M.; Treiber, K.D.; Zarrinkar, P.P.; Treiber, K.D. Comprehensive analysis of kinase inhibitor selectivity. *Nat. Biotechnol.* **2011**, *29*, 1046–1051. [[CrossRef](#)] [[PubMed](#)]
28. Schmitt, S.; Kuhn, D.; Klebe, G. A New Method to Detect Related Function among Proteins Independent of Sequence and Fold Homology. *J. Mol. Biol.* **2002**, *323*, 387–406. [[CrossRef](#)]
29. Krotzky, T.; Fober, T.; Hüllermeier, E.; Klebe, G. Extended Graph-Based Models for Enhanced Similarity Search in Cavbase. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2014**, *11*, 878–890. [[CrossRef](#)]
30. Wood, E.R.; Truesdale, A.T.; McDonald, O.B.; Yuan, D.; Hassell, A.; Dickerson, S.H.; Ellis, B.; Pennisi, C.; Horne, E.; Lackey, K.; et al. A Unique Structure for Epidermal Growth Factor Receptor Bound to GW572016 (Lapatinib). *Cancer Res.* **2004**, *64*, 6652–6659. [[CrossRef](#)]
31. Aertgeerts, K.; Skene, R.; Yano, J.; Sang, B.C.; Zou, H.; Snell, G.; Jennings, A.; Iwamoto, K.; Habuka, N.; Hirokawa, A.; et al. Structural Analysis of the Mechanism of Inhibition and Allosteric Activation of the Kinase Domain of HER2 Protein. *J. Biol. Chem.* **2011**, *286*, 18756–18765. [[CrossRef](#)] [[PubMed](#)]
32. Ishikawa, T.; Seto, M.; Banno, H.; Kawakita, Y.; Oorui, M.; Taniguchi, T.; Ohta, Y.; Tamura, T.; Nakayama, A.; Miki, H.; et al. Design and Synthesis of Novel Human Epidermal Growth Factor Receptor 2 (HER2)/Epidermal Growth Factor Receptor (EGFR) Dual Inhibitors Bearing a Pyrrolo[3,2-d]pyrimidine Scaffold. *J. Med. Chem.* **2011**, *54*, 8030–8050. [[CrossRef](#)]
33. Wan, P.T.C.; Garnett, M.J.; Roe, S.M.; Lee, S.; Niculescu-Duvaz, D.; Good, V.M.; Project, C.G.; Jones, C.M.; Marshall, C.J.; Springer, C.J.; et al. Mechanism of activation of the RAF-ERK signaling pathway by oncogenic mutations of B-RAF. *Cell* **2004**, *116*, 855–867. [[CrossRef](#)]
34. Ren, L.; Wenglow, S.; Miknis, G.; Rast, B.; Buckmelter, A.J.; Ely, R.J.; Schlachter, S.; Laird, E.R.; Randolph, N.; Callejo, M.; et al. Non-oxime inhibitors of B-RafV600E kinase. *Bioorg. Med. Chem. Lett.* **2011**, *21*, 1243–1247. [[CrossRef](#)] [[PubMed](#)]
35. Furet, P.; Guagnano, V.; Fairhurst, R.A.; Imbach-Weese, P.; Bruce, I.; Knapp, M.; Fritsch, C.; Blasco, F.; Blanz, J.; Aichholz, R.; et al. Discovery of NVP-BYL719 a potent and selective phosphatidylinositol-3 kinase alpha inhibitor selected for clinical evaluation. *Bioorg. Med. Chem. Lett.* **2013**, *23*, 3741–3748. [[CrossRef](#)] [[PubMed](#)]
36. Okamoto, K.; Ikemori-Kawada, M.; Jestel, A.; von König, K.; Funahashi, Y.; Matsushima, T.; Tsuruoka, A.; Inoue, A.; Matsui, J. Distinct Binding Mode of Multikinase Inhibitor Lenvatinib Revealed by Biochemical Characterization. *ACS Med. Chem. Lett.* **2015**, *6*, 89–94. [[CrossRef](#)] [[PubMed](#)]
37. Kolb, P.; Rosenbaum, D.M.; Irwin, J.J.; Fung, J.J.; Kobilka, B.K.; Shoichet, B.K. Structure-based discovery of  $\beta_2$ -adrenergic receptor ligands. *Proc. Natl. Acad. Sci. USA* **2009**, *106*, 6843–6848. [[CrossRef](#)]
38. Irwin, J.J.; Sterling, T.; Mysinger, M.M.; Bolstad, E.S.; Coleman, R.G. ZINC: A Free Tool to Discover Chemistry for Biology. *J. Chem. Inf. Model.* **2012**, *52*, 1757–1768. [[CrossRef](#)]
39. Kuntz, I.D.; Meng, E.C.; Oatley, S.J.; Langridge, R.; Ferrin, T.E. A geometric approach to macromolecule-ligand interactions. *J. Mol. Biol.* **1982**, *161*, 269–288. [[CrossRef](#)]
40. Meng, E.C.; Shoichet, B.K.; Kuntz, I.D. Automated docking with grid-based energy evaluation. *J. Comp. Chem.* **1992**, *13*, 505–524. [[CrossRef](#)]
41. Shoichet, B.K.; Kuntz, I.D. Matching chemistry and shape in molecular docking. *Protein Eng. Des. Sel.* **1993**, *6*, 723–732. [[CrossRef](#)]
42. Shoichet, B.K.; Leach, A.R.; Kuntz, I.D. Ligand solvation in molecular docking. *Proteins* **1999**, *34*, 4–16. [[CrossRef](#)]
43. Mysinger, M.M.; Shoichet, B.K. Rapid context-dependent ligand desolvation in molecular docking. *J. Chem. Inf. Model.* **2010**, *50*, 1561–1573. [[CrossRef](#)]

44. Mysinger, M.M.; Carchia, M.; Irwin, J.J.; Shoichet, B.K. Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking. *J. Med. Chem.* **2012**, *55*, 6582–6594. [[CrossRef](#)] [[PubMed](#)]
45. Weber, W.; Bertics, P.J.; Gill, G.N. Immunoaffinity purification of the epidermal growth factor receptor. Stoichiometry of binding and kinetics of self-phosphorylation. *J. Biol. Chem.* **1984**, *259*, 14631–14636. [[CrossRef](#)]
46. Quian, X.L.; Decker, S.J.; Greene, M.I. p185c-neu and epidermal growth factor receptor associate into a structure composed of activated kinases. *Proc. Natl. Acad. Sci. USA* **1992**, *89*, 1330–1334. [[CrossRef](#)] [[PubMed](#)]
47. Sinnamon, R.H.; McDevitt, P.; Pietrak, B.L.; Leydon, V.R.; Xue, Y.; Lehr, R.; Qi, H.; Burns, M.; Elkins, P.; Ward, P.; et al. Baculovirus production of fully-active phosphoinositide 3-kinase alpha as a p85 $\alpha$ -p110 $\alpha$  fusion for X-ray crystallographic analysis with ATP competitive enzyme inhibitors. *Protein Expr. Purif.* **2010**, *73*, 167–176. [[CrossRef](#)]
48. Kupcho, K.R.; Bruinsma, R.; Hallis, T.M.; Lasky, D.A.; Somberg, R.L.; Turek-Etienne, T.; Vogel, K.W.; Huwiler, K.G. Fluorescent Cascade and Direct Assays for Characterization of RAF Signaling Pathway Inhibitors. *Curr. Chem. Genom.* **2008**, *1*, 43–53. [[CrossRef](#)]
49. Mendez, D.; Gaulton, A.; Bento, A.P.; Chambers, J.; De Veij, M.; Félix, E.; Magarinos, M.P.; Mosquera, J.F.; Mutowo, P.; Nowotka, M.; et al. ChEMBL: Towards direct deposition of bioassay data. *Nucleic Acids Res.* **2018**, *47*, D930–D940. [[CrossRef](#)] [[PubMed](#)]
50. Groom, C.R.; Bruno, I.J.; Lightfoot, M.P.; Ward, S.C. The Cambridge Structural Database. *Acta Crystallogr. B Struct. Sci. Cryst. Eng. Mater.* **2016**, *72*, 171–179. [[CrossRef](#)]
51. Hendlich, M.; Rippmann, F.; Barnickel, G. LIGSITE: Automatic and efficient detection of potential small molecule-binding sites in proteins. *J. Mol. Graph. Model.* **1997**, *15*, 359–363. [[CrossRef](#)]
52. Lyu, J.; Wang, S.; Balias, T.E.; Singh, L.; Levit, A.; Moroz, Y.S.; O'Meara, M.J.; Che, T.; Alga, E.; Tolmachova, K.; et al. Ultra-large library docking for discovering new chemotypes. *Nature* **2019**, *566*, 224–229. [[CrossRef](#)]
53. Chevillard, F.; Stotani, S.; Karawajczyk, A.; Hristeva, S.; Pardon, E.; Steyaert, J.; Tzalis, D.; Kolb, P. Interrogating dense ligand chemical space with a forward-synthetic library. *Proc. Natl. Acad. Sci. USA* **2019**, *116*, 11496–11501. [[CrossRef](#)]

## Supplementary Materials: Analyzing kinase similarity in small molecule and protein structural space to explore the limits of multi-target screening

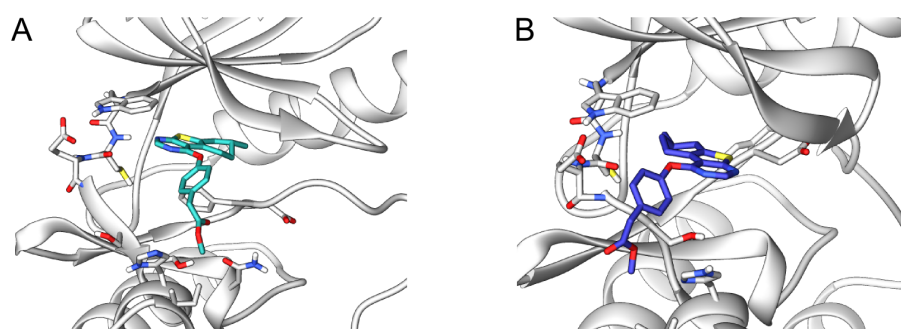
Denis Schmidt<sup>1</sup>, Magdalena M. Scharf<sup>1</sup>, Dominique Sydow<sup>1</sup>, Eva Aßmann, Maria Martí-Solano, Marina Keul, Andrea Volkamer<sup>1</sup> and Peter Kolb<sup>1</sup>

### 1. Supplementary Methods

#### 1.1. DiscoverX kinase assay

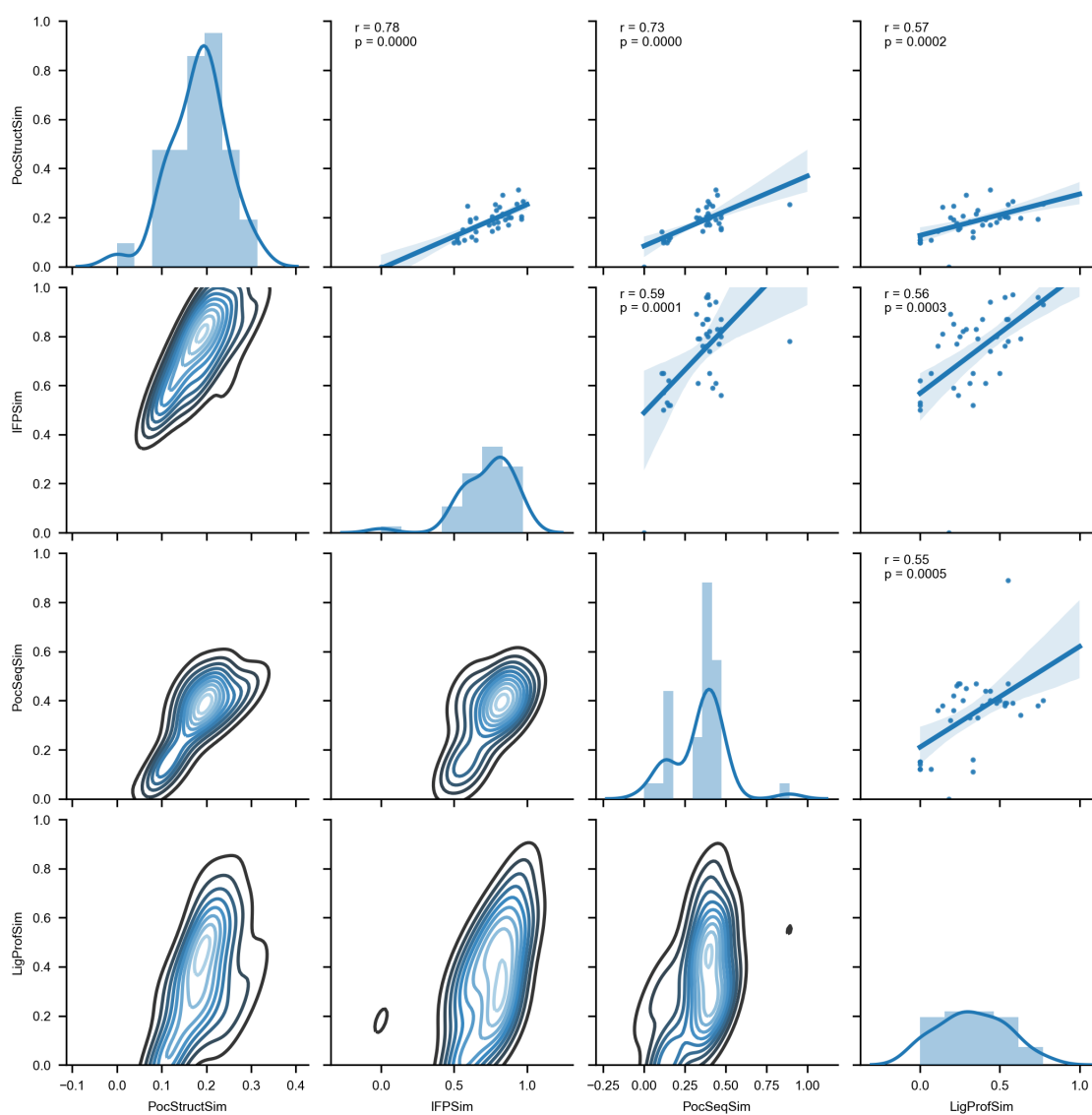
For most assays, kinase-tagged T7 phage strains were grown in parallel in 24-well blocks in an *E.coli* host derived from the BL21 strain. *E.coli* were grown to log-phase and infected with T7 phage from a frozen stock (multiplicity of infection = 0.4) and incubated with shaking at 32°C until lysis (90-150 minutes). The lysates were centrifuged (6,000 × g) and filtered (0.2 μm) to remove cell debris. The remaining kinases were produced in HEK-293 cells and subsequently tagged with DNA for qPCR detection. Streptavidin-coated magnetic beads were treated with biotinylated small molecule ligands for 30 minutes at room temperature to generate affinity resins for the kinases. The liganded beads were blocked with excess biotin and washed with blocking buffer (SeaBlock [Pierce], 1% BSA, 0.05% Tween 20, 1 mM DTT) to remove unbound ligands and to reduce non-specific phage binding. Binding reactions were assembled by combining kinases, liganded affinity beads, and test compounds in 1x binding buffer (20% SeaBlock, 0.17x PBS, 0.05% Tween 20, 6 mM DTT). Test compounds were prepared as 40x stocks in 100% DMSO and directly diluted into the assay. All reactions were performed in polypropylene 384-well plates in a final volume of 0.02 ml. The assay plates were incubated at room temperature with shaking for 1 hour and the affinity beads were washed with wash buffer (1x PBS, 0.05% Tween 20). The beads were then re-suspended in elution buffer (1x PBS, 0.05% Tween 20, 0.5 μM non-biotinylated affinity ligand) and incubated at room temperature with shaking for 30 minutes. The kinase concentration in the eluates was measured by qPCR.

### 2. Supplementary Figures



**Figure S1.** Structure of ligand DS39984 bound to the BRAF structures PDB 1UWH, DFG-out (A) and PDB 3PPK, DFG-in (B). The protein structure is shown as cartoon, colored in grey. The compound and interacting binding site residues are represented as sticks.



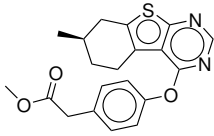
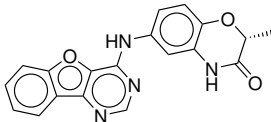
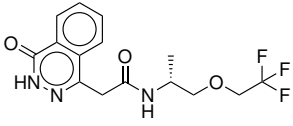
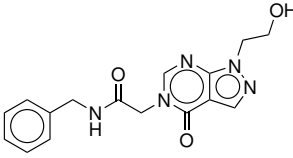
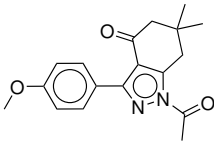
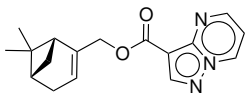
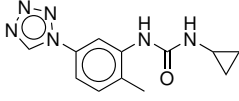
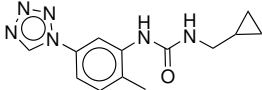


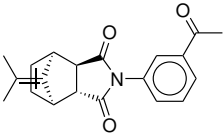
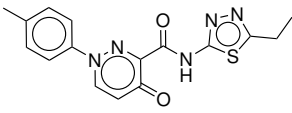
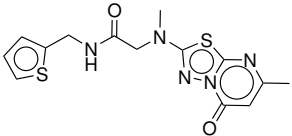
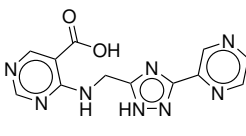
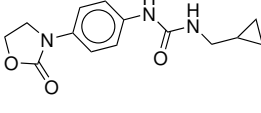
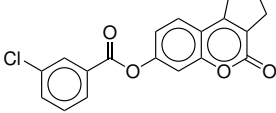
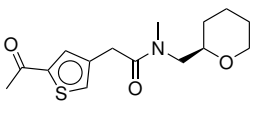
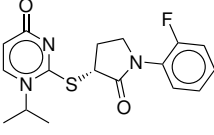
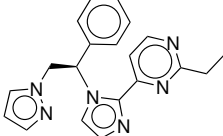
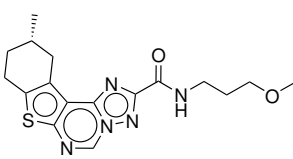
**Figure S2.** Comparison of different similarity measures for pairwise kinase structure comparisons. Diagonal: Distributions of structure similarities for the herein described similarity measures. Lower triangular matrix: Bivariate distributions of similarities per pairs of similarity measures, shown as isocontours with light blue indicating high densities and dark blue indicating low densities. Upper triangular matrix: Scatter plots of similarities per pairs of similarity measures with fitted regression lines (dark lines) and 95% CI intervals of regression (light blue shades)

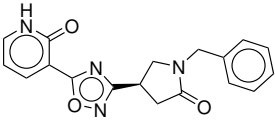
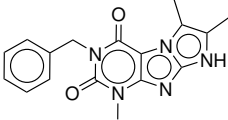
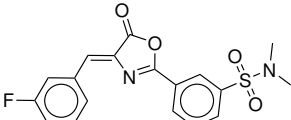
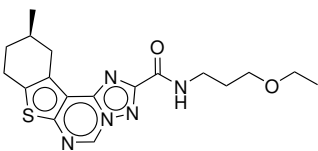
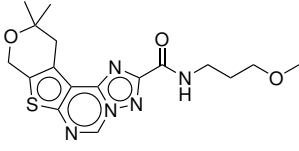
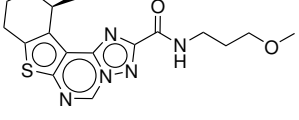
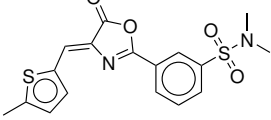
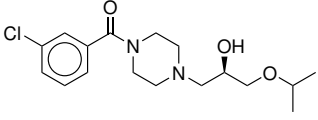
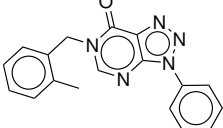
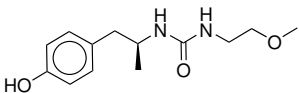
## 3. Supplementary Tables

## 3.1. Compound lists and experimental results

**Table S1.** IDs and 2D depictions of all compounds tested in the different kinase assays as well as the docking profile they were selected from. Three compounds were selected independently from two different profiles and are marked accordingly.

Mol ID	ZINC ID	2D structure	Profile
<b>Actives</b>			
DS39984	C03283998		+EGFR+ErbB2-BRAF
K001MM011	C32808493		+VEGFR2
<b>Inactives</b>			
DS04644	C84640464		+EGFR+ErbB2-BRAF
DS05168	C04940516		+EGFR+ErbB2-BRAF
DS18339	C71281833		+EGFR+ErbB2-BRAF
DS34376	C95373437		+EGFR+ErbB2-BRAF
DS44245	C47934424		+EGFR+ErbB2-BRAF
DS44738 <sup>b</sup>	C48954473		+EGFR+ErbB2-BRAF

DS57124	C17055712		+EGFR+ErbB2-BRAF
DS59212	C09205921		+EGFR+ErbB2-BRAF
DS72975	C06807297		+EGFR+ErbB2-BRAF
DS74417	C95387441		+EGFR+ErbB2-BRAF
DS75739	C48697573		+EGFR+ErbB2-BRAF
DS76514	C00137651		+EGFR+ErbB2-BRAF
DS84326	C20858432		+EGFR+ErbB2-BRAF
DS99367	C71899936		+EGFR+ErbB2-BRAF
DS23815	C71422381		+EGFR+PI3K-BRAF
DS31939	C02343193		+EGFR+PI3K-BRAF

DS52225	C44425222		+EGFR+PI3K-BRAF
DS62156	C08706215		+EGFR+PI3K-BRAF
DS74631	C07397463		+EGFR+PI3K-BRAF
DS82066	C02228206		+EGFR+PI3K-BRAF
DS11689 <sup>a</sup>	C02341168		+EGFR+PI3K-BRAF/ +EGFR+ErbB2-BRAF
DS66846 <sup>a</sup>	C02226684		+EGFR+PI3K-BRAF/ +EGFR+ErbB2-BRAF
DS74871 <sup>a</sup>	C07397487		+EGFR+PI3K-BRAF/ +EGFR+ErbB2-BRAF
K001MM002	C97100024		+EGFR+VEGFR2-BRAF
K001MM003	C04266692		+EGFR+VEGFR2-BRAF
K001MM004	C48922370		+EGFR+VEGFR2-BRAF

K001MM005	C76064467		+EGFR+VEGFR2-BRAF
K001MM006	C96153842		+EGFR+VEGFR2-BRAF
K001MM007	C97142813		+EGFR+VEGFR2-BRAF
K001MM008	C40067740		+EGFR+VEGFR2-BRAF
K001MM009 <sup>b</sup>	C48954473		+EGFR+VEGFR2-BRAF
K001MM010	C96160364		+VEGFR2
K001MM012	C03453350		+VEGFR2
K001MM013	C23551796		+VEGFR2

<sup>a</sup> Compounds were selected independently from docking campaigns against two profiles.

<sup>b</sup> Compound was selected independently from two docking profiles and tested twice in separate test rounds during experimental validation.

**Table S2.** Experimental % control values from the DiscoverX kinase assay. Compounds were tested against the nine kinases EGFR, ErbB2, LCK, CDK2, BRAF, MET, p38 $\alpha$ , PI3K and VEGFR2, unless otherwise stated. Binding of kinase and compound were tested at a compound concentration of 10  $\mu$ M and in comparison to a control compound. Lower values indicate a higher affinity of the compound to the protein and values below 35% indicate significant binding according to information of the CRO.

Mol ID	ZINC ID	BRAF	EGFR	ErbB2	VEGFR2	CDK2	LCK	MET	p38 $\alpha$	PI3K
Actives										
DS39984	C03283998	99	17	21	99	100	89	100	100	100
K001MM011 <sup>a</sup>	C32808493	100	1.4	53	99	n.t.	n.t.	n.t.	n.t.	n.t.
Inactives										
DS04644	C84640464	97	100	100	100	99	96	100	100	100
DS05168	C04940516	98	100	100	100	99	95	94	100	100
DS18339	C71281833	100	97	100	97	100	89	100	99	100
DS34376	C95373437	92	90	100	100	99	100	100	100	100
DS44245	C47934424	99	100	100	98	100	100	90	100	100
DS44738 <sup>b</sup>	C48954473	100	100	91	100	100	100	84	100	94
DS57124	C17055712	100	100	91	100	100	100	94	100	100
DS59212	C09205921	100	100	100	100	100	100	97	99	87
DS72975	C06807297	100	96	87	99	100	100	78	100	100
DS74417	C95387441	86	89	100	100	100	100	100	100	97
DS75739	C48697573	100	100	94	100	100	100	85	87	100
DS76514	C00137651	89	87	96	100	100	100	90	97	100
DS84326	C20858432	100	100	97	100	100	100	95	90	100
DS99367	C71899936	100	93	100	100	100	100	87	91	100
DS23815	C71422381	99	100	100	100	99	100	99	100	97
DS31939	C02343193	100	100	100	100	100	99	100	100	73
DS52225	C44425222	100	100	94	98	100	100	91	100	100
DS62156	C08706215	95	90	96	100	100	100	87	97	66
DS74631	C07397463	100	97	95	100	100	87	100	83	100
DS82066	C02228206	83	92	90	100	100	100	100	93	89
DS11689	C02341168	88	100	99	100	100	100	99	100	99
DS66846	C02226684	100	100	96	100	100	100	92	100	100
DS74871	C07397487	100	100	100	100	100	90	95	81	100
K001MM002 <sup>a</sup>	C97100024	89	97	92	100	n.t.	n.t.	n.t.	n.t.	n.t.
K001MM003 <sup>a</sup>	C04266692	95	99	99	100	n.t.	n.t.	n.t.	n.t.	n.t.
K001MM004 <sup>a</sup>	C48922370	100	100	100	100	n.t.	n.t.	n.t.	n.t.	n.t.
K001MM005 <sup>a</sup>	C76064467	99	100	100	100	n.t.	n.t.	n.t.	n.t.	n.t.
K001MM006 <sup>a</sup>	C96153842	100	96	100	100	n.t.	n.t.	n.t.	n.t.	n.t.
K001MM007 <sup>a</sup>	C97142813	87	90	94	100	n.t.	n.t.	n.t.	n.t.	n.t.
K001MM008 <sup>a</sup>	C40067740	100	100	100	100	n.t.	n.t.	n.t.	n.t.	n.t.
K001MM009 <sup>a,b</sup>	C48954473	96	100	98	98	n.t.	n.t.	n.t.	n.t.	n.t.
K001MM010 <sup>a</sup>	C96160364	100	95	90	97	n.t.	n.t.	n.t.	n.t.	n.t.
K001MM012 <sup>a</sup>	C03453350	100	90	100	91	n.t.	n.t.	n.t.	n.t.	n.t.
K001MM013 <sup>a</sup>	C23551796	100	100	94	90	n.t.	n.t.	n.t.	n.t.	n.t.

<sup>a</sup> Compounds were only tested against four kinases (EGFR, ErbB2, BRAF and VEGFR2).

<sup>b</sup> Compound was selected independently from two docking profiles and tested separately during experimental validation.

n.t.: not tested.

**Table S3.** Experimental results from the Eurofins assay. Inhibition of four kinases (EGFR, PI3K, ErbB2, BRAF) was measured at compound concentrations of 20  $\mu$ M. Inhibition was calculated as % inhibition of control activity. According to CRO, values above 50% inhibition represent significant inhibition, values above 25% weak inhibition effect and values below 25% as well as negative values are usually not significant. Results are reported in *mean (SD)* format.

Mol ID	ZINC ID	EGFR	PI3K	ErbB2	BRAF
Actives					
DS39984	C03283998	58.9 (3.2)	-6.7 (2.3)	-1.3 (1.8)	-0.1 (0.1)
Inactives					
DS04644	C84640464	3.4 (1.8)	-2.8 (0.6)	-7.4 (1.0)	-14.7 (7.4)
DS05168	C04940516	11.1 (1.8)	-1 (1.6)	2.8 (1.6)	-12.4 (10.9)
DS18339	C71281833	15.3 (7.7)	-3.7 (0.3)	0.7 (1.1)	-21.8 (9.5)
DS34376	C95373437	16.1 (7.5)	-2 (0.8)	-1.6 (1.9)	-11.3 (4.2)
DS44245	C47934424	2.4 (14.7)	1.8 (4.4)	-3.4 (1.0)	-13.7 (10.5)
DS44738	C48954473	4.4 (14.0)	0.4 (1.7)	10.5 (23.6)	-20.8 (0.4)
DS57124	C17055712	23.9 (7.5)	7.8 (5.0)	-3 (1.8)	-20.2 (19.8)
DS59212	C09205921	7.8 (3.3)	0.2 (3.6)	5.8 (0.2)	-22.9 (4.9)
DS72975	C06807297	19.8 (5.0)	-1.1 (2.1)	9.8 (21.9)	-12.4 (1.8)
DS74417	C95387441	18 (3.2)	2.1 (0.4)	-2.8 (0.7)	-0.3 (1.6)
DS75739	C48697573	4.3 (6.9)	1.7 (0.8)	-0.6 (1.1)	-11.1 (4.9)
DS76514	C00137651	3.2 (10.7)	17.4 (0.4)	-0.6 (1.0)	-33.6 (18.1)
DS84326	C20858432	23.4 (0.6)	0.7 (0.8)	0.2 (2.8)	-8.2 (13.6)
DS99367	C71899936	17.3 (0.1)	1.4 (0.1)	-3.3 (0.5)	-19.6 (11.8)
DS23815	C71422381	10.7 (0.4)	-4.7 (3.1) <sup>a</sup>	-7.2 (5.7)	-19.7 (2.4)
DS31939	C02343193	9.6 (0.2)	-9.8 (7.8) <sup>a</sup>	-1.7 (0.8)	-27.8 (10.8)
DS52225	C44425222	22.7 (15.5)	0.9 (3.6)	4.5 (4.9)	-17.7 (15.6)
DS62156	C08706215	16.6 (0.4)	1.5 (5.8)	3.1 (3.3)	-6.3 (1.6)
DS74631	C07397463	11.3 (4.5)	20 (5.2)	-4.6 (2.9)	-22.8 (15.9)
DS82066	C02228206	12.6 (1.9)	3.6 (2.3)	-1.4 (0.8)	-18.5 (1.8)
DS11689	C02341168	18.1 (7.9)	-0.2 (0.4)	0.1 (2.1)	-8.3 (20.8)
DS66846	C02226684	14.5 (8.6)	4.6 (7.7)	-2.5 (1.6)	-18.7 (11.9)
DS74871	C07397487	23.3 (4.2)	22.1 (2.3)	-8.3 (1.2)	-44.9 (24.1)

<sup>a</sup> Compound interfered with assay readout.

## 3.2. Raw data for LigProfSim, PocSeqSim, IFPSim, and PocStructSim

## 3.2.1. LigProfSim

**Table S4.** LigProfSim matrix: Similarity values per kinase pair

kinase	EGFR	ErbB2	BRAF	CDK2	LCK	MET	p38a	KDR	p110a
EGFR	0.59	0.55	0.53	0.19	0.29	0.23	0.48	0.35	0.07
ErbB2	0.55	0.62	0.50	0.31	0.21	0.24	0.44	0.41	0.00
BRAF	0.53	0.50	0.82	0.36	0.58	0.39	0.74	0.77	0.33
CDK2	0.19	0.31	0.36	0.55	0.14	0.21	0.25	0.63	0.33
LCK	0.29	0.21	0.58	0.14	0.63	0.27	0.54	0.44	0.00
MET	0.23	0.24	0.39	0.21	0.27	0.79	0.11	0.55	0.00
p38a	0.48	0.44	0.74	0.25	0.54	0.11	0.77	0.53	0.00
KDR	0.35	0.41	0.77	0.63	0.44	0.55	0.53	0.70	0.18
p110a	0.07	0.00	0.33	0.33	0.00	0.00	0.00	0.18	0.65

**Table S5.** LigProfSim counts: Number of ChEMBL compounds commonly tested in each kinase pair

kinase	EGFR	ErbB2	BRAF	CDK2	LCK	MET	p38a	KDR	p110a
EGFR	5702	1199	70	47	129	82	46	875	180
ErbB2	1199	1690	22	29	28	29	9	189	1
BRAF	70	22	3625	14	38	31	42	268	3
CDK2	47	29	14	1520	22	24	8	122	12
LCK	129	28	38	22	1552	66	136	419	5
MET	82	29	31	24	66	2851	18	348	2
p38a	46	9	42	8	136	18	3581	125	5
KDR	875	189	268	122	419	348	125	7426	175
p110a	180	1	3	12	5	2	5	175	4150

**Table S6.** LigProfSim common actives: Number of ChEMBL compounds commonly active in each kinase pair

kinase	EGFR	ErbB2	BRAF	CDK2	LCK	MET	p38a	KDR	p110a
EGFR	3382	658	37	9	38	19	22	303	13
ErbB2	658	1048	11	9	6	7	4	77	0
BRAF	37	11	2968	5	22	12	31	207	1
CDK2	9	9	5	837	3	5	2	77	4
LCK	38	6	22	3	976	18	73	183	0
MET	19	7	12	5	18	2248	2	193	0
p38a	22	4	31	2	73	2	2753	66	0
KDR	303	77	207	77	183	193	66	5197	32
p110a	13	0	1	4	0	0	0	32	2706



## 3.2.2. PocSeqSim

**Table S7.** Kinase sequence identity of binding site residues (MSA of 85 binding site residues from KLIFS)

kinase	EGFR	ErbB2	BRAF	CDK2	LCK	MET	p38	PI3K	VEGFR2
EGFR	1	0.89	0.38	0.32	0.45	0.46	0.39	0.12	0.47
ErbB2	0.89	1	0.4	0.33	0.42	0.47	0.4	0.12	0.44
BRAF	0.38	0.4	1	0.33	0.39	0.38	0.38	0.16	0.4
CDK2	0.32	0.33	0.33	1	0.38	0.36	0.47	0.11	0.34
LCK	0.45	0.42	0.39	0.38	1	0.4	0.39	0.15	0.44
MET	0.46	0.47	0.38	0.36	0.4	1	0.36	0.12	0.47
p38	0.39	0.4	0.38	0.47	0.39	0.36	1	0.14	0.39
PI3K	0.12	0.12	0.16	0.11	0.15	0.12	0.14	1	0.15
VEGFR2	0.47	0.44	0.4	0.34	0.44	0.47	0.39	0.15	1

## 3.2.3. IFPSim

**Table S8.** IFPSim matrix: Similarity values per kinase pair

kinase1	EGFR	ErbB2	BRAF	CDK2	LCK	MET	p38a	PI3K	VEGFR2
EGFR	1.0	0.78	0.76	0.89	0.83	0.77	0.8	0.65	0.83
ErbB2	0.78	0.71	0.65	0.61	0.59	0.56	0.74	0.5	0.61
BRAF	0.76	0.65	0.96	0.79	0.97	0.87	0.96	0.52	0.93
CDK2	0.89	0.61	0.79	1.0	0.81	0.85	0.8	0.65	0.79
LCK	0.83	0.59	0.97	0.81	0.91	0.82	0.87	0.62	0.94
MET	0.77	0.56	0.87	0.85	0.82	1.0	0.76	0.57	0.87
p38a	0.8	0.74	0.96	0.8	0.87	0.76	1.0	0.53	0.96
PI3K	0.65	0.5	0.52	0.65	0.62	0.57	0.53	0.91	0.52
VEGFR2	0.83	0.61	0.93	0.79	0.94	0.87	0.96	0.52	1.0

## 3.2.4. PocStructSim


**Table S9.** PocStructSim matrix: Similarity values per kinase pair

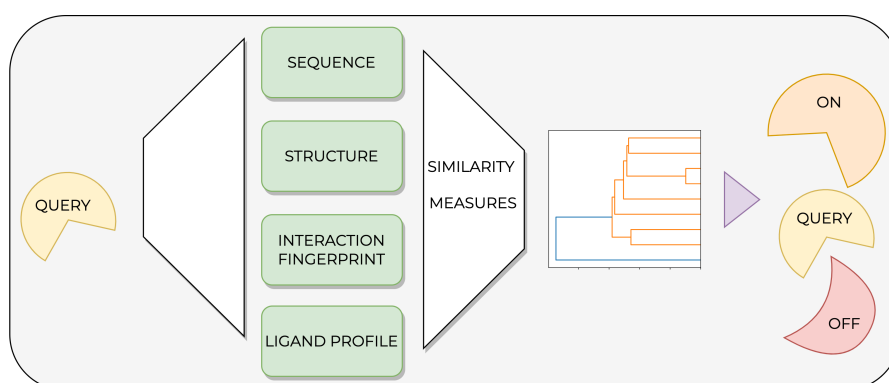
	EGFR	ErbB2	VEGFR2	PI3K	BRAF	CDK2	LCK	MET	p38a
EGFR	1.000	0.400	0.478	0.241	0.518	0.500	0.534	0.451	0.466
ErbB2	0.400	1.000	0.291	0.164	0.318	0.300	0.373	0.279	0.259
VEGFR2	0.478	0.291	1.000	0.290	0.607	0.427	0.615	0.435	0.565
PI3K	0.241	0.164	0.290	1.000	0.259	0.282	0.225	0.242	0.322
BRAF	0.518	0.318	0.607	0.259	1.000	0.589	0.491	0.409	0.522
CDK2	0.500	0.300	0.427	0.282	0.589	1.000	0.456	0.460	0.433
LCK	0.534	0.373	0.615	0.225	0.491	0.456	1.000	0.433	0.419
MET	0.451	0.279	0.435	0.242	0.409	0.460	0.433	1.000	0.409
p38a	0.466	0.259	0.565	0.322	0.522	0.433	0.419	0.409	1.000

### 3.1.3 Kinase Similarity Assessment Pipeline for Off-Target Prediction Publication D

Computational target prediction methods are limited, amongst others, by the availability of data. Hence, the parallel assessment of kinase similarity is best conducted from different perspectives that are based on different data resources. In this study, we present a production-ready pipeline that allows users to define a kinase set of interest and to compare the kinases based on their pocket sequences, pocket structures (KiSSim [141]), interaction fingerprints, and ligand profiles. Finally, all perspectives are visually summarized to enable a quick and easy assessment of the results.

 <https://github.com/volkamerlab/teachopencadd>

 <https://projects.volkamerlab.org/teachopencadd/talktorials.html#kinase-similarity>



Contribution:

#### Co-first author

Conceptualization (45%)

Data Curation (50%)

Formal Analysis (50%)

Investigation (50%)

Methodology (50%)

Software (50%)

Validation (50%)

Visualization (50%)

Writing — Original Draft (40%)

Writing — Review & Editing (40%)

Reprinted from Kimber TB\*, Sydow D\*, Volkamer A. Kinase Similarity Assessment Pipeline for Off-Target Prediction. *Living Journal of Computational Molecular Science*. **2022**; 3(1):1599. (\*contributed equally) 10.33011/livecoms.3.1.1599

Open access article licensed under a CC BY 4.0 license.

# Kinase Similarity Assessment Pipeline For Off-Target Prediction [Article v1.0]

Talia B. Kimber<sup>1,2†</sup>, Dominique Sydow<sup>1†‡</sup>, Andrea Volkamer<sup>1\*</sup>

<sup>1</sup>*In silico* Toxicology and Structural Bioinformatics, Institute of Physiology, Charité-Universitätsmedizin Berlin, Charitéplatz 1, 10117, Berlin, Germany;

<sup>2</sup>Computational and Systems Biology Program, Sloan Kettering Institute, Memorial Sloan Kettering Cancer Center, New York, New York 10065, United States

This LiveCoMS document is maintained online on GitHub at [https://github.com/volkamerlab/kinase\\_similarity\\_pipeline\\_paper](https://github.com/volkamerlab/kinase_similarity_pipeline_paper); to provide feedback, suggestions, or help improve it, please visit the GitHub repository and participate via the issue tracker.

This version dated June 23, 2022

**Abstract** Kinases are established drug targets to combat cancer and inflammatory diseases. Despite decades of kinase research, challenges still remain, such as the under-exploration of a large fraction of the kinome and the promiscuous binding of many kinase inhibitors. Due to the highly conserved orthosteric ATP binding site in kinases, ligands may bind not only to their designated kinase (on-target) but also to other kinases (off-targets). Such promiscuous binding can cause mild to severe side effects, and the prediction of these off-targets is highly non-trivial. Therefore, we propose a pipeline that allows the study of kinase similarities from four different angles in an automated and modular fashion. The first method considers the binding site sequence. The second method uses structural information via KiSSim, a newly developed fingerprint that considers both physico-chemical and spatial properties of the binding site. The third method involves kinase-ligand interaction fingerprints as provided by KLIFS, and the last method utilizes the measured activity of ligands on kinases based on ChEMBL data. Finally, results for a given set of kinases are collected and analyzed to gain insight into potential off-targets from the different aforementioned perspectives. Since the pipeline is set up as a series of Jupyter notebooks covering both theoretical and practical aspects, the target audience ranges from beginners to advanced users working in the field of natural and computer sciences. The pipeline is part of the TeachOpenCADD project and extends it with this special kinase edition. All code is free, open-source, and made available at <https://github.com/volkamerlab/teachopencadd>.

**\*For correspondence:**

[andrea.volkamer@charite.de](mailto:andrea.volkamer@charite.de) (AV)

<sup>†</sup>These authors contributed equally to this work

**Present address:** <sup>‡</sup>Sosei Heptares, Steinmetz Building, Granta Park, Cambridge CB21 6DG, United Kingdom

## 1 Introduction

Kinases are involved in most cellular processes by phosphorylating—and thereby activating—themselves or other proteins. This family is among the most frequently mutated proteins in tumors and kinases have been successfully studied as drug targets for many decades [1]. Thanks to the longstanding research, a plethora of kinase data is freely available, i.e., as part of databases such as UniProt [2], PDB [3] or ChEMBL [4], and has been made easily accessible via kinase resources such as the KLIFS—Kinase-Ligand Interaction Fingerprints and Structures—database [5]. As of February 2022, 5,911 X-ray structures of human kinases have been resolved (see the KLIFS database [6]) and 70 FDA-approved small molecule protein kinase inhibitors are on the market [7]. Most of the approved drugs bind in the ATP binding pocket and intermediate surroundings (orthosteric binding site).

Although structural data provides rich information, kinases have been widely classified based on sequence. Manning et al. [8] clustered the human protein kinases based on their sequence similarity into eight major groups (AGC, CAMK, CK1, CMGC, STE, TK, TKL, and "Other") as well as atypical kinases. The resulting Manning kinome tree depicts kinase clustering (see Figure 1).

Despite decades of kinase research, challenges still remain [9]. For example:

1. A large fraction of the kinome is un-/underexplored. Figure 1a shows the number of PDB structures per kinase, unveiling a vast imbalance between structurally resolved kinases and unexplored ones. For example, CDK2 has been resolved in 426 PDB structures, while only 313 kinases [6] out of approximately 540 in the kinome [9] have been structurally resolved.
2. Many kinase inhibitors are promiscuous binders, causing off-target effects or enabling polypharmacology [1, 10]. For example, the Epidermal Growth Factor Receptor (EGFR) inhibitor erlotinib shows affinities to other kinases in the highly sequentially-similar TK kinase group, but also strongly affects off-targets in more remote kinase groups (see Figure 1b).

Therefore, assessing kinase similarity from different angles may be a crucial step in understanding and predicting off-targets to help to design more selective drugs and to avoid side effects.

### 1.1 Scope

In this study, similarities between a set of kinases are investigated based on methods offering different perspectives on this challenging topic with a focus on orthosteric binding sites (here referred to as binding sites), as summarized

in Table 1. The first method considers the binding site sequence as deposited in the KLIFS database. The second method uses KiSSim [11], a recently developed fingerprint that considers physico-chemical as well as spatial properties of the binding site. The third method involves protein-ligand interaction fingerprints as provided in the KLIFS database, and the last method utilizes the measured activity of ligands against kinases based on ChEMBL data [4]. The different methods are preceded by a general introduction to kinases and the challenges faced in kinase-centric drug design, and succeeded by a comparison between the different kinase similarity methods.

Note that this study focuses on the similarities between ATP binding sites. Therefore, kinase polypharmacology and off-targets can only be assessed within the scope of orthosteric binding sites, even though the promiscuity of some ligands may be explained by binding to allosteric binding sites. Potential allosteric binding sites are summarized in the Kinase Atlas [12].

This study has been assembled into a modular pipeline that enables the research of kinase similarities in an automated fashion, allowing users to simply use it out of the box, or adapt it to their needs.

This workflow is integrated in the context of TeachOpenCADD [15, 16], a teaching platform for computer-aided drug design (CADD) using open-source packages and data. Specific tasks in cheminformatics and structural bioinformatics are described and solved using Python-based Jupyter notebooks [17] as interactive platform. All code has been deposited on GitHub, see <https://github.com/volkamerlab/teachopencadd>. The project website can be found at this link, <https://projects.volkamerlab.org/teachopencadd>.

## 2 Prerequisites

### 2.1 Target audience

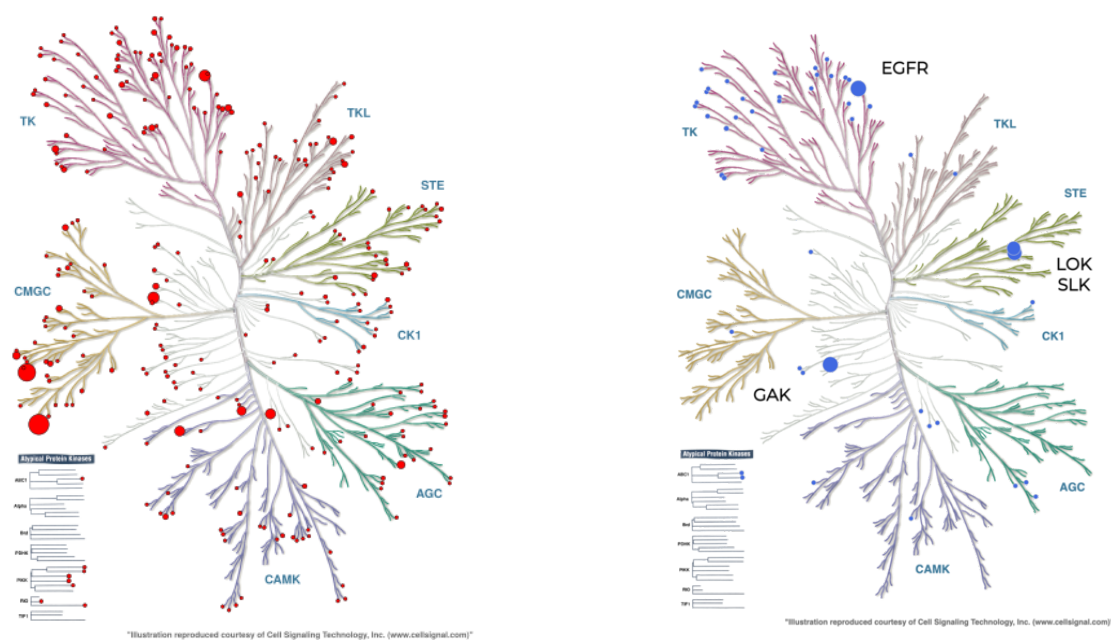
The notebooks were developed to support researchers interested in kinase-centric computational drug design, with a focus on understanding and predicting kinase off-targets. As this collection is part of the TeachOpenCADD training material [15, 16], we also recommend the notebooks to teachers as pedagogical interactive material in structural bioinformatics and cheminformatics.

### 2.2 Background knowledge

The notebooks are constructed in a way that no in depth prior knowledge besides an affinity for the natural or computer sciences is required. Each notebook eases into the topic of kinase drug development and kinase similarity with a lot of theoretical background and comments on all content as well

Topic	Description	Hyperlink
What is a kinase?	Introduction to kinases and challenges in drug discovery.	<a href="#">T023</a>
Pocket sequence	Pairwise similarities/identities between 85 residue long KLIFS pocket sequences.	<a href="#">T024</a>
Pocket structure	Pairwise similarities between 1,032-bit long KiSSim fingerprints, which encode spatial and physico-chemical pocket properties.	<a href="#">T025</a>
Pocket-ligand interactions	Pairwise similarities between 595-bit long KLIFS kinase-ligand interaction fingerprints (IFP).	<a href="#">T026</a>
Ligand profile	Pairwise similarity based on the ratio of compounds tested active against kinase pairs.	<a href="#">T027</a>
Kinase similarity	Comparison between predicted off-targets based on calculated kinase similarities using aforementioned methods.	<a href="#">T028</a>

**Table 1.** TeachOpenCADD kinase edition overview: Notebook topics, description, and index with a hyperlink to the associated notebook.



**(a)** Number of PDB structures per kinase. The figure shows the imbalance between highly explored kinases, for example, the groups TK and CMGC. The CDK2 kinase in the CMGC group has the most structures, with 426. The red circle is proportional to the number of PDB structures for each kinase, such that the greater is the circle, the higher is the number of structures.

**(b)** Developing selective kinase inhibitors is non-trivial since kinases are highly conserved in the ATP binding site. EGFR inhibitor erlotinib binds not only to its intended target EGFR, but also to kinases in remote groups, such as SLK/LOK in the STE group and GAK in the "Other" group. The blue circle is proportional to the  $K_d$  value in nM taken from the Karaman et al. [13] dataset.

**Figure 1.** Visual representation using the Manning tree of existing challenges in kinase research: un-/underexplored kinase groups (left) and the promiscuous behavior of kinases (right). The figure is taken from [https://projects.volkamerlab.org/teachopencadd/talktorials/T023\\_what\\_is\\_a\\_kinase.html](https://projects.volkamerlab.org/teachopencadd/talktorials/T023_what_is_a_kinase.html) and is generated using KinMap [14].

as programming-related steps in great detail. Nevertheless, users will benefit from a basic understanding of the Python programming language and the usage of Jupyter notebooks. If such basic introduction is needed, please refer to training material as listed on the TeachOpenCADD website [18].

### 2.3 Software requirements

The notebooks are written in Python and rely on open-source packages such as pandas [19], numpy [20], scipy [21], matplotlib [22], seaborn [23], scikit-learn [24], rdkit [25], biotite [26], opencadd [27], kissim [28], and requests [29].

The user only needs to install the *teachopencadd* conda-forge package [30] (see installation [31]), which will install all relevant packages and save a copy of all TeachOpenCADD notebooks—including the kinase edition discussed in this paper—on the user's local machine. A read-only mode of the notebooks is accessible via the TeachOpenCADD website at <https://projects.volkamerlab.org/teachopencadd/>. Online execution can be done via Binder [32], using the following link <https://mybinder.org/v2/gh/volkamerlab/TeachOpenCADD/master>.

## 3 Method

In this section, the four methods that are introduced to measure kinase similarity are described, namely the pocket sequence, the KiSSim fingerprint, the interaction fingerprint, and the ligand profile. Note that the theoretical and practical aspects of each method are also covered in great detail in the individual notebooks of this kinase collection (Table 1). As discussed in the "Scope" section of this manuscript, we focus on kinase similarity based on orthosteric binding sites.

### 3.1 Pocket sequence

The full amino acid sequence is often used to assess similarities between kinases (see the phylogenetic tree developed by Manning et al. [8]). Since binding sites are often more conserved than the whole protein, van Linden et al. [33] defined as part of KLIFS a 85-long pocket sequence that is aligned across the kinome. Using a sequence that focuses on the binding site seems appropriate in the case of kinases, since this is where the ligand is likely to bind. Moreover, working with a fixed length sequence is practical from a computational point of view.

In this study, two methods are used to compute relationships based on sequence, namely the sequence identity and the sequence similarity, which are described below.

#### 3.1.1 Sequence identity

The pairwise sequence identity, or simply sequence identity, is a similarity based on character-wise discrepancy, in other

terms, the number of residues that match in two aligned sequences [34]. More formally, given two kinase sequences  $S$  and  $S'$  of same lengths  $L$ , the sequence identity can be defined as

$$\text{sequence identity}(S, S') = \frac{1}{L} \sum_{n=1}^L I(S[n], S'[n]), \quad (1a)$$

where  $I$  is the identity matrix of the amino acids, and  $S[n]$  the amino acid at position  $n$  of the kinase sequence  $S$ . Note that not all kinases have residues present at each of the 85 alignment positions. Such gaps are represented by "-" and count as mismatch to any amino acid.

#### 3.1.2 Sequence similarity

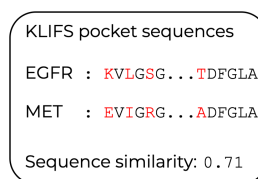
Unlike sequence identity which treats all residues uniformly, pairwise sequence similarity, or sequence similarity, takes into account the change of the amino acids over evolutionary time, thus, reflecting relationships between amino acids. It is based on a substitution matrix  $M$ , where each entry gives a score between two amino acids. In this study, the BLOSUM substitution matrix [35], as implemented in biotite [36], is used. Formally, the following is defined:

$$\text{sequence similarity}(S, S') = \frac{1}{L} \sum_{n=1}^L M'(S[n], S'[n]), \quad (1b)$$

where  $M'$  is the translated and rescaled version of the substitution matrix  $M$ .

For both the sequence identity and similarity, the closer the value is to 1, the more similar are the kinases.

Figure 2 shows the sequence similarity between the KLIFS pocket sequence of EGFR and MET kinases. Sequence similarity is used by default in the pipeline for further analysis.

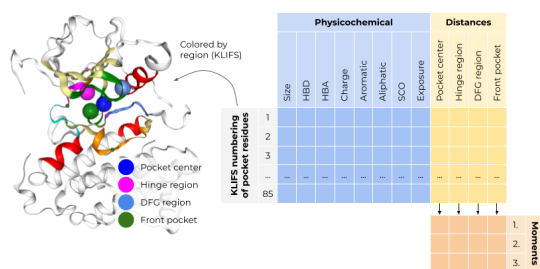


**Figure 2.** Sequence similarity between EGFR and MET. The 85-residue pocket sequence is retrieved from KLIFS. The pairwise sequence similarity takes into account the change of the amino acids over evolutionary time.

### 3.2 The KiSSim fingerprint

In order to assess the pairwise similarity of kinases from a structural point of view, the newly developed KiSSim (Kinase

Structure Similarity) fingerprint [11, 28] is used. This fingerprint describes the physico-chemical and spatial properties of structurally resolved kinases, while focusing on the KLIFS pocket residues. Each structure is mapped to a fingerprint composed of 1,032 bits, the first 680 (=  $85 \times 8$ ) bits describing physico-chemical features and the remaining 352 (=  $85 \times 4 + 12$ ) bits spatial information (see Figure 3).



**Figure 3.** The 1,032-long KiSSim fingerprint encodes physico-chemical and spatial properties of the kinase's pocket, adding a structural perspective on kinases. The figure is adapted from [28].

### 3.2.1 From several structures to one kinase

A kinase can be represented by one or even a hundred resolved crystal structures in the PDB (see Figure 1a). In this study, we aim at comparing different kinases and not individual structures. Since KiSSim generates a fingerprint for each structure, the following mapping from structures to kinase is applied:

Given two kinases  $K$  and  $K'$ , all available structures in KLIFS for these kinases are fetched using `opencadd` [27], namely  $s_1, \dots, s_m$  for kinase  $K$ , and  $s'_1, \dots, s'_n$  for kinase  $K'$ , noting that the number of structures might be different for each kinase. Each structure  $s_i, s'_i$  is then mapped to its corresponding KiSSim fingerprint  $fp_i, fp'_i$ , see Figure 4. The fingerprints  $fp, fp'$  corresponding to kinases  $K, K'$  respectively, are the ones for which the Euclidean distance is minimized (Figure 4). Note that these *minimal distance* fingerprints vary for each kinase depending on the compared  $K, K'$  pair.

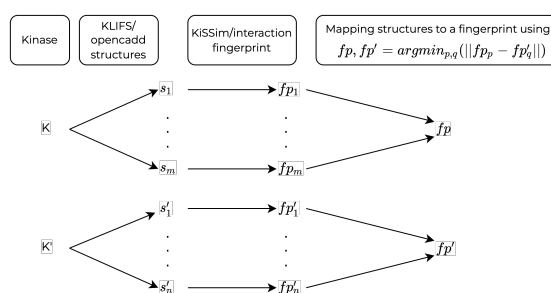
Finally, two kinases  $K, K'$  are compared based on their respective *minimal distance* between KiSSim fingerprint  $fp, fp'$  using the Euclidean norm:

$$\text{KiSSim dissimilarity}(fp, fp') = \|fp - fp'\|_2. \quad (2)$$

In this case, the closer the value to 0, the more similar the kinases.

## 3.3 The interaction fingerprint

Interaction fingerprints (IFPs) encode the binding mode of a ligand in a binding site, i.e., the protein-ligand interactions



**Figure 4.** Associating one structural fingerprint per kinase. All available structures are retrieved for two given kinases and all fingerprints are computed. The fingerprints selected to be associated with the kinase in the present kinase pair are the ones for which the computed distance is minimized.

that are present in a structurally resolved complex. If a ligand can form similar interaction patterns in proteins other than its designated protein (off- vs. on-target), it is possible that this ligand will cause unintended side effects. Knowledge about binding mode similarities can therefore help to avoid such off-target effects.

The KLIFS interaction fingerprint describes seven possible interactions for each of the 85 residues in the binding pocket. Interactions include 1. hydrophobic contacts, 2. aromatic interactions, face to face, 3. aromatic interactions, edge to face, 4. H-bond donors, 5. H-bond acceptors, 6. cationic interactions, and 7. anionic interactions. The 595-bit long vector describes the presence or absence of such interactions for all 85 residues (see Figure 5).

1							2							3							85						
HYD	F-F	F-E	DON	ACC	ION+	ION-	HYD	F-F	F-E	DON	ACC	ION+	ION-	HYD	F-F	F-E	DON	ACC	ION+	ION-	HYD	F-F	F-E	DON	ACC	ION+	ION-
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0

**Figure 5.** The KLIFS interaction fingerprint encodes seven interaction types for each of the 85 residues in the binding site. Interaction types include: hydrophobic contacts (HYD), face to face aromatic interactions (F-F), face to edge aromatic interactions (F-E), protein H-bond donors (DON), protein H-bond acceptors (ACC), protein cationic interactions (ION+), and protein anionic interactions (ION-). The figure is taken from [37].

Similarly to the KiSSim comparison, given two kinases  $K$  and  $K'$ , all available structures in KLIFS for these kinases are fetched using `opencadd` [27]. Each structure is mapped to its corresponding IFP. The interaction fingerprints  $fp, fp'$  corresponding to kinases  $K, K'$  respectively are the ones for which the Jaccard distance [38] is minimized (Figure 4). Note that the Euclidean distance is used in case of the KiSSim fingerprint, which contains continuous and discrete values, while the Jaccard distance is employed in case of the binary IFPs.

Finally, two kinases  $K$ ,  $K'$  are compared using their respective *minimal distance* between interaction fingerprint  $fp$ ,  $fp'$  and calculating the Jaccard distance:

$$\text{IFP dissimilarity}(fp, fp') = d_j(fp, fp'), \quad (3)$$

where  $d_j$  is the Jaccard distance.

In this case, the closer the value to 0, the more similar the kinases.

### 3.4 Ligand profile

In the context of drug design, the following assumption is often made: if a compound was tested active on two different kinases, it is suspected that these two kinases may have some degree of similarity [39]. This is the rationale behind the ligand profile similarity. Given bioactivity data for a set of compounds measured against a set of targets—in this case kinases—and two kinases  $K$ ,  $K'$ , ligand profile similarity is defined as

$$\text{lig. profile similarity}(K, K') = \frac{\# \text{ actives on both } K \text{ and } K'}{\# \text{ tested on both } K \text{ and } K'}. \quad (4)$$

The closer the value is to 1, the more similar are the kinases. If no compounds were commonly tested on two kinases, then the similarity is set to 0. Computing the similarity between a kinase and itself may be interpreted as kinase promiscuity, where the similarity described above would therefore represent the fraction of active compounds over all tested compounds for this kinase.

#### 3.4.1 Bioactivity data

The bioactivity data used for this method comes from Kinodata [40], from the Openkinome organization [41]. It is a pre-processed kinase subset of the ChEMBL data [4], version 29. Further processing includes keeping only  $IC_{50}$  values given in nM, and converting them to  $pIC_{50}$  values. If there are several measurements for a kinase-compound pair, then the most active value, i.e., the entry with the highest  $pIC_{50}$  value, is kept. Finally, the  $pIC_{50}$  values are binarized using a 6.3 cut-off to discriminate between an active or inactive compound as described in [42].

In the pipeline, one can additionally compute the non-reduced ratio of number of active compounds against the total number of compounds to gain insight into the actual number of measurements for each kinase pair.

### 3.5 Kinase comparison and clustering

To assess kinase similarities based on the calculated (dis)similarity matrices, two visualization methods are used, namely heatmaps and dendrograms.

#### 3.5.1 Heatmaps

The heatmaps are generated using matplotlib [22] to depict the similarity between a set of kinases. The maximum value is 1, indicating exact similarity, as is the case for diagonal entries. The value 0 indicates total dissimilarity. Plotting such figures allows to see and extract patterns thanks to the gradient of colors, see top row in Figure 6.

#### 3.5.2 Dendrograms

Clustering algorithms are used to identify groups such that the similarities within clusters are higher than compared to other clusters [43]. In this study, hierarchical clustering is used, and, unlike heatmaps, it is based on distance (or dissimilarity). Hierarchical clustering can be graphically displayed using a dendrogram (see bottom row in Figure 6), where the height of each node is proportional to the dissimilarity between its two daughter clusters. The clustering and plotting is done using scikit-learn [24] and matplotlib [22], respectively.

For fair comparison, the distance matrices for all four methods are normalized so that each entry lives between 0 and 1. Similarity matrices—as used for the heatmaps—are then computed using 1-distance matrix. Contrary to the dendrograms, that use the distance matrix.

## 4 Pipeline

Measuring kinase similarity is a non-trivial task; distinct measures can provide different insights, which can be complementary, confirmatory, or contradictory, and therefore expand our knowledge on the target(s) at hand. However, implementing multiple methods can be time-consuming and comparing results across many output types can be laborious. Turning such processes into a functional pipeline helps to avoid the scattering of scripts and to speed up iterations of the design-make-test-analyze cycle [44] of drug design campaigns. Moreover, following the findable, accessible, interoperable, and reusable (FAIR) principles [45] makes such pipelines long-lasting and available to the community.

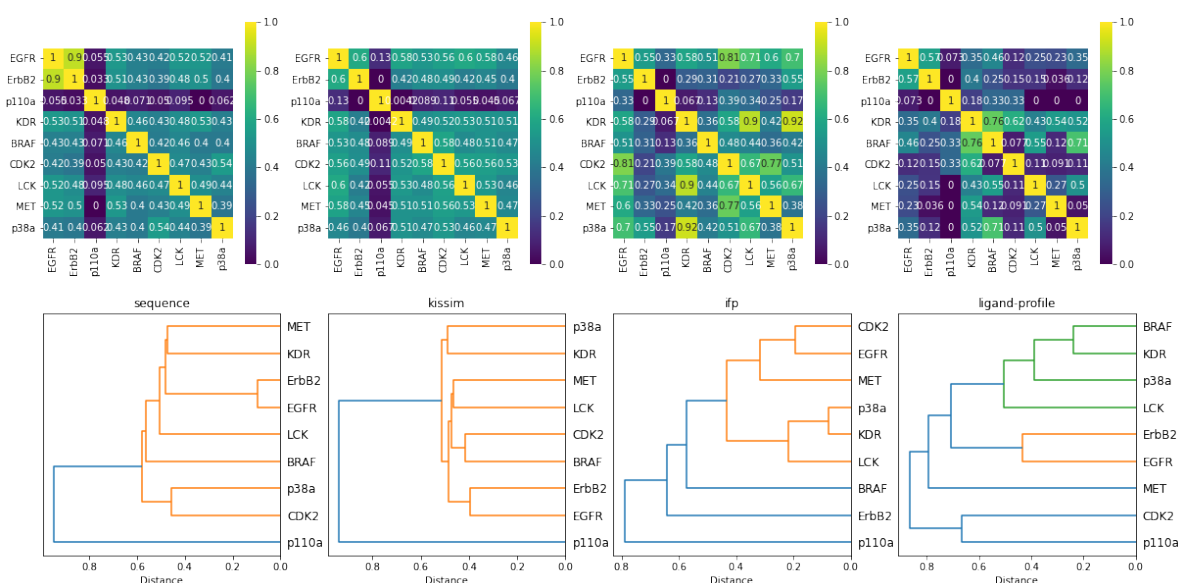
In the pipeline presented herein, we implemented the different methods once and streamlined each method's results into a standardized output with a pre-defined set of visualization tools for easy comparison. Moreover, the pipeline is flexible enough so that adding new methods or new visualization tools is effortless, making the whole process easy to understand, maintain, and expand.

### 4.1 Means of the pipeline

The proposed pipeline is a collection of six Jupyter notebooks [17] that allows the study of kinase similarity from four different angles in an automated and modular fashion (Figure 7).



## A LiveCoMS Training Article



**Figure 6.** Visualization of kinase similarity from four different angles: sequence, KiSSim, interaction fingerprint (ifp) as well as ligand-profile. The top, bottom row shows four heatmaps, dendrograms respectively for a set of nine study kinases.

## 4.2 Structure of the notebooks

The structure of all notebooks is as follows: the first section covers the theory written in Markdown and summarizes the necessary concepts to understand the task. Relevant references are also mentioned. The second part of a notebook deals with the actual implementation of the task in a pedagogical manner, including motivation for practical steps and detailed comments on coding decisions. Finally, a discussion and a quiz section wrap up the notebook. This structure is very well suited from a teaching perspective, since it contains both theory and hands on programming. The notebook can easily be used as a medium for a presentation, and it allows for self-study as well as usage in own research projects.

## 4.3 About the code

The programming section is done in Python exclusively and the code follows the latest software best practices. It is written pythonically and contains lots of code comments. Thanks to the continuous integration (CI), all outputs and results are fully reproducible and the maintenance of the pipeline is facilitated.

## 4.4 Content of the pipeline

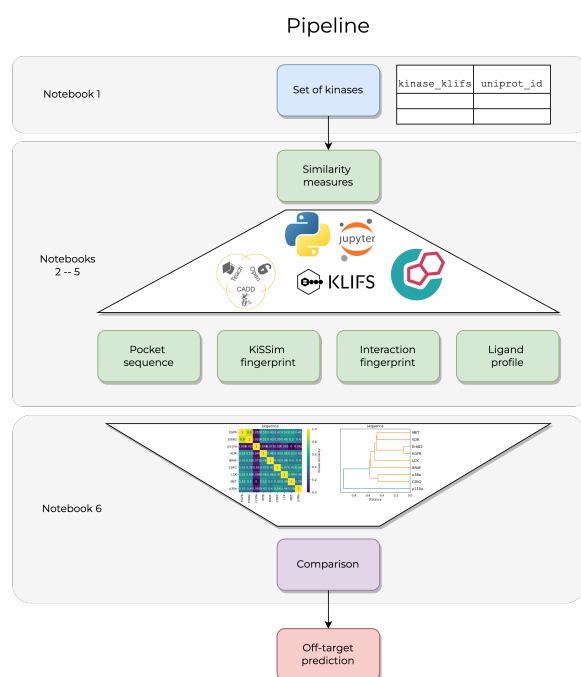
As mentioned previously, the proposed pipeline contains six notebooks, described below:

The first notebook sets the stage with a kinase introduction and references/tools on where to find kinase-related

information. It is also in this first notebook that a set of kinases of interest is defined. In this study, nine kinases are selected, the same nine as in the paper by Schmidt et al. [46], where the authors discussed the challenges and advantages of tackling kinase similarity from multiple perspectives. Table 2 summarizes the information used for these kinases. The pipeline can be executed out of the box with the defined set of kinases, but it can equally be run with a different user defined set of kinases. The only condition is that the uploaded CSV file with the kinases of interest contains two mandatory columns, namely `kinase_klif`s, which is the KLIFS name of the kinase, and `uniprot_id`, the Uniprot identifier (ID) [2] of the kinase (Figure 7).

The four following notebooks describe one similarity method at a time as discussed in Section 3: the pocket sequence, the KiSSim fingerprint, the interaction fingerprint, and the ligand profile.

The final notebook collects the information from the previous ones and compares the different perspectives with easy-to-understand visualization such as heatmaps and dendrograms (see Section 3.5). Additionally, an equally weighted average to combine distance and similarity matrices from all four perspectives can be computed, yielding a single heatmap, and a single dendrogram. The user can easily extend this to a knowledge-informed weighting scheme based on their own research focus.



**Figure 7.** The proposed pipeline consists of six Jupyter notebooks [17]. Given a set of kinases in a CSV format, four similarity measures are implemented, and kinases are compared using heatmaps and dendrograms. The project is part of TeachOpenCADD [15, 16] and uses open-source tools and databases such as KLIFS [5] and ChEMBL [4].

#### 4.5 Features of the pipeline

The developed pipeline contains many useful features. Firstly, it is part of the TeachOpenCADD project [15, 16] and extends it with this special kinase edition. Being part of TeachOpenCADD has the following advantages:

1. TeachOpenCADD is open-source and freely available at <https://github.com/volkamerlab/teachopencadd>, under the Attribution 4.0 International (CC BY 4.0) license.
2. A dedicated conda package [47] facilitates installation.
3. Online execution is possible via the Binder project [32].
4. The teaching approach makes the notebooks easy to follow.

Moreover, the pipeline is easily adaptable to new sets of kinases as well as new similarity methods, defined by a user.

## 5 Conclusion

In this study, a full pipeline for the assessment of kinase similarity is presented, using four methods of comparison. The pipeline is composed of six Jupyter notebooks:

1. An introduction to kinases and their central role in drug discovery, as well as the collection of the kinase set for the downstream notebooks.
2. The similarity from a pocket sequence point of view.
3. The similarity based on the KISSim fingerprint, which encodes physico-chemical and spatial properties of the kinase pocket.
4. The similarity based on KLIFS interaction fingerprints between the kinase pocket residues and a co-crystallized ligand.
5. The similarity based on ligand profiling data collected from ChEMBL, measuring a compound's activity on a kinase.
6. An analysis notebook which collects the proximity matrices calculated for the four methods, visualizes the similarities with heatmaps and the clusters with dendrograms, and finally discusses the results.

We encourage users to develop their own similarity methods and to contribute to the existing pipeline.

This paper could be of interest to

1. researchers who want to gain insights into off-target prediction and kinase similarity, and integrate their new comparison methods to a working workflow,
2. beginners in software development who need inspiration to set up a fully functional pipeline,
3. teachers who want a starting point for lecture material,
4. students with a background in bioinformatics, cheminformatics, and the life sciences in general,
5. anyone who is curious.

#### Author Contributions

Conceptualization: TBK, DS, AV; Methodology: TBK, DS, AV; Software: TBK, DS, AV; Validation: TBK, DS, AV; Formal Analysis: TBK, DS, AV; Investigation: TBK, DS, AV; Writing – Original Draft: TBK, DS, AV; Writing – Review & Editing: TBK, DS, AV; Visualization: TBK, DS, AV; Project Administration: TBK, DS, AV; Funding Acquisition, Supervision: AV.

For a more detailed description of author contributions, see the GitHub issue tracking and changelog at [https://github.com/volkamerlab/kinase\\_similarity\\_pipeline\\_paper](https://github.com/volkamerlab/kinase_similarity_pipeline_paper).

#### Potentially Conflicting Interests

The authors declare no conflict of interests.

#### Abbreviations

List of abbreviations used in the paper.

kinase	<b>kinase_klifs</b>	<b>uniprot_id</b>	group	full kinase name
EGFR	EGFR	P00533	TK	Epidermal growth factor receptor
ErbB2	ErbB2	P04626	TK	Erythroblastic leukemia viral oncogene homolog 2
PI3K	p110a	P42336	Atypical	Phosphatidylinositol-3-kinase
VEGFR2	KDR	P35968	TK	Vascular endothelial growth factor receptor 2
BRAF	BRAF	P15056	TKL	Rapidly accelerated fibrosarcoma isoform B
CDK2	CDK2	P24941	CMGC	Cyclic-dependent kinase 2
LCK	LCK	P06239	TK	Lymphocyte-specific protein tyrosine kinase
MET	MET	P08581	TK	Mesenchymal-epithelial transition factor
p38a	p38a	Q16539	CMGC	p38 mitogen activated protein kinase alpha

**Table 2.** Set of defined kinases. The table lists the kinases used in the pipeline, the same nine as in the study by Schmidt et al. [46]. It is noteworthy that the pipeline is applicable to an arbitrary set of kinases, the only condition being that the input CSV file should contain two columns, **kinase\_klifs** and **uniprot\_id**, displayed in bold.

KLIFS	Kinase-Ligand Interaction Fingerprints and Structures	JP, Papadatos G, Smit I, Leach AR. The ChEMBL database in 2017. <i>Nucleic Acids Research</i> . 2016; 45(D1):D945–D954. <a href="https://doi.org/10.1093/nar/gkw1074">https://doi.org/10.1093/nar/gkw1074</a> .
EGFR	Epidermal Growth Factor Receptor	
KiSSim	Kinase Structure Similarity	
IFP	Interaction Fingerprint	[5] Kanev GK, de Graaf C, Westerman BA, de Esch IJP, Kooistra AJ. KLIFS: an overhaul after the first 5 years of supporting kinase research. <i>Nucleic Acids Research</i> . 2020; 49(D1):D562–D569. <a href="https://doi.org/10.1093/nar/gkaa895">https://doi.org/10.1093/nar/gkaa895</a> .
ID	Identifier	
CI	Continuous Integration	

## Funding Information

TBK received funding from the Stiftung Charité in the context of the Einstein BIH Visiting Fellow Project, DS from the Deutsche Forschungsgemeinschaft (grant VO 2353/1-1), and AV from the Bundesministerium für Bildung und Forschung (grant number 031A262C).

## Author Information

### ORCID:

Talia B. Kimber: [0000-0002-8881-920X](https://orcid.org/0000-0002-8881-920X)

Dominique Sydow: [0000-0003-4205-8705](https://orcid.org/0000-0003-4205-8705)

Andrea Volkamer: [0000-0002-3760-580X](https://orcid.org/0000-0002-3760-580X)

## References

- [1] Cohen P, Cross D, Jänne PA. Kinase drug discovery 20 years after imatinib: progress and future directions. *Nature Reviews Drug Discovery*. 2021; 20(7):551–569. <https://doi.org/10.1038/s41573-021-00195-4>.
- [2] Consortium TU. UniProt: the universal protein knowledge-base in 2021. *Nucleic Acids Research*. 2020; 49(D1):D480–D489. <https://doi.org/10.1093/nar/gkaa1100>.
- [3] Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE. The Protein Data Bank. *Nucleic Acids Research*. 2000; 28(1):235–242. <https://doi.org/10.1093/nar/28.1.235>.
- [4] Gaulton A, Hersey A, Nowotka M, Bento AP, Chambers J, Mendez D, Motow P, Atkinson F, Bellis LJ, Cibrián-Uhalte E, Davies M, Dedman N, Karlsson A, Magariños MP, Overington
- [5] Kanev GK, de Graaf C, Westerman BA, de Esch IJP, Kooistra AJ. KLIFS: an overhaul after the first 5 years of supporting kinase research. *Nucleic Acids Research*. 2020; 49(D1):D562–D569. <https://doi.org/10.1093/nar/gkaa895>.
- [6] KLIFS; 2022. [Online; accessed 01-February-2022]. <https://klifs.net/>.
- [7] Blue Ridge Institute for Medical Research in Horse Shoe NCU, FDA-approved small molecule protein kinase inhibitors; 2022. [Online; accessed 01-February-2022]. <http://www.brimr.org/PKI/PKIs.htm>.
- [8] Manning G, Whyte DB, Martinez R, Hunter T, Sudarsanam S. The Protein Kinase Complement of the Human Genome. *Science*. 2002; 298(5600):1912–1934. <https://doi.org/10.1126/science.1075762>.
- [9] Kooistra AJ, Volkamer A. Kinase-Centric Computational Drug Development. In: *Annual Reports in Medicinal Chemistry* Elsevier; 2017.p. 197–236. <https://doi.org/10.1016/bs.armc.2017.08.001>.
- [10] Morphy R. Selectively Nonselective Kinase Inhibition: Striking the Right Balance. *J Med Chem*. 2009; 53(4):1413–1437. <https://doi.org/10.1021/JM901132V>.
- [11] Sydow D, Aßmann E, Kooistra AJ, Rippmann F, Volkamer A. KiSSim: Predicting Off-Targets from Structural Similarities in the Kinome. *Journal of Chemical Information and Modeling*. 2022; 62(10):2600–2616. <https://doi.org/10.1021/acs.jcim.2c00050>.
- [12] Yueh C, Rettenmaier J, Xia B, Hall DR, Alekseenko A, Porter KA, Barkovich K, Keseru G, Whitty A, Wells JA, Vajda S, Kozakov D. Kinase Atlas: Druggability Analysis of Potential Allosteric Sites in Kinases. *J Med Chem*. 2019; 62(14):6512–6524. <https://doi.org/10.1021/acs.jmedchem.9b00089>.
- [13] Karaman MW, Herrgard S, Treiber DK, Gallant P, Atteridge CE, Campbell BT, Chan KW, Ciceri P, Davis MI, Edeen PT, Faraoni R, Floyd M, Hunt JP, Lockhart DJ, Milanov ZV, Morrison MJ, Pallares

- G, Patel HK, Pritchard S, Wodicka LM, et al. A quantitative analysis of kinase inhibitor selectivity. *Nature Biotechnology*. 2008; 26(1):127–132. <https://doi.org/10.1038/nbt1358>.
- [14] Eid S, Turk S, Volkamer A, Rippmann F, Fulle S. KinMap: a web-based tool for interactive navigation through human kinome data. *BMC Bioinformatics*. 2017; 18(1). <https://doi.org/10.1186/s12859-016-1433-7>.
- [15] Sydow D, Morger A, Driller M, Volkamer A. TeachOpenCADD: a teaching platform for computer-aided drug design using open source packages and data. *Journal of Cheminformatics*. 2019; 11(1). <https://doi.org/10.1186/s13321-019-0351-x>.
- [16] Sydow D, Rodríguez-Guerra J, Kimber TB, Schaller D, Taylor CJ, Chen Y, Leja M, Misra S, Wichmann M, Ariamajd A, Volkamer A. TeachOpenCADD 2022: open source and FAIR Python pipelines to assist in structural bioinformatics and cheminformatics research. *Nucleic Acids Research*. 2022; <https://doi.org/10.1093/nar/gkac267>, [gkac267](https://doi.org/10.1093/nar/gkac267).
- [17] Kluyver T, Ragan-Kelley B, Pérez F, Granger B, Bussonnier M, Frederic J, Kelley K, Hamrick J, Grout J, Corlay S, Ivanov P, Avila D, Abdalla S, Willing C, development team J. Jupyter Notebooks - a publishing format for reproducible computational workflows. In: Loizides F, Schmidt B, editors. *Positioning and Power in Academic Publishing: Players, Agents and Agendas* IOS Press; 2016. p. 87–90. <https://doi.org/10.3233/978-1-61499-649-1-87>.
- [18] List of Python introduction resources; 2022. [Online; accessed 21-February-2022]. <https://github.com/volkamerlab/teachopencadd#python-programming-introduction>.
- [19] The pandas development team, pandas-dev/pandas: Pandas. Zenodo; 2020. <https://doi.org/10.5281/zenodo.3509134>.
- [20] Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, Wieser E, Taylor J, Berg S, Smith NJ, Kern R, Picus M, Hoyer S, van Kerkwijk MH, Brett M, Haldane A, del Río JF, Wiebe M, Peterson P, Gérard-Marchant P, et al. Array programming with NumPy. *Nature*. 2020; 585(7825):357–362. <https://doi.org/10.1038/s41586-020-2649-2>.
- [21] Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, van der Walt SJ, Brett M, Wilson J, Millman KJ, Mayorov N, Nelson ARJ, Jones E, Kern R, Larson E, Carey CJ, et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*. 2020; 17:261–272. <https://doi.org/10.1038/s41592-019-0686-2>.
- [22] Hunter JD. Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*. 2007; 9(3):90–95. <https://doi.org/10.1109/mcse.2007.55>.
- [23] Waskom ML. seaborn: statistical data visualization. *Journal of Open Source Software*. 2021; 6(6):3021. <https://doi.org/10.21105/joss.03021>.
- [24] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. 2011; 12:2825–2830.
- [25] RDKit, RDKit: Open-Source Cheminformatics. <http://www.rdkit.org>, [Online; accessed 2022-02-02].
- [26] Kunzmann P, Hamacher K. Biotite: a unifying open source computational biology framework in Python. *BMC Bioinformatics*. 2018; 19(1):346. <https://doi.org/10.1186/s12859-018-2367-z>.
- [27] Sydow D, Rodríguez-Guerra J, Volkamer A. OpenCADD-KLIFS: A Python package to fetch kinase data from the KLIFS database. *Journal of Open Source Software*. 2022; 7(70):3951. <https://doi.org/10.21105/joss.03951>.
- [28] Volkamerlab, KiSSim open-source Python package; 2022. [Online; accessed 01-February-2022]. <https://github.com/volkamerlab/kissim>.
- [29] requests, requests. <https://docs.python-requests.org/>; <https://docs.python-requests.org/>, [Online; accessed 2022-02-02]. <https://docs.python-requests.org/>.
- [30] TeachOpenCADD conda-forge package; 2022. [Online; accessed 2022-02-02]. <https://anaconda.org/conda-forge/teachopencadd>.
- [31] TeachOpenCADD, TeachOpenCADD installation instructions. <https://volkamerlab.org/>; [Online; accessed 2022-02-02]. <https://projects.volkamerlab.org/teachopencadd/installing.html>.
- [32] Project Jupyter, Matthias Bussonnier, Jessica Forde, Jeremy Freeman, Brian Granger, Tim Head, Chris Holdgraf, Kyle Kelley, Gladys Nalvarte, Andrew Osherooff, Pacer M, Yuvi Panda, Fernando Perez, Benjamin Ragan Kelley, Carol Willing. Binder 2.0 - Reproducible, interactive, sharable environments for science at scale. In: Fatih Akici, David Lippa, Dillon Niederhut, Pacer M, editors. *Proceedings of the 17th Python in Science Conference*; 2018. p. 113 – 120. <https://doi.org/10.25080/Majora-4af1f417-011>.
- [33] van Linden OPJ, Kooistra AJ, Leurs R, de Esch IJP, de Graaf C. KLIFS: A Knowledge-Based Structural Database To Navigate Kinase–Ligand Interaction Space. *Journal of Medicinal Chemistry*. 2013; 57(2):249–277. <https://doi.org/10.1021/jm400378w>.
- [34] Rost B. Twilight zone of protein sequence alignments. *Protein Engineering, Design and Selection*. 1999; 12(2):85–94. <https://doi.org/10.1093/protein/12.2.85>.
- [35] Henikoff S, Henikoff JG. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*. 1992; 89(22):10915–10919. <https://doi.org/10.1073/pnas.89.22.10915>.
- [36] Kunzmann P, Hamacher K. Biotite: a unifying open source computational biology framework in Python. *BMC Bioinformatics*. 2018; 19(1). <https://doi.org/10.1186/s12859-018-2367-z>.
- [37] TeachOpenCADD, TeachOpenCADD website. <https://volkamerlab.org/>; [Online; accessed 2022-02-02]. <https://projects.volkamerlab.org/teachopencadd/>.
- [38] Kosub S. A note on the triangle inequality for the Jaccard distance. *Pattern Recognition Letters*. 2019; 120:36–38. <https://doi.org/https://doi.org/10.1016/j.patrec.2018.12.007>.
- [39] Barelier S, Sterling T, O'Meara MJ, Shoichet BK. The Recognition of Identical Ligands by Unrelated Proteins. *ACS Chemical Biology*. 2015; 10(12):2772–2784. <https://doi.org/10.1021/acschembio.5b00683>.

- [40] Kinodata; 2022. [Online; accessed 01-February-2022]. <https://github.com/openkinome/kinodata>.
- [41] OpenKinome; 2022. [Online; accessed 01-February-2022]. <http://openkinome.org/>.
- [42] Merget B, Turk S, Eid S, Rippmann F, Fulle S. Profiling Prediction of Kinase Inhibitors: Toward the Virtual Assay. *Journal of Medicinal Chemistry*. 2017; 60(1):474-485. <https://doi.org/10.1021/acs.jmedchem.6b01611>.
- [43] Hastie T, Tibshirani R, Friedman J. *The Elements of Statistical Learning*. Springer New York; 2009. <https://doi.org/10.1007/978-0-387-84858-7>.
- [44] Schneider P, Walters WP, Plowright AT, Sieroka N, Listgarten J, Goodnow RA, Fisher J, Jansen JM, Duca JS, Rush TS, Zentgraf M, Hill JE, Krutoholow E, Kohler M, Blaney J, Funatsu K, Luebke-mann C, Schneider G. Rethinking drug design in the artificial intelligence era. *Nature Reviews Drug Discovery*. 2019; 19(5):353-364. <https://doi.org/10.1038/s41573-019-0050-3>.
- [45] Wilkinson MD, Dumontier M, Aalbersberg IJ, Appleton G, Axton M, Baak A, Blomberg N, Boiten JW, da Silva Santos LB, Bourne PE, Bouwman J, Brookes AJ, Clark T, Crosas M, Dillo I, Dumon O, Edmunds S, Evelo CT, Finkers R, Gonzalez-Beltran A, et al. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*. 2016; 3(1). <https://doi.org/10.1038/sdata.2016.18>.
- [46] Schmidt D, Scharf MM, Sydow D, Aßmann E, Marti-Solano M, Keul M, Volkamer A, Kolb P. Analyzing Kinase Similarity in Small Molecule and Protein Structural Space to Explore the Limits of Multi-Target Screening. *Molecules*. 2021; 26(3):629. <https://doi.org/10.3390/molecules26030629>.
- [47] conda-forge community, The conda-forge Project: Community-based Software Distribution Built on the conda Package Format and Ecosystem. Zenodo; 2015. <https://doi.org/10.5281/zenodo.4774216>.



### 3.2 Exploring Kinome-Wide Subpocket Fragment Spaces

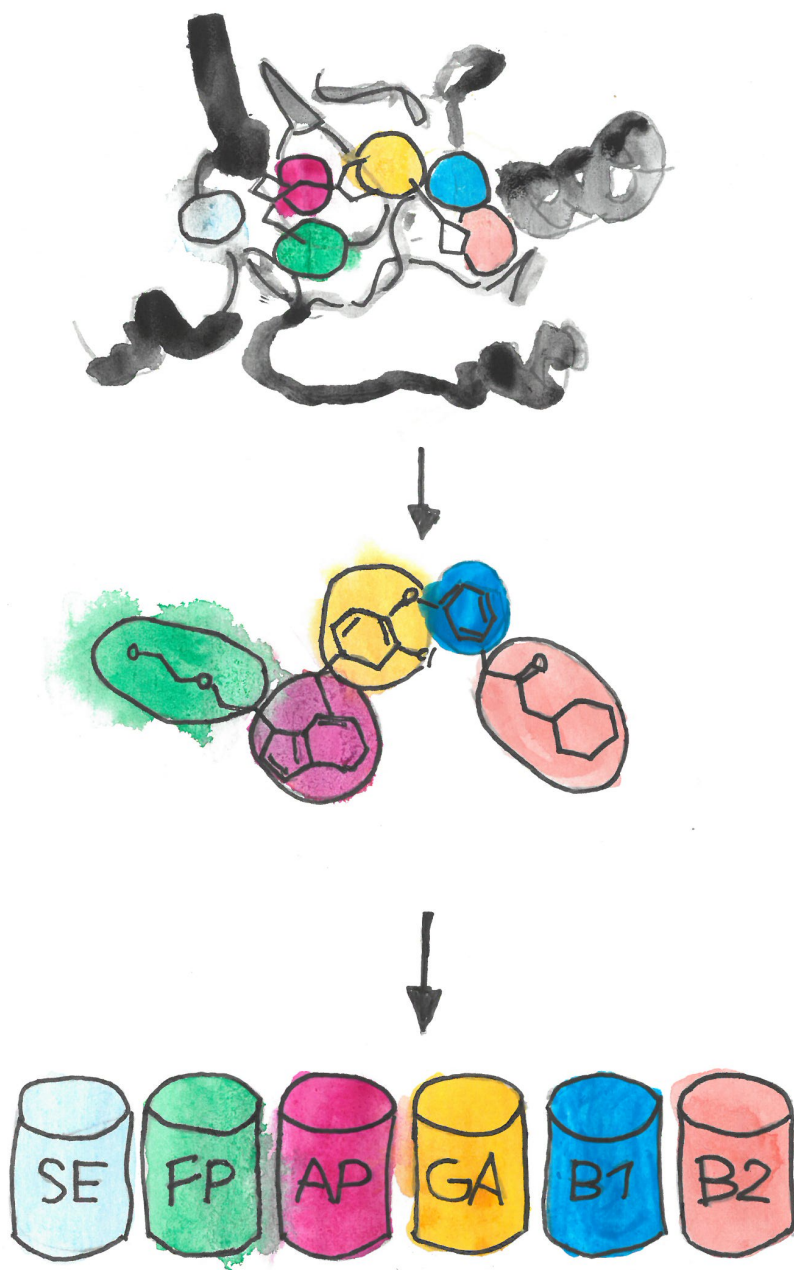

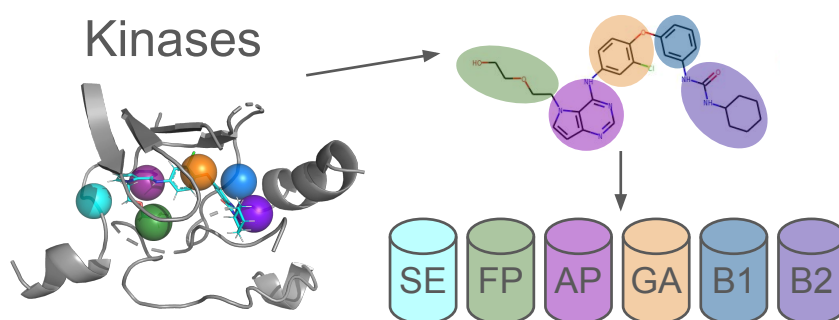


Figure 3.3: Exploring kinome-wide subpocket fragment spaces as illustrated by Ferdinand Krupp, adapted from Sydow et al. [143].

### 3.2.1 KinFragLib: Exploring the Kinase Inhibitor Space Using Subpocket-Focused Fragmentation and Recombination Publication E

The KinFragLib project as published in this article reports a subpocket-based fragmentation and recombination strategy for kinase-ligand complexes in the KLIFS database [63]. The structurally available kinase inhibitor space is decomposed based on the subpockets that they occupy, yielding a fragment space for each relevant kinase subpocket. The resulting fragment libraries are explored regarding their chemical space and are used to guide subpocket-informed recombination to generate novel kinase-focused molecules.

 <https://github.com/volkamerlab/kinfraglib>



Contribution:

#### Co-first author

Conceptualization (25%)

Data Curation (80%)

Formal Analysis (50%)

Investigation (50%)

Methodology (50%)

Software (50%)

Visualization (50%)

Writing — Original Draft (50%)

Writing — Review & Editing (50%)

Reprinted with permission from Sydow D\*, Schmiel P\*, Mortier J, Volkamer A. KinFragLib: Exploring the Kinase Inhibitor Space Using Subpocket-Focused Fragmentation and Recombination. *Journal of Chemical Information and Modeling*. **2020**; 60(12):6081-6094. 10.1021/acs.jcim.0c00839. (\*contributed equally)

Copyright © 2020 American Chemical Society.



# KinFragLib: Exploring the Kinase Inhibitor Space Using Subpocket-Focused Fragmentation and Recombination

Dominique Sydow,<sup>§</sup> Paula Schmiel,<sup>§</sup> Jérémie Mortier, and Andrea Volkamer\*Cite This: *J. Chem. Inf. Model.* 2020, 60, 6081–6094

Read Online

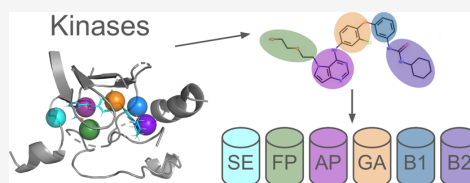
ACCESS |

Metrics &amp; More

Article Recommendations

Supporting Information

**ABSTRACT:** Protein kinases play a crucial role in many cell signaling processes, making them one of the most important families of drug targets. In this context, fragment-based drug design strategies have been successfully applied to develop novel kinase inhibitors. These strategies usually follow a knowledge-driven approach to optimize a focused set of fragments to a potent kinase inhibitor. Alternatively, KinFragLib explores and extends the chemical space of kinase inhibitors using data-driven fragmentation and recombination. The method builds on available structural kinome data from the KLIFS database for over 2500 kinase DFG-in structures cocrystallized with noncovalent kinase ligands. The computational fragmentation method splits the ligands into fragments with respect to their 3D proximity to six predefined functionally relevant subpocket centers. The resulting fragment library consists of six subpocket pools with over 7000 fragments, available at <https://github.com/volkamerlab/KinFragLib>. KinFragLib offers two main applications: on the one hand, in-depth analyses of the chemical space of known kinase inhibitors, subpocket characteristics, and connections, and on the other hand, subpocket-informed recombination of fragments to generate potential novel inhibitors. The latter showed that recombining only a subset of 624 representative fragments generated 6.7 million molecules. This combinatorial library contains, besides some known kinase inhibitors, more than 99% novel chemical matter compared to ChEMBL and 63% molecules compliant with Lipinski's rule of five.



## 1. INTRODUCTION

**1.1. Protein Kinases and Kinase Inhibitors.** Protein kinases constitute one of the largest protein families, with roughly 518 kinases encoded in the human genome.<sup>1</sup> Kinases share a catalytic domain for adenosine triphosphate (ATP) binding and are responsible for protein phosphorylation, a mechanism fundamental to most aspects of cell life. A variety of diseases, including cancer, inflammation, and autoimmune disorders, are associated with aberrant regulation of protein kinases. Thus, over the past 20 years, they have become one of the most important classes of drug targets, especially in the field of oncology.<sup>2–5</sup>

Protein kinases are generally divided into eukaryotic and atypical protein kinases. Eukaryotic kinases share a similar sequence and structure, whereas atypical kinases have biochemical kinase activity but lack sequence similarity to the typical kinase domain. Eukaryotic protein kinases can be further classified based on their sequence identity into eight main kinase groups: AGC, CAMK, CK1, CMGC, STE, TK, TKL, and Other.<sup>1,6</sup>

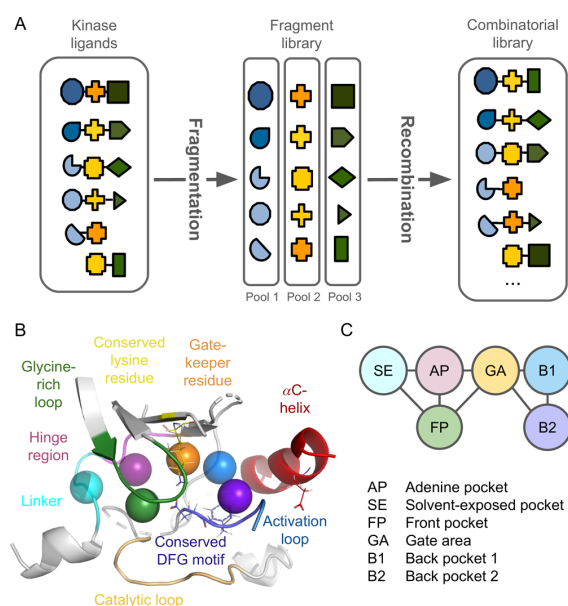
Protein kinase structures consist of two domains, the N- and C-lobes, connected *via* a hinge region. The majority of kinase inhibitors target the catalytic cleft between these lobes, which contains a highly conserved ATP-binding site. Based on over 1200 kinase-ligand crystal structures, van Linden *et al.*<sup>7</sup> have defined that the binding site comprises 85 residues and 19 well-defined regions and motifs, covering the front cleft, gate area, and back cleft (see Figure 1B). This information is listed

in the KLIFS<sup>7,8</sup> database and we provide the KLIFS numbering in brackets where applicable. The front cleft accommodates ATP and contains the hinge region (46–48), linker (49–52), glycine-rich loop (4–9), and catalytic loop (68–75). ATP's adenine group as well as most kinase inhibitors form hydrogen bonds to the hinge region. The gate area contains the conserved DFG motif (81–83), the conserved lysine residue K17 (17), and the gatekeeper residue (45), which is often used for inhibitor selectivity and precedes the hinge region. The back cleft contains among others the  $\alpha$ C-helix (20–30), including the conserved glutamine residue E24 (24), which forms the conserved K17-E24 salt bridge in the  $\alpha$ C-in (as opposed to the  $\alpha$ C-out) conformations. Furthermore, the DFG motif can undergo a significant conformational change, which results in an inactive state of the kinase (DFG-out instead of DFG-in conformation). This DFG-flip opens a hydrophobic region in the back cleft targeted by inhibitors stabilizing the inactive state.<sup>7,9</sup> The KLIFS database<sup>7,8</sup> has made this and further information about kinase structures, their bound ligands, and the interactions between them freely available.

Received: July 23, 2020

Published: November 6, 2020





**Figure 1.** (A) Simplified schematic depiction of the KinFragLib approach. Based on their location in the binding site, known kinase ligands are fragmented and placed into subpocket pools, which can then be used to generate a combinatorial library. (B) As an example, the kinase-binding site of EGFR (PDB:3W2S) is shown with important regions and the six defined subpocket centers as spheres. (C) Schematic depiction of the six subpockets and the predefined allowed connections between these subpockets. Colors of the subpockets are matching in (B) and (C).

Kinase inhibitors are classified by their binding modes.<sup>10</sup> Type I and II inhibitors occupy mainly the front cleft and form hydrogen bonds with the hinge region. Type I inhibitors bind to the active DFG-in conformation, type II inhibitors stabilize the inactive DFG-out conformation, and type II/III inhibitors extend into the back pockets in the DFG-in conformation. Allosteric inhibitors bind only next to the ATP-binding site (type III) or outside of the catalytic cleft (type IV). Type V inhibitors are bivalent, that is, binding different regions simultaneously. Type I–V inhibitors bind reversibly, whereas covalent inhibitors are classified as type VI.

**1.2. Fragmentation and Recombination of Kinase Inhibitors.** Fragment-based drug discovery (FBDD) has been successfully applied to develop novel and selective compounds, including kinase inhibitors.<sup>11,12</sup> Fragments are low-molecular-weight compounds targeting a specific subpocket within the active site of a protein. They usually bind to their target with weaker activity than traditional drug-like molecules but with a good binding efficiency, that is, a higher proportion of the atoms are interacting with the protein.<sup>13,14</sup>

In drug design, molecules can be viewed as combinations of multiple fragments. Growing, linking, and merging fragments is the essence of FBDD.<sup>15</sup> Fragments can be generated computationally by decomposing larger compounds. Clearly, the choice of the fragmentation technique will have an impact on the resulting fragment library. RECAP (REtrosynthetic Combinatorial Analysis Procedure)<sup>16</sup> and BRICS (Breaking of Retrosynthetically Interesting Chemical Substructures)<sup>17</sup> aim to cut only synthetically meaningful chemical bonds. eMolFrag<sup>18</sup> builds on top of BRICS to generate a set of

(larger) “bricks” and (smaller) connecting linkers. Alternatively, the BREED<sup>19</sup> algorithm immediately produces recombined molecules for proteins with similar pockets such as kinases. First, two structures, and thereby also their cocrystallized ligands, are superimposed. If bonds in each of the two ligands are in close proximity, the adjacent fragments are swapped, producing two recombined molecules.

Typically, FBDD starts with the screening of a fragment library to identify binders to specific targets, and only these hits are optimized into larger compounds by fragment linking or fragment growing. The screening step can be performed experimentally or *in silico*.<sup>20</sup> In the context of kinase inhibition, Urlich *et al.*<sup>21</sup> extracted ~6000 fragments with hinge-binding motifs from a kinase-unfocused library of 2.3 million compounds and docked them against 46 kinase structures to identify potential hinge binders. Fragment expansion of promising hits yielded a number of potent kinase inhibitors. Rachman *et al.*<sup>22</sup> reported a potent hinge-binding fragment, selected from a kinase-unfocused fragment library (624 fragments). The fragments were docked against the JAK2 ATP-binding site and filtered based on (i) pharmacophoric restraints (restrained docking) at the hinge region and (ii) interaction strength measured by the work necessary to break a defined hinge hydrogen bond (dynamic undocking). However, it is also possible to start off directly with a kinase-focused library of fragments that provide optimal interaction patterns with the ATP-binding site. Based on kinase-ligand crystal structures, Mukherjee *et al.*<sup>23</sup> extracted the smallest possible fragment with hydrogen bonds to the hinge region, yielding about 1000 fragments from 2250 ligands (Kinase Crystal Miner). Substructure searches for these fragments in large molecule databases supplied molecules with kinase binding potential. Vidović *et al.*<sup>24</sup> have used the aforementioned BREED strategy to reshuffle ligand functionalities between ligand pairs from 936 cocrystallized kinase ligands. This produced a total of ~150,000 recombined molecules, including ~26,000 lead-/drug-like molecules and ~300 known kinase inhibitors. Note that all the aforementioned approaches make use of 3D structural information and (except the last study) focus on hinge-binding fragments to be used for fragment expansion or substructure searches in compound libraries.

An alternative approach is to decompose a compound library based on kinase-focused criteria and to recombine the resulting fragments into a kinase-focused molecule library. Recently, Yang *et al.*<sup>25</sup> reported a ligand-based fragmentation and recombination strategy, which was applied on both a kinase-focused (194 kinase inhibitors from PKIDB<sup>26</sup>) and a kinase-unfocused library of ~4.6 million compounds. The fragments were assigned to three different fragment pools (core, connecting, and modifying fragments) representing three designated parts of a kinase inhibitor. Without using 3D structural information, fragments were assigned to the core fragment pool if a donor–acceptor hinge recognition pattern could be found. Enumerating different combinations of core–connector–modifying fragments yielded two virtual kinase-focused recombined molecule libraries (~500,000 and ~40 million recombined molecules), based on the aforementioned kinase-focused and kinase-unfocused input data.

**1.3. KinFragLib Methodology.** KinFragLib, which is introduced here, takes advantage of the large amount of structural data on kinase ligands from KLIFS<sup>7,8</sup> for subpocket-based fragmentation and recombination (Figure 1). Organizing kinase ligand fragments by subpockets enables not only a

Table 1. Dataset Filtering Steps during Preprocessing and Fragmentation

	Discarded structures	Remaining structures
Preprocessing Steps		
(A.1) KLIFS download (human, DFG-in, ligand within main pocket)	-	7370
(A.2) Discard atypical kinases	216	7154
(A.3) Choose best quality entry for each PDB	3775	3379
(A.4) Discard mol2 files not readable with RDKit	22	3357
(A.5) Discard substrates and substrate derivatives	429	2928
(A.6) Discard complexes with multiple ligands	17	2911
(A.7) Discard covalent inhibitors	110	2801
Additional Filtering Steps		
(B.1) Discard structures with important atoms missing	7	2794
(B.2) Discard ligands with large BRICS fragments	134	2660
(B.3) Discard ligands not occupying AP	100	2560
(B.4) Discard ligands with unwanted subpocket connections	7	2553

detailed subpocket-specific analysis of their fragment space but also a better understanding of the composition and spatial arrangement of ligands in reported kinase complexes. Moreover, this kinase-focused fragment library allows a subpocket-controlled recombination of fragments, revealing unexplored territory in the chemical space of kinase inhibitors.

## 2. DATA AND METHODS

The following sections describe the procedure for (2.1) collecting and preprocessing the dataset of kinase complex structures, (2.2) defining subpockets, (2.3) fragmenting each of the cocrystallized ligands in the dataset, (2.4) analyzing the fragment library, (2.5) recombining fragments, and (2.6) studying the combinatorial library.

**2.1. Data Collection and Preprocessing.** Structures of kinase-inhibitor complexes were collected from the KLIFS database<sup>7,8</sup> (downloaded on 2019-11-06), which offers superimposed kinase structures from the PDB<sup>27</sup> with 85 residues defined as kinase-binding sites. In KLIFS, several entries can exist for one PDB code because crystal structures were split into all existing alternate location models and all kinase domain-containing chains of heteromeric protein complexes. Each KLIFS entry comes with structural details, including a quality score (the higher the better), see Details S1 in the Supporting Information.

The structural data were preprocessed as described in the following steps (A.1–A.7) (see also Table 1): (A.1) Only human kinases in the DFG-in conformation and with a ligand lying within the main pocket (type I and II/2) were selected for download from the KLIFS website,<sup>28</sup> yielding a starting set of 7370 complex structures. (A.2) Atypical kinases were discarded because of the large difference in the binding site compared to eukaryotic kinases. (A.3) For each PDB code, the KLIFS entry (specified by PDB code, chain identifier, and alternative location) with the best quality score, or the first entry if there were multiple structures with an equal score, was extracted. (A.4) Mol2 files containing the binding site and the ligand of each chosen structure were loaded into RDKit.<sup>29</sup> Because of inconsistencies in the supported mol2 formats, some files were not readable and thus discarded. (A.5) Kinase structures in complex with adenine or any molecule containing a phosphate group or a ribose substructure were discarded (covering among others the PDB ligand IDs AMP, ADP, ATP, ACP, ANP, ADN, and ADE). These are kinase substrates or substrate analogues and were therefore not in the focus for the design of novel kinase inhibitors. (A.6) Some kinase structures

in the database were in complex with multiple disconnected molecules in the ATP-binding site. If one of these ligands was a substrate or substrate analogue, the complete structure was discarded because the ligand binding is not substrate-competitive. If multiple ligands consisting of more than 14 heavy atoms existed, the structure was also discarded. Otherwise, only the largest ligand was extracted. (A.7) Finally, as the current approach focuses on the discovery of reversible inhibitors, covalent ligands were also excluded, see Details S2 in the Supporting Information.

The dataset after preprocessing consists of 2801 kinase-ligand structures. Further filtering steps during the fragmentation procedure, as described in “2.3 Molecule Fragmentation”, result in a final dataset of 2553 complex structures (see B1–B4 in Table 1).

**2.2. Subpocket Definition and Allowed Connections.** In this work, the kinase-binding site was divided into six subpockets, which were selected based on their location and function in known kinase-inhibitor structures. Each subpocket is described by the geometric center of the C $\alpha$  atoms within the newly identified anchor residues chosen from the 85 binding site residues defined by KLIFS.<sup>7</sup> The respective subpocket-spanning anchor residues (Table 2) were selected

Table 2. Subpockets of the Kinase-Binding Site as Defined in This Work<sup>a</sup>

Subpocket	Abbreviation	Anchor residues	KLIFS correspondence
Adenine pocket	AP	15, 46, 51, 75	AP
Solvent-exposed pocket	SE	51	none
Front pocket	FP	10, 51, 72, 81	FP-I &; FP-II
Gate area	GA	17, 45, 81	BP-I-A &; BP-I-B
Back pocket I	B1	28, 38, 43, 81	BP-II-A, BP-II-in &;
Back pocket II	B2	18, 24, 70, 83	BP-II-B

<sup>a</sup>Each subpocket is described by the geometric center of its anchor residues' C $\alpha$  atoms (KLIFS residue numbering). For comparison, the corresponding KLIFS subpockets<sup>7</sup> are annotated (approximate manual assignment).

manually after visual inspection of several structures. The aim was to define a location that overlays with important parts of known kinase ligands and to provide a good distribution of centers within the pocket (see Figure 1B). Later, fragments are assigned to the closest subpocket, by measuring their distance to the subpocket centers, and stored in subpocket-specific

library pools (*subpocket pools*). In the following, the residue numbering refers to the numbering used in KLIFS.

**2.2.1. Subpocket Locations.** The *adenine pocket* (AP), located at the geometric center of the spanning residues 15, 46, 51, and 75, lies next to the hinge region. It is usually occupied by adenine in the ATP-bound state of the kinase and anchors substrates or other compounds by forming up to three hydrogen bonds. The *solvent-exposed pocket* (SE), defined here by the single residue 51 at the entrance of the binding site adjacent to AP, was also called the selectivity entrance by Zhao *et al.*,<sup>30</sup> as it shows diverse characteristics in different kinases and can therefore be used to achieve improved selectivity. The *front pocket* (FP), here represented by the geometric center of residues 10, 51, 72, and 81, is occupied by the ribose and phosphate groups of ATP and is partially solvent-exposed.<sup>9</sup> The *gate area* (GA) acts as a gate between the front cleft (containing AP, FP, and SE) and the back cleft. The GA pocket is defined by the region between the gatekeeper (residue 45), the conserved lysine (residue 17), and the aspartic acid (residue 81) in the DFG motif. The back cleft is split into two subpockets, *back pocket I and II* (B1 and B2), both lying next to the  $\alpha$ C-helix, spanned by residues 28, 38, 43, and 81, as well as 18, 24, 70, and 83, respectively. In addition to the six subpocket pools, a seventh pool X was created to hold fragments that cannot be assigned clearly to a subpocket because the distance to their closest subpocket center exceeds 8 Å. Selecting anchor residues and handling structures with missing anchor residues is described in Details S3 in the Supporting Information.

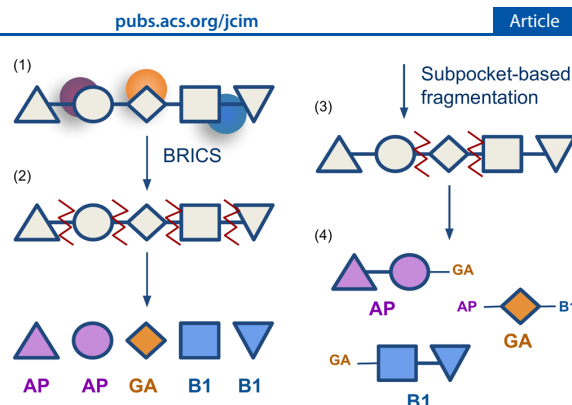
**2.2.2. Allowed Subpocket Connections.** In order to set up the fragment library, first, the connections between the above defined subpockets were investigated. After manual inspection of the typical structure of known kinase inhibitors (type I and II/2 only), eight allowed subpocket connections were identified as schematically depicted in Figure 1C. A first investigation of the generated fragments revealed that 95.2% of the molecules comply with this scheme. The remaining 4.8% ligands were handled as described in “2.3 Molecule Fragmentation”.

**2.3. Molecule Fragmentation.** A fragmentation algorithm was implemented to generate fragments from a given ligand in complex with a kinase structure, assign them to subpockets, and thereby populate the fragment library's subpocket pools (see Figure 1). Each kinase-ligand complex is processed successively in the following way (see Figure 2).

**2.3.1. Subpocket Center Calculation.** The aforementioned six subpocket centers are calculated for the binding site of the respective kinase structure (see “2.2 Subpocket Definition and Allowed Connections”).

**2.3.2. Initial BRICS Fragmentation.** The BRICS algorithm<sup>17</sup> was chosen for fragmentation. BRICS employs 16 rules to cleave bond types by taking the chemical environment and neighboring substructures into account. This ensures that structural features of organic compounds stay intact, increasing the chance of synthetic accessibility of the recombined fragments.

To determine the potential cleaving positions, the cocrystallized ligand of the structure in hand is submitted to an initial fragmentation step, using the RDKit implementation of the BRICS algorithm. Next, each of the resulting fragments needs to be assigned to a subpocket. Therefore, the geometric center of all atoms (including hydrogens) in the fragment, and its distance to all subpocket centers, is calculated. Then, the



**Figure 2.** Implemented fragmentation algorithm splits a given kinase ligand based on the subpockets that it occupies. (1) Subpocket centers within the kinase-binding site are calculated. (2) The BRICS algorithm is used for an initial fragmentation. The resulting BRICS fragments are then assigned to the closest subpocket. (3) Finally, the molecule is fragmented only at those bonds that separate fragments assigned to two different subpockets. (4) At the fragmented bonds, the information on the originally adjacent subpocket is stored.

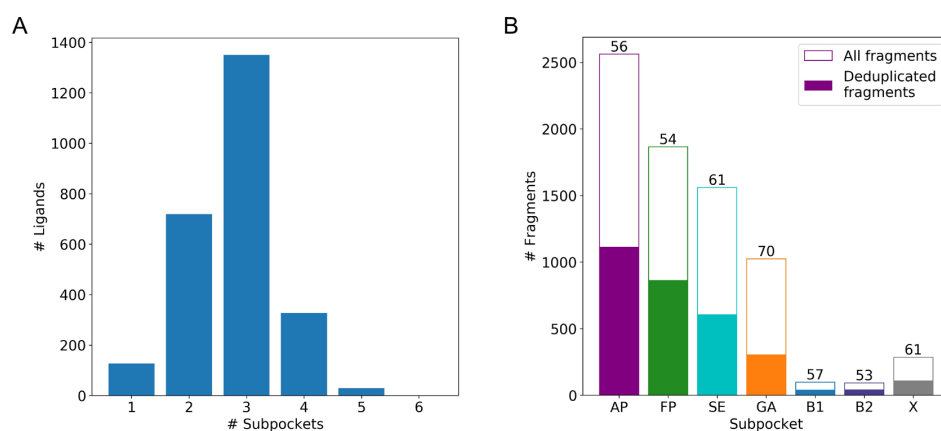
fragment is assigned to the subpocket with the closest subpocket center. However, if the closest subpocket to a fragment is more than 8 Å away, this fragment is considered as lying outside of the binding site and assigned to the outlier pool X. Note that the information on the BRICS environment type of each fragment is kept for later recombination.

Subsequently, the cleavage assignments are revised in order to avoid fragments that are too small in the final fragment library. For each fragment with less than three atoms, the neighboring fragments are checked. If all neighboring fragments are assigned to the same subpocket, nothing needs to be done because by default they will be merged in the next step. If the subpockets of the neighboring fragments differ, the current small fragment is assigned to the subpocket of the largest neighboring fragment. This procedure is repeated until all fragments with less than three atoms are reassigned.

Finally, for each bond between two BRICS fragments, the subpockets of the two fragments are compared. If the two subpockets differ, this bond is stored as a cleaving position for the final fragmentation.

**2.3.3. Final Subpocket-Based Fragmentation.** The original ligand is now fragmented only at bonds crossing two subpockets, while storing for each fragment the subpocket that it occupies. The subpockets of neighboring fragments are compared in order to detect unwanted subpocket connections (see section “2.2 Subpocket Definition and Allowed Connections”). (i) If a connection between subpockets FP and B1 or FP and B2 is detected, the distance of the FP fragment to the GA subpocket center is calculated. If this distance is smaller than 5 Å, this fragment is reassigned to GA instead (applied to only 15 cases). Else, the fragment in B1 or B2, respectively, is assigned to pool X. (ii) If any unwanted subpocket connection is still present after this procedure, the complete ligand is excluded from the fragment library.

**2.3.4. Fragment Information Storage.** Fragments are stored in one structure-data file (sdf) per subpocket. Together with the structural data, information about each (dummy) atom's subpocket annotation, BRICS details, original kinase affiliation, and more is stored to enable detailed analyses and



**Figure 3.** (A) Distribution of the number of subpockets (excluding pool X) occupied by the ligands. (B) Number of fragments and deduplicated fragments (fragments remaining after removal of duplicates) in each subpocket pool, with the percentage of duplicate fragments on top of each subpocket's bar.

later recombination of the fragments (see Details S4 in the Supporting Information).

**2.3.5. Summary of Removed Ligands during Fragmentation.** During the fragmentation procedure, some complexes were discarded because of the following reasons (Table 1): (B.1) A few kinase structures were missing required atom positions and thus their subpocket centers could not be calculated. (B.2) Some ligands, such as staurosporine, are not suitable for fragmentation, as they contain large, unfragmentable portions. Thus, structures with BRICS fragment(s) with more than 22 heavy atoms were discarded. (B.3) Ligands not occupying AP were excluded from the fragment library, as this work focuses on ligands targeting the ATP-binding site and most kinase inhibitors developed so far bind in the AP subpocket.<sup>7</sup> (B.4) Ligands displaying unwanted subpocket connections were discarded. Consequently, 2553 ligands remained and their fragments were included in the fragment library (available at <https://github.com/volkamerlab/KinFragLib>).

**2.4. Fragment Analysis.** The following paragraphs describe the different analyses that were performed on the fragment level.

**2.4.1. Deduplicated Fragments.** Several fragments were found more than once in a subpocket. Hence, a unified set was created for further analysis. First, fragments were simplified by replacing dummy atoms with hydrogens and removing all non-explicit hydrogens (*simplified fragments*). Second, fragments within one subpocket pool were deduplicated based on their canonical SMILES representation, that is, in the case of identical fragments, only one was kept (*deduplicated fragments*).

Fragment similarity was calculated to allow analyses of the fragment diversity within subpockets as well as within and across kinase groups. For subpocket-based analyses, fragments were deduplicated per subpocket and similarities between all pairwise fragment combinations per subpocket were calculated. To this end, the topological RDKit molecular fingerprint<sup>31</sup> was generated for each fragment and the Tanimoto similarity metric was applied. Self-comparisons of fragments were omitted.

To analyze similarities within and across kinase groups, fragments were categorized by subpocket and kinase group

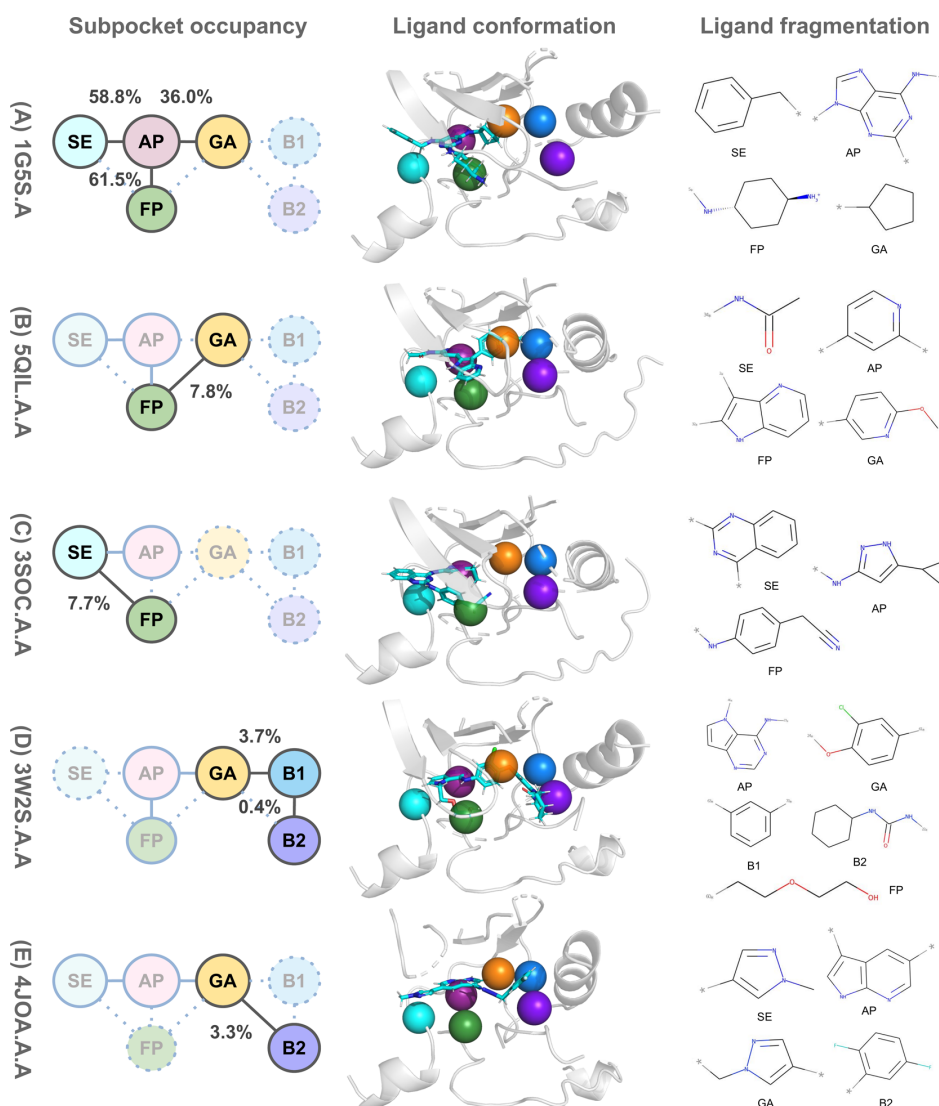
(according to the structure they were bound to) and deduplicated per category. For each subpocket (excluding pool X), similarities between all pairwise fragment combinations, within and across all kinase groups, were calculated as described in the previous paragraph.

**2.4.2. Common Fragment Motifs per Subpocket.** In order to identify the most common fragments in each subpocket (excluding pool X), the number of occurrences of each fragment was calculated before deduplication based on the *simplified fragments*. The 50 most common fragments in each subpocket were then clustered based on the Butina algorithm<sup>32</sup> using topological RDKit molecular fingerprints<sup>31</sup> and a distance threshold of 0.6. Note that subpockets B1 and B2 contain less than 50 deduplicated fragments, and thus, all fragments were chosen for clustering. Furthermore, representative fragments were extracted manually for each subpocket in order to provide a visual overview on chemical differences and overlaps between subpockets. Each selected fragment represents a variety of common fragments with similar scaffolds and R-groups.

**2.5. Fragment Recombination.** Novel molecules can be created by recombining fragments from the fragment library. For a proof-of-concept study, only a subset of the fragment library was used. The individual steps for data reduction and fragment recombination are explained in this section.

**2.5.1. Data Reduction.** The full fragment library contains 7486 fragments. In order to reduce the combinatorial library size and run time, a diverse subset of fragments was chosen. Fragments unsuitable for recombination, for example, unfragmented ligands, duplicates, or fragments in pool X, were discarded. Only fragments complying with the rule of three<sup>33</sup> (fragment-like) and fragments with hinge-like properties in case of the AP pool were kept. The Butina algorithm<sup>32</sup> was applied to cluster each subpocket's filtered fragments and the most common fragments were selected. The final reduced fragment library consists of 624 fragments (AP: 145, FP: 192, SE: 140, GA: 93, B1: 24, and B2: 30). A detailed description is given in Details S2 in the Supporting Information.

**2.5.2. Recombination Procedure.** All possible fragment combinations of the above described reduced set were enumerated, while preserving the original subpocket connections when connecting the fragmented bonds using the



**Figure 4.** Subpocket connectivity for example ligands in KLIFS ([PDB code].[chain].[alternate model]): (left) Subpockets and allowed connections with solid/dotted lines if present/not present in the example ligand, including the frequency of ligands showing the highlighted connection. (Middle) Ligand conformation in the example kinase structure, including subpocket centers (spheres). (Right) Ligand fragmentation with assigned subpockets and dummy atoms (gray).

subpocket-labeled dummy atoms. Recombination started from AP fragments only, while adding fragments from other subpockets consecutively and thereby excluding any recombined molecules not occupying AP. Fragments were combined by adding a bond between two atoms adjacent to dummy atoms, while removing the dummy atoms. Thereby, two fragments were connected *via* a new bond between two atoms if the following conditions were fulfilled. (i) The first fragment's dummy atom was associated with the same subpocket as the second fragment and vice versa. (ii) The BRICS environment types of the atoms to be connected were matching according to the BRICS rules,<sup>17</sup> in order to preserve synthetic accessibility. In addition, the bond type (single or double bond) between dummy atoms was preserved when connecting the fragments. (iii) While connecting the frag-

ments, it was ensured that the resulting molecule did not contain two fragments from the same subpocket, so that no subpocket is occupied twice.

Recombination was deemed complete if either the molecule had no dummy atoms left to another subpocket (excluding pool X), the molecule's remaining dummy atoms could not be replaced by any matching fragment, or the molecule consisted of four fragments. This upper limit of occupied subpockets was introduced because the majority of kinase ligands occupies only up to four subpockets (see Figure 3A) and molecules occupying more subpockets will most likely not fulfill the requirements of a drug-like molecule because of their size (*e.g.*, Lipinski's rule of five<sup>34</sup>). Finally, if the resulting recombined molecule contained any remaining dummy atoms, they were

replaced with hydrogen atoms. This recombination strategy produced over 6.7 million ligands based on 624 fragments.

**2.6. Recombined Molecule Analysis.** All molecules discussed in the following were standardized<sup>35</sup> and neutralized<sup>36</sup> using RDKit's `rdMolStandardize` module.<sup>37</sup>

The recombined molecules were compared against two sets of ligands: (i) the 542 ligands, from which the reduced set of 624 fragments originated (*reduced original ligands*), were searched for exact and substructure matches, and (ii) the ChEMBL database<sup>38</sup> was screened for exact matches and the most similar molecules. From the latter, all 1,870,461 molecules from the ChEMBL 25<sup>39</sup> dataset were downloaded. If an entry contained a mixture, the largest molecule was extracted. Duplicates were dropped (based on canonical SMILES) and only molecules with more than four heavy atoms were kept. Standardization resulted in 1,782,229 molecules, which were stored as InChI<sup>40</sup> strings to be used for the exact match search between the combinatorial library and ChEMBL. Furthermore, for each recombined ligand, a Tanimoto comparison based on topological RDKit fingerprints<sup>31</sup> yielded the most similar ChEMBL molecule.

**2.7. Used Software and Libraries.** The project was implemented in Python 3.6.8. RDKit<sup>29</sup> (2020.03.3) was used to perform most molecule-related calculations, matplotlib<sup>41</sup> (3.2.2) and seaborn<sup>42</sup> (0.10.1) to generate plots, and PyMol<sup>43</sup> (1.9.0.0) to visualize structures and subpocket centers.

### 3. RESULTS AND DISCUSSION

The main objective of this work has been to decompose kinase ligands with respect to 3D information and to assign each resulting fragment to the kinase subpocket it interacts with. Only kinase-ligand complex structures with molecules targeting the ATP-binding site in the DFG-in conformation were selected, such as type I and II/2 inhibitors, to reduce the conformational space of the kinase structures. After filtering the 7370 starting structures assembled from the KLIFS database, 2553 protein kinase-ligand structures were chosen for this study.

In a first step, inspired by the functional subpocket annotation in KLIFS, six functionally relevant subpockets were defined covering the ATP-binding site. Note that KLIFS specifies eight subpockets, some of which describe relatively small subpockets that were combined into one subpocket in KinFragLib. Subpockets, which are too small, are algorithmically less desired in this case because either very small fragments would be generated or large fragments would span over several of these small subpockets. Additionally, a solvent-exposed pocket (SE) was introduced in KinFragLib, a region of the binding site occupied by many kinase inhibitors (see subpocket definitions in Table 2).

In a second step, the cocrystallized kinase ligands were fragmented with respect to the subpockets that they occupy. This resulted in a kinase-focused *fragment library* with six subpocket pools (plus the pool X) and 7486 fragments, which is analyzed in depth in the following paragraphs.

In the last step, a subset of this kinase-focused fragment library was used to create a *combinatorial library* by enumerating all feasible fragment combinations. The potential of the combinatorial library is shown in comparison with the KLIFS ligands, from which the fragment subset originates, and with the ChEMBL database.

The generated fragment and combinatorial libraries alongside Jupyter<sup>44</sup> notebooks illustrating library usage as well as the

analyses of both libraries, as discussed in the following, are available on GitHub: <https://github.com/volkamerlab/KinFragLib>.

**3.1. Subpockets and Fragment Library.** The generated kinase-focused fragment library allows analyses of kinase-ligand interactions and exploration of the chemical space of kinase ligands. In total, 7486 fragments (7201 fragments without pool X) originating from 2553 cocrystallized ligands were generated by the fragmentation procedure. After subpocket-based deduplication, 2977 fragments remained (without pool X). The following analyses aim to provide a better understanding of kinase-inhibitor binding and may serve as a valuable starting point for the design of novel kinase inhibitors.

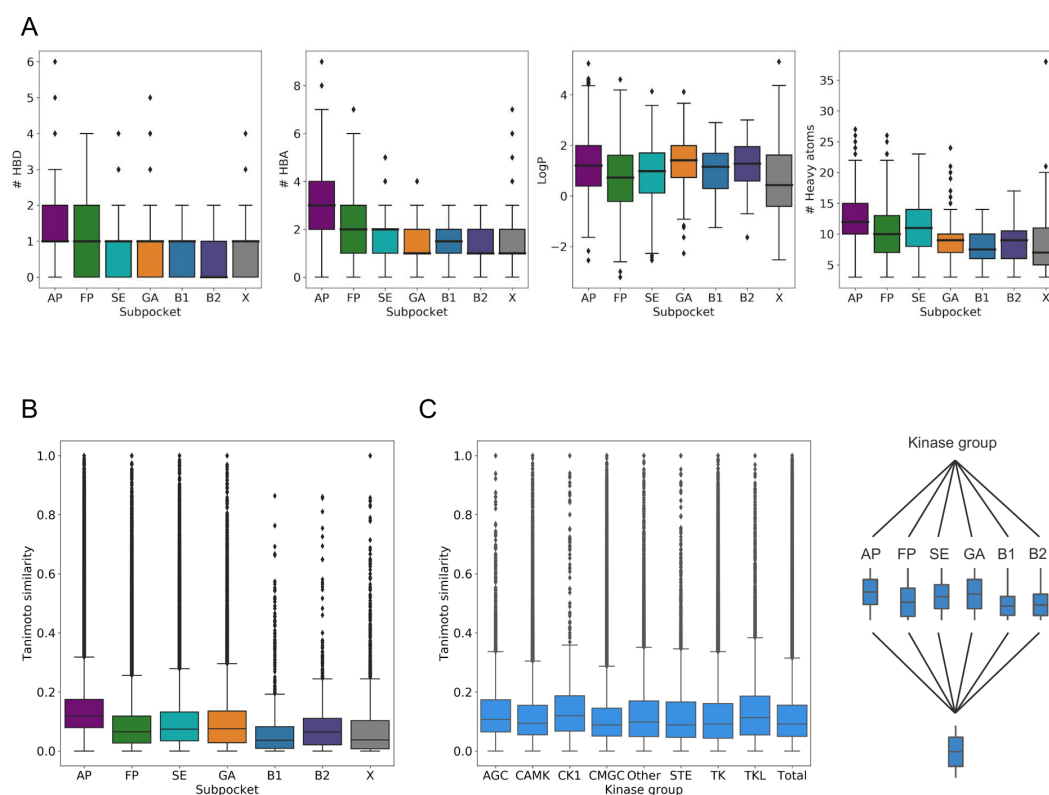
**3.1.1. Ligand Occupancy across Subpockets.** The compiled fragment library enables an in-depth analysis of the number of subpockets occupied by the original ligands (Figure 3A).

*Ligands occupying 2–4 subpockets.* The majority of ligands occupy two (28%) or three (53%) of the six subpockets. In another 13% of the cases, the ligand spans over four subpockets (examples of such can be seen in Figure 4A,B,E). This illustrates that kinase ligands usually do not fully exploit the available space in the kinase-binding site but target only specific subpockets.

*Ligands occupying 5–6 subpockets.* Very few ligands (1%) occupy five subpockets, and only one visits all six subpockets. For instance, in the ALK kinase structure PDB:4FNZ,<sup>45</sup> the cocrystallized ligand (NZF, ChEMBL2023556<sup>46</sup>) covers all six subpockets (see Figure S1A) and was indeed measured to be active on ALK ( $pIC_{50} = 7.2$ ) as well as IGF1R ( $pIC_{50} = 6.9$ ). An example of a ligand covering five subpockets is the cocrystallized active compound (W2R, ChEMBL2322330<sup>47</sup>) in the EGFR structure PDB:3W2S<sup>48</sup> ( $pIC_{50} = 8.2$ ), as shown in Figure 4D.

*Ligands occupying 1 subpocket.* Additionally, 127 ligands (5%) target only one subpocket and were left unfragmented during the fragmentation procedure. Because this study focuses on ligands covering the AP subpocket, all these unfragmented ligands are located in AP. They have an average number of 15 heavy atoms, which is higher than the average over all AP fragments (11 heavy atoms). As shown in Figure S1B–D, these molecules represent either (i) small fragment-like molecules or (ii) large rigid molecules that contain a large fraction of rings, which are difficult to split for most fragmentation algorithms. An example for the former group (i) is the series of halogenated pyrazoles that stem from a fragment-based approach for druggability assessment and hit generation,<sup>49</sup> see Figure S1B(1–8). The latter group (ii) contains complete drug-like molecules that could not be divided because BRICS fragmentation rules, see Figure S1D(1), or KinFragLib cleavage bond annotations did not apply. Or the molecules were simply too rigid to be fragmented, containing mainly fused ring systems with small decorations, such as quinalizarin (see Figure S1C(1,2) and Details S6 in the Supporting Information).

Note that the unfragmented ligands cannot be used in the recombination algorithm, because no attachment point resulting from the fragmentation could be assigned. This could be seen as a restriction in available chemical space of the current approach because each fragment-like molecule can be viewed as a potential starting point for fragment growing. Nevertheless, roughly 28% of the unfragmented ligands were found to be substructures of other original ligands. More than



**Figure 5.** (A) Chemical descriptor statistics for each subpocket pool. Calculated descriptors are the number of hydrogen-bond donors and acceptors (HBDS and HBAs), logP, and the number of heavy atoms, while excluding duplicate fragments. (B) Distribution of Tanimoto similarities between all pairwise fragment combinations per subpocket, while excluding duplicate fragments per subpocket. (C) Distribution of Tanimoto similarities between pairwise fragment combinations in each kinase group and across all kinase groups (total), while excluding duplicate fragments within each kinase group and subpocket as well as comparing only fragments within the same subpocket.

half of these unfragmented ligands are fragment-like (*i.e.*, fulfill the rule of three<sup>35</sup>). Thus, they are implicitly used in the introduced recombination approach. The remaining 72% of the unfragmented ligands are however not considered, a limitation which could be addressed by manually adding attachment points on relevant positions.

**3.1.2. Ligand Connectivity across Subpockets.** The fragmentation of existing kinase inhibitors yields an overview of how the fragments are arranged within the binding site and throughout the individual subpockets, revealing which subpockets are connected most frequently by kinase ligands.

**Disallowed subpocket connections.** As described in “Data and Methods”, a few design choices were made to only allow the subpocket connections as depicted in Figure 1C that were defined based on prior investigation of known kinase inhibitors. The majority (95.2%) of the analyzed molecules follow this scheme. Another 4.5% of molecules initially contained disallowed FP-B1 or FP-B2 connections, which could be rescued with additional fragmentation rules. The remaining molecules (0.3%) contained non-adjacent subpocket connections and were discarded in this analysis (see Details S7 in the Supporting Information).

**Subpocket connections and fragment arrangements.** The fragment connectivity of the cocrystallized ligands was analyzed to identify the typical layout of kinase inhibitors. Examples of ligands representing different subpocket con-

nections are illustrated in Figure 4. The central connections starting from AP are observed most often. The AP-FP connection is present most frequently in 61.5% of the analyzed ligands, closely followed by the AP-SE and the AP-GA connections with 58.8% and 36.0%, respectively (see Figure 4A). This agrees with the finding that subpocket pools AP, FP, SE, and GA contain the most fragments in descending order (Figure 3B). FP-GA and FP-SE connections also occur in more than 7% of the ligands each (see Figure 4B,C). Generally, the back pockets B1 and B2 are covered less often in the fragment set and they can only be reached through GA. Thus, the GA-B1 or GA-B2 connections appear only in 3.7% and 3.3% of the cases, respectively (see Figure 4D,E). B1-B2 connections are present in only 10 ligands (0.4%, see Figure 4D).

These findings seem to be in good agreement with the inhibitor binding modes reported in KLIFS (Table 5 in the original publication,<sup>7</sup> see also Table 2). The majority of ligands are described to be front cleft binders in both approaches, occupying mostly AP-GA and AP-FP subpockets (because SE is not defined in KLIFS, AP-SE and FP-SE connections are part of the KLIFS equivalent of the AP and FP subpockets). In contrast, back cleft binders describing ligands that occupy the back pockets (AP-GA-B1/2 combinations) occur by far less often. While the KLIFS binding mode annotation is based on kinase-ligand interaction fingerprints, the analysis reported



here shows that KinFragLib's automated subpocket-based procedure generates reasonable fragments.

**3.1.3. Fragment Occurrence per Subpocket.** The number of fragments per subpocket is reported in Figure 3B and Table S1. Containing 35.6% of the 7201 fragments (excluding pool X), AP is the most frequently occupied subpocket. Remember that by design, this study focuses on ligands covering the AP subpocket, and all ligands not occupying AP were discarded beforehand. Also note that AP contains 8 fragments more than the actual number of fragmented ligands, that is, 2553. This is possible because the fragmentation algorithm allows two not neighboring fragments of a ligand to occupy the same subpocket because not all ligands fit perfectly to the defined subpockets (this happens only rarely). The second most occupied subpocket is FP (25.9% of fragments), followed by SE (21.7%) and GA (14.2%). The back pockets B1 and B2 are occupied by only 2.6% of the fragments in total. According to this, known type I and II/2 kinase ligands mostly target the same subpockets as the kinase substrate ATP (AP and FP) to gain potency, followed by the neighboring subpockets such as GA, targeted to increase selectivity. In this dataset, the remote back pocket is targeted less frequently because of two reasons. First, 69% of the underlying kinase structures show the  $\alpha$ C-in conformation, limiting the available space for ligands in B1 and B2. Second, 73% of the front cleft binders target the  $\alpha$ C-in conformation, compared to only 25% in the case of the back cleft binders. Finally, pool X contains 285 additional fragments. These fragments were classified as lying outside of the main binding site or showing disallowed subpocket connections.

**3.1.4. Fragment Properties per Subpocket.** In the following, the fragment pools are analyzed with respect to duplicate fragments and physicochemical properties per subpocket.

**Duplicates.** On average, 59% of the fragments in each subpocket were present in more than one structure (referred to as duplicates). This can be explained by the traditional medicinal chemistry approach, studying a wide range of decorating groups around a shared molecular scaffold and thereby exploring structure-activity relationships. Such approaches can result in the crystallization of multiple analogs from the same series. However, this finding also highlights the limited chemical diversity of the known kinase inhibitor space (considering molecules with available crystal structures only). The highest relative number of duplicates was identified in GA (70%); for the other subpockets the values do not differ largely from the average of 59% (Figure 3B). The higher share of duplicates in GA could be explained by the generally smaller fragment size in this subpocket (compared to AP, FP, and SE, see Figure 5A).

**Physicochemical properties.** In order to identify particularities in the chemical space of the different subpocket pools, standard chemical descriptors were calculated. These include (i) hydrogen-bond donors and acceptors (HBDs and HBAs), (ii) logP values, and (iii) molecule size as in the number of heavy atoms. The distributions of these descriptors for each subpocket pool are displayed as box plots in Figure 5A, while excluding duplicates.

AP fragments generally have a higher number of HBDs and HBAs, as this part of the inhibitor usually forms hydrogen bonds to the hinge region and acts as an anchor to position the ligand.<sup>9</sup> The logP values vary widely in all subpocket pools. X, FP, and SE fragments have the lowest median logP, meaning they tend to be more hydrophilic. For SE, this can be explained

by the solvent exposure of this part of the kinase binding site. The same holds for FP, which is also partially solvent-exposed.<sup>9</sup> While the AP fragments usually do provide hydrogen bonds as anchors, they are often surrounded by a hydrophobic pocket, which could explain the need for lipophilic moieties in the fragment and thus a higher logP. Furthermore, AP, FP, and SE fragments tend to be larger in terms of the number of heavy atoms, with AP having the highest median value. Note that most of the outliers in AP refer to unfragmented ligands, as shown in Figure S1C,D, while outliers in FP mostly refer to large fragments that extend widely into the solvent.

This analysis reflects the general knowledge medicinal chemists have about kinase inhibitors: an HBD-HBA recognition motif is required for binding to the hinge region, the SE subpocket is used to attach functional groups that increase compound solubility, and the GA region accommodates small and hydrophobic moieties. This highlights the KinFragLib method's ability to automatically capture the chemical properties of kinase inhibitors.

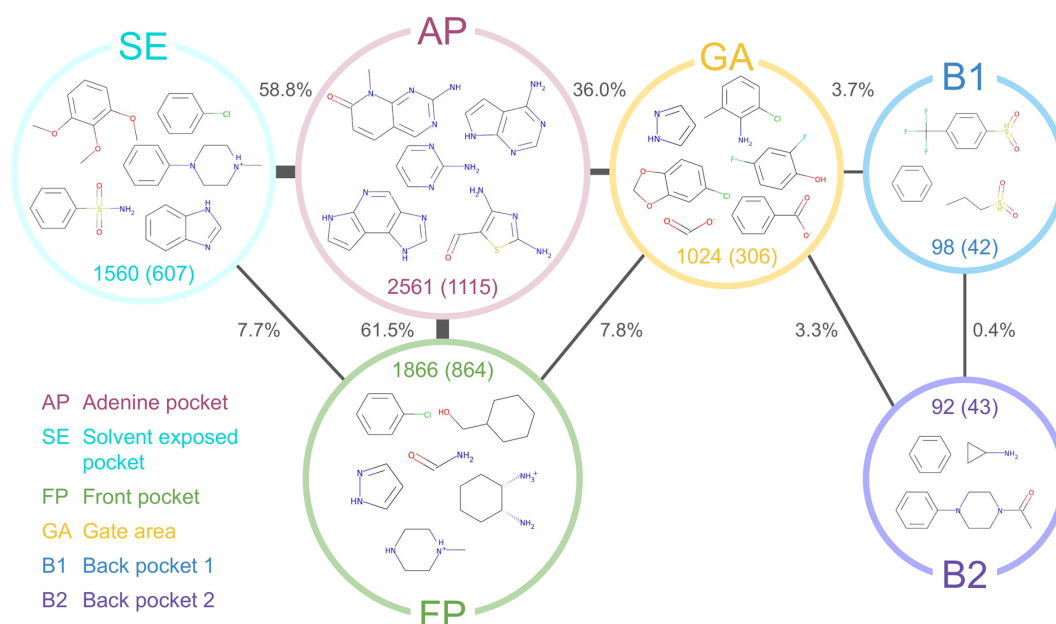
**3.1.5. Fragment Similarity per Subpocket.** In the following, the fragment similarity was analyzed within each subpocket to assess if certain subpockets are occupied by more similar fragments than others. Overall, the intra-subpocket fragment similarity does not differ largely between the subpockets and is generally rather low (Figure 5B, Table S1). The highest average intra-subpocket similarity was observed in AP with a mean of 0.14, the lowest similarities in B1 (0.07), B2 (0.09), and FP (0.09). A higher similarity in AP can be explained by the lower flexibility of this kinase region and the targeted design of chemical moieties interacting specifically with the hinge region. The low average similarity within FP might be observed because of the larger space around the FP center compared to the other subpockets, allowing a higher diversity in FP fragments. The low similarity in B1 and B2 is probably the result of the small amount of data available for these subpockets.

In general, this analysis indicates that the fragments in the subpocket pools have a high structural diversity (after deduplication), which underlines the potential of KinFragLib to generate novel chemical matter.

**3.1.6. Fragment Promiscuity.** Fragment promiscuity was addressed from two angles. Are fragments more similar within kinase groups than across kinase groups? If fragments are observed multiple times in the same subpocket pool, are the respective ligands cocrystallized with different kinases (or kinases from the same group)?

To address the first question, all fragments were grouped by subpockets (excluding pool X) and kinase groups. Within each of these subsets, fragments were deduplicated and similarities for all pairwise fragment combinations were calculated and pooled by kinase groups. This results in fragment similarities per kinase group, while in each kinase group, only fragments were compared that occupy the same subpocket. If fragments were indeed selective for specific kinase groups, a higher fragment similarity would be observed within kinase groups compared to across all kinase groups (*i.e.*, pooling all similarities from all subpockets). Nevertheless, no significant difference was observed (Figure 5C). This result indicates that the collected fragments are potentially useful for the design of an inhibitor of any target kinase.

To address the second question, all fragments were grouped by and deduplicated within subpockets (excluding pool X), while the number of duplicates was kept per deduplicated



**Figure 6.** Representative kinase ligand composition: the representative fragments (manual selection) of the most common fragments are shown per subpocket. The subpockets' circle size illustrates the number of fragments (number of deduplicated fragments in brackets) per subpocket. Fragment connections between subpockets are shown as lines, including the percentage of ligands showing each connection. The full list of the top 50 most common fragments per subpocket is shown in Figures S2–S7. Note that dummy atoms were replaced by hydrogen atoms.

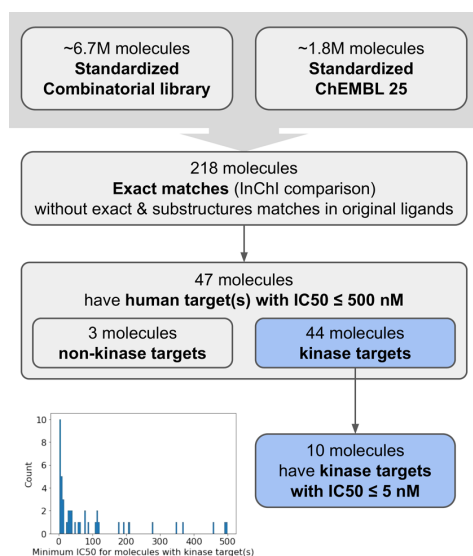
fragment: 67% of the fragments represent singletons (appear only once per subpocket) and 12% originate from different molecules that were bound to the exact same kinase and subpocket. One interpretation of this result can be that 79% of the collected fragments have the potential to be part of a molecule that specifically inhibits one kinase. This is in line with the arguments by Xing *et al.*<sup>50</sup> and Hu and Bajorath<sup>51</sup> after exploring kinase hinge-binding scaffolds. Another interpretation can be that 4 out of 5 fragments have never been explored on kinase targets from a different family. Using this information to create kinase-focused chemical matter could therefore be extremely useful. The remaining 21% of the fragments were bound to more than one kinase. More than three-quarters of this fragment set were also cocrystallized with kinases from more than one kinase group. This result supports the conclusion that fragments can be promiscuous, that is, identical fragments can interact with multiple different kinase targets. Instead, the combination of different fragments could be the key for kinase selectivity.

**3.1.7. Common Fragments and Motifs per Subpocket.** In order to illustrate the chemical nature of the fragments within each subpocket pool and highlight differences and similarities across them, representative fragments are shown in Figure 6.

The AP subpocket binds mainly heteroaromatic systems based on single or fused five- or six-membered rings, mostly showing the prominent donor-acceptor patterns for hinge binding. The SE subpocket is predominantly occupied by single aromatic rings, while the FP subpocket shows both single aromatic and non-aromatic rings with different substitutions. Both subpockets show residual groups rich in nitrogen, oxygen, and halogen. The GA subpocket binds mostly benzene rings with oxygen- and halogen-rich residual groups. Both GA and FP also accommodate smaller linear fragments, which are mostly terminal fragments, because a

large fraction of molecules are front pocket binders and thus do not extend further into the back pockets. For B1 and B2, much less data are available (about 90 vs 1000–2500 molecules per subpocket), and thus, the fragments are less representative for the chemical matter that could be accommodated by these pockets. The B1 subpocket pool contains many sulfonyl groups and is rich in halogen substitutions (*e.g.*, trifluoromethyl groups), whereas the B2 subpocket shows a quite diverse set of ligands. An overview of the 50 most common fragments per subpocket is shown in Figures S2–S7. The identified common fragments are in good agreement with the representative scaffolds reported for the different KLIFS subpockets by van Linden *et al.*<sup>7</sup> (Table 6 in the original publication).

In order to assess overlaps and differences in results from different approaches, hinge-binding fragments from the literature are compared to fragments from the hinge-equivalent subpocket in this study, that is, the AP subpocket. Xing *et al.*<sup>50</sup> and Mukherjee *et al.*<sup>23</sup> both report their 10 most common hinge scaffolds/fragments (Figures 1 and 7 in the original publications, respectively). Excluding adenine and staurosporine from the comparison, which were removed from this library, see Table 1 (A.5) and (B.2), all 8 fragments reported by Xing *et al.*<sup>50</sup> and 5 (out of 7) fragments reported by Mukherjee *et al.*<sup>23</sup> have exact matches in the AP subpocket pool reported in this work. When also considering highly similar (difference in one atom) AP fragments, all fragments from both studies are in the top 15 of the most common AP fragments in this study (Table S3). While both reported methods check for hydrogen bonding between the fragment and the hinge region in crystal structures, KinFragLib is able to retrieve hinge-contacting fragments without specifically searching for hinge contacts but by checking the position within the binding site. As a further comparison, Yang *et al.*<sup>25</sup> report 15



**Figure 7.** Number of exact matches (based on standardized InChI comparison) of recombined molecules in the ChEMBL 25 dataset, including the number of active molecules (activity is here defined as  $IC_{50} \leq 500$  nM). The histogram shows the  $IC_{50}$  values for those molecules that are active against kinases.

examples of hinge-binding fragments extracted using hinge-like donor–acceptor patterns from kinase inhibitors (Figure 5 in the original publication). More than half of these fragments are similar to fragments in KinFragLib’s top 21 AP fragments (only few exact matches), the remaining fragments were not substructures of KinFragLib’s original ligands and thus are not part of the fragment library.

**3.2. Recombined Molecules.** To exemplify the power of the combinatorial library, molecules were enumerated based on a reduced and diverse subset of the fragment library consisting of 624 fragments (see subsection “2.5 Fragment Recombination”). The recombination algorithm generated 6,752,232 molecules, of which only 31,595 molecules were duplicates, yielding 6,720,637 distinct molecules. This means that only 0.005% of the library contains duplicates, that is, equal molecules that were generated coincidentally from different fragment combinations.

**3.2.1. Recombined Original Ligands from KLIFS.** An important way to control the relevance of the generated chemical matter is to demonstrate this workflow’s ability to reconstruct the ligands from which the reduced set of 624 fragments originate (542 “reduced original ligands”): 35 recombined molecules have exact matches and 324 recombined molecules are substructures. Note that only a subset of fragments (624 out of 2977) was used for recombination, and thus, only a fraction of original ligands can be retrieved.

**3.2.2. Recombined ChEMBL Molecules.** The search for exact matches in ChEMBL<sup>38</sup> (1,782,229 molecules) revealed that only 298 of the over 6.7 million recombined molecules have already been described in ChEMBL. Only 218 matching molecules remain after removing the exact and substructure matches in the “reduced original ligands” used for the fragmentation. Consulting bioactivity data available in ChEMBL, 47 out of these 218 molecules have been shown to be active against at least one human target (activity is here

defined as  $IC_{50} \leq 500$  nM): 44 are active against kinases, two against cytochrome P450, and one against a voltage-gated ion channel. In total, 10 molecules show a high activity against kinases with an  $IC_{50} \leq 5$  nM (see Figure 7). More details on the ChEMBL IDs and molecular structures are shown in Table S4 and Figure S8. This shows strong evidence that the KinFragLib library contains molecules with a high chance of exhibiting kinase activity.

**3.2.3. Chemical Novelty (with Respect to the KLIFS Subset and ChEMBL).** Excluding the 359 original ligands (35 exact and 324 substructure matches in KLIFS) and the 218 exact matches in ChEMBL (without KLIFS matches), the recombination generated 6,720,058 novel molecules out of 6,720,637 deduplicated recombined molecules, that is, 99.99% of chemical matter with no precedent in ChEMBL nor the KLIFS subset. Furthermore, comparison of the recombined ligands with their most similar ChEMBL molecules revealed that the combinatorial library is not highly similar to the ChEMBL chemical space (mean similarity of 0.54 with a standard deviation of 0.07, see Figure S9).

At the same time, as discussed before, 35 original kinase inhibitors from KLIFS and 44 additional potent kinase inhibitors in ChEMBL could be recombined, while using only a subset of the fragment library. This indicates that KinFragLib can be used to generate large libraries of novel chemical matter, while being tailored for the design of kinase inhibitors.

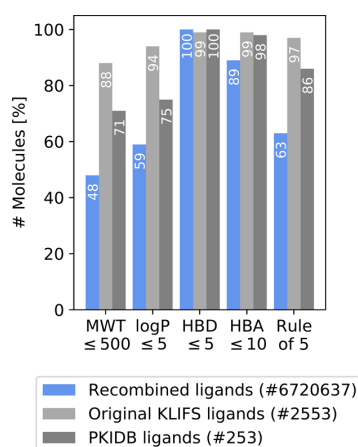
**3.2.4. Properties of Recombined Molecules.** The majority of the 6.7 million recombined molecules include fragments able to occupy four subpockets (90%), whereas the majority of original ligands are smaller and occupy three (53%) or two (28%) subpockets only. This is the consequence of a choice made in order to illustrate the power of exhaustive *in silico* library enumeration, that is, allowing fragments only to be linked until they occupy up to four subpockets. However, most importantly, the presented workflow allows for tailored library design that can easily be adapted to fulfill the requirements of a particular project.

While 86% of all kinase inhibitors in clinical trials (dataset from 2020-07-15 downloaded from PKIDB<sup>26</sup>) fulfill Lipinski’s rule of five, 63% of the combinatorial library (4.2 million molecules) complies with Lipinski’s rule of five (Figure 8), representing a large kinase-focused library to be used for virtual screening studies.

Note that only a subset of fragments was used to generate the recombined library. Thus, even larger libraries could be generated by taking into account all fragments identified in this study.

## 4. CONCLUSIONS

Kinases are one of the most studied protein families in medicinal chemistry, resulting in an amount of available data too large to be handled by a human brain. By computationally combining a precise cartography of the ATP-binding site and a tailored fragmentation method, KinFragLib allows inhibitors cocrystallized with a kinase in the DFG-in conformation to be read, fragmented, and organized by subpockets. The subsequent analysis of the chemical matter of the compiled fragments is in agreement with the general knowledge of medicinal chemists, identifying small and lipophilic fragments in the gatekeeper area, solubilizing fragments in the front pocket, and typical hinge binders for the adenine pocket. While this analysis is also in line with previous work conducted for



**Figure 8.** Lipinski's rule of five criteria applied to the 6.7 million molecules in the recombined library (blue) in comparison with the (i) 253 kinase inhibitors in clinical trials from PKIDB (light gray) and (ii) 2553 original KLIFS ligands used for building the kinase-focused fragment library (dark gray). The recombined molecules are overall molecules with a larger molecular weight (MWT) and more hydrogen-bond acceptors (HBAs), whereas the partition coefficient (logP) and the number of hydrogen-bond donors (HBDs) stay relatively the same. In total, 63% of the recombined library (4.2 million molecules) fulfills Lipinski's rule of five.

the hinge-binding fragments, this study provides for the first time a fragment library that is organized by subpocket and thus unveiling subpocket occupation and connection frequencies. It was found that chemically diverse fragments can bind the same subpocket. Furthermore, 79% of the identified fragments were only observed in one kinase structure, while the other 21% could bind the same subpocket of different kinase groups. This result indicates that a fragment binding one kinase subpocket is likely to bind the same region of other kinases. Therefore, the high chemical diversity of the generated fragment library is a rich source of inspiration for building novel kinase inhibitors. To investigate this possibility, a library of recombined fragments was enumerated *in silico* (using a diverse subset of the fragments only). The resulting virtual library, containing over 6.7 million molecules, was compared to the ChEMBL database (exact matches), indicating 99.99% of novel chemical matter. The rare exceptions of compounds with precedence in the literature include predominately known kinase inhibitors. These results clearly highlight the enormous potential of this fragment library for the design of novel kinase inhibitors.

The reported method focuses on two types of kinase inhibitors (type I and II/2); however, other libraries could be generated by fragmenting other kinase inhibitor types. Similarly, the same protocol could be applied to a more specific set of ligands, for example, to design a library of fragments specific of a kinase group, or a different dataset of ligand-kinase 3D structures. Finally, this workflow is also perfectly suited to support a fragment-growing approach after one novel fragment has been validated in a kinase subpocket.

## ■ ASSOCIATED CONTENT

### Supporting Information

The Supporting Information is available free of charge at <https://pubs.acs.org/doi/10.1021/acs.jcim.0c00839>.

**Data and Methods:** KLIFS data (Details S1); Structures with covalent ligands that were excluded from the KinFragLib dataset (Details S2); Exceptions for anchor residue definitions (Details S3); Fragment information storage (Details S4); Fragment data reduction before recombination (Details S5). **Fragment library:** Number of fragments, deduplicated fragments, singletons per subpocket pool and average pairwise Tanimoto similarity between fragments in each subpocket (Table S1); Ligands occupying 1 subpocket (Details S6); Ligands/fragments which show special subpocket occupancies (Figure S1); Disallowed subpocket connections/special cases (Details S7); Disallowed subpocket connections/special cases (Table S2); 50 most common fragments in the AP, FP, SE, GA, BL, and B2 subpockets (Figure S2-S7); Comparison of AP fragments reported in this study to hinge fragments from literature (Table S3). **Combinatorial library analysis:** ChEMBL details on recombined molecules with reported activity in ChEMBL against at least one kinase with  $IC_{50} \leq 5$  nM (Table S4); Structures of recombined molecules with reported activity in ChEMBL against at least one kinase with  $IC_{50} \leq 500$  nM (Figure S8); Distribution of Tanimoto similarities for recombined ligands each to their most similar molecule in ChEMBL (Figure S9) (PDF)

## ■ AUTHOR INFORMATION

### Corresponding Author

**Andrea Volkamer** – *In Silico Toxicology and Structural Bioinformatics, Institute of Physiology, Charité—Universitätsmedizin Berlin, 10117 Berlin, Germany;*  
 ● [orcid.org/0000-0002-3760-580X](https://orcid.org/0000-0002-3760-580X);  
 Email: [andrea.volkamer@charite.de](mailto:andrea.volkamer@charite.de)

### Authors

**Dominique Sydow** – *In Silico Toxicology and Structural Bioinformatics, Institute of Physiology, Charité—Universitätsmedizin Berlin, 10117 Berlin, Germany;*  
 ● [orcid.org/0000-0003-4205-8705](https://orcid.org/0000-0003-4205-8705)  
**Paula Schmiel** – *In Silico Toxicology and Structural Bioinformatics, Institute of Physiology, Charité—Universitätsmedizin Berlin, 10117 Berlin, Germany;*  
 ● [orcid.org/0000-0002-9671-837X](https://orcid.org/0000-0002-9671-837X)  
**Jérémie Mortier** – *Digital Technologies, Computational Molecular Design, Bayer AG, 13342 Berlin, Germany;*  
 ● [orcid.org/0000-0001-8707-3867](https://orcid.org/0000-0001-8707-3867)

Complete contact information is available at:

<https://pubs.acs.org/doi/10.1021/acs.jcim.0c00839>

### Author Contributions

<sup>§</sup>D.S. and P.S. contributed equally to this paper. Conceptualization, A.V. and J.M.; Methodology, A.V., P.S., and D.S.; Software, P.S. and D.S.; Formal Analysis, D.S. and P.S.; Investigation, P.S., D.S., J.M., and A.V.; Writing—Original Draft, Review and Editing, D.S., P.S., A.V., and J.M.; Visualization, P.S. and D.S.; Supervision, A.V., J.M., and D.S.; Funding Acquisition, A.V.

### Notes

The authors declare no competing financial interest. The generated fragments and recombined ligands, the full fragment and combinatorial library analysis, and a quick start

notebook on how to access the data are freely available at <https://github.com/volkamerlab/KinFragLib> (v1.0.0, DOI: <https://10.5281/zenodo.3956638>).

## ACKNOWLEDGMENTS

A.V. and D.S. gratefully acknowledge funding from the Deutsche Forschungsgemeinschaft (grant VO 2353/1-1). A.V. acknowledges support from the Bundesministerium für Bildung und Forschung (Grant 031A262C). The authors thank Albert Kooistra for many fruitful discussions about the fragmentation procedure and the ZEDAT HPC service (Freie Universität Berlin) for cluster time. The authors also thank Rebecca Hunt for her feedback on the manuscript.

## ABBREVIATIONS

ATP, adenosine triphosphate; GK, gatekeeper residue; FBDD, fragment-based drug discovery; BRICS, breaking of retrosynthetically interesting chemical substructures; KLIFS, kinase–ligand interaction fingerprints and structures database; PDB, RCSB Protein Data Bank; AP, adenine pocket; SE, solvent-exposed pocket; FP, front pocket; GA, gate area; B1, back pocket I; B2, back pocket II; SDF, structure-data file; MWT, molecular weight; HBD, hydrogen-bond donor; HBA, hydrogen-bond acceptor

## REFERENCES

- (1) Manning, G.; Whyte, D. B.; Martinez, R.; Hunter, T.; Sudarsanam, S. The Protein Kinase Complement of the Human Genome. *Science* **2002**, *298*, 1912–1934.
- (2) Cohen, P. Protein kinases - the major drug targets of the twenty-first century? *Nat. Rev. Drug Discovery* **2002**, *1*, 309.
- (3) Cohen, P.; Alessi, D. R. Kinase Drug Discovery - What's Next in the Field? *ACS Chem. Biol.* **2013**, *8*, 96–104.
- (4) Kooistra, A. J.; Volkamer, A. In *Annual Reports in Medicinal Chemistry*; Goodnow, R. A., Jr, Ed.; Elsevier, 2017; Vol. 50, pp 197–236.
- (5) Fabbro, D.; Cowan-Jacob, S. W.; Moebitz, H. Ten Things You Should Know About Protein Kinases: IUPHAR Review 14. *Br. J. Pharmacol.* **2015**, *172*, 2675–2700.
- (6) Sakkiath, S.; Ping Cao, G.; P Gupta, S.; Woo Lee, K. Overview of the Structure and Function of Protein Kinases. *Curr. Enzym. Inhib.* **2017**, *13*, 81–88.
- (7) van Linden, O. P. J.; Kooistra, A. J.; Leurs, R.; De Esch, I. J. P.; De Graaf, C. KLIFS: A Knowledge-Based Structural Database to Navigate Kinase-Ligand Interaction Space. *J. Med. Chem.* **2014**, *57*, 249–277.
- (8) Kooistra, A. J.; Kanev, G. K.; van Linden, O. P. J.; Leurs, R.; de Esch, I. J. P.; de Graaf, C. KLIFS: A Structural Kinase-Ligand Interaction Database. *Nucleic Acids Res.* **2015**, *44*, D365–D371.
- (9) Liao, J. J.-L. Molecular Recognition of Protein Kinase Binding Pockets for Design of Potent and Selective Kinase Inhibitors. *J. Med. Chem.* **2007**, *50*, 409–424.
- (10) Roskoski, R., Jr Classification of Small Molecule Protein Kinase Inhibitors Based Upon the Structures of their Drug-Enzyme Complexes. *Pharmacol. Res.* **2016**, *103*, 26–48.
- (11) Mortenson, P. N.; Berdini, V.; O'Reilly, M. Fragment-Based Approaches to the Discovery of Kinase Inhibitors. *Methods Enzymol.* **2014**, *548*, 69–92.
- (12) Erlanson, D. A.; De Esch, I. J. P.; Jahnke, W.; Johnson, C. N.; Mortenson, P. N. Fragment-to-Lead Medicinal Chemistry Publications in 2018. *J. Med. Chem.* **2020**, *63*, 4430–4444.
- (13) Rabal, O.; Urbano-Cuadrado, M.; Oyarzabal, J. Computational Medicinal Chemistry in Fragment-Based Drug Discovery: What, How and When. *Future Med. Chem.* **2011**, *3*, 95–134.
- (14) Mortier, J.; Rakers, C.; Frederick, R.; Wolber, G. Computational Tools for in Silico Fragment-Based Drug Design. *Curr. Top. Med. Chem.* **2012**, *12*, 1935–1943.
- (15) de Souza Neto, L. R.; Moreira-Filho, J. T.; Neves, B. J.; Maidana, R. L. B. R.; Guimarães, A. C. R.; Furnham, N.; Andrade, C. H.; Paes Silva, F., Jr In silico Strategies to Support Fragment-to-Lead Optimization in Drug Discovery. *Front. Chem.* **2020**, *8*, 93.
- (16) Lewell, X. Q.; Judd, D. B.; Watson, S. P.; Hann, M. M. RECAPRetrosynthetic Combinatorial Analysis Procedure: A Powerful New Technique for Identifying Privileged Molecular Fragments with Useful Applications in Combinatorial Chemistry. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 511–522.
- (17) Degen, J.; Wegscheid-Gerlach, C.; Zaliani, A.; Rarey, M. On the Art of Compiling and Using “Drug-Like” Chemical Fragment Spaces. *ChemMedChem* **2008**, *3*, 1503–1507.
- (18) Liu, T.; Naderi, M.; Alvin, C.; Mukhopadhyay, S.; Brylinski, M. Break Down in Order to Build Up: Decomposing Small Molecules for Fragment-Based Drug Design with eMolFrag. *J. Chem. Inf. Model.* **2017**, *57*, 627–631.
- (19) Pierce, A. C.; Rao, G.; Bemis, G. W. BREED: Generating Novel Inhibitors through Hybridization of Known Ligands. Application to CDK2, P38, and HIV Protease. *J. Med. Chem.* **2004**, *47*, 2768–2775.
- (20) Lamoree, B.; Hubbard, R. E. Current Perspectives in Fragment-Based Lead Discovery (FBLD). *Essays Biochem.* **2017**, *61*, 453–464.
- (21) Urich, R.; Wishart, G.; Kiczun, M.; Richters, A.; Tidten-Luksch, N.; Rauh, D.; Sherborne, B.; Wyatt, P. G.; Brenk, R. De Novo Design of Protein Kinase Inhibitors by in Silico Identification of Hinge Region-Binding Fragments. *ACS Chem. Biol.* **2013**, *8*, 1044–1052.
- (22) Rachman, M.; Bajusz, D.; Hetényi, A.; Scarpino, A.; Merő, B.; Egyed, A.; Buday, L.; Barril, X.; Keserű, G. M. Discovery of a Novel Kinase Hinge Binder Fragment by Dynamic Undocking. *RSC Med. Chem.* **2020**, *11*, 552–558.
- (23) Mukherjee, P.; Bentzien, J.; Bosanac, T.; Mao, W.; Burke, M.; Muegge, I. Kinase Crystal Miner: A Powerful Approach to Repurposing 3D Hinge Binding Fragments and Its Application to Finding Novel Bruton Tyrosine Kinase Inhibitors. *J. Chem. Inf. Model.* **2017**, *57*, 2152–2160.
- (24) Vidović, D.; Muskal, S. M.; Schürer, S. C. Novel Kinase Inhibitors by Reshuffling Ligand Functionalities Across the Human Kinome. *J. Chem. Inf. Model.* **2012**, *52*, 3107–3115.
- (25) Yang, Y.; Zhang, Y.; Hua, Y.; Chen, X.; Fan, Y.; Wang, Y.; Liang, L.; Deng, C.; Lu, T.; Chen, Y.; Liu, H. In Silico Design and Analysis of a Kinase-Focused Combinatorial Library Considering Diversity and Quality. *J. Chem. Inf. Model.* **2020**, *60*, 92–107.
- (26) Carles, F.; Bourg, S.; Meyer, C.; Bonnet, P. PKIDB: A Curated, Annotated and Updated Database of Protein Kinase Inhibitors in Clinical Trials. *Molecules* **2018**, *23*, 908.
- (27) Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. The Protein Data Bank. *Nucleic Acids Res.* **2000**, *28*, 235–242.
- (28) Division of Medicinal Chemistry, Vrije Universiteit Amsterdam. KLIFS—Kinase-Ligand Interaction Fingerprints and Structures Database. <https://klifs.vu-compmedchem.nl/> (accessed Nov 06, 2019).
- (29) RDKit. RDKit Version 2020.03.3. 2018, <http://www.rdkit.org> (accessed April 05, 2020).
- (30) Zhao, Z.; Xie, L.; Xie, L.; Bourne, P. E. Delineation of Polypharmacology Across the Human Structural Kinome Using a Functional Site Interaction Fingerprint Approach. *J. Med. Chem.* **2016**, *59*, 4326–4341.
- (31) RDKit. RDKit Fingerprint Version 2020.03.3. <https://www.rdkit.org/docs/source/rdkit.Chem.rdfingerprintgenerator.html> (accessed April 05, 2020).
- (32) Butina, D. Unsupervised Data Base Clustering Based on Daylight's Fingerprint and Tanimoto Similarity: A Fast and Automated Way To Cluster Small and Large Data Sets. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 747–750.
- (33) Congreve, M.; Carr, R.; Murray, C.; Jhoti, H. A “Rule of Three” for fragment-based lead discovery? *Today* **2003**, *8*, 876–877.

- (34) Lipinski, C. A.; Lombardo, F.; Dominy, B. W.; Feeney, P. J. Experimental and Computational Approaches to Estimate Solubility and Permeability in Drug Discovery and Development Settings. *Adv. Drug Deliv. Rev.* **1997**, *23*, 3–25.
- (35) RDKit. rdkit.Chem.MolStandardize.rdmolstandardize.standardizeSmiles Version 2020.03.3. <https://www.rdkit.org/docs/source/rdkit.Chem.MolStandardize.rdmolstandardize.html> (accessed April 05, 2020).
- (36) RDKit. rdkit.Chem.MolStandardize.rdmolstandardize.uncharge.uncharge Version 2020.03.3. <https://www.rdkit.org/docs/source/rdkit.Chem.MolStandardize.rdmolstandardize.html> (accessed April 05, 2020).
- (37) RDKit. rdkit.Chem.MolStandardize.rdmolstandardize Version 2020.03.3. <https://www.rdkit.org/docs/source/rdkit.Chem.MolStandardize.rdmolstandardize.html> (accessed April 05, 2020).
- (38) Gaulton, A.; Hersey, A.; Nowotka, M.; Bento, A. P.; Chambers, J.; Mendez, D.; Motow, P.; Atkinson, F.; Bellis, L. J.; Cibrián-Uhalte, E.; Davies, M.; Dedman, N.; Karlsson, A.; Magariños, M. P.; Overington, J. P.; Papadatos, G.; Smit, I.; Leach, A. R. The ChEMBL Database in 2017. *Nucleic Acids Res.* **2017**, *45*, D945–D954.
- (39) ChEMBL. ChEMBL25 download. <http://doi.org/10.6019/CHEMBL.database.25> (accessed July 04, 2019).
- (40) Heller, S. R.; McNaught, A.; Pletnev, I.; Stein, S.; Tchekhovskoi, D. InChI, the IUPAC International Chemical Identifier. *J. Cheminf.* **2015**, *7*, 23.
- (41) Hunter, J. D. Matplotlib: A 2D Graphics Environment. *Comput. Sci. Eng.* **2007**, *9*, 90–95.
- (42) Seaborn. Seaborn v0.9.0. 2018, <https://seaborn.pydata.org/> (accessed April 05, 2020).
- (43) DeLano, W. L. PyMol: An Open-Source Molecular Graphics Tool (Version 1.9). *CCP4 Newsletter on Protein Crystallography*; 2002; Vol. 40, pp 82–92.
- (44) Kluyver, T.; Ragan-Kelley, B.; Pérez, F.; Granger, B.; Bussonnier, M.; Frederic, J.; Kelley, K.; Hamrick, J.; Grout, J.; Corlay, S.; Ivanov, P.; Avila, D.; Abdalla, S.; Willing, C.; Jupyter Development Team. *Positioning and Power in Academic Publishing: Players, Agents and Agendas*; Loizides, F., Schmidt, B., Eds.; IOS Press: Amsterdam, The Netherlands, 2016; pp 87–90.
- (45) Epstein, L. F.; Chen, H.; Emkey, R.; Whittington, D. A. The R1275Q Neuroblastoma Mutant and Certain ATP-competitive Inhibitors Stabilize Alternative Activation Loop Conformations of Anaplastic Lymphoma Kinase. *J. Biol. Chem.* **2012**, *287*, 37447–37457.
- (46) ChEMBL. Compound ID CHEMBL2023556. [https://www.ebi.ac.uk/chembl/compound\\_report\\_card/CHEMBL2023556/](https://www.ebi.ac.uk/chembl/compound_report_card/CHEMBL2023556/) (accessed March 20, 2020).
- (47) ChEMBL. Compound ID CHEMBL2322330. [https://www.ebi.ac.uk/chembl/compound\\_report\\_card/CHEMBL2322330/](https://www.ebi.ac.uk/chembl/compound_report_card/CHEMBL2322330/) (accessed March 20, 2020).
- (48) Sogabe, S.; Kawakita, Y.; Igaki, S.; Iwata, H.; Miki, H.; Cary, D. R.; Takagi, T.; Takagi, S.; Ohta, Y.; Ishikawa, T. Structure-Based Approach for the Discovery of Pyrrolo[3,2-d]pyrimidine-Based EGFR T790M/L858R Mutant Inhibitors. *ACS Med. Chem. Lett.* **2013**, *4*, 201–205.
- (49) Wood, D. J.; Lopez-Fernandez, J. D.; Knight, L. E.; Al-Khawaldeh, I.; Gai, C.; Lin, S.; Martin, M. P.; Miller, D. C.; Cano, C.; Endicott, J. A.; Hardcastle, I. R.; Noble, M. E. M.; Waring, M. J. FragLites-Minimal, Halogenated Fragments Displaying Pharmacophore Doublets. An Efficient Approach to Druggability Assessment and Hit Generation. *J. Med. Chem.* **2019**, *62*, 3741–3752.
- (50) Xing, L.; Klug-Mcleod, J.; Rai, B.; Lunney, E. A. Kinase Hinge Binding Scaffolds and Their Hydrogen Bond Patterns. *Bioorg. Med. Chem.* **2015**, *23*, 6520–6527.
- (51) Hu, Y.; Bajorath, J. Exploring the Scaffold Universe of Kinase Inhibitors. *J. Med. Chem.* **2014**, *58*, 315–332.

Supporting Information:  
KinFragLib: Exploring the Kinase Inhibitor  
Space Using Subpocket-Focused  
Fragmentation and Recombination

Dominique Sydow,<sup>†,¶</sup> Paula Schmiel,<sup>†,¶</sup> Jérémie Mortier,<sup>‡</sup> and Andrea Volkamer<sup>\*,†</sup>

*†In Silico Toxicology and Structural Bioinformatics, Institute of Physiology, Charité -  
Universitätsmedizin Berlin, Charitéplatz 1, 10117 Berlin, Germany*

*‡Bayer AG, Digital Technologies, Computational Molecular Design, 13342 Berlin, Germany*

*¶Authors contributed equally to this paper.*

E-mail: andrea.volkamer@charite.de

## Data and Methods

### Details S1: KLIFS data

Each KLIFS entry comes with the following details: species, kinase name, kinase group, PDB code of the complex and the ligand, sequence alignment of the 85 binding site residues, DFG conformation (in, out, or out-like), ligand position (within or outside the main pocket), and KLIFS quality score. The latter ranges from 0 (bad) to 10 (flawless) and describes the quality of the alignment as well as structure based on each structure's alignment to a reference as well as its number of missing residues and atoms, respectively.

### Details S2: Structures with covalent ligands

The KLIFS dataset was downloaded on 2019-11-06 from the KLIFS website's search page with the following restrictions: "Organism" = "Human", "DFG conformation" = "IN", and "Ligand-bound" = "Within main pocket".

Covalent ligands were identified by downloading the PDB file corresponding to the KLIFS structure and checking the CONECT records for any connection between the kinase and the ligand. Note that after personal communication with A. Kooistra,<sup>1</sup> two PDB entries were excluded manually (2clx, 4cfn), since the ligand was found to be not covalently bound; and three PDB entries (4d9t, 4hct, 4kio) were added, because the ligands bind covalently but the CONECT entries were missing.

The following structures (<complex PDB>.<ligand PDB>.<chain>) were excluded from the KLIFS dataset because they contain covalent ligands (110 in total):

4yhf.4C9.B, 5j87.N42.D, 5p9j.8E8.A, 5p9k.7G8.A, 5p9l.7G9.A, 5p9m.7GB.A, 6di1.GJD.A, 6di5.GJ7.A, 6di9.GJJ.A, 6j6m.BA0.A, 6n9p.KHD.A, 6o8i.LTJ.A, 5cyi.55S.A, 5oo1.9Z2.A, 5oo3.9ZB.A, 5osm.AEQ.A, 5acb.5I1.C, 2j5f.DJK.A, 2jiv.HKI.A, 3ika.0UN.A, 3w2p.W2P.A, 3w2q.HKI.A, 4g5j.0WN.A, 4g5p.0WN.A, 4i24.1C9.A, 4li5.1WY.A, 4ll0.YUN.A, 4lqm.DJK.A, 4lrm.YUN.D, 4r5s.FI3.A, 4wd5.3LH.A, 5fed.5X4.A, 5fee.5X4.A, 5feq.5XH.A, 5gmp.F62.A,



5gnk.80U.A, 5gty.816.A, 5gtz.81C.A, 5j9y.6HL.A, 5j9z.6HJ.A, 5xdk.8JC.A, 5xdl.8JC.A, 5y25.8LU.A, 5y9t.8RC.A, 5yu9.1E8.C, 6d8e.FZP.A, 5mjb.7O3.B, 5l6o.6P6.A, 5l6p.6P8.A, 2r4b.GW7.B, 6ges.6H3.A, 4zzm.CQ6.A, 4zzo.CQ3.A, 5lcl.6TS.A, 5lck.6TT.A, 6g54.6H3.A, 5vnd.9ES.B, 6mzw.TZ0.A, 6nvl.XL6.A, 6p68.O1Y.C, 6p69.O21.A, 4qqc.37O.A, 4xcu.40M.A, 5nud.99K.A, 5nwz.9CT.B, 6iuo.AWX.A, 6jpp.FGF.A, 6nvg.XL8.A, 6nvh.XL6.A, 6nvi.XL7.A, 6nvj.XL5.A, 6nvk.XL9.A, 6h0u.FKB.A, 3t9t.IAQ.A, 4hct.18R.A, 4hcu.13L.A, 4hcv.13J.A, 4kio.G5K.C, 4qps.37Q.C, 4v0g.G9B.A, 4z16.4LH.D, 5lwn.79S.A, 5toz.7H4.A, 5tts.7KU.A, 5ttu.7KV.A, 5ttv.7KX.A, 5wfi.9Z4.A, 6da4.G4V.A, 6db4.G4Y.A, 6dud.HB4.A, 3v6r.CQQ.B, 3v6s.0F0.B, 4x21.3WH.B, 5z1d.95U.A, 6ib0.H8Z.A, 6ib2.862.A, 6qft.J0B.A, 6qg4.J0E.A, 6qg7.6HL.A, 6qho.J3H.A, 6qhr.J3N.A, 3pwy.SYP.A, 4d9t.0JG.A, 4d9u.0JH.A, 4jg6.1LB.A, 4jg7.1LC.A, 4jg8.1LE.A, 6ate.6H3.A, 6e6e.HVY.B, 4gs6.1FM.A

### Details S3: Exceptions for anchor residue definitions

The definition of the 85 binding site residues in the KLIFS database is based on a multiple sequence alignment, which can have gaps. It was therefore avoided to set residues with a high gap rate among the structures as anchor residue. Furthermore, some coordinates of an amino acid or a single atom may be missing because they could not be resolved by crystallography. If the coordinates of an anchor residue's  $C\alpha$  atom was missing, the following procedure was applied: If possible, the coordinates were replaced with the geometric center of the two neighboring residues'  $C\alpha$  atoms. If one of those was absent as well, the coordinates of the other neighboring residue were used instead. If both adjacent  $C\alpha$  atoms were missing, the structure was discarded.

### Details S4: Fragment information storage

Fragment atoms are labeled with the name of the subpockets that they occupy. Each original attachment point of each fragment is stored as *dummy atom* and the subpocket of the former adjacent fragment is stored as a property. This enables retracing of the subpocket that the

adjacent fragment was targeting in the original ligand (needed for the later recombination). Fragments are stored in structure-data files (sdf), using one file for each subpocket pool as well as pool X. In addition to the structural information (3D coordinates, elements, and bonds), the following data are stored for each fragment: (i) the PDB code of the original kinase-ligand complex and the name of the ligand itself, (ii) the chain and alternate model of this complex in KLIFS, (iii) the kinase, kinase family, and kinase group, (iv) the subpocket of each atom, including dummy atoms, and (v) the BRICS environment type for each atom.

### **Details S5: Data reduction**

The full fragment library contains 7,486 fragments. In order to reduce the combinatorial library size and run time, a diverse subset of fragments was chosen. (i) All fragments that are not suitable for recombination were removed, i.e. duplicates, fragments in pool X, fragments without dummy atoms (unfragmented ligands), and fragments with dummy atoms only connecting to pool X. Furthermore, only fragments complying with the rule of three,<sup>2</sup> a filter for fragment-likeness, and hinge-like AP fragments were kept. The latter filter checks for at least one hydrogen bond donor or acceptor in the AP fragment, together with at least one aliphatic or aromatic ring. The filtering steps in (i) result in 2,029 fragments. (ii) Per subpocket, a diverse set of fragments was selected for recombination to avoid enumerating highly similar fragments. The Butina algorithm<sup>3</sup> was applied to cluster each subpocket's filtered fragments using topological RDKit molecular fingerprints<sup>4</sup> and a distance threshold of 0.6. Per cluster, the most common fragments were selected. The larger the cluster the more fragments were chosen (one fragment per 10 cluster members, whereby clusters with less than 10 fragments are represented with one fragment). The final reduced fragment library consists of 624 fragments (AP: 145, FP: 192, SE: 140, GA: 93, B1: 24, and B2: 30).

## Fragment library analysis

Find here supporting information regarding the fragmentation library analysis.

Table S1: Number of fragments, deduplicated fragments, and singletons (fragments occurring only once) per subpocket pool in the fragment library, and average pairwise Tanimoto similarity between fragments in each subpocket.

<b>Subpocket</b>	<b>All fragments</b>	<b>Deduplicated fragments</b>	<b>Singletons</b>	<b>Average similarity</b>
AP	2,561	1 115	762	0.139
FP	1,866	864	607	0.089
SE	1,560	607	397	0.103
GA	1,024	306	181	0.105
B1	98	42	29	0.074
B2	92	43	27	0.089
Total	7,201	3,011	2,003	

**Details S6: Ligand occupancy across subpockets: Ligands occupying 1 subpocket**

As shown in Figure S1.B-D, molecules that occupy only the AP subpocket represent either (i) small fragment-like molecules or (ii) large rigid molecules that contain a large fraction of rings, which are difficult to split for most fragmentation algorithms.

An example for the former group (i) is the series of halogenated pyrazoles that stem from a fragment-based approach for druggability assessment and hit generation,<sup>5</sup> see Figure S1.B1-B8. The latter group (ii) contains complete drug-like molecules that either could not be divided because none of the BRICS rules applied or they had a potential BRICS cleavage bond in the initial fragmentation step, which was not broken because the two potential fragments were located in the same subpocket. Furthermore, there are rigid molecules that only contain fused rings with small decorations and, thus do not apply to any fragmentation approach (such as quinalizarin, a CK2 inhibitor, and derivatives, see Figure S1.C1-C2). An example of a molecule that could not be fragmented by BRICS is the co-crystallized ligand HK4 (ChEMBL248396,<sup>6</sup> pIC<sub>50</sub> = 8.3) bound to the CHK1 structure (PDB:4FST,<sup>7</sup> see Figure S1.D1). The two ring moieties clearly cover distinct subpockets (AP and GA), but could not be assigned to them since no rule exists that allows splitting next to a triple bond between two carbon atoms.

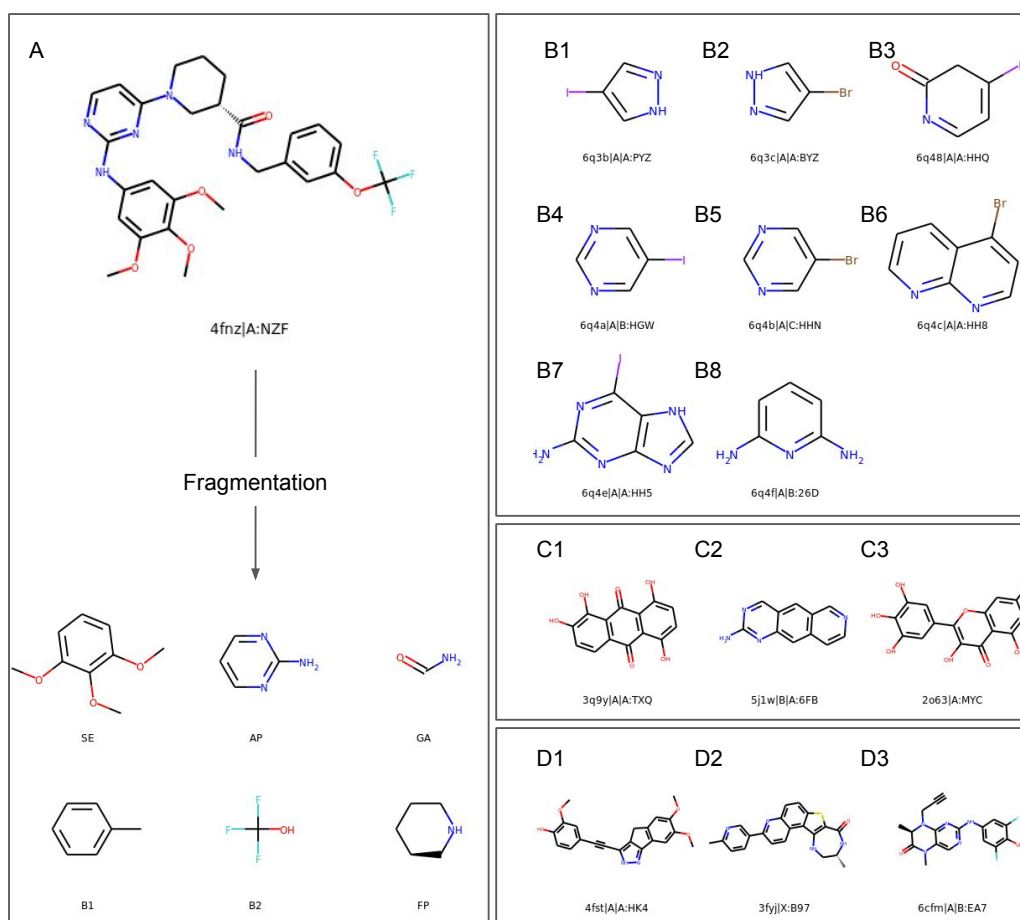


Figure S1: Ligands and fragments which represent special cases in subpocket occupancy, discussed in the main manuscript: (A) Ligand occupying all six subpockets, shown in complete and fragmented state. (B-D) Ligands that were not fragmented either because they are small fragment-like molecules (B) or because they are large rigid molecules which cannot be fragmented: (C) Very rigid molecules containing mostly fused rings with small decorations, and (D) further molecules where no BRICS rule applied.

**Details S7: Ligand connectivity across subpockets: Disallowed subpocket connections/special cases**

Not all ligands showed the allowed subpocket connections as described in Figure 1.C in the main manuscript. Some of those could be rescued by adding additional rules, others had to be discarded (see an overview of disallowed subpocket connections in Table S2). The cases are discussed in the following.

In 113 cases, FP–B2 connections were detected initially. Manual inspection revealed two different methodological drawbacks that could be resolved by the introduced rules: First, in some cases a fragment was assigned to FP because its centroid was slightly closer to FP than GA, although visual inspection showed that the fragment acts as a gate from the front to the back cleft, and should therefore belong to GA (14 cases). Thus, the molecules containing these fragments could be included by reassigning them to GA (see "Molecule fragmentation" in the "Data and Methods" part of the manuscript). Second, the FP–B2 connection was observed when the FP fragment was relatively large. Although some FP-connected fragments pointed mostly into the solvent, they were still close enough and thus falsely assigned to B2. Furthermore, very rare cases were manually observed where the fragment actually covered B2. Since the latter two cases could not be distinguished algorithmically, and the FP–B2 connection is rather unexpected, these B2 fragments were reassigned to pool X (99 cases). The same applies for FP–B1 connections, where each of the two cases described above occurred once.

Connections between non-adjacent subpockets (e.g. SE-GA, AP-B1) usually occur when one of the two subpockets contains a large BRICS fragment (that cannot be further fragmented), which also spans the respective subpocket in between. This happened only rarely, i.e. for AP-B1 and AP–B2 connections in 4 and 3 cases, respectively. Note that potential SE–GA connections were not counted as these ligands do not contain an AP fragment and were excluded from the study beforehand.

Table S2: Disallowed subpocket connections/special cases.

Initial connection	Reassigned fragment	Final connection	# Cases
FP-B1	FP > GA	GA-B1	1
FP-B2	FP > GA	GA-B2	14
FP-B1	B1 > X	FP-X	1
FP-B2	B2 > X	FP-X	99
AP-B1	-	-	4
AP-B2	-	-	3

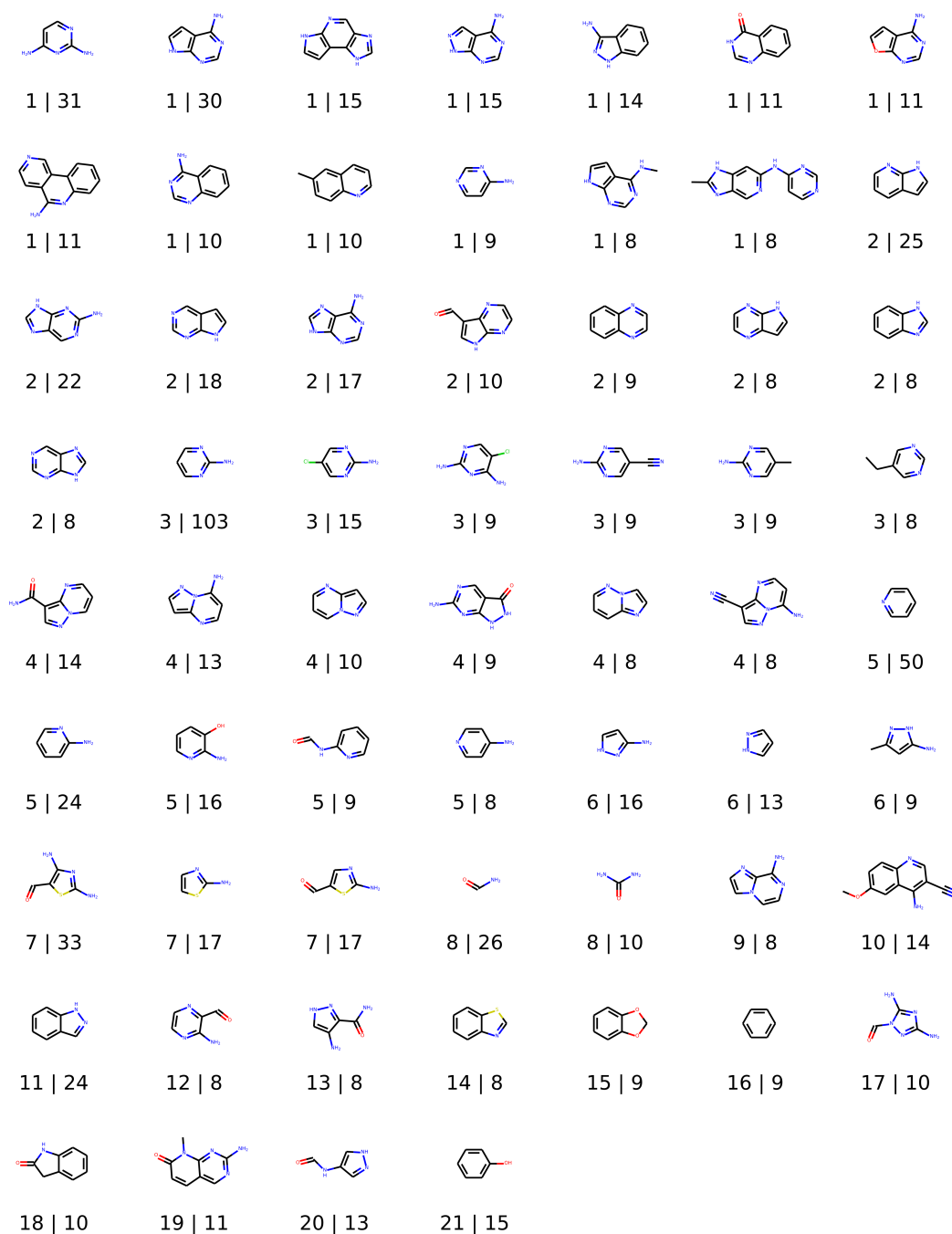


Figure S2: 50 most common fragments in the adenine pocket (AP). Fragments are sorted by and labeled with the cluster number (clusters were sorted by size in descending order) and the number of occurrences. Dummy atoms are replaced with hydrogens. Notebooks to perform this analysis are available at <https://github.com/volkamerlab/kinfraglib>.



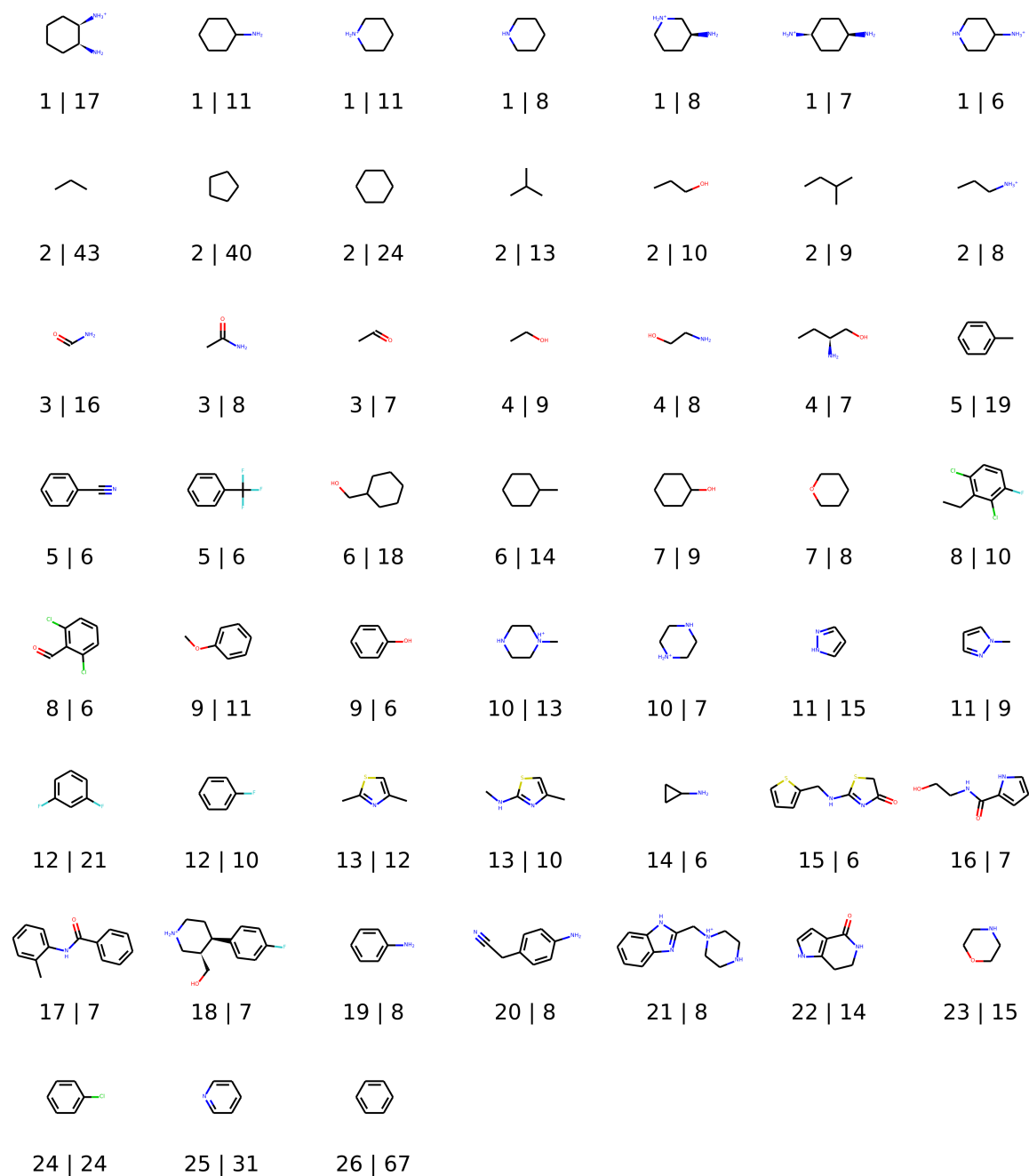


Figure S3: 50 most common fragments in the front pocket (FP). Fragments are sorted by and labeled with the cluster number (clusters were sorted by size in descending order) and the number of occurrences. Dummy atoms are replaced with hydrogens. Notebooks to perform this analysis are available at <https://github.com/volkamerlab/kinfraglib>.

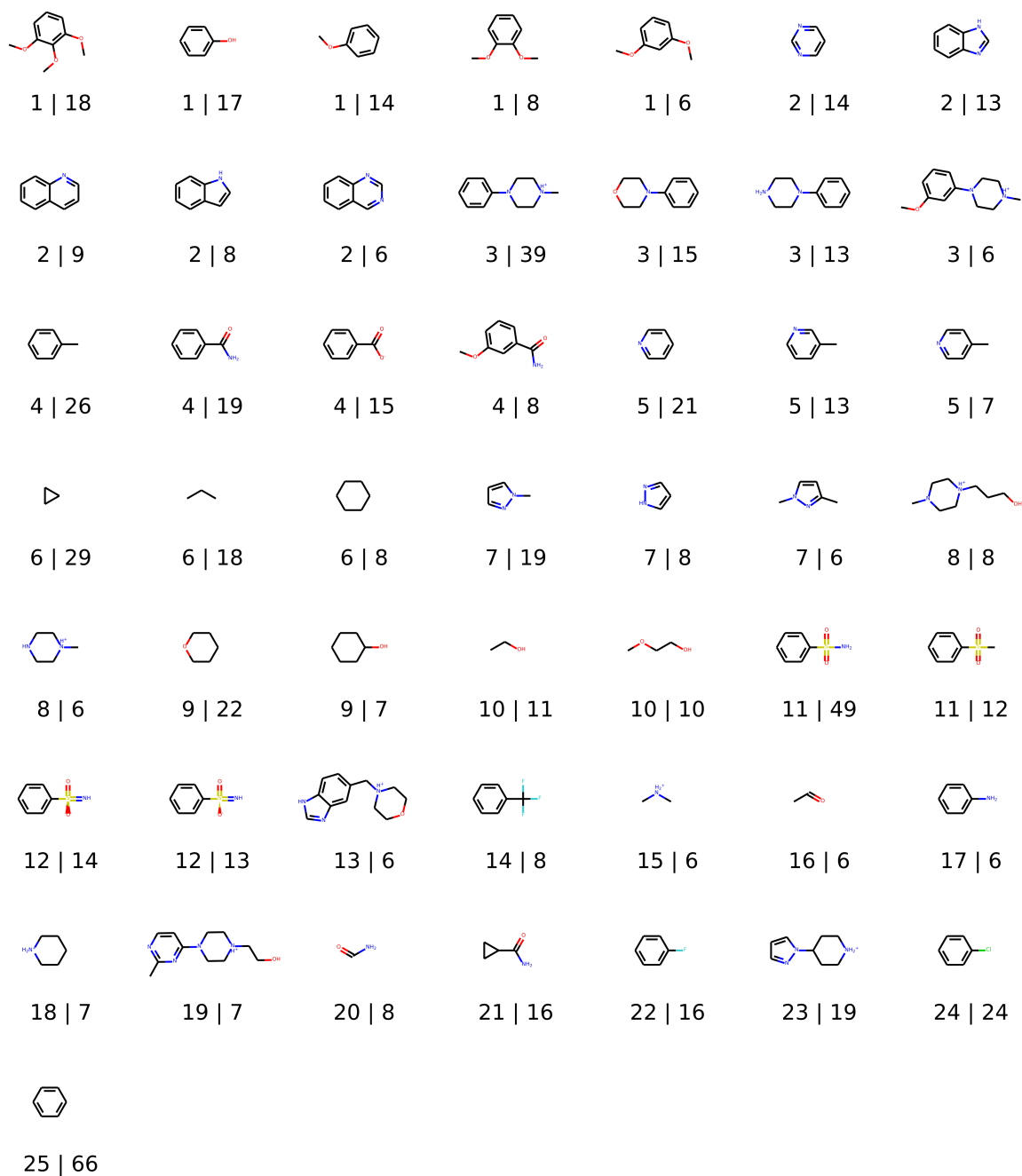


Figure S4: 50 most common fragments in the solvent-exposed pocket (SE). Fragments are sorted by and labeled with the cluster number (clusters were sorted by size in descending order) and the number of occurrences. Dummy atoms are replaced with hydrogens. Notebooks to perform this analysis are available at <https://github.com/volkamerlab/kinfraglib>.

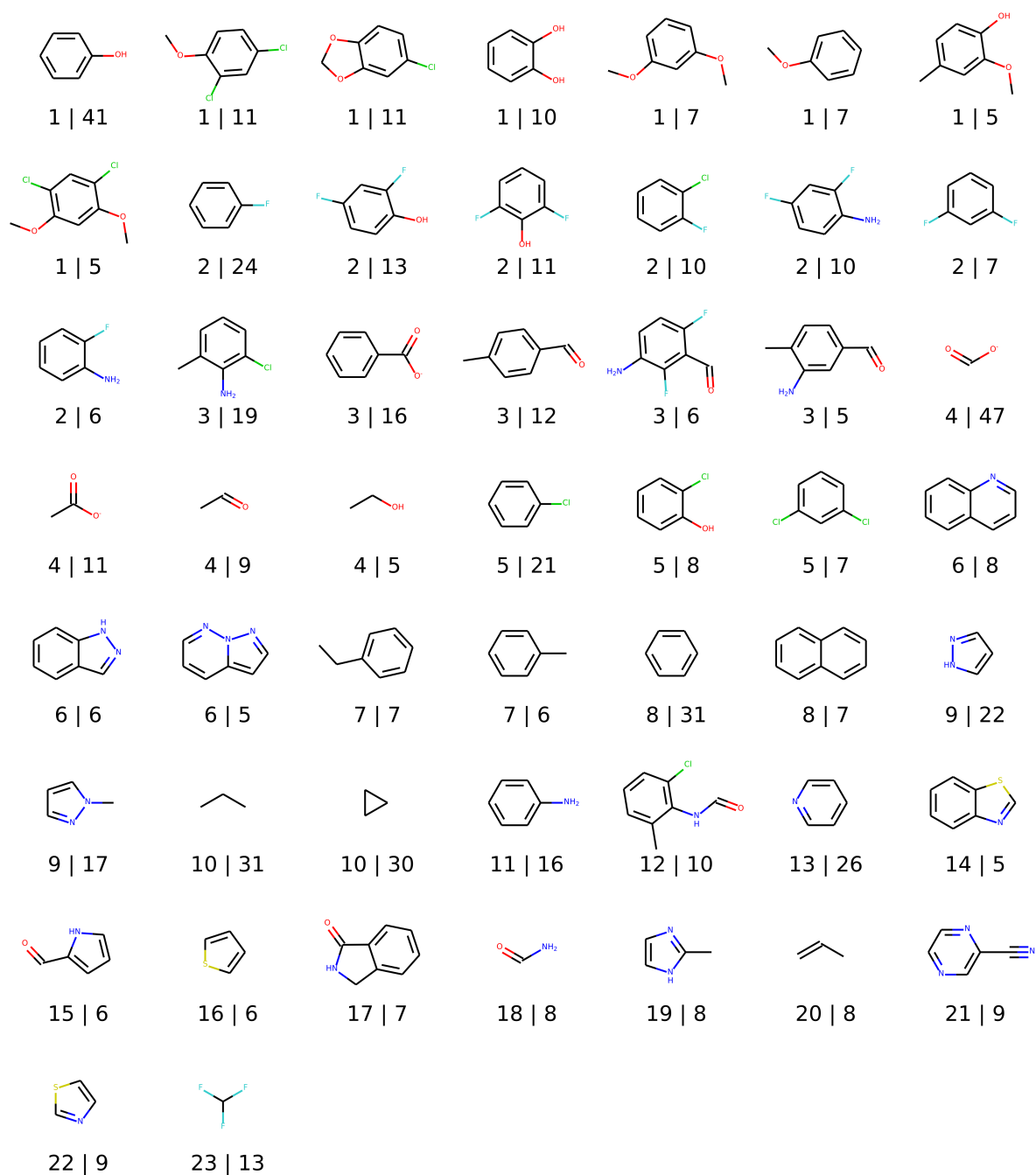


Figure S5: 50 most common fragments in the gate area (GA). Fragments are sorted by and labeled with the cluster number (clusters were sorted by size in descending order) and the number of occurrences. Dummy atoms are replaced with hydrogens. Notebooks to perform this analysis are available at <https://github.com/volkamerlab/kinfraglib>.

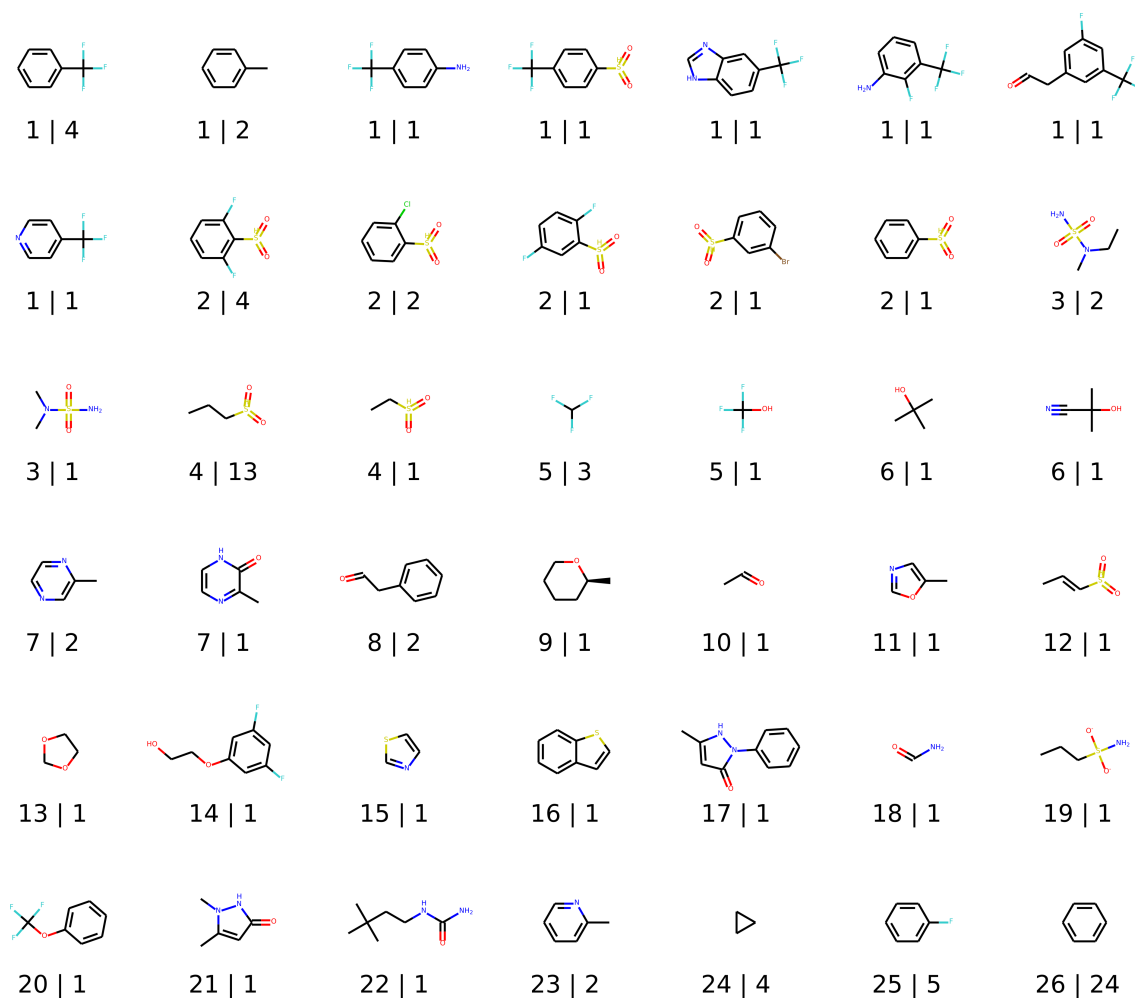


Figure S6: Fragments in the back pocket I (B1). Fragments are sorted by and labeled with the cluster number (clusters were sorted by size in descending order) and the number of occurrences. Dummy atoms are replaced with hydrogens. Notebooks to perform this analysis are available at <https://github.com/volkamerlab/kinfraglib>.

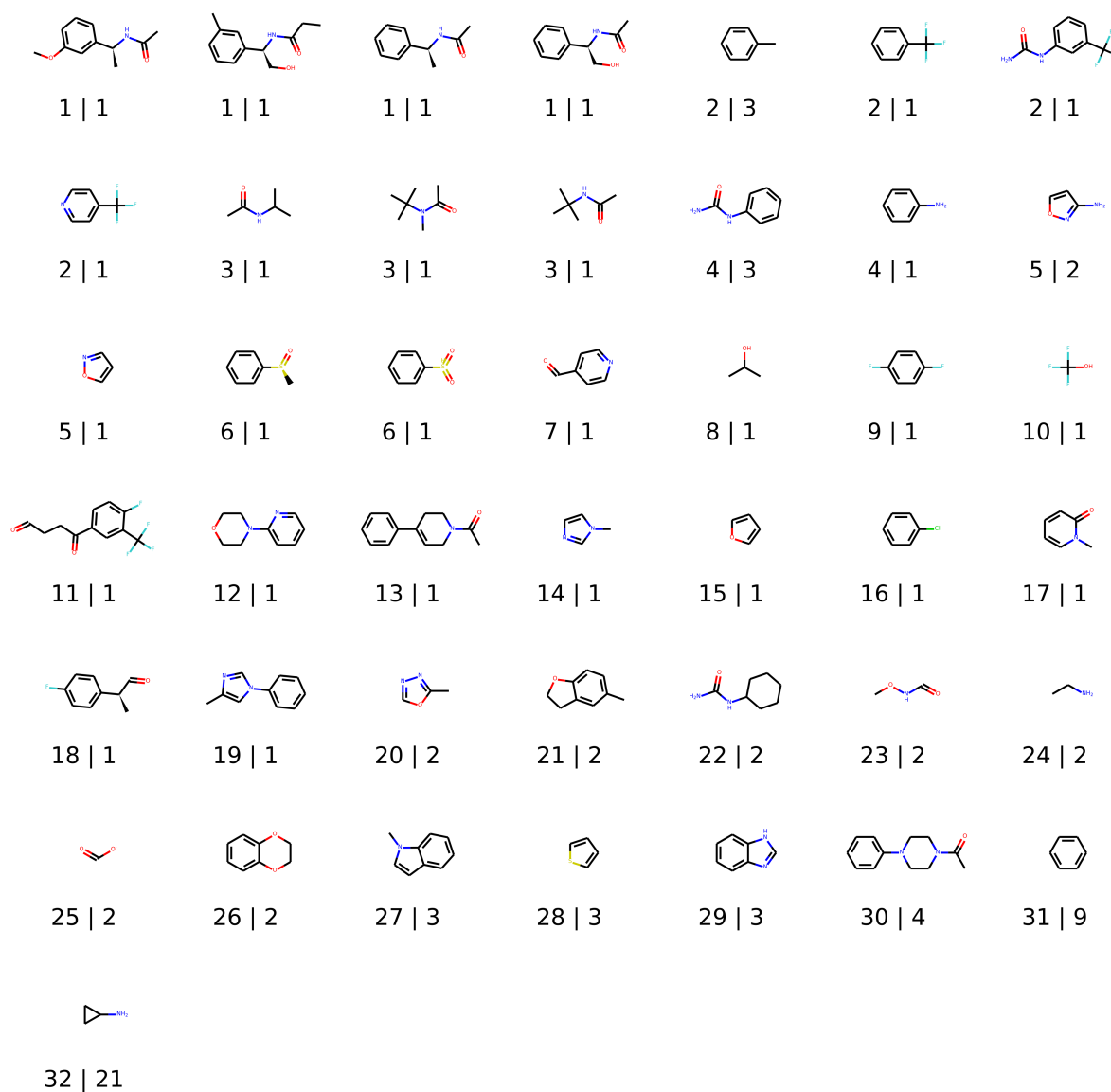

















Figure S7: Fragments in the back pocket II (B2). Fragments are sorted by and labeled with the cluster number (clusters were sorted by size in descending order) and the number of occurrences. Dummy atoms are replaced with hydrogens. Notebooks to perform this analysis are available at <https://github.com/volkamerlab/kinfraglib>.

Table S3: Comparison of our AP fragments to the top 10 hinge fragments from literature (residual groups replaced with hydrogen). Multiple AP fragments can have the same ranks, if they appear equally often, e.g. counts 10-10-8-8-5-4-1 translate to ranks 1-1-3-3-5-6-7. If no or only low ranked exact matches were found in the AP pool, similar AP fragments were selected; rank/count shown in brackets and structure shown in "AP fragment similar structure". Ligands with staurosporine (2. row) and most of adenine (1. row) were removed from KinFragLib and thus do not appear or only with a low AP fragment rank.

Fragment structure	Xing et al. <sup>8</sup> rank	Mukherjee et al. <sup>9</sup> rank	AP fragment rank	AP fragment count	AP fragment similar structure
	1	1, 7	205	2	
	2	4	–	0	
	3	–	15	16	
	4	2, 6	1	103	
	5	9	7	25	
	6	–	8	24	
	7	–	5	30	
	8	–	80 (4)	5 (31)	
	9	3, 5	2	50	
	10	–	11	18	
	–	8	– (11)	0 (18)	
	–	10	– (5)	0 (30)	

## Combinatorial library analysis

Find here supporting information regarding the combinatorial library analysis.

Table S4: Recombined molecules with reported activity in ChEMBL against at least one kinase (activity is here defined as  $IC_{50} \leq 5$  nM). If a compound was measured against the same kinase more than once, all values and respective assay IDs are reported here. Note that the structures of these compounds are shown in Figure S8. The notebook performing this analysis is available at <https://github.com/volkamerlab/kinfraglib>.

Molecule ChEMBL ID	Kinase name	Kinase group	Assay ChEMBL ID	IC50 [nM]
CHEMBL1287863	Serine/threonine-protein kinase Chk1	CAMK	CHEMBL1291622	2.0
CHEMBL1288009	Serine/threonine-protein kinase Chk1	CAMK	CHEMBL1291622	3.0
CHEMBL1288278	Serine/threonine-protein kinase Chk1	CAMK	CHEMBL1291622	1.0
CHEMBL1652706	Casein kinase II	Other	CHEMBL1663323	4.0
CHEMBL1652706	Casein kinase II alpha	Other	CHEMBL3706356	4.0
CHEMBL2030386	Serine/threonine-protein kinase PIM3	CAMK	CHEMBL2038010	4.0
CHEMBL2030386	Serine/threonine-protein kinase PIM1	CAMK	CHEMBL2038008	5.0
CHEMBL2385579	TGF-beta receptor type II	TKL	CHEMBL2390518	1.37
CHEMBL2385579	Vascular endothelial growth factor receptor 2	TK	CHEMBL2390517	1.68
CHEMBL3403541	Tyrosine-protein kinase JAK2	TK	CHEMBL3404501	1.0
CHEMBL3409588	MAP kinase ERK2	CMGC	CHEMBL3705207	3.9
CHEMBL4080944	MAP kinase ERK2	CMGC	CHEMBL4051356	1.2
CHEMBL4114404	MAP kinase ERK2	CMGC	CHEMBL3887970	3.1

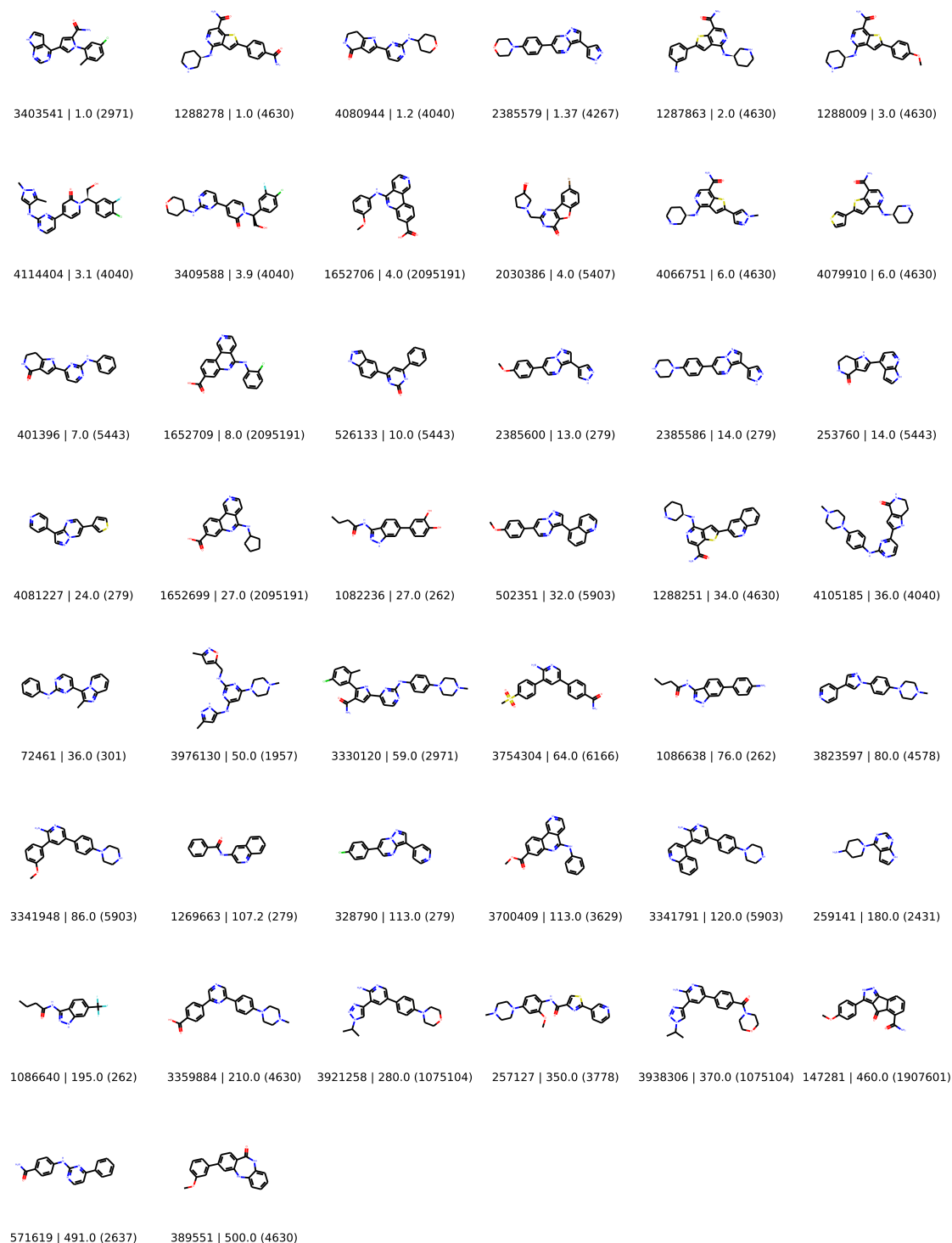


Figure S8: Recombined molecules with reported activity in ChEMBL against at least one kinase (activity is here defined as  $IC_{50} \leq 500$  nM). Legend: molecule ChEMBL ID | minimum  $IC_{50}$  value for kinase (target ChEMBL ID). Add the prefix "ChEMBL" to all ChEMBL IDs.



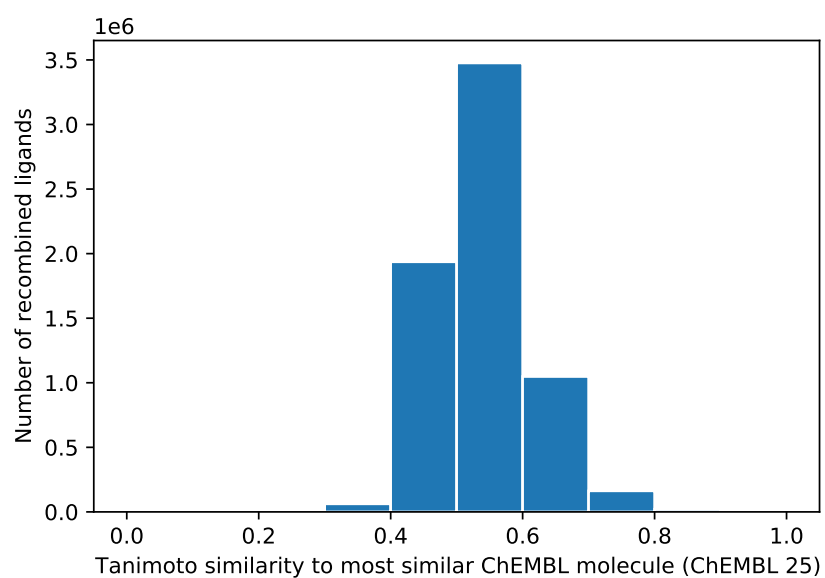


Figure S9: Distribution of Tanimoto similarities for recombined ligands each to their most similar molecule in ChEMBL (ChEMBL 25 dataset), using the RDKit fingerprint.<sup>4</sup> The mean similarity is 0.54 with a standard deviation of 0.07.

## References

- (1) Kooistra, A. Personal communication, 2019.
- (2) Congreve, M.; Carr, R.; Murray, C.; Jhoti, H. A ‘Rule of Three’ for Fragment-Based Lead Discovery? *Drug Discov. Today* **2003**, *8*, 876–877.
- (3) Butina, D. Unsupervised Data Base Clustering Based on Daylight’s Fingerprint and Tanimoto Similarity: A Fast and Automated Way to Cluster Small and Large Data Sets. *J. Chem. Inf. Comput.Sci.* **1999**, *39*, 747–750.
- (4) RDKit, RDKit Fingerprint Version 2020.03.3. <https://www.rdkit.org/docs/source/rdkit.Chem.rdFingerprintGenerator.html> (accessed 2020-04-05).
- (5) Wood, D. J.; Lopez-Fernandez, J. D.; Knight, L. E.; Al-Khawaldeh, I.; Gai, C.; Lin, S.; Martin, M. P.; Miller, D. C.; Cano, C.; Endicott, J. A.; Hardcastle, I. R.; Noble, M. E. M.; Waring, M. J. FragLites—Minimal, Halogenated Fragments Displaying Pharmacophore Doublets. An Efficient Approach to Druggability Assessment and Hit Generation. *Journal of Medicinal Chemistry* **2019**, *62*, 3741–3752, PMID: 30860382.
- (6) ChEMBL, Compound ID ChEMBL248396. [https://www.ebi.ac.uk/chembl/compound\\_report\\_card/ChEMBL248396/](https://www.ebi.ac.uk/chembl/compound_report_card/ChEMBL248396/) (accessed 2020-03-20).
- (7) PDB, Entry ID 4FST. <https://www.rcsb.org/structure/4fst> (accessed 2020-04-05).
- (8) Xing, L.; Klug-Mcleod, J.; Rai, B.; Lunney, E. A. Kinase Hinge Binding Scaffolds and Their Hydrogen Bond Patterns. *Bioorg. Med. Chem.* **2015**, *23*, 6520–6527.
- (9) Mukherjee, P.; Bentzien, J.; Bosanac, T.; Mao, W.; Burke, M.; Muegge, I. Kinase Crystal Miner: A Powerful Approach to Repurposing 3D Hinge Binding Fragments and Its Application to Finding Novel Bruton Tyrosine Kinase Inhibitors. *J. Chem. Inf. Model.* **2017**, *57*, 2152–2160.

### 3.3 FAIR Pipelines and Tools in Kinase-Centric Drug Design

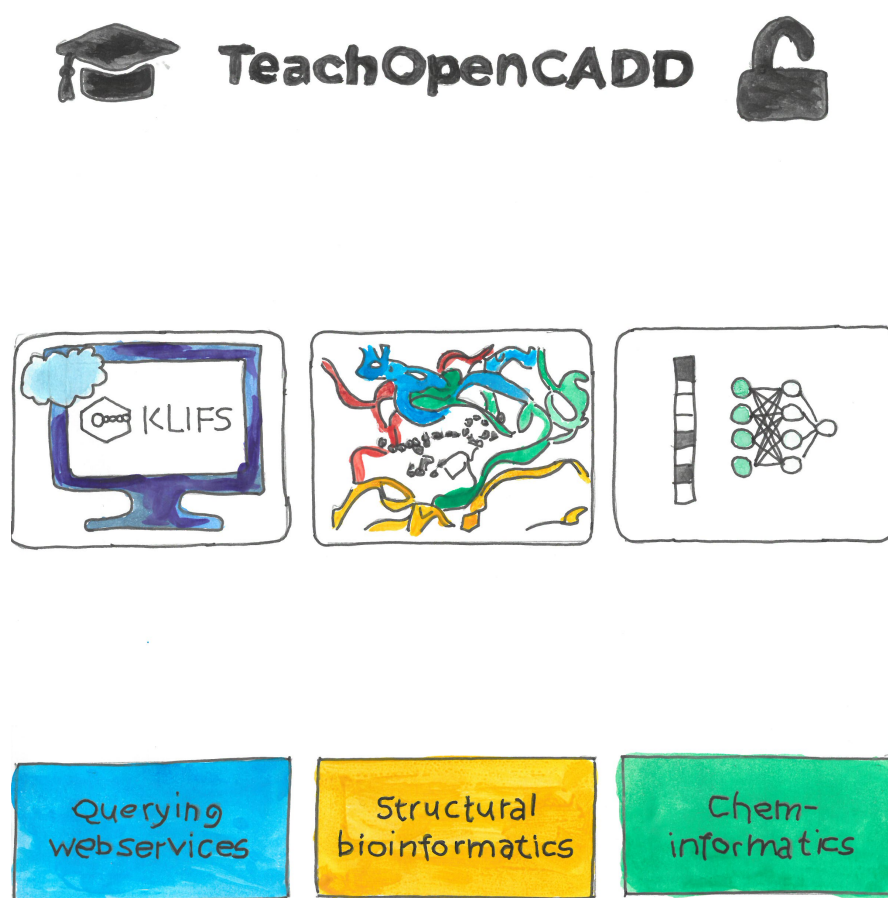



Figure 3.4: FAIR pipelines and tools in kinase-centric drug design as illustrated by Ferdinand Krupp, adapted from Sydow et al. [145].

### 3.3.1 TeachOpenCADD: A Teaching Platform for Computer-Aided Drug Design Using Open Source Packages and Data Publication F

This article is the first publication reporting our TeachOpenCADD platform that initially contained ten Jupyter Notebooks [130] covering common tasks in cheminformatics and structural bioinformatics, including how to programmatically access the ChEMBL [71] and PDB [70] databases. Each Jupyter Notebook covers the topic's aim, theoretical background, practical implementation of the task at hand, a short discussion, and a final quiz. We discuss how this material can be used for novices in the field but also as a starting point for researchers' scientific questions.

 <https://github.com/volkamerlab/teachopencadd>

 <https://projects.volkamerlab.org/teachopencadd/talktorials.html#edition-2019-jcim>



Contribution:

#### First author

Conceptualization (50%)

Data Curation (40%)

Formal Analysis (40%)

Investigation (40%)

Methodology (40%)

Software (40%)

Validation (40%)

Visualization (100%)

Writing — Original Draft (90%)

Writing — Review & Editing (33%)

Reprinted from Sydow D, Morger A, Driller M, Volkamer A. TeachOpenCADD: A Teaching Platform for Computer-Aided Drug Design Using Open Source Packages and Data. *Journal of Cheminformatics*. **2019**; 11(1):29. 10.1186/s13321-019-0351-x.

Open access article licensed under a CC BY 4.0 license.

## SOFTWARE

## Open Access



# TeachOpenCADD: a teaching platform for computer-aided drug design using open source packages and data

Dominique Sydow , Andrea Morger , Maximilian Driller and Andrea Volkamer\*

## Abstract

Owing to the increase in freely available software and data for cheminformatics and structural bioinformatics, research for computer-aided drug design (CADD) is more and more built on modular, reproducible, and easy-to-share pipelines. While documentation for such tools is available, there are only a few freely accessible examples that teach the underlying concepts focused on CADD, especially addressing users new to the field. Here, we present TeachOpenCADD, a teaching platform developed by students for students, using open source compound and protein data as well as basic and CADD-related Python packages. We provide interactive Jupyter notebooks for central CADD topics, integrating theoretical background and practical code. TeachOpenCADD is freely available on GitHub: <https://github.com/volkamerlab/TeachOpenCADD>.

**Keywords:** Computer-aided drug design, Python, RDKit, Open source, Teaching, Learning, Cheminformatics, Structural bioinformatics

## Introduction

Open access resources for cheminformatics and structural bioinformatics as well as public platforms for code deposition such as GitHub are increasingly used in research. This combination facilitates and promotes the generation of modular, reproducible, and easy-to-share pipelines for computer-aided drug design (CADD). Comprehensive lists of open resources are reviewed by Pirhadi et al. [1], or presented in the form of the web-based search tool Click2Drug [2], aiming to cover the full CADD pipeline.

While documentation for open access resources is available, freely accessible teaching platforms for concepts and applications in CADD are rare. Available examples include the following: On the one hand, graphical user interface (GUI) based tutorials teach CADD basics, such as the web-based educational Drug Design Workshop [3, 4]. On the other hand, examples for educational coding tutorials are the Java-based Chemistry

Development Kit (CDK) [5–9] and the Teach–Discover–Treat (TDT) initiative [10], which launched challenges to develop tutorials, such as a Python-based virtual screening (VS) workflow to identify malaria drugs [11, 12].

Complementing these resources, we developed the TeachOpenCADD platform to provide students and researchers new to CADD and/or programming with step-by-step tutorials suitable for self-study training as well as classroom lessons, covering both ligand- and structure-based approaches. TeachOpenCADD is a novel teaching platform developed by students for students, using open source data and Python packages to tackle various common tasks in cheminformatics and structural bioinformatics. Interactive Jupyter notebooks [13] are presented for central topics, integrating detailed theoretical background and well-documented practical code. Topics build upon one another in the form of a pipeline, which is illustrated at the example of the epidermal growth factor receptor (EGFR) kinase, but can easily be adapted to other query proteins. TeachOpenCADD is publicly available on GitHub and open to contributions from the community: <https://github.com/volka>

\*Correspondence: [andrea.volkamer@charite.de](mailto:andrea.volkamer@charite.de)

In Silico Toxicology, Institute of Physiology, Charité – Universitätsmedizin Berlin, Charitéplatz 1, 10117 Berlin, Germany

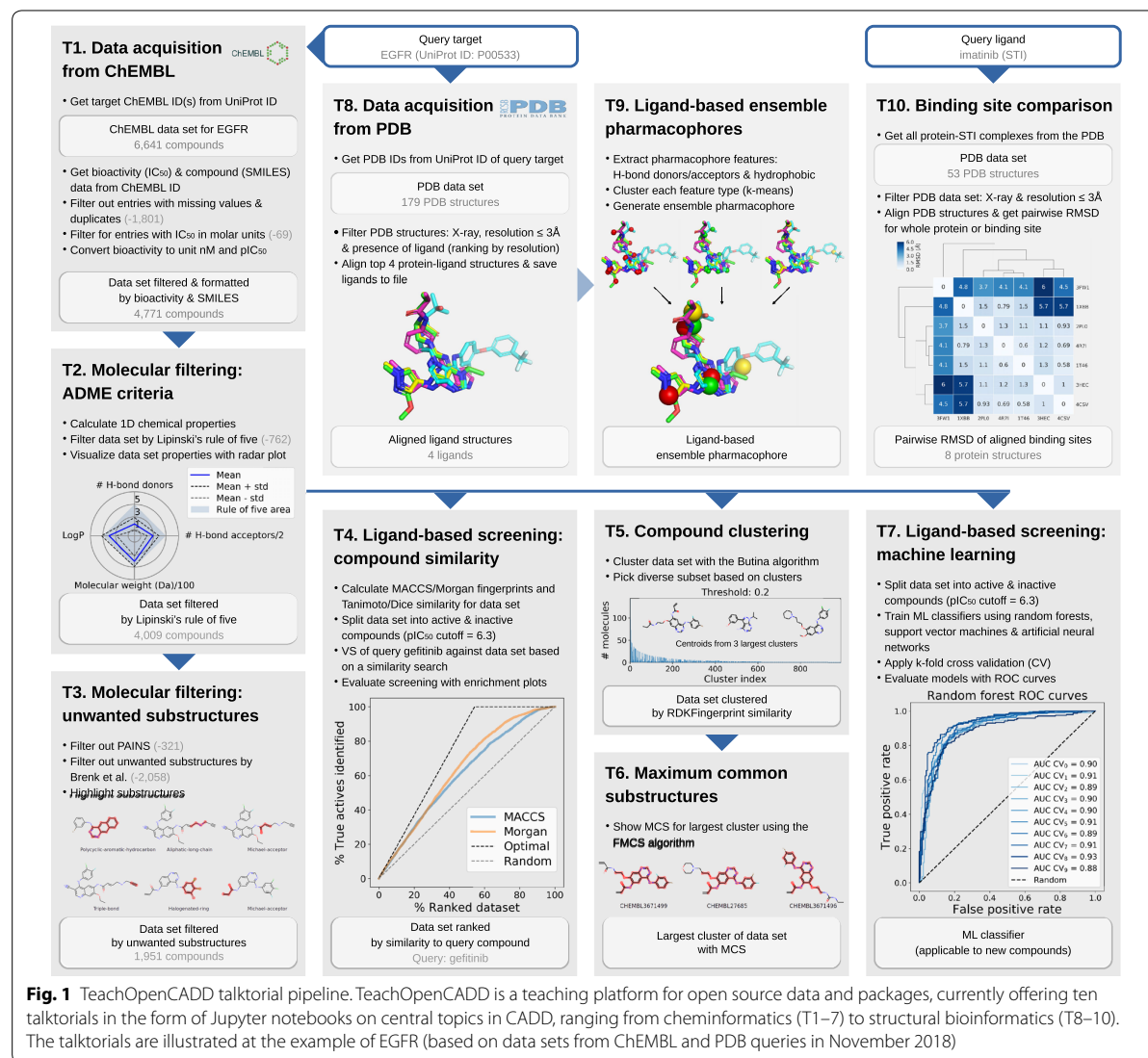


[merlab/TeachOpenCADD](https://doi.org/10.5281/zenodo.2600909) (current release: <https://doi.org/10.5281/zenodo.2600909>).

## Methods

TeachOpenCADD currently consists of ten *talktorials* covering central topics in CADD, see Fig. 1. Talktorials are offered as interactive Jupyter notebooks that can be used as tutorials but also for oral presentations, e.g. in student CADD seminars (talk + tutorial = talktorial). They start with a topic motivation and learning goals, continue with the main part composed of theoretical background and practical code, and end with a short discussion and quiz, see Fig. 2.

Open data resources employed are the ChEMBL [14] and PDB [15] databases for compound and protein structure data acquisition, respectively. Open source libraries utilized are RDKit [16] (cheminformatics), the ChEMBL webresource client [17] and PyPDB [18] (ChEMBL and PDB application programming interface access), BioPandas [19] (loading and manipulating molecular structures), and PyMOL [20] (structural data visualization). Additionally, basic Python computing libraries employed include numpy [21, 22] and pandas [23, 24] (high-performance data structures and analysis), scikit-learn [25] (machine learning), as well as matplotlib [26] and seaborn [27] (plotting). Furthermore, the user is instructed how



**Fig. 1** TeachOpenCADD talktorial pipeline. TeachOpenCADD is a teaching platform for open source data and packages, currently offering ten talktorials in the form of Jupyter notebooks on central topics in CADD, ranging from cheminformatics (T1–7) to structural bioinformatics (T8–10). The talktorials are illustrated at the example of EGFR (based on data sets from ChEMBL and PDB queries in November 2018)

The screenshot shows a Jupyter notebook interface with the following content:

**Aim of this talktorial**  
 In this talktorial, we use known EGFR ligands, which were selected and aligned in the previous talktorial, to identify donor, acceptor, and hydrophobic pharmacophoric features for each ligand. Those features are then clustered to define an ensemble pharmacophore, which represents the properties of the set of known EGFR ligands and can be used to search for novel EGFR ligands via virtual screening.

**Practical**  
**Show ensemble pharmacophore**  
 In this last step, we combine the clustered pharmacophoric features (i.e. hydrogen bond donors and acceptors as well as hydrophobic contacts), to one ensemble pharmacophore, representing the pharmacophoric properties of the four selected ligands.

```
In [46]: # Initialize PyMol in order to remove all previous objects
objPMV.server.do("reinitialize")

# Load ligands
rangeMols = range(1, len(mols)+1)
for mol, i in zip(mols, rangeMols):
    objPMV.ShowMol(mol, name='mol_%d'%i, showOnly=False)
    toStickCmd='cmd.show("sticks", "mol_'+str(i)+'")'
    objPMV.server.do(toStickCmd)
    i += 1

# Load clusters
for feature_type in cluster_indices_sel.keys():
    centers = cluster_centers_sel[feature_type]
    for i in range(len(centers)):
        loc = centers[i]
        sphere_radius = 1
        feature_color = feature_colors[feature_type]
        label = feature_type + '_c%d'%(i+1)
        objPMV.server.sphere(loc, sphere_radius, feature_color, label, 1)

# Turn camera
objPMV.server.do("turn x, -40")

# Set PyMol styling
objPMV.server.do("bg_color white")
objPMV.server.do("zoom")
objPMV.server.do("ray 1800, 1000")

# Export as PNG file
outputPNG = objPMV.GetPNG(w=1800, h=1000)
outputPNG.save("../data/T9/ligands_ensemble_ph4.png")

# Display in Jupyter notebook
objPMV.GetPNG(h=300)
```

**Talktorial sections**  
 Aim of this talktorial  
 Learning goals  
 References  
 Theory  
 Practical  
 Discussion  
 Quiz

**Explanation**  
**Code**  
**Output**

**All-in-one Jupyter notebook**

**Out[46]:**

**Fig. 2** Screenshot of TeachOpenCADD talktorial composition. TeachOpenCADD talktorials are Jupyter notebooks that cover one CADD topic each, composed of (i) a topic motivation, (ii) learning goals, (iii) references to literature, (iv) theoretical background, (v) practical code, (vi) a short discussion, and (vii) a quiz—all in one place. Shown here is a screenshot of parts of talktorial T9 to generate pharmacophores

to work with conda [28], a widely used package, dependency and environment management tool. A conda yml file is provided to ensure an easy and quick setup of an environment containing all required packages.

The talktorial topics include how to acquire data from ChEMBL (T1), filter compounds for drug-likeness (T2), and identify unwanted substructures (T3). Furthermore, measures for compound similarity are introduced and applied for VS of kinase inhibitor gefitinib (T4) as well as for compound clustering (T5), including the use of maximum common substructures (T6). Machine learning approaches are employed to build models for predicting active compounds (T7). Lastly, protein-ligand complexes are fetched from the PDB (T8), used to

generate ligand-based ensemble pharmacophores (T9). Geometry-based binding site comparison of kinase inhibitor imatinib binding proteins is performed to analyse potential off-targets (T10). In summary, the presented talktorials build a pipeline with starting points being (i) a query protein to study associated compound data (T1 and T8) and (ii) a query ligand to investigate associated on- and off-targets (T10), see Fig. 1. These talktorials can be studied independently from each other or as a pipeline.

As an example, the talktorial pipeline is used to identify novel EGFR kinase inhibitors. EGFR kinase is a transmembrane protein, which activates several signaling cascades to convert extracellular signals into cellular

responses. Dysfunctional signaling of EGFR is associated with diseases such as cancer, making it a frequent target in drug development projects (the reader is referred to a review by Chen et al. [29] for more information on EGFR). Furthermore, the pipeline can easily be adapted to other examples by simply exchanging the query protein (T1 and T8: protein UniProt ID) and query ligand (T10: ligand names in the PDB).

## Results

In the following, the content of each talktorial is briefly discussed and summarized in Fig. 1. If not noted otherwise, tasks are conducted with RDKit or basic Python libraries as stated in the Methods section. Note that reported numbers and results are based on data sets from ChEMBL and PDB queries conducted in November 2018.

*T1. Data acquisition from ChEMBL.* Compound information on structure, bioactivity and associated targets is organized in databases such as ChEMBL, PubChem [30], or DrugBank [31]. For the query target EGFR (UniProt ID P00533), compound data including molecular structure (SMILES) and bioactivity data is automatically fetched from the ChEMBL database, using the ChEMBL web-resource client, and is filtered for e.g. binding assays and  $IC_{50}$  measurements (6,641 compounds). The data set is formatted and further filtered: e.g. duplicates and entries with missing values are dropped and only bioactivity values in molar units are kept and converted to  $pIC_{50}$  values (4,771 compounds retained, referred to as *data set T1*), see Fig. 1.T1.

*T2. Molecular filtering: ADME criteria.* Not all compounds are suitable starting points for drug development due to undesirable pharmacokinetic properties, which for instance negatively affect a drug's absorption, distribution, metabolism, and excretion (ADME). Therefore, such compounds are usually not included in data sets for VS. *Data set T1* is filtered by lead-likeness criteria, i.e. Lipinski's rule of five [32], in order to remove less drug-like molecules from the EGFR data set (4009 compounds retained, referred to as *data set T2*). This data set is visualized using radar plots demonstrating their ADME properties, see Fig. 1.T2, and serves as starting point for several talktorials discussed in the following.

*T3. Molecular filtering: unwanted substructures.* Compounds can contain unwanted substructures that may cause mutagenic, reactive, or other unfavorable pharmacokinetic effects [33] or that may lead to non-specific interactions with assays (PAINS) [34]. Such unwanted substructures are detected and highlighted in *data set T2*. This knowledge can be integrated into cheminformatics pipelines to either perform an additional filtering step before screening (1,951 compounds retained) or – more often – to set alert flags to compounds being

potentially problematic. They can be manually evaluated by medicinal chemists if reported as hits after screening, see Fig. 1.T3.

*T4. Ligand-based screening: compound similarity.* In VS, compounds similar to known ligands of a target under investigation often constitute the starting point for drug development. This approach follows the similar property principle stating that structurally similar compounds are more likely to exhibit similar biological activities [35, 36] (exceptions are so-called activity cliffs [37]). For computational representation and processing, compound properties can be encoded in the form of bit arrays, so-called molecular fingerprints, e.g. MACCS [38] and Morgan fingerprints [39, 40]. Compound similarity can be assessed by comparison measures, such as the Tanimoto and Dice similarity [41]. Using these encoding and comparison methods, VS is conducted based on a similarity search: the EGFR inhibitor gefitinib is used to find its most similar compounds in data set T2. With the data being split into active and inactive compounds based on the chosen  $pIC_{50}$  cutoff of 6.3, screening results are evaluated with enrichment plots, see Fig. 1.T4. In the top 5% of the compounds ranked by similarity, called the enrichment factor at 5% ( $EF_{5\%}$ ), 8.3% of actives can be retrieved, while the random and optimal  $EF_{5\%}$  of this data set are 5.0% and 9.2%, respectively.

*T5. Compound clustering.* The similar property principle can also be used to identify groups of similar compounds via clustering, in order to pick a set of diverse compounds from these clusters for e.g. non-redundant experimental testing. In this talktorial, Butina clustering [42] based on the RDKFingerprint [43] is applied to cluster *data set T2* at a Tanimoto distance cutoff of 0.2, resulting in 988 clusters with the largest cluster consisting of 143 compounds, see Fig. 1.T5. Following the example in the TDT pipeline by Riniker et al. [11], a maximum of 1000 compounds is subsequently picked by selecting the ten most similar compounds per cluster (or 50% for clusters with fewer compounds), starting with the largest cluster. Thereby, compound diversity is ensured (representatives of each cluster), while structure-activity relationship (SAR) information is retained (most similar compounds selected from clusters).

*T6. Maximum common substructures.* In order to visualize shared scaffolds and thereby emphasize the extent and type of chemical similarities or differences of a compound cluster, the maximum common substructure (MCS) [44] can be calculated and highlighted. The MCS for the largest cluster from T5 is calculated using the FMCS algorithm [45], see Fig. 1.T6. Different parameters can be applied, e.g. a threshold to set the percentage of compounds in the set that need to share the same MCS,



or a restriction to match ring bonds only with other ring bonds.

**T7. Ligand-based screening: machine learning.** With the continuously increasing amount of available data, machine learning (ML) gained momentum in drug discovery and especially in ligand-based VS to predict the activity of novel compounds against a target of interest. The EGFR compound data set is split into active and inactive compounds as described in T4, and used to train ML classifiers based on random forests (RF) [46], support vector machines (SVM) [47], and artificial neural networks (ANN) [48], applying 10-fold cross validation. Models are evaluated using receiver operating characteristic (ROC) curves and mean area under the curve (AUC) values (mean AUC results for RF, SVM, and ANN are 90%, 87%, and 87%, respectively), see Fig. 1.T7. The trained models can be used to perform a classification of an unknown screening data set to predict novel potential EGFR inhibitors.

**T8. Data acquisition from PDB.** The PDB database holds 3D structural data and meta information on experimentally resolved proteins. Using PyPDB, all EGFR structures are automatically fetched from the PDB (by UniProt ID) and filtered by ligand-bound structures resolved with X-ray crystallography, retaining four EGFR-ligand structures with good structural resolution. Using the Python integration of the molecular visualization tool PyMOL, those structures are subsequently aligned to each other in 3D. Ligands are extracted, see Fig. 1.T8, and saved to be used in T9 for the generation of a ligand-based ensemble pharmacophore.

**T9. Ligand-based ensemble pharmacophores.** Another approach for ligand-based VS – besides a similarity search (T4) or machine learning classifiers (T7) – are ligand-based (ensemble) pharmacophore models. They describe important steric and physicochemical properties of a ligand (or a set of ligands) to bind a target under investigation. Examples for physicochemical properties are so-called donor, acceptor, and hydrophobic pharmacophoric features present in a molecule [49, 50]. For the EGFR ligands selected and aligned in T8, pharmacophoric features are identified for each ligand and subsequently clustered with k-means clustering [51] in order to define an ensemble pharmacophore, see Fig. 1.T9. Such a pharmacophore represents the properties of the set of known EGFR ligands and can be used to search for novel EGFR ligands via VS, as described in an RDKit pharmacophore tutorial by Stiefl et al. [52].

**T10. Off-target prediction and binding site comparison.** Off-targets are proteins that interact with a drug or (one of) its metabolite(s) without being the designated target, potentially causing unwanted side effects. Off-targets mainly occur because they share similar structural motifs

in their binding site with on-targets, and are therefore able to bind similar ligands. Computational off-target prediction using binding site comparison is an established approach in early stages of drug development [53, 54]. In T10, structural similarity is exemplarily accessed using a basic measure, i.e. the geometrical variation between structures by calculating the root mean square deviation (RMSD) between pairs of aligned structures using PyMOL, including either the whole proteins or focusing on their binding sites. Pairwise RMSD comparison of seven protein structures binding imatinib, a small molecule tyrosine kinase inhibitor for cancer treatment, is able to separate tyrosine kinases (on-targets) from quinone reductase (reported off-target [55]), see Fig. 1.T10.

## Conclusion

The presented teaching platform TeachOpenCADD aims at introducing interested students and researchers to the ease and benefit of using open access resources for cheminformatics and structural bioinformatics. Jupyter notebooks (talktorials) offer detailed theoretical background and Python code examples, forming an automated pipeline that saves and reloads results from one topic to another. The pipeline is illustrated using the example of EGFR, but can easily be adapted to other examples by exchanging the input protein and ligand. Beyond their teaching purpose for self-study training and classroom lessons, the talktorials can serve as starting point for users' project-directed modifications and extensions. TeachOpenCADD intends to expand existing and add new topics continuously, and is open for contributions and ideas from the community.

## Abbreviations

CADD: computer-aided drug design; GUI: graphical user interface; CDK: Chemistry Development Kit; TDT: Teach-Discover-Treat; VS: virtual screening; EGFR: epidermal growth factor receptor; ADME: absorption, distribution, metabolism, excretion; SAR: structure-activity relationship; MCS: maximum common substructure; ML: machine learning; RF: random forest; SVM: support vector machine; ANN: artificial neural network; ROC: receiver operating characteristic; AUC: area under the curve; RMSD: root mean square deviation; EF: enrichment factor.

## Authors' contributions

All authors (DS, AM, MD, and AV) contributed to implementing the platform, finalizing the talktorials, and editing/reviewing the manuscript. DS was responsible for management and major writing, and AV for conceptualization, management, and writing. All authors read and approved the final manuscript.

## Acknowledgements

The authors thank the participants of the CADD seminar courses in 2017 and 2018 (joint bioinformatics study program at the Freie Universität Berlin and the Charité) for working on the reported talktorials: Svetlana Leng and Paula Junge (T1), Mathias Wajnberg and Michele Ritschel (T2), Maximilian Driller and Sandra Krüger (T3), Andrea Morger and Franziska Fritz (T4), Gizem Spriewald and Calvinna Caswara (T5), Oliver Nagel (T6), Jacob Gora and Jan Philipp Albrecht (T7), Majid Vafadar and Anja Georgi (T8), Pratik Dhakal and

Florian Gusewski (T9), as well as Angelika Szengel and Marvis Sydow (T10). Additionally, the authors acknowledge Greg Landrum and Boran Adas for their feedback on the talktorials. Finally, the authors express their gratitude to the Freie Universität Berlin for supporting the TeachOpenCADD project (SUPPORT für die Lehre: Förderung innovativer Lehrvorhaben).

#### Competing interests

The authors declare that they have no competing interests.

#### Availability and requirements

Project name: TeachOpenCADD. Project home page: <https://github.com/volkamerlab/TeachOpenCADD>. Operating system(s): Platform independent. Programming language: Python. Other requirements: Databases: ChEMBL and PDB. Python packages: RDKit, ChEMBL webresource client, PyPDB, BioPandas, PyMOL, numpy, pandas, scikit-learn, matplotlib, seaborn, and conda. License: <http://creativecommons.org/licenses/by/4.0/>. Any restrictions to use by non-academics: Not applicable.

#### Availability of data and materials

TeachOpenCADD talktorial material is available at <https://github.com/volkamerlab/TeachOpenCADD>. Compound and protein structure data used as EGFR example in the talktorials are fetched from the ChEMBL (query by UniProt ID "P00533") and PDB (query by UniProt ID "P00533", "STI", and "imatinib") databases.

#### Funding

The authors receive funding from the Bundesministerium für Bildung und Forschung (AV: Grant Number 031A262C), Deutsche Forschungsgemeinschaft (DFG) (AV and DS: Grant Number 391684253), and the HaVo-Stiftung, Ludwigshafen, Germany (AM). The authors acknowledge support from the German Research Foundation (DFG) and the Open Access Publication Fund of Charité – Universitätsmedizin Berlin.

#### Publisher's Note

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations

Received: 19 December 2018 Accepted: 27 March 2019

Published online: 08 April 2019

#### References

- Pirhadi S, Sunseri J, Koes DR (2016) Open source molecular modeling. *J Mol Graph Modell* 69:127–43
- Swiss Institute of Bioinformatics (2013) Click2Drug website. <http://www.click2drug.org/>. Accessed 18 Dec 2018
- Daina A, Blatter MC, Baillie Gerritsen V, Palagi PM, Marek D, Xenarios I, Schwede T, Michielin O, Zoete V (2017) Drug design workshop: a web-based educational tool to introduce computer-aided drug design to the general public. *J Chem Educ* 94:335–344
- Swiss Institute of Bioinformatics (2015) Drug Design Workshop website. [www.drug-design-workshop.ch](http://www.drug-design-workshop.ch). Accessed 18 Dec 2018
- Steinbeck C, Han Y, Kuhn S, Horlacher O, Luttmann E, Willighagen E (2003) The Chemistry Development Kit (CDK): an open-source java library for chemo- and bioinformatics. *J Chem Inf Comput Sci* 43:493–500
- Steinbeck C, Hoppe C, Kuhn S, Floris M, Guha R, Willighagen EL (2006) Recent developments of the Chemistry Development Kit (CDK)—an open-source java library for chemo- and bioinformatics. *Curr Pharm Des* 12:2111–20
- May JW, Steinbeck C (2014) Efficient ring perception for the Chemistry Development Kit. *J Cheminf* 6:3
- Willighagen EL, Mayfield JW, Alvarsson J, Berg A, Carlsson L, Jeliakova N, Kuhn S, Pluskal T, Rojas-Chertó M, Torrance G, Evelo CT, Guha R, Steinbeck C (2017) The Chemistry Development Kit (cdk) v2.0: atom typing, depiction, molecular formulas, and substructure searching. *J Cheminf* 9:33
- Chemistry Development Kit (2017) Chemistry Development Kit (CDK) website. <https://cdk.github.io/>, Accessed 18 Dec 2018
- Jansen JM, Cornell W, Tseng YJ, Amaro RE (2012) Teach–Discover–Treat (TDT): collaborative computational drug discovery for neglected diseases. *J Mol Graph Modell* 38:360–2
- Riniker S, Landrum GA, Montanari F, Villalba SD, Maier J, Jansen JM, Walters WP, Shelat AA (2017) Virtual-screening workflow tutorials and prospective results from the Teach–Discover–Treat competition 2014 against malaria. *F1000Research* 6:1136
- Riniker S, Landrum GA, Montanari F, Villalba SD, Maier J, Jansen JM, Walters WP, Shelat AA (2017) Tutorial for the Teach–Discover–Treat (TDT) Competition 2014—Challenge 1: anti-malaria hit finding using classifier-fusion boosted predictive models. <https://github.com/sriniker/TDT-tutorial-2014>. Accessed 18 Dec 2018
- Kluyver T, Ragan-Kelley B, Pérez F, Granger B, Bussonnier M, Frederic J, Kelley K, Hamrick J, Grout J, Corlay S, Ivanov P, Avila D, Abdalla S, Willing C, Team Jupyter Development (2016) Jupyter Notebooks—a publishing format for reproducible computational workflows. Agents and agendas. In: Loizides F, Schmidt B (eds) Positioning and power in academic publishing: players. IOS Press, Amsterdam, pp 87–90
- Gaulton A, Bellis LJ, Bento AP, Chambers J, Davies M, Hersey A, Light Y, McGlinchey S, Michalovich D, Al-Lazikani B, Overington JP (2012) ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Res* 40:1100–7
- Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE (2000) The protein data bank. *Nucleic Acids Res* 28:235–42
- RDKit (2018) RDKit: Open-Source Cheminformatics, Version 2018.09.1. <http://www.rdkit.org>
- Davies M, Nowotka M, Papadatos G, Dedman N, Gaulton A, Atkinson F, Bellis L, Overington JP (2015) ChEMBL web services: streamlining access to drug discovery data and utilities. *Nucleic Acids Res* 43:W612–W620
- Gilpin W (2015) PyPDB: a Python API for the protein data bank. *Bioinformatics* 32:159–60
- Raschka S (2017) BioPandas: working with molecular structures in pandas DataFrames. *J Open Source Softw* 2:279
- Schrödinger L (2015) The PyMOL molecular graphics system. Version 1.8
- Oliphant T (2006) A guide to NumPy. Trelgol Publishing
- van der Walt S, Colbert SC, Varoquaux G (2011) The NumPy array: a structure for efficient numerical computation. *Comput Sci Eng* 13(2):22–30
- McKinney W (2010) Data structures for statistical computing in Python. In: van der Walt S, Millman J (eds) Proceedings of the 9th Python in science conference, pp 51–56
- McKinney W (2011) pandas: a foundational Python library for data analysis and statistics
- Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in Python. *J Mach Learn Res* 12:2825–2830
- Hunter JD (2007) Matplotlib: a 2D graphics environment. *Comput Sci Eng* 9:90–95
- Waskom M (2018) seaborn v0.9.0
- Continuum Analytics Inc (dba Anaconda Inc) (2017) conda. <https://www.anaconda.com>. Accessed 18 Dec 2018
- Chen J, Zeng F, Forrester SJ, Eguchi S, Zhang MZ, Harris RC (2016) Expression and function of the epidermal growth factor receptor in physiology and disease. *Physiol Rev* 96:1025–1069
- Kim S, Thiessen PA, Bolton EE, Chen J, Fu G, Gindulyte A, Han L, He J, He S, Shoemaker BA, Wang J, Yu B, Zhang J, Bryant SH (2016) PubChem substance and compound databases. *Nucleic Acids Res* 44:D1202–D1213
- Wishart DS, Knox C, Guo AC, Shrivastava S, Hassanali M, Stothard P, Chang Z, Woolsey J (2006) DrugBank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic Acids Res* 34:D668–D672
- Lipinski CA, Lombardo F, Dominy BW, Feeney PJ (1997) Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Adv Drug Deliv Rev* 23:3–25
- Brenk R, Schipani A, James D, Krasowski A, Gilbert IH, Frearson J, Wyatt PG (2008) Lessons learnt from assembling screening libraries for drug discovery for neglected diseases. *ChemMedChem* 3:435–444
- Baell JB, Holloway GA (2010) New substructure filters for removal of pan assay interference compounds (PAINS) from screening libraries and for their exclusion in bioassays. *J Med Chem* 53:2719–2740

35. Johnson MA, Maggiora GM (1990) Concepts and applications of molecular similarity, 1st edn. Wiley, New York
36. Bender A, Glen RC (2004) Molecular similarity: a key technique in molecular informatics. *Org Biomol Chem* 2:3204
37. Bajorath J (2017) Representation and identification of activity cliffs. *Expert Opin Drug Discov* 12:879–883
38. Accelrys Inc, San Diego, CA, USA (2011) MACCS structural keys
39. Morgan HL (1965) The generation of a unique machine description for chemical structures—a technique developed at Chemical Abstracts Service. *J Chem Doc* 5:107–113
40. Rogers D, Hahn M (2010) Extended-connectivity fingerprints. *J Chem Inf Model* 50:742–754
41. Maggiora G, Vogt M, Stumpfe D, Bajorath J (2014) Molecular similarity in medicinal chemistry. *J Med Chem* 57:3186–3204
42. Butina D (1999) Unsupervised data base clustering based on Daylight's fingerprint and Tanimoto similarity: a fast and automated way to cluster small and large data sets. *J Chem Inf and Model* 39:747–750
43. RDKit (2018) RDKFingerprint. <http://rdkit.org/docs/source/rdkit.Chem.rdmolops.html>. Accessed 18 Dec 2018
44. Raymond JW, Willett P (2002) Maximum common subgraph isomorphism algorithms for the matching of chemical structures. *J Comput-Aided Mol Des* 16:521–33
45. Dalke A, Hastings J (2013) FMCS: a novel algorithm for the multiple MCS problem. *J Cheminf* 5:O6
46. Ho TK (1995) Random decision forests. In: Proceedings of 3rd international conference on document analysis and recognition, vol 1. IEEE Comput Soc Press, Los Alamitos, California, pp 278–282
47. Cortes C, Vapnik V (1995) Support-vector networks. *Mach Learn* 20:273–297
48. van Gerven M, Bohte S (2017) Editorial: artificial neural networks as models of neural information processing. *Front Comput Neurosci* 11:114
49. Wermuth CG, Ganellin CR, Lindberg P, Mitscher LA (1998) Glossary of terms used in medicinal chemistry (IUPAC Recommendations 1998). *Pure Appl Chem* 70:1129–1143
50. Seidel T, Wolber G, Murgueitio MS (2018) Pharmacophore perception and applications. *Applied chemoinformatics*. Wiley, Weinheim, pp 259–282
51. Macqueen J (1967) Some methods for classification and analysis of multivariate observations. In: 5th Berkeley symposium on mathematical statistics and probability, pp 281–297
52. Stiefl N (2016) 3D pharmacophores in the RDKit. [https://github.com/rdkit/UGM\\_2016/blob/master/Notebooks/Stiefl\\_RDKitPh4FullPublication.ipynb](https://github.com/rdkit/UGM_2016/blob/master/Notebooks/Stiefl_RDKitPh4FullPublication.ipynb). Accessed 18 Dec 2018
53. Kellenberger E, Schalon C, Rognan D (2008) How to measure the similarity between protein ligand-binding sites? *Curr Comput-Aided Drug Des* 4:209–220
54. Ehrt C, Brinkjost T, Koch O (2016) Impact of binding site comparisons on medicinal chemistry and rational molecular design. *J Med Chem* 59:4121–4151
55. Winger JA, Hantschel O, Superti-Furga G, Kuriyan J (2009) The structure of the leukemia drug imatinib bound to human quinone reductase 2 (NQO2). *BMC Struct Biol* 9:7

Ready to submit your research? Choose BMC and benefit from:

- fast, convenient online submission
- thorough peer review by experienced researchers in your field
- rapid publication on acceptance
- support for research data, including large and complex data types
- gold Open Access which fosters wider collaboration and increased citations
- maximum visibility for your research: over 100M website views per year

At BMC, research is always in progress.


Learn more [biomedcentral.com/submissions](https://biomedcentral.com/submissions)

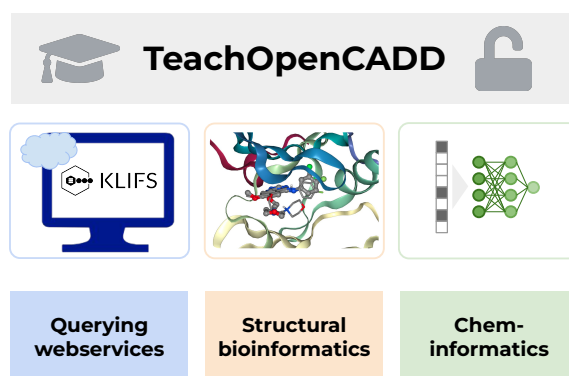


### 3.3.2 TeachOpenCADD 2022: Open Source and FAIR Python Pipelines to Assist in Structural Bioinformatics and Cheminformatics Research Publication G

This article is the follow-up publication three years after TeachOpenCADD's initial publication in 2019 [144]. Here, we outline the next batch of CADD topics composed of 12 new Jupyter Notebooks [130] with a focus on structure-based methods and database queries. This new release included also the restructuring of the GitHub repository and code to follow Python software best practices and launched a new website for easy online browsing through the TeachOpenCADD content.

 <https://github.com/volkamerlab/teachopencadd>

 <https://projects.volkamerlab.org/teachopencadd/talktorials.html#edition-2021>



Contribution:

#### Co-first author

Conceptualization (33%)

Data Curation (50%)

Formal Analysis (33%)

Investigation (33%)

Methodology (33%)

Software (50%)

Validation (33%)

Visualization (80%)

Writing — Original Draft (90%)

Writing — Review & Editing (33%)

Reprinted from Sydow D\*, Rodríguez-Guerra J\*, Kimber TB, Schaller D, Taylor CJ, Chen Y, Leja M, Misra S, Wichmann M, Ariamajd A, Volkamer A. TeachOpenCADD 2022: Open Source and FAIR Python Pipelines to Assist in Structural Bioinformatics and Cheminformatics Research. *Nucleic Acids Research*. **2022**; 50(W1):W753–W760. 10.1093/nar/gkac267 (\*contributed equally)

Open access article licensed under a CC BY 4.0 license.

# TeachOpenCADD 2022: open source and FAIR Python pipelines to assist in structural bioinformatics and cheminformatics research

Dominique Sydow<sup>†</sup>, Jaime Rodríguez-Guerra<sup>†</sup>, Talia B. Kimber<sup>‡</sup>, David Schaller<sup>‡</sup>, Corey J. Taylor<sup>‡</sup>, Yonghui Chen<sup>‡</sup>, Mareike Leja<sup>‡</sup>, Sakshi Misra<sup>‡</sup>, Michele Wichmann<sup>‡</sup>, Armin Ariamajd<sup>‡</sup> and Andrea Volkamer<sup>‡\*</sup>

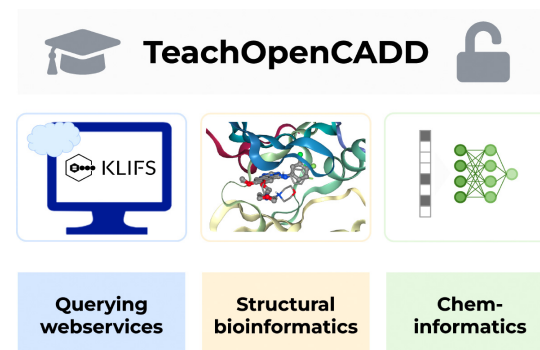
In Silico Toxicology and Structural Bioinformatics, Institute of Physiology, Charité – Universitätsmedizin Berlin, corporate member of Freie Universität Berlin and Humboldt-Universität zu Berlin, Germany

Received March 08, 2022; Revised March 30, 2022; Editorial Decision April 02, 2022; Accepted April 06, 2022

## ABSTRACT

Computational pipelines have become a crucial part of modern drug discovery campaigns. Setting up and maintaining such pipelines, however, can be challenging and time-consuming—especially for novice scientists in this domain. TeachOpenCADD is a platform that aims to teach domain-specific skills and to provide pipeline templates as starting points for research projects. We offer Python-based solutions for common tasks in cheminformatics and structural bioinformatics in the form of Jupyter notebooks, based on open source resources only. Including the 12 newly released additions, TeachOpenCADD now contains 22 notebooks that cover both theoretical background as well as hands-on programming. To promote reproducible and reusable research, we apply software best practices to our notebooks such as testing with automated continuous integration and adhering to the idiomatic Python style. The new TeachOpenCADD website is available at <https://projects.volkamerlab.org/teachopencadd> and all code is deposited on GitHub.

## GRAPHICAL ABSTRACT



## INTRODUCTION

Computational methods play an integral role in the design-make-test-analyze (DMTA) cycle that drives real-world drug design projects (1). To address questions raised during this cycle, a single method does not suffice to deliver an answer; instead, a pipeline combining different methods can produce complementary and useful insights. Setting up such complex pipelines, however, can be difficult and time-consuming for many reasons: the scientist may not have had the training necessary to tackle these tasks (2), tools and their usage are constantly evolving (or becoming deprecated), and feeding the output from one tool into another is often not straightforward. On top of these considerations, sustainable pipelines need to be findable, accessible, interoperable, and reusable (FAIR principles (3))—not only today but in many years from now—to drive reproducible research.

In 2019, we launched the teaching platform TeachOpenCADD (4) on [GitHub](#) to help face these challenges.

\*To whom correspondence should be addressed. Email: [andrea.volkamer@charite.de](mailto:andrea.volkamer@charite.de)

<sup>†</sup>The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.

## 2 Nucleic Acids Research, 2022

TeachOpenCADD teaches by example how to build Python pipelines with open source resources used in the fields of cheminformatics and structural bioinformatics to answer central questions in computer-aided drug design (CADD). With these ready-to-use pipelines, we target students and teachers who need training material for CADD-related topics, as well as researchers who need a template or an inspiration to tackle their research questions. The theoretical and practical aspects of each topic are covered in an interactive Jupyter notebook (5). This setup makes it easy for users from different fields to understand the computational concepts and to get started with hands-on Python programming. We call these Jupyter notebooks *talktorials* (talk + tutorial) because their format is suited for presentations as well. The initial stack of talktorials T001–T010 covers common CADD tasks involving webserver queries, cheminformatics, and structural bioinformatics (4). We show how to fetch chemical and structural data from the ChEMBL (6) and PDB (7,8) databases and how to encode, filter, cluster, and screen such datasets to find novel drug candidates and off-targets (4). The talktorials are inspired by several online resources recommended for further reading such as Teach-Discover-Treat and CDK (9,10) and the blogs [Practical Cheminformatics](#), [RDKit blog](#), and [Is live worth living?](#). Over the last two years, the TeachOpenCADD GitHub repository underwent many additions and changes: we now have more than doubled our content and extended the application of software best practices rigorously. The full collection of talktorials is easily accessible on the new [TeachOpenCADD website](#). We comply with software best practices regarding the code style as well as maintenance and facilitate installation with a dedicated conda package.

### NEW TALKTORIALS

The new stack of talktorials showcases data acquisition from additional CADD-relevant databases, adds many examples for structure-based tasks, and extends the cheminformatics side with straightforward deep learning (DL) applications. Our example use case is the EGFR kinase (19) but the talktorials are easily adaptable to other targets as long as sufficient data is available. Besides the domain-specific resources described below, we rely in all talktorials on established Python packages for data science and visualization such as NumPy (20), pandas (21), scikit-learn (22), matplotlib (23), and seaborn (24).

### Webservices queries

Over the last decades, the scientific community has produced an incredible amount of data and analysis software, and adapted modern technologies to make these resources easily available via online webservices (25). However, it might not always be obvious to the beginner how to use a web application programming interface (API) to access such data and how to integrate them into larger pipelines. TeachOpenCADD dedicates several talktorials to the usage of different webservices relevant for the life sciences.

In the first TeachOpenCADD release from 2019, we already showed how to query the ChEMBL (6) and PDB (7,8)

databases. From the ChEMBL webservice, compounds and bioactivities are fetched for the EGFR kinase using the ChEMBL webresource client (26) (T001). This dataset is used in many downstream talktorials for common cheminformatics tasks (T002–T007). From the PDB webservice, we fetch a set of EGFR kinase structures based on criteria such as ‘ligand-bound structures from X-ray experiments with a resolution <3.0 Å’ using the biotite (27) and PyPDB (28) (T008) packages.

In the latest release, we now have added three more notebooks covering the usage of additional online API webservices (Figure 1, T011–T013).

**T011: Querying online API webservices.** We added a broad introduction on how to programmatically use online webservices from Python with a focus on REST services and web scraping. The usage of several libraries is demonstrated; e.g. we use [requests](#) to retrieve content from UniProt (29), [bravado](#) to generate a Python client for OpenAPI-compatible services—exemplified for the KLIFS database (11)—, and [Beautiful Soup](#) to scrape (parse) HTML content from the web.

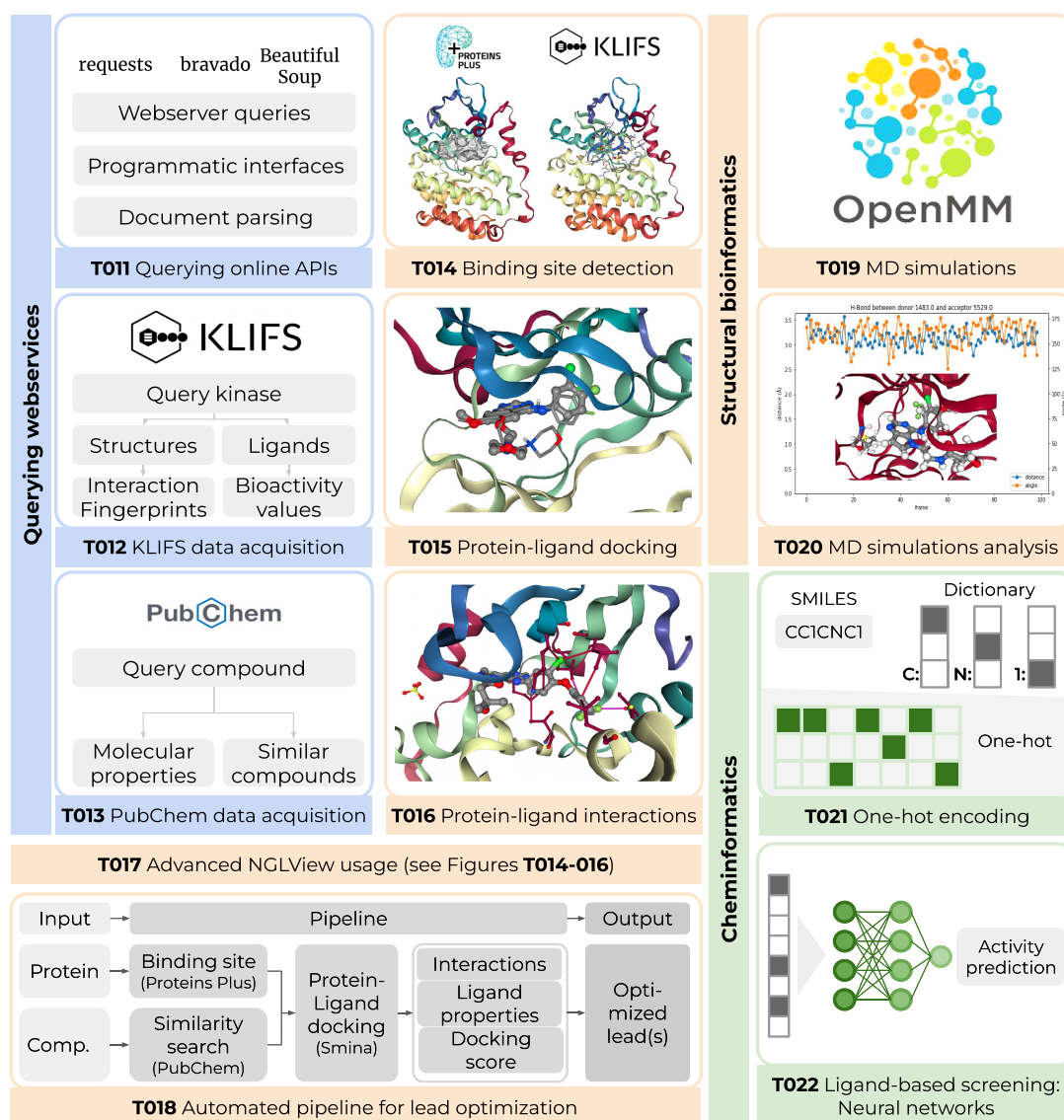
**T012: Data acquisition from KLIFS.** KLIFS (11) is a kinase database gathering information on experimental kinase structures and interacting inhibitors. The talktorial shows how to quickly fetch data from KLIFS given a query kinase or ligand. For example, we spot frequent key ligand-interactions in EGFR based on KLIFS interaction fingerprints and we assess kinome-wide bioactivity values for the inhibitor gefitinib. These queries are demonstrated by using the [KLIFS OpenAPI](#) directly with [bravado](#), or by using the KLIFS-dedicated wrapper [OpenCADD-KLIFS](#) (30), implemented in the Python package [OpenCADD](#).

**T013: Data acquisition from PubChem.** PubChem (12) is a database holding chemical information on over 100 million compounds. We demonstrate how to fetch data from PubChem’s PUG-REST API (31), given the name or SMILES (32) of a query ligand. For example, we show how to fetch molecular properties for a ligand of interest by name (aspirin) and how to query PubChem for the most similar compounds given a query SMILES (gefitinib).

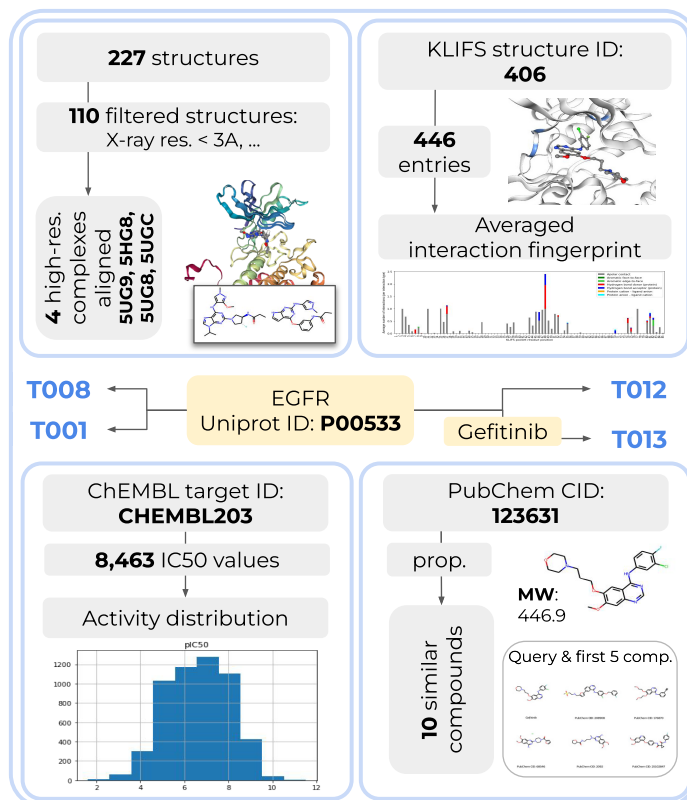
**Data acquisition case study.** A summary of the information that can be acquired automatically for a target of interest using these webservices is exemplified in Figure 2. Using the Uniprot ID of EGFR kinase as input query only, (i) 227 available EGFR structures from the PDB can be obtained and further filtered (T008); (ii) 446 available complex structures and their interaction fingerprints can be fetched from KLIFS (T012), or (iii) a total of 8463 IC50 values of molecules measured against EGFR can be acquired from ChEMBL (T001). Finally, (iv) a PubChem query with the molecule name ‘gefitinib’ showcases how to gather ligand properties or to perform a similarity search (T013).

### Pocket detection, ligand–protein docking and interactions

During a drug discovery campaign, frequent questions are: What should I test next? Can you suggest a diverse set of



**Figure 1.** Overview of 12 new talktorials. (i) Querying webservices (blue): T011 gives a broad introduction to programmatic access to webservices from Python, T012 and T013 demonstrate how to query the KLIFS (11) and PubChem (12) databases for kinase and compound data, respectively. (ii) Structural bioinformatics (orange): T014 detects the binding site in an EGFR kinase structure and compares the prediction to the binding site defined by KLIFS (11). T015 performs a re-docking for an EGFR–ligand complex with Smina (13). T016 detects protein–ligand interactions in an EGFR–ligand complex structure with PLIP (14). T017 introduces basic and advanced usages of the molecular visualization tool NGLView (15), used throughout most of TeachOpenCADD’s talktorials. T018 outlines a fully automated lead optimization pipeline: Based on an input structure, the pocket is detected and a set of compounds similar to a selected ligand are fetched from PubChem (12). These compounds are docked into the selected binding site. The most promising compounds w.r.t. docking scores and interaction profiles are proposed as optimized compounds. T019 demonstrates how to set up and run a molecular dynamics (MD) simulation on Google Colab with OpenMM (16). T020 analyzes the resulting MD trajectory with a focus on the root-mean-square deviation (RMSD) between trajectory frames and the dynamics of protein–ligand interactions using MDAnalysis (17,18). (iii) Cheminformatics (green): T021 exhibits the steps to numerically encode a small molecule from its SMILES representation. T022 lays the groundwork for deep learning and focuses on a simple feed-forward neural network for activity prediction using molecular fingerprints.



**Figure 2.** Data and information that can be automatically gathered for the EGFR kinase using the different web query talktorials as of September 2021, created based on ChEMBL v.27 (6) (T001), PDB (8) (T008), PubChem (12) (T013), and KLIFS (11) (T012). Input: yellow boxes, output: gray boxes, plots and molecule visualizations (using NGLView (15) and RDKit).

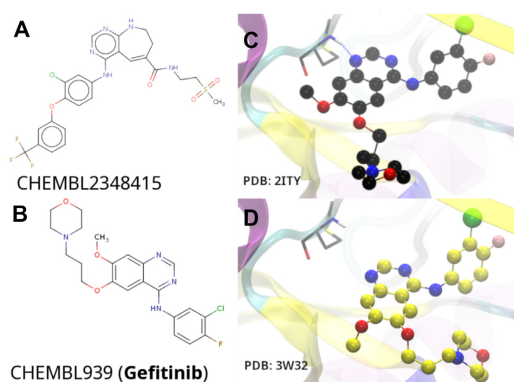
small molecules likely to bind to this protein? How should I modify the lead compound to increase the binding affinity? Answering these questions involves multiple scientific observations, and thus, multiple computational steps as addressed in talktorials T014–T017. Finally, an automated pipeline is compiled (T018) to process a protein structure and a lead compound, and propose several similar ligands with optimized estimated affinities and interactions based on the docked protein–ligand structures.

**T014: Binding site detection.** First, we need to know where ligands may bind to a protein of interest. Sometimes the binding site is known from experimental protein–ligand structures. If only experimental apo structures are available, putative binding sites can be predicted with computational methods. We demonstrate how to use the REST API of the ProteinsPlus webservice (33) to detect the main pocket of an EGFR structure using the DoGSiteScorer (34) pocket detection algorithm. To validate our results, the predicted pocket is compared with the KLIFS-defined kinase pocket, which encompasses 85 residues shown to be in contact with ligands based on X-ray complex structures (35).

**T015: Protein–ligand docking.** Next, we introduce molecular docking to predict the binding mode of a ligand to its protein target by explaining several sampling algorithms and scoring functions, as well as commenting on limitations and interpretation of docking results. The theoretical background is then applied in a re-docking experiment aiming to reproduce the binding mode observed in a published X-ray structure of EGFR. Protein and ligand are prepared using Pybel (36), the ligand is docked into the protein using Smina (13), and finally, the docking poses are visually inspected using NGLView (15). We refer to [JupyterDock](#) for further reading on different docking protocols run from Jupyter notebooks.

**T016: Protein–ligand interactions.** Understanding which forces and interactions drive molecular recognition is important for drug design (37). In this talktorial, we give an introduction to relevant protein–ligand interactions and their programmatic detection using the protein–ligand interaction profiler PLIP (14). To this end, all interactions in an EGFR–ligand complex fetched from the PDB are detected and visualized in 3D using NGLView.





**Figure 3.** Case study for talktorial T018 depicting (A) 2D structure of the input ligand for the pipeline that was used with an EGFR crystal structure (PDB: 3W32, IC<sub>50</sub> = 75nM); (B) 2D structure of gefitinib (IC<sub>50</sub> = 0.17nM), an EGFR ligand found during similarity searches; (C) crystal structure of gefitinib co-crystallized with EGFR (PDB: 2ITY, black CPK representation); (D) docked pose of gefitinib (yellow CPK representation). Some segments of the protein structure have been removed for clarity. The ligand RMSD between (C) and (D) and the discovery of a higher-affinity ligand demonstrate the utility of the fully automated pipeline for early stage drug discovery.

**T017: Advanced NGLView usage.** Since the molecular visualization package NGLView is invoked in many talktorials, we give a dedicated overview of its usage and show some advanced cases on how to customize residue coloring, and how to create interactive interfaces with IPyWidgets. In addition, access to the JavaScript layer NGL (38,39) is showcased to perform operations that are not exposed to the Python wrapper NGLView.

**T018: Automated pipeline for lead optimization.** All previous talktorials are composed of stand-alone tasks that can be completed independently. Proposing ligand modifications that will improve interaction patterns with target proteins in a complete end-to-end process, however, necessitates orchestration of code and concepts implemented in the previously discussed talktorials T014–T017. A docking pipeline is constructed in T018 that is comprised of both a step-by-step demonstration and a fully automated procedure. Given a query protein and a lead compound, similar ligands fetched from PubChem are suggested, which show optimized affinity estimates and interaction profiles based on generated docking poses.

**Lead optimization case study.** As a case study, an EGFR crystal structure (PDB: 3W32) and its co-crystallized ligand were used as inputs for the pipeline. A similarity search led to the generation of a small library of compounds from PubChem for docking and further analysis to find compounds ideally more affine than the co-crystallized ligand. Using the pipeline, an approved breast cancer drug, gefitinib, was found in the top 50 of docked poses (Figure 3). Gefitinib (IC<sub>50</sub> = 0.17 nM (40)) is at least an order of magnitude more affine for EGFR than the measured affinity of the input ligand (IC<sub>50</sub> = 75 nM (41)). Gefitinib's predicted geometry was <2 Å RMSD from a crystal structure of wild-type

EGFR (PDB: 2ITY). This retrospective example demonstrates the utility of a fully automated pipeline and potential application as prospective tool.

### Molecular dynamics

Experimentally resolved structures offer immense insights for drug design but can only provide a static snapshot of the full conformational space that represents the flexible nature of biological systems. Molecular dynamics (MD) simulations approximate such flexibility *in silico* with a trajectory of atom positions over a series of time steps (frames). These trajectories thereby reveal a more detailed—albeit still incomplete—picture of drug-target recognition and binding by providing access to protein-ligand interaction patterns over time (42–44). These insights can for example help in lead discovery to examine the stability and validity of a predicted ligand docking pose, and in lead optimization phases to estimate the effect of a chemical modification on binding affinity.

**T019: MD simulations.** We explain the key concepts behind MD simulations and provide the code to run a short MD simulation of EGFR in complex with a ligand on a local machine or on Google Colab with [condacolab](#), which allows for GPU-accelerated simulations. The protein and ligand are thereby separately prepared with [pdbfixer](#) and [RDKit](#), and subsequently combined using [MDTraj](#) (45) and [openff-toolkit](#). The simulation is performed with [OpenMM](#) (16), a high-performance toolkit for molecular simulation. The talktorial produces a 100 ps trajectory if run on Google Colab. On a local machine, only 20 fs are generated by default to keep computational efforts reasonable. We refer to the work by Arantes *et al.* (46) for further reading on different MD protocols run with [OpenMM](#) using Jupyter notebooks on Google Colab.

**T020: Analyzing MD simulations.** We analyze and visualize the trajectory using the Python packages [MDAnalysis](#) (17,18) and [NGLView](#). First, the protein is structurally aligned across all trajectory frames, followed by calculating the root-mean-square deviation (RMSD) for different system components, i.e. protein, backbone, and ligand. Then, we take a closer look at a selected interaction between ligand and protein atoms, showcasing the contribution of distance and angle to the hydrogen bond strengths.

### Deep learning

Machine learning and more specifically deep learning have gained in popularity over the last few decades thanks to powerful computational resources (GPUs), novel algorithms, and the growing amount of available data (47). Applications to CADD are diverse, ranging from molecular property prediction (48) to *de novo* molecular design (49). Here, the focus is the featurization of molecular entities (T021) and ligand-based screening (T022).

**T021: One-hot encoding.** In CADD, machine learning algorithms require as input a numerical representation of small molecules. Besides molecular fingerprints (see T004),

## 6 Nucleic Acids Research, 2022

a popular featurization is the SMILES notation (32). However, these representations are composed of strings and therefore cannot simply be input to an algorithm. One-hot encoding provides a solution for SMILES usage, explained in T021.

*T022: Ligand-based screening: neural networks.* We introduce the basics of neural networks and build a simple two-layer neural network. A model is trained on a subset of ChEMBL data to predict the pIC50 values of compounds against EGFR using MACCS keys as input. This talktorial is meant as groundwork for the understanding of neural networks. More complex architectures such as convolutional and recurrent neural networks will be explored in future notebooks. Such models may use the one-hot encoding of SMILES as input (50).

### BEST PRACTICES

We provide reliable and reproducible TeachOpenCADD pipelines, periodically checked via automated testing mechanisms, and a streamlined and easy-to-understand code style across all talktorials.

*Testing.* Reproducibility is ensured by testing if the notebooks can run without errors and whether the output of specific operations can be reproduced. For this purpose, we use the tools `pytest` and `nbval`.

*Continuous integration.* We are testing the talktorials regularly for Linux, OSX, and Windows and different Python versions on [GitHub Actions](#). This ensures identical behavior across different operating systems and Python versions and also spots issues like conflicting dependency updates or changing outputs.

*Repository structure.* The repository structure is based on the `cookiecutter-cms` template, which provides a Python-focused project scaffold with pre-configured settings for packaging, continuous integration, `Sphinx`-based documentation, and much more. We have adapted the template to our notebook-focused needs.

*Code style.* We aim to adhere to the [PEP8](#) style guide for Python code, which defines how to write idiomatic Python (Pythonic) code. Such rules are important so that new developers—or in our case talktorial users—can quickly read and understand the code. Furthermore, we use `black-nb` to format the Python notebooks compliant with PEP8.

### TEACHOPENCADD USAGE

There are many ways to use the talktorials. If users simply want to go through the material, they can use the read-only website version. If users would rather like to execute and modify the Jupyter notebooks, this can be done online thanks to the [Binder](#) integrations or locally using the new conda package.

*New website.* Firing up Jupyter notebooks can entail unexpected complications if one wants to simply read through a talktorial. To make the access easy and fast, we launched a new [TeachOpenCADD website](#). The website statically renders the talktorials for immediate online reading using `sphinx-nb` and provides detailed documentation for local usage, contributions and external resources.

*New Binder support.* The [Binder project](#) offers a place to share computing environments via a single link. The environment setup of TeachOpenCADD can take a couple of minutes but does not require any kind of action on the user's end. This access option is recommended if the user plans on executing the material but does not need to save the changes.

*New conda package.* To make the local installation of TeachOpenCADD as easy as possible, we offer a conda package that ships all Jupyter notebooks with all necessary dependencies. The installation instructions are lined out in the [TeachOpenCADD documentation](#). This access option is recommended if the user plans on adapting the material for individual use cases.

### CONCLUSION

The increasing amount of data and the focus on data-driven methods call for reproducible and reliable pipelines for computer-aided drug design (CADD). Knowing how to access and use these resources programmatically, however, requires domain-specific training and inspiration. The TeachOpenCADD platform showcases webserver-based data acquisition and common tasks in the fields of cheminformatics and structural bioinformatics. The theoretical and programmatic aspects of each topic are outlined side-by-side in Jupyter notebooks (talktorials) using open source resources only. To foster FAIR research, we apply software best practices such as testing, continuous integration, and idiomatic coding throughout the whole project. The talktorials are accessible via our website, Binder, and conda package to accommodate different use cases such as reading, executing, and modifying, respectively. We believe that TeachOpenCADD is not only a rich resource for CADD pipelines and teaching material on computational concepts and programming but as well a good example of how to set up websites, automated testing, and packaging for notebook-centric repositories. TeachOpenCADD is a living resource; problems can be voiced via GitHub issues and contributions can be made in the form of pull requests on GitHub. TeachOpenCADD is meant to grow; everyone is welcome to add new topics. Whenever you explore a new topic for your work, we invite you to fill our talktorial template with what one learns along the way and to submit it to TeachOpenCADD.

### DATA AVAILABILITY

- TeachOpenCADD website: <https://projects.volkamerlab.org/teachopencadd/>.
- TeachOpenCADD GitHub repository: <https://github.com/volkamerlab/teachopencadd>.

## ACKNOWLEDGEMENTS

The authors thank Piedro Gerletti (T019), Ahmed Atta (T022), Melanie Vogel (T018), Abishek Laxmanan Ravi Shankar (T014), Maria Trofimova (T015) and Jeffrey R. Wagner (T019) for the initial drafts or contributions to the above-mentioned talktorials.

The authors are grateful to the PyPDB and biotite maintainers for their work on updating their packages according to the new RCSB PDB API (special thanks to Patrick Kunzmann), and Albert Kooistra for helping with questions regarding the KLIFS database. The authors thank Hai Nguyen for dedicated and helpful NGLView support. The authors are thankful for a fruitful hackathon at the RDKit UGM in 2019, where we started to tackle a few TeachOpenCADD enhancements: structural superimposition and visualization without PyMol with Richard Gowers and testing Jupyter notebooks with Floriane Montanari. Finally, the authors thank Pat Walters and Hai Nguyen for endorsing the TeachOpenCADD platform on their websites, and we thank Greg Landrum for giving TeachOpenCADD a spot at the RDKit UGMs 2019, 2020, and 2021. *Author contributions:* Conceptualization: D.S., J.R.G., A.V.; Data Curation, Formal Analysis, Investigation, Software, Validation and Visualization: D.S., J.R.G., T.B.K., Da.S., C.T., Y.C., M.L., S.M., M.W., A.A., A.V.; Funding Acquisition: A.V.; Methodology and Maintenance: D.S., J.R.G., T.B.K., Da.S., A.V.; Project Administration: D.S., J.R.G., A.V.; Resources: A.V.; Supervision: D.S., J.R.G., Da.S., T.B.K., A.V.; Writing-Original Draft: D.S., T.B.K., Da.S., J.R.G., A.V.; Writing - Review and Editing: D.S., J.R.G., T.B.K., Da.S., C.T., Y.C., M.L., S.M., M.W., A.A., A.V.

## FUNDING

Note that the TeachOpenCADD project has been a group effort and has received no explicit funding, while the positions of individual authors were supported by diverse funding agencies; the Volkamer Lab received funding from the Bundesministerium für Bildung und Forschung [031A262C to A.V.]; Deutsche Forschungsgemeinschaft [VO 2353/1-1 to D.S.]; Stiftung Charité in the context of the Einstein BIH Visiting Fellow Project [to T.B.K., J.R.G. and C.T.]; Bayer in the context of the MIAME project (DaS); China Scholarship Council Project [201906210079 to Y.C.]. We acknowledge financial support from the Open Access Publication Fund of Charité - Universitätsmedizin Berlin and the German Research Foundation (DFG).

*Conflict of interest statement.* None declared.

## REFERENCES

- Schneider,P., Walters,W.P., Plowright,A.T., Sieroka,N., Listgarten,J., Goodnow,R.A., Fisher,J., Jansen,J.M., Duca,J.S., Rush,T.S. *et al.* (2020) Rethinking drug design in the artificial intelligence era. *Nat. Rev. Drug Disc.*, **19**, 353–364.
- Ringer McDonald,A. (2021) In: *Teaching Programming across the Chemistry Curriculum. Teaching Programming across the Chemistry Curriculum: A Revolution or a Revival?* American Chemical Society pp. 1–11.
- Wilkinson,M.D., Dumontier,M., Aalbersberg,I.J., Appleton,G., Axton,M., Baak,A., Blomberg,N., Boiten,J.-W., da Silva Santos,L.B., Bourne,P.E. *et al.* (2016) The FAIR guiding principles for scientific data management and stewardship. *Scientific Data*, **3**, 160018.
- Sydow,D., Morger,A., Driller,M. and Volkamer,A. (2019) TeachOpenCADD: A Teaching Platform For Computer-Aided Drug Design Using Open Source Packages And Data. *J. Cheminform.*, **11**, 29.
- Kluyver,T., Ragan-Kelley,B., Pérez,F., Granger,B., Bussonnier,M., Frederic,J., Kelley,K., Hamrick,J., Grout,J., Corlay,S. *et al.* (2016) Jupyter Notebooks - A Publishing Format For Reproducible Computational Workflows. In: Loizides,F. and Schmidt,B. (eds). *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. IOS Press, Netherlands, pp. 87–90.
- Mendez,D., Gaulton,A., Bento,A.P., Chambers,J., De Veij,M., Félix,E., Magariños,M., Mosquera,J., Mutowo,P., Nowotka,M. *et al.* (2018) ChEMBL: towards direct deposition of bioassay data. *Nucleic Acids Res.*, **47**, D930–D940.
- Berman,H.M., Westbrook,J., Feng,Z., Gilliland,G., Bhat,T.N., Weissig,H., Shindyalov,I.N. and Bourne,P.E. (2000) The Protein Data Bank. *Nucleic Acids Res.*, **28**, 235–242.
- Burley,S.K., Bhikadiya,C., Bi,C., Bittrich,S., Chen,L., Crichlow,G.V., Christie,C.H., Dalenberg,K., Di Costanzo,L., Duarte,J.M. *et al.* (2020) RCSB Protein Data Bank: powerful new tools for exploring 3D structures of biological macromolecules for basic and applied research and education in fundamental biology, biomedicine, biotechnology, bioengineering and energy sciences. *Nucleic Acids Res.*, **49**, D437–D451.
- Riniker,S., Landrum,G., Montanari,F., Villalba,S., Maier,J., Jansen,J., Walters,W. and Shelat,A. (2018) Virtual-screening workflow tutorials and prospective results from the Teach-Discover-Treat competition 2014 against malaria [version 2; peer review: 3 approved]. *F1000Research*, **6**, 1136.
- Willighagen,E.L., Mayfield,J.W., Alvarsson,J., Berg,A., Carlsson,L., Jeliazkova,N., Kuhn,S., Pluskal,T., Rojas-Chertó,M., Spjuth,O. *et al.* (2017) The Chemistry Development Kit (CDK) v2.0: atom typing, depiction, molecular formulas, and substructure searching. *J. Cheminform.*, **9**, 33.
- Kanev,G.K., de Graaf,C., Westerman,B.A., de Esch,I. J.P. and Kooistra,A.J. (2020) KLIFS: an overhaul after the first 5 years of supporting kinase research. *Nucleic Acids Res.*, **49**, D562–D569.
- Kim,S., Chen,J., Cheng,T., Gindulyte,A., He,J., He,S., Li,Q., Shoemaker,B.A., Thiessen,P.A., Yu,B. *et al.* (2020) PubChem in 2021: new data content and improved web interfaces. *Nucleic Acids Res.*, **49**, D1388–D1395.
- Koes,D.R., Baumgartner,M.P. and Camacho,C.J. (2013) Lessons learned in empirical scoring with smina from the CSAR 2011 benchmarking exercise. *J. Chem. Inf. Model.*, **53**, 1893–1904.
- Salentin,S., Schreiber,S., Haupt,V.J., Adasme,M.F. and Schroeder,M. (2015) PLIP: fully automated protein–ligand interaction profiler. *Nucleic Acids Res.*, **43**, W443–W447.
- Nguyen,H., Case,D.A. and Rose,A.S. (2017) NGLView - Interactive Molecular Graphics For Jupyter Notebooks. *Bioinformatics*, **34**, 1241–1242.
- Eastman,P., Swails,J., Chodera,J.D., McGibbon,R.T., Zhao,Y., Beauchamp,K.A., Wang,L.-P., Simmonett,A.C., Harrigan,M.P., Stern,C.D. *et al.* (2017) OpenMM 7: Rapid development of high performance algorithms for molecular dynamics. *PLoS Comput. Biol.*, **13**, e1005659.
- Michaud-Agrawal,N., Denning,E.J., Woolf,T.B. and Beckstein,O. (2011) MDAAnalysis: a toolkit for the analysis of molecular dynamics simulations. *J. Comput. Chem.*, **32**, 2319–2327.
- Gowers,R.J., Linke,M., Barnoud,J., Reddy,T.J.E., Melo,M.N., Seyler,S.L., Domański,J., Dotson,D.L., Buchoux,S., Kenney,I.M. *et al.* (2016) MDAAnalysis: A Python Package for the Rapid Analysis of Molecular Dynamics Simulations. In: Sebastian,B. and Scott,R. (eds). *Proceedings of the 15th Python in Science Conference*. pp. 98–105.
- Herbst,R.S. (2004) Review of epidermal growth factor receptor biology. *Int. J. Radiat. Oncol.*, **59**(Suppl. 2), S21–S26.
- Harris,C.R., Millman,K.J., van der Walt,S.J., Gommers,R., Virtanen,P., Cournapeau,D., Wieser,E., Taylor,J., Berg,S., Smith,N.J. *et al.* (2020) Array programming with NumPy. *Nature*, **585**, 357–362.
- McKinney,W. (2010) Data structures for statistical computing in Python. In: van der Walt, S. and Millman,J. (eds). *Proceedings of the 9th Python in Science Conference*. pp. 56–61.
- Pedregosa,F., Varoquaux,G., Gramfort,A., Michel,V., Thirion,B., Grisel,O., Blondel,M., Prettenhofer,P., Weiss,R., Dubourg,V. *et al.*


8 *Nucleic Acids Research*, 2022


- (2011) Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.*, **12**, 2825–2830.
23. Hunter, J.D. (2007) Matplotlib: A 2D graphics environment. *Comput. Sci. Eng.*, **9**, 90–95.
  24. Waskom, M.L. (2021) seaborn: statistical data visualization. *J. Open Source Softw.*, **6**, 3021.
  25. Ireland, S.M. and Martin, A. C.R. (2021) GraphQL for the delivery of bioinformatics web APIs and application to ZincBind. *Bioinformatics Adv.*, **1**, vbab023.
  26. Davies, M., Nowotka, M., Papadatos, G., Dedman, N., Gaulton, A., Atkinson, F., Bellis, L. and Overington, J.P. (2015) ChEMBL Web Services: Streamlining Access To Drug Discovery Data And Utilities. *Nucleic Acids Res.*, **43**, W612–W620.
  27. Kunzmann, P. and Hamacher, K. (2018) Biotite: a unifying open source computational biology framework in Python. *BMC Bioinformatics*, **19**, 346.
  28. Gilpin, W. (2015) PyPDB: a Python API for the Protein Data Bank. *Bioinformatics*, **32**, 159–160.
  29. Consortium, T.U. (2020) UniProt: the universal protein knowledgebase in 2021. *Nucleic Acids Res.*, **49**, D480–D489.
  30. Sydow, D., Rodríguez-Guerra, J. and Volkamer, A. (2022) OpenCADD-KLIFS: A Python package to fetch kinase data from the KLIFS database. *J. Open Source Softw.*, **7**, 3951.
  31. Kim, S., Thiessen, P.A., Cheng, T., Yu, B. and Bolton, E.E. (2018) An update on PUG-REST: RESTful interface for programmatic access to PubChem. *Nucleic Acids Res.*, **46**, W563–W570.
  32. Weininger, D. (1988) SMILES, A Chemical Language And Information System. 1. Introduction To Methodology And Encoding Rules. *J. Chem. Inf. Model.*, **28**, 31–36.
  33. Fährrolfes, R., Bietz, S., Flachsenberg, F., Meyder, A., Nittinger, E., Otto, T., Volkamer, A. and Rarey, M. (2017) ProteinsPlus: a web portal for structure analysis of macromolecules. *Nucleic Acids Res.*, **45**, W337–W343.
  34. Volkamer, A., Kuhn, D., Grombacher, T., Rippmann, F. and Rarey, M. (2012) Combining global and local measures for structure-based druggability predictions. *J. Chem. Inf. Model.*, **52**, 360–372.
  35. van Linden, O. P.J., Kooistra, A.J., Leurs, R., de Esch, I. J.P. and de Graaf, C. (2014) KLIFS: a knowledge-based structural database to navigate kinase–ligand interaction space. *J. Med. Chem.*, **57**, 249–277.
  36. O’Boyle, N.M., Banck, M., James, C.A., Morley, C., Vandermeersch, T. and Hutchison, G.R. (2011) Open Babel: an open chemical toolbox. *J. Cheminformatics*, **3**, 33.
  37. Klebe, G. (2013) In: *Drug Design: Methodology, Concepts, and Mode-of-Action chapter Protein–Ligand Interactions as the Basis for Drug Action*. Springer Berlin Heidelberg, pp. 61–88.
  38. Rose, A.S. and Hildebrand, P.W. (2015) NGL Viewer: a web application for molecular visualization. *Nucleic Acids Res.*, **43**, W576–W579.
  39. Rose, A.S., Bradley, A.R., Valasatava, Y., Duarte, J.M., Prlić, A. and Rose, P.W. (2018) NGL viewer: web-based molecular graphics for large complexes. *Bioinformatics*, **34**, 3755–3758.
  40. Kawakita, Y., Seto, M., Ohashi, T., Tamura, T., Yusa, T., Miki, H., Iwata, H., Kamiguchi, H., Tanaka, T., Sogabe, S. *et al.* (2013) Design and synthesis of novel pyrimido[4,5-b]azepine derivatives as HER2/EGFR dual inhibitors. *Bioorg. Med. Chem.*, **21**, 2250–2261.
  41. Yang, J., Tu, Z., Xu, X., Luo, J., Yan, X., Ran, C., Mao, X., Ding, K. and Qiao, C. (2017) Novel conjugates of endoperoxide and 4-anilinoquinazoline as potential anticancer agents. *Bioorgan. Med. Chem. Lett.*, **27**, 1341–1345.
  42. Mortier, J., Rakers, C., Bermudez, M., Murgueitio, M.S., Riniker, S. and Wolber, G. (2015) The impact of molecular dynamics on drug design: applications for the characterization of ligand–macromolecule complexes. *Drug Discov. Today*, **20**, 686–702.
  43. De Vivo, M., Masetti, M., Bottegioni, G. and Cavalli, A. (2016) Role of molecular dynamics and related methods in drug discovery. *J. Med. Chem.*, **59**, 4035–4061.
  44. Salmasso, V. and Moro, S. (2018) Bridging molecular docking to molecular dynamics in exploring ligand–protein recognition process: an overview. *Front. Pharm.*, **9**, 923.
  45. McGibbon, R.T., Beauchamp, K.A., Harrigan, M.P., Klein, C., Swails, J.M., Hernández, C.X., Schwantes, C.R., Wang, L.-P., Lane, T.J. and Pande, V.S. (2015) MDTraj: a modern open library for the analysis of molecular dynamics trajectories. *Biophys. J.*, **109**, 1528–1532.
  46. Arantes, P.R., Polêto, M.D., Pedebos, C. and Ligabue-Braun, R. (2021) Making it rain: cloud-based molecular simulations for everyone. *J. Chem. Inf. Model.*, **61**, 4852–4856.
  47. Goodfellow, I., Bengio, Y. and Courville, A. (2016) In: *Deep Learning*. MIT Press.
  48. Wu, Z., Ramsundar, B., Feinberg, E.N., Gomes, J., Geniesse, C., Pappu, A.S., Leswing, K. and Pande, V. (2018) MoleculeNet: a benchmark for molecular machine learning. *Chem. Sci.*, **9**, 513–530.
  49. Brown, N., Fiscato, M., Segler, M.H. and Vaucher, A.C. (2019) GuacaMol: benchmarking models for de novo molecular design. *J. Chem. Inf. Model.*, **59**, 1096–1108.
  50. Kimber, T.B., Engelke, S., Tetko, I.V., Bruno, E. and Godin, G. (2018) Synergy effect between convolutional neural networks and the multiplicity of SMILES for improvement of molecular prediction. arXiv doi: <https://arxiv.org/abs/1812.04439>, 11 December 2018, preprint: not peer reviewed.

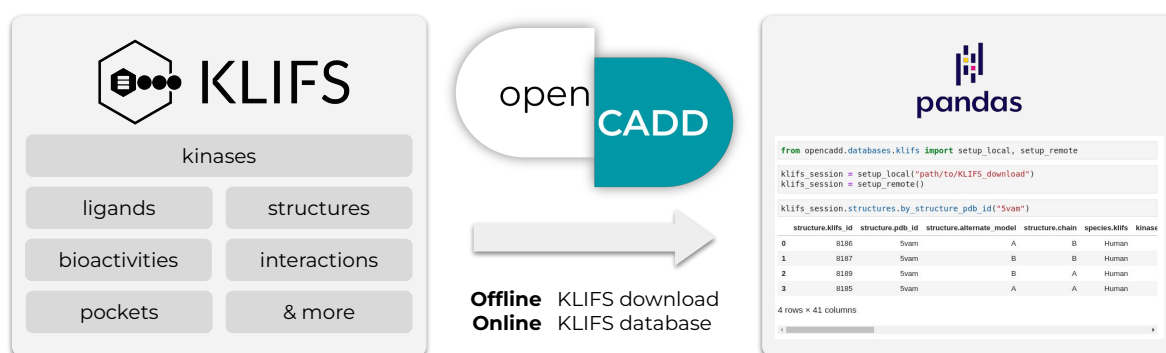


### 3.3.3 OpenCADD-KLIFS: A Python Package to Fetch Kinase Data from the KLIFS Database Publication H

This software paper covers OpenCADD-KLIFS, a Python module that allows easy access to data from the KLIFS database [63]. KLIFS data types such as kinases, structures, ligands, bioactivities, interactions, pockets, and more can be fetched with a clean and user-friendly Python API in the form of Pandas DataFrames (tables) [93]. This setup was extensively used in several of the kinase-focused projects of this thesis by providing a faster, more reproducible, and easier-to-maintain code base, circumventing the need for code duplications.

 <https://github.com/volkamerlab/opencadd>

 [https://opencadd.readthedocs.io/en/latest/databases\\_klifs.html](https://opencadd.readthedocs.io/en/latest/databases_klifs.html)



Contribution:

#### First author

Conceptualization (90%)

Software (90%)

Validation (90%)

Visualization (90%)

Writing — Original Draft (90%)

Writing — Review & Editing (90%)

Reprinted from [Sydow D, Rodríguez-Guerra J, Volkamer A. OpenCADD-KLIFS: A Python Package to Fetch Kinase Data from the KLIFS Database. \*Journal of Open Source Software\*. 2022; 7\(70\):3951. 10.21105/joss.03951](#)

Open access article licensed under a CC BY 4.0 license.



## OpenCADD-KLIFS: A Python package to fetch kinase data from the KLIFS database

Dominique Sydow<sup>\*1</sup>, Jaime Rodríguez-Guerra<sup>1</sup>, and Andrea Volkamer<sup>†1</sup>

<sup>1</sup> *In Silico* Toxicology and Structural Bioinformatics, Institute of Physiology, Charité – Universitätsmedizin Berlin, corporate member of Freie Universität Berlin and Humboldt-Universität zu Berlin, Augustenburger Platz 1, 13353 Berlin, Germany

DOI: [10.21105/joss.03951](https://doi.org/10.21105/joss.03951)

### Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

**Editor:** Richard Gowers ↗

### Reviewers:

- [@ojeda-e](#)
- [@andrewtarzia](#)
- [@mcs07](#)

**Submitted:** 09 November 2021

**Published:** 17 February 2022

### License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).

### Summary

Protein kinases are involved in most aspects of cell life due to their role in signal transduction. Dysregulated kinases can cause severe diseases such as cancer, inflammation, and neurodegeneration, which has made them a frequent target in drug discovery for the last decades (Cohen et al., 2021). The immense research on kinases has led to an increasing amount of kinase resources (Kooistra & Volkamer, 2017). Among them is the KLIFS database, which focuses on storing and analyzing structural data on kinases and interacting ligands (Kanev et al., 2020). The OpenCADD-KLIFS Python module offers a convenient integration of the KLIFS data into workflows to facilitate computational kinase research.

OpenCADD-KLIFS (`opencadd.databases.klifs`) is part of the OpenCADD package, a collection of Python modules for structural cheminformatics.

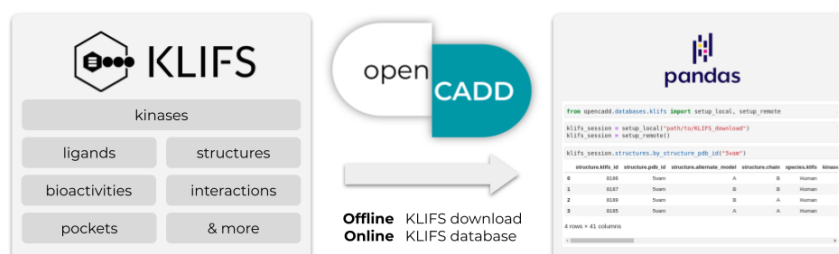
### Statement of need

The KLIFS resource (Kanev et al., 2020) contains information about kinases, structures, ligands, interaction fingerprints, and bioactivities. KLIFS thereby focuses especially on the ATP binding site, defined as a set of 85 residues and aligned across all structures using a multiple sequence alignment (van Linden et al., 2014). Fetching, filtering, and integrating the KLIFS content on a larger scale into Python-based pipelines is currently not straight-forward, especially for users without a background in online queries. Furthermore, switching between data queries from a *local* KLIFS download and the *remote* KLIFS database is not readily possible.

OpenCADD-KLIFS is aimed at current and future users of the KLIFS database who seek to integrate kinase resources into Python-based research projects. With OpenCADD-KLIFS, KLIFS data can be queried either locally from a KLIFS download or remotely from the KLIFS webserver. The presented module provides identical APIs for the remote and local queries and streamlines all output into standardized Pandas DataFrames (The pandas development team, 2020) to allow for easy and quick downstream data analyses (Figure 1). This Pandas-focused setup is ideal if you work with Jupyter notebooks (Kluyver et al., 2016).

<sup>\*</sup>corresponding author

<sup>†</sup>corresponding author



**Figure 1:** OpenCADD-KLIFS fetches KLIFS data (Kanev et al., 2020) offline from a local KLIFS download or online from the KLIFS database and formats the output as user-friendly Pandas DataFrames (The pandas development team, 2020).

## State of the field

The KLIFS database is unique in the structure-based kinase field in terms of integrating and annotating different data resources in a kinase- and pocket-focused manner. Kinases, structures, and ligands have unique identifiers in KLIFS, which makes it possible to fetch and filter cross-referenced information for a query kinase, structure, or ligand.

- Kinase structures are fetched from the PDB, split by chains and alternate models, annotated with the KLIFS pocket of 85 residues, and aligned across the fully structurally covered kinome.
- Kinase-ligand interactions seen in experimental structures are annotated for the 85 pocket residues in the form of the KLIFS interaction fingerprint (KLIFS IFP).
- Bioactivity data measured against kinases are fetched from ChEMBL (Mendez et al., 2018) and linked to kinases, structures, and ligands available in KLIFS.
- Kinase inhibitor metadata are fetched from the PKIDB (Carles et al., 2018) and linked to co-crystallized ligands available in KLIFS.

The KLIFS data integrations and annotations can be accessed in different ways, which are all open source:

- Manually via the [KLIFS website](#) interface: This mode is preferable when searching for information on a specific structure or smaller set of structures.
- Automated via the [KLIFS KNIME](#) nodes (Kooistra et al., 2018; McGuire et al., 2017): This mode is extremely useful if the users' projects are embedded in KNIME workflows; programming is not needed.
- Programmatically using the REST API and KLIFS OpenAPI specifications: This mode is needed for users who seek to perform larger scale queries or to integrate different queries into programmatic workflows. In the following, we will discuss this mode in context of Python-based projects and explain how OpenCADD-KLIFS improves the user experience.

The KLIFS database offers standardized URL schemes (REST API), which allows users to query data by defined URLs, using e.g., the Python package `requests` (requests, 2021). Instead of writing customized scripts to generate such KLIFS URLs, the KLIFS OpenAPI specifications, a document that defines the KLIFS REST API scheme, can be used to generate a Python client, using e.g., the Python package `bravado` (bravado, 2021). This client offers a Python API to send requests and receive responses. This setup is already extremely useful, however,





it has a few drawbacks: the setup is technical; the output is not easily readable for humans and not ready for immediate downstream integrations, requiring similar but not identical reformatting functions for different query results; and switching from remote requests to local KLIFS download queries is not possible. Facilitating and streamlining these tasks is the purpose of OpenCADD-KLIFS as discussed in more detail in the next section.

## Key Features

The KLIFS database offers a REST API compliant with the OpenAPI specification (KLIFS, 2021). Our module OpenCADD-KLIFS uses bravado to dynamically generate a Python client based on the OpenAPI definitions and adds wrappers to enable the following functionalities:

- A session is set up automatically, which allows access to various KLIFS *data sources* by different *identifiers* with the API `session.data_source.by_identifier`. *Data sources* currently include kinases, structures and annotated conformations, modified residues, pockets, ligands, drugs, and bioactivities; *identifiers* refer to kinase names, PDB IDs, KLIFS IDs, and more. For example, `session.structures.by_kinase_name` fetches information on all structures for a query kinase.
- The same API is used for local and remote sessions, i.e., interacting with data from a KLIFS download folder and from the KLIFS website, respectively.
- The returned data follows the same schema regardless of the session type (local/remote); all results obtained with bravado are formatted as Pandas DataFrames with standardized column names, data types, and handling of missing data.
- Files with the structural 3D coordinates deposited on KLIFS include full complexes or selections such as proteins, pockets, ligands, and more. These files can be downloaded to disc or loaded via biopandas (Raschka, 2017) or RDKit (RDKit, 2021).

OpenCADD-KLIFS is especially convenient whenever users are interested in multiple or more complex queries such as “fetching all structures for the kinase EGFR in the DFG-in conformation” or “fetching the measured bioactivity profiles for all ligands that are structurally resolved in complex with EGFR.” Formatting the output as DataFrames facilitates subsequent filtering steps and DataFrame merges in case multiple KLIFS datasets need to be combined.

OpenCADD-KLIFS is currently used in several projects from the Volkamer Lab (Volkamer Lab, 2021) including TeachOpenCADD (TeachOpenCADD, 2021), OpenCADD-pocket (OpenCADD, 2021), KiSSim (KiSSim, 2021), KinoML (OpenKinome, 2021), and PLIPify (PLIPify, 2021). For example, OpenCADD-KLIFS is applied in a TeachOpenCADD tutorial to demonstrate how to fetch all kinase-ligand interaction profiles for all available EGFR kinase structures to visualize the per-residue interaction types and frequencies with only a few lines of code.

## Acknowledgements

We thank the whole KLIFS team for providing such a great kinase resource with an easy-to-use API and especially Albert Kooistra for his help with questions and wishes regarding the KLIFS database. We thank David Schaller for his feedback on the OpenCADD-KLIFS module. We acknowledge the contributors involved in software programs and packages used by OpenCADD-KLIFS, such as bravado, RDKit, Pandas, Jupyter, and Pytest, and Sphinx.

## References

bravado. (2021). bravado. In *GitHub repository*. GitHub. <https://github.com/Yelp/bravado>



- Carles, F., Bourg, S., Meyer, C., & Bonnet, P. (2018). PKIDB: A Curated, Annotated and Updated Database of Protein Kinase Inhibitors in Clinical Trials. *Molecules*, 23(4), 908. <https://doi.org/10.3390/molecules23040908>
- Cohen, P., Cross, D., & Jänne, P. A. (2021). Kinase drug discovery 20 years after imatinib: Progress and future directions. *Nature Reviews Drug Discovery*, 20(7), 551–569. <https://doi.org/10.1038/s41573-021-00195-4>
- Kanev, G. K., de Graaf, C., Westerman, B. A., de Esch, I. J. P., & Kooistra, A. J. (2020). KLIFS: an overhaul after the first 5 years of supporting kinase research. *Nucleic Acids Research*, 49(D1), D562–D569. <https://doi.org/10.1093/nar/gkaa895>
- KiSSim. (2021). KiSSim: Subpocket-based fingerprint for kinase pocket comparison. In *GitHub repository*. GitHub. <https://github.com/volkamerlab/kissim>
- KLIFS. (2021). *KLIFS OpenAPI*. <https://dev.klifs.net>. [https://dev.klifs.net/swagger\\_v2/](https://dev.klifs.net/swagger_v2/)
- Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., Willing, C., & team, J. development. (2016). Jupyter notebooks - a publishing format for reproducible computational workflows. In F. Loizides & B. Schmidt (Eds.), *Positioning and power in academic publishing: Players, agents and agendas* (pp. 87–90). IOS Press. <https://eprints.soton.ac.uk/403913/>
- Kooistra, A. J., Vass, M., McGuire, R., Leurs, R., Esch, I. J. P. de, Vriend, G., Verhoeven, S., & Graaf, C. de. (2018). 3D-e-Chem: Structural Cheminformatics Workflows for Computer-Aided Drug Discovery. *ChemMedChem*, 13(6), 614–626. <https://doi.org/10.1002/cmdc.201700754>
- Kooistra, A. J., & Volkamer, A. (2017). Chapter six - kinase-centric computational drug development. In R. A. Goodnow (Ed.), *Platform technologies in drug discovery and validation* (Vol. 50, pp. 197–236). Academic Press. <https://doi.org/10.1016/bs.armc.2017.08.001>
- McGuire, R., Verhoeven, S., Vass, M., Vriend, G., Esch, I. J. P. de, Lusher, S. J., Leurs, R., Ridder, L., Kooistra, A. J., Ritschel, T., & Graaf, C. de. (2017). 3D-e-chem-VM: Structural cheminformatics research infrastructure in a freely available virtual machine. *Journal of Chemical Information and Modeling*, 57(2), 115–121. <https://doi.org/10.1021/acs.jcim.6b00686>
- Mendez, D., Gaulton, A., Bento, A. P., Chambers, J., De Veij, M., Félix, E., Magariños, M. P., Mosquera, J. F., Mutowo, P., Nowotka, M., Gordillo-Marañón, M., Hunter, F., Junco, L., Mugumbate, G., Rodriguez-Lopez, M., Atkinson, F., Bosc, N., Radoux, C. J., Segura-Cabrera, A., ... Leach, A. R. (2018). ChEMBL: towards direct deposition of bioassay data. *Nucleic Acids Research*, 47(D1), D930–D940. <https://doi.org/10.1093/nar/gky1075>
- OpenCADD. (2021). OpenCADD-Pocket: Identification and analysis of protein (sub)pockets. In *GitHub repository*. GitHub. <https://github.com/volkamerlab/opencadd>
- OpenKinome. (2021). KinoML: Structure-informed machine learning for kinase modeling. In *GitHub repository*. GitHub. <https://github.com/openkinome/kinoml>
- PLIPify. (2021). PLIPify: Protein-ligand interaction frequencies across multiple structures. In *GitHub repository*. GitHub. <https://github.com/volkamerlab/plipify>
- Raschka, S. (2017). BioPandas: Working with molecular structures in pandas DataFrames. *The Journal of Open Source Software*, 2(14). <https://doi.org/10.21105/joss.00279>
- RDKit. (2021). RDKit: Open-Source Cheminformatics. In *RDKit website*. RDKit. <http://www.rdkit.org>
- requests. (2021). requests. In *GitHub repository*. GitHub. <https://github.com/psf/requests>



- TeachOpenCADD. (2021). TeachOpenCADD: a teaching platform for computer-aided drug design (CADD) using open source packages and data. In *GitHub repository*. GitHub. <https://github.com/volkamerlab/teachopencadd>
- The pandas development team. (2020). Pandas-dev/pandas: pandas. In *Zenodo repository*. Zenodo. <https://doi.org/10.5281/zenodo.3509134>
- van Linden, O. P. J., Kooistra, A. J., Leurs, R., Esch, I. J. P. de, & Graaf, C. de. (2014). KLIFS: A knowledge-based structural database to navigate kinase–ligand interaction space. *Journal of Medicinal Chemistry*, 57(2), 249–277. <https://doi.org/10.1021/jm400378w>
- Volkamer Lab. (2021). Volkamer Lab website. In *Volkamer Lab website*. Volkamer Lab. <https://volkamerlab.org/>



# Chapter 4

## Discussion

### 4.1 Computational Target Prediction

Target identification is the initial step in early drug discovery campaigns and aims to define disease-relevant targets as well as to determine the most similar targets to a target of interest. Identifying similar targets is useful for (i) finding similar structurally resolved targets for homology modeling, if no structural information is available for the target of interest, and (ii) detecting potential off-targets to inform selective drug design. Other applications involve understanding the target’s mode of action, polypharmacology, and possibilities for drug repurposing. Computational approaches have the potential to save time and costs during target identification.

In the following, I discuss the remaining challenges in the field (Publication A [22]), while outlining how we addressed these challenges in the context of kinase-focused research (Publications B–D [95, 141, 142]) and in the context of unpublished work on the proteome-wide binding site comparison method Ratar (Appendix 5.2.1).

#### 4.1.1 Remaining Challenges of State-of-the-Art Approaches

Targets can be predicted in many different ways, i.e., from a ligand, structure, or hybrid point of view, which we reviewed in detail in **Publication A** [22] (Section 1.2.1). While the field of computational target prediction has made great progress and reported a plethora of methods, five main challenges remain that are of technical nature and include method and data availability. We discuss how this thesis has addressed these challenges in the following (and in more detail in the next Section 4.2).

**Challenge 1: FAIR methods.** Although we have many published methods at our disposal, only a limited number is practically available to us for incorporation into a pipeline. Many methods are not available at all or need a license or manual request, while others are only accessible via a webserver or are technically difficult to set up. All methods reported in this thesis are developed based on FAIR principles, i.e., they are findable, accessible, interoperable, and reusable. For example, the kinase-focused off-target prediction tools as discussed in Publications B and D [95, 141] are freely available on GitHub [108], distributed as conda packages on conda-forge [146], and operate on Linux, macOS, and Windows.

**Challenge 2: Explainable hits.** Method results are often not easily interpretable; similarity scores determine the ranking of targets but rationalizing the ranking can be difficult. For example, the scores from alignment-free binding site comparison methods do not explain which encoded features are responsible for high or low similarity scores. Hence, the structurally informed comparison between binding sites is not possible although it is important to translate

the results into rational design decisions. For this reason, the KiSSim method presented in Publication B [141] offers a 3D visualization of (dis)similarities between kinases on a residue level.

**Challenge 3: Target data availability.** Computational target prediction is limited by data availability; structure-based binding site comparison will only detect similarities between structurally resolved binding sites (e.g., roughly 300 out of 500 kinases are structurally resolved). The structural gaps within the proteome are addressed by research and industry groups as well as organizations such as the Structural Genomics Consortium [147] but this takes time. In the meantime, it is worth considering multiple perspectives during target identification as we suggest in Publications C and D [95, 142]; kinase similarities are here viewed not only based on pocket structures but also based on pocket sequences, protein-ligand interactions, and ligand profiles. Furthermore, predicted structures from AlphaFold2 [148] could be considered as well to fill the structural gaps. However, studies have shown that the predicted binding sites are overall not accurate enough and hence the predicted structures should at this stage probably only be used after pocket-specific refinement and manual curation [149, 150].

**Challenge 4: Activity cliffs.** Underlying principles such as "similar ligands bind similar targets" and "similar pockets bind similar ligands" are often useful estimates but are not always true as known for so-called activity cliffs, where two ligands with only slight chemical differences show massive changes in their activity towards the desired target [151]. Alternatively, two highly similar targets can show different ligand profiles. In this thesis, we propose to help detect such activity cliffs by not relying upon a single similarity measure but on a set of orthogonal similarity measures as outlined in Publications C and D [95, 142], as previously argued in the context of challenge 3.

**Challenge 5: Target flexibility.** Proteins are flexible; they exist in an ensemble of conformational states. However, only a subset of these states is receptive to ligand binding. Ideally, target prediction methods capture targets in those relevant conformational states. Some methods consider flexibility implicitly by a coarse-grained pocket encoding step or by a tolerant or partial matching (comparison) step as outlined in Publication A [22]. Furthermore, flexibility can be represented by protein ensembles based on experimentally determined structures or molecular dynamics simulations. In the case of the KiSSim method presented in Publication C [142], flexibility is considered in terms of experimentally resolved kinase ensembles. This is advantageous for kinases with a high and representative structural coverage, while the flexibility of kinases with only one or a few structures remains un- oder underexplored.

In the next section, we discuss the KiSSim methodology and kinase similarity pipeline presented in this thesis in more detail (Publications B–D), including considerations of these five challenges.

## 4.2 Predicting Kinome-Wide (Sub)Pocket-Based Off-Targets

Kinome-wide (off-)target prediction is a key step in early-stage kinase drug design campaigns to define desired kinase profiles and their tractability as well as to identify undesired off-targets. Although a plethora of methods has been published for computational target prediction, many of them are not publicly available and none are tailored to kinases, a protein class with high structural coverage and a highly conserved binding site. The KLIFS database processes kinase structures to provide residue-by-residue alignments of kinase pockets. Instead of applying kinase-unspecific binding site comparison methods, we aimed to use this pocket knowledge to enable kinase-specific and fast binding site representation and comparison.

The resulting KiSSim procedure and analyses are fully open-sourced within the `kissim` package [152] and the `kissim_app` repository [153], respectively, which makes it possible to include the KiSSim method in other applications and to reproduce all data (addressing challenge 1 in Section 4.1).

#### 4.2.1 KiSSim: Enabling Kinase-Specific Encoding and Comparison

We developed the kinase- and subpocket-focused tool KiSSim in **Publication B** [141] (Section 3.1.1) with the aim to (i) encode their binding site more accurately, and (ii) enable a simple setup, maintenance, and incorporation into a larger pipeline.

The KiSSim fingerprint is composed of physicochemical and spatial bits encoding each of the 85 KLIFS pocket residues, which are aligned across the full structurally covered kinome and can therefore easily be compared bit by bit. Physicochemical features include each residue’s size, hydrogen bond donors/acceptors, charge, aromatic and aliphatic properties as well as the side chain orientation and solvent exposure. Spatial bits include the distance of each residue’s C $\alpha$  atom to important subpocket centers, i.e., the pocket centroid, the hinge region, the DFG region, and the front pocket. Each subpocket’s distance distribution is furthermore described by its first three moments (following the USR approach [154] as described in Appendix 5.2.1). The resulting 1032-bit fingerprint can be directly used for pairwise kinase pocket comparisons.

In Publication B, we showed that the fingerprints’ feature space (i) reflects sequence-related similarities between kinases on a generalized level through the defined physicochemical properties, and (ii) catches information on reported flexible and stable regions through the defined spatial properties, which reflects the differences in 3D space of crystallized structures.

To map kinase-kinase relationships, we performed an all-against-all comparison for 4112 structures covering 257 kinases. The resulting  $4112 \times 4112$  structure distance matrix was reduced to a  $257 \times 257$  kinase distance matrix by representing each kinase pair with the most similar structure pair amongst these kinases’ structural ensemble (addressing challenge 5 in Section 4.1). The resulting kinase distance matrix is visualized in the form of a circular phylogenetic tree.

This KiSSim dataset covers all structurally resolved kinases (of satisfying quality as defined in [141]) in the DFG-in conformation; restricting the kinase conformation to the predominant DFG-in conformation reflects the research focus on type I and I $^{1/2}$  inhibitors and covers the majority ( $\sim 85\%$ ) of PDB kinase structures. Alternatively, we also provided KiSSim datasets including all conformations or DFG-out conformations only, while users can produce KiSSim datasets with any other structure subset of interest using the open-sourced `kissim` Python package [152]. To include also kinases without any resolved structure, predicted kinase structures such as those proposed by AlphaFold2 [148] could be included in the future. We recommend flagging such structures for the user to indicate that the kinase similarity predictions are based on one or more predicted structures. The following considerations are based on the KiSSim dataset composed of DFG-in structures.

#### 4.2.2 KiSSim: Detecting Expected and Unexpected Kinase Relationships

Based on the phylogenetic tree of the structurally covered kinome, we showed that the KiSSim dataset retrieves the sequence-based kinome tree by Manning et al. [67] including TK and CMGC subbranches, which is probably attributed to the physicochemical KiSSim fingerprint bits that generalize the pocket sequence. In contrast, some kinases show inter-group proximities, of which some can be rationalized such as the CaMKK2 and DRAK2 off-target relationship [155]. Thus, the addition of structural information in the KiSSim fingerprint allows us to group more

distantly related kinases together.

To explain KiSSim results, we implemented a 3D visualization that colors the kinase pocket residue by residue with the following values (addressing challenge 2 in Section 4.1): (i) the KiSSim fingerprint bits, allowing us to investigate physicochemical and spatial properties within a pocket, (ii) the difference between two KiSSim fingerprint bits, allowing us to understand the kinase similarity on a residue level, and (iii) the standard deviation of KiSSim fingerprints from kinase structure ensembles, allowing us to detect spatial variations within experimentally resolved structures of a single kinase. These visual aids can guide the design of selective ligands during lead optimization phases.

We evaluated KiSSim’s specificity and sensitivity with profiling data [97, 98, 101] as a surrogate for expected kinase (dis)similarities. Profiling-based evaluation has its shortcomings such as unbalanced data availability per kinase-ligand pair and different experimental setups across profiling datasets (see discussion in [141]). Nonetheless, this profiling-based perspective reflects in a retrospective way how the KiSSim approach is applied to real-world questions, i.e., how well can the KiSSim fingerprint reflect a ligand’s kinase profile. Across 48 kinase-ligand pairs involving 21 ligands, the AUCs range from 0.49 to 1.0 with a mean of  $0.75 \pm 0.12$ . For example, KiSSim was able to explain Erlotinib/EGFR’s unexpected off-targets LOK and SLK (both STE kinase group) but not the off-target GAK ("Other" group). Furthermore, the method was able to retrieve all of Imatinib/ABL1’s off-targets LCK, KIT, and FMS (TK group).

We compared KiSSim to other kinase similarity measures, i.e., the KLIFS pocket sequence and interaction fingerprints (IFPs) as defined in detail in the next section, as well as the protein-wide binding site comparison tool SiteAlign, whose size and pharmacological residue features have been implemented in the KiSSim fingerprint. We observed the following: (i) KiSSim compares well with these established methods while often improving predictions between kinase pairs without an obvious relationship based on the sequence. (ii) The pocket sequence- and IFP-based methods are much faster than the structure-based methods KiSSim and SiteAlign, however, the overall kinase similarity assessment benefits from the added structural pocket information. (iii) KiSSim’s setup and runtime are more convenient and faster than the SiteAlign method while yielding slightly better results. In contrast to SiteAlign, KiSSim relies on KLIFS’ 85-residue pocket alignment. On the one hand, this is advantageous because the method builds on curated residue-by-residue alignments in the KLIFS database and therefore allows fast and kinase-tailored comparisons. On the other hand, this restricts comparisons to the residue-by-residue KLIFS alignment and (at this point) excludes the comparison of structures without the KLIFS assignment. The latter disadvantage could be solved by providing the KLIFS alignment as functionality within the KiSSim methodology or —preferably— as functionality within the KLIFS database itself to allow its usage in other applications as well.

In the next two sections, we motivate why and outline how the KiSSim methodology should and can be integrated with other similarity measures to mitigate individual method shortcomings and data scarcity by using complementary data resources covering structural, chemical, and pharmacological datasets.

### 4.2.3 Assessing Kinase Similarity from Different Perspectives

In a study, which was conducted before the development of the KiSSim method, we saw that different perspectives on kinase similarity can yield complementary insights on kinase relationships. The study was conducted in collaboration with the Kolb Lab in Marburg, Germany, and is described in **Publication C** [142] (Section 3.1.2).

The initial goal of this study was to find selective kinase inhibitors with a specific profile of



on- and off-targets. Candidate ligands were determined based on docking screens and assayed to determine their experimental binding affinities. Compared to previous studies the resulting hit rates were low, which prompted a re-analysis of the selected kinase profiles concerning kinase similarities. We assessed the similarities between EGFR, ErbB2, p110a (PI3K), KDR (VEGFR2), BRAF, CDK2, LCK, MET, and p38a based on different measures:

- (i) **"Pocket sequence" similarity** was defined as the identity between the 85 KLIFS pocket residues of two kinases.
- (ii) **"Pocket structure" similarity** was defined as the similarity between two kinase pockets as detected with LigSite [156] and calculated using an extension of the graph-based CavBase method [157, 158].
- (iii) **"Interaction fingerprint" (IFP) similarity** was defined as the Tanimoto similarity between two KLIFS interaction fingerprints, which describe interactions between the 85 pocket residues and associated co-crystallized ligands.
- (iv) **"Ligand profile" similarity** was defined as the ratio of the number of compounds that are active on both kinases divided by the total number of compounds that are tested on both kinases.

While the overall trend of calculated similarities is conserved across the different perspectives, individual conclusions regarding selected kinase profiles differ. For example, the high similarity between the TK kinases EGFR and ErbB2 and their low similarity is overall conserved to the atypical kinase p110a (PI3K), while it is less pronounced based on the "pocket structure". In fact, the pocket structure perspective showed comparably low similarity between EGFR and ErbB2 (which are known to be highly similar), while showing comparably high similarities to BRAF. Furthermore, while the "ligand profile" and "pocket sequence" would favor a profile with on-targets EGFR and KDR (VEGFR2) and off-target BRAF, the other two perspectives would not.

Based on these findings and the observations from our KiSSim evaluation compared to other methods, we argue that it is advantageous to consider kinase similarity from multiple perspectives that —ideally— cover multiple data sources. Therefore, we decided to build an automated pipeline that calculates kinase similarities based on the measures presented in Publication C, while exchanging the kinase-unspecific CavBase method with the open-sourced and kinase-specific KiSSim method.

#### 4.2.4 Integrating Kinase Similarity Measures as an Automated Pipeline

The findings from Publications B and C led to the idea of an integrated pipeline that calculates similarity measures from different perspectives including the KiSSim encoding as outlined in **Publication D** [95] (Section 3.1.3)

We developed a pipeline composed of Jupyter Notebooks that allows users to define their kinase set of interest based on UniProt IDs. Their similarities are thereafter measured with the following similarity methods as outlined in Publication C if not otherwise specified: (i) KLIFS pocket sequence similarity, (ii) KLIFS pocket structure similarity using the novel kinase-specific KiSSim method, (iii) KLIFS IFP similarity, and (iv) ligand profile similarity. These different approaches are based on different data sources, addressing two challenges: This multi-perspective can compensate for missing data points, e.g., sequence data is available even if a kinase is unexplored in structural and profiling data; it might also flag activity cliffs, e.g.,

pockets might be highly similar but still not bind the same ligand, which could be indicated by the ligand profile or IFPs similarity if such data is available (challenges 3 and 4 in Section 4.1).

As the final step, similarity matrices from the previous perspectives are collected and compared in a final Jupyter Notebook with easy-to-understand visualizations such as heatmaps and dendrograms. Additionally, an equally weighted average can be computed to combine distance and similarity matrices from all four perspectives, yielding a single heatmap and dendrogram. This pipeline has been published within the TeachOpenCADD platform, which is described in more detail in Section 4.4.

The setup of this kinase similarity pipeline and its integration into the TeachOpenCADD platform has several advantages: (i) The chosen similarity measures are commonly used; for kinase research, this pipeline can be used out-of-the-box in the context of KLIFS structures and summarizes the protocol for these tasks in one place. (ii) Thanks to its integration into the TeachOpenCADD platform, which is discussed in more detail in Section 4.4, this kinase similarity pipeline is maintained within a larger software project and offers greater visibility for potential users. (iii) Thanks to its modular setup, additional similarity measures of interest can be added to this pipeline, following the same logic as for the existing measures. Such additional measures can remain with the user or can be integrated into the TeachOpenCADD platform.

#### 4.2.5 Generalizing Pocket Comparison Concepts from the Kinome to Proteome

KiSSim’s advantage—the residue-by-residue comparison based on the KLIFS alignment—also has a downside: This approach is restricted to kinases and therefore cannot detect similarities of kinases to non-kinases, which are also relevant to off-target considerations. Existing kinase-unrestricted tools and their challenges have been discussed in Publication A and Section 4.1: a FAIR, fast, and pipeline-integrable comparison method with interpretable results is still missing.

In the following, we present the first implementation of a novel binding site comparison tool, Ratar, that encodes binding sites based on distance distributions to defined reference points within the pocket similarly to the KiSSim approach; since we cannot define subpockets proteome-wide as we did for kinases with KiSSim, we follow the definition of reference points as described for the Ultrafast Shape Recognition (USR) method [154].

The Ratar project transfers the principles of the fast and transformation-invariant encoding USR method from ligands to binding sites. USR encodes the distances between ligand atoms to defined ligand reference points, while its extension ElectroShape includes the atoms’ charge as a 4<sup>th</sup> dimension. In the context of Ratar, different USR derivatives for binding sites (instead of ligands) have been implemented, as well as an extension that incorporates more physicochemical information in the form of Z-scales [159]. The performance of these baseline methods yields an average AUC of about 0.61 on FuzCav’s dataset of similar and dissimilar binding sites [160]. This is a good first step but requires further work to improve the method’s discriminative power.

In the next step of this project, the encoding procedure is intended to be applied—instead of to the full binding site as currently implemented—to overlapping binding site patches. During the binding site comparison step, such a procedure could detect more fine-grained regional similarities; this could improve the performance as well as the explainability of results because similarities could be traced back to specific binding site regions (addressing challenge 2 in Section 4.1). The baseline Ratar methods have been implemented as part of the open-sourced `ratar` Python package [161], which follows FAIR principles and allows for fast encoding with less than half a second per structure (addressing challenge 1 in Section 4.1).

## 4.3 Exploring Kinome-Wide Subpocket Fragment Spaces

Drug design for kinases is challenging: New drugs need to (i) compete against mM levels of ATP, (ii) be highly selective, (iii) be flat and hydrophobic, two challenging properties for later stages in drug development, and (iv) be novel because the IP space is restrictive due to 20 years of pharmaceutical research. Fragment-based drug design has been shown to help with at least the latter two challenges due to its sampling character and has contributed to producing two FDA-approved kinase inhibitors [26]. Since kinases are so well-studied, a vast amount of structural data is available, which can be exploited for a data-driven *in silico* fragmentation and recombination strategy.

In the previous Section 4.2, we used the characteristics of KLIFS kinase pockets for off-target prediction. In this section, we assess the pockets in the context of structurally resolved bound ligands for fragment-based drug design. The KinFragLib method described in **Publication E** [143] (Section 3.2.1) (i) fragments co-crystallized kinase ligands *in silico* with respect to the subpockets that they occupy, (ii) explores the chemical space of the resulting fragment subpocket pools, and (iii) uses these fragment pools for subpocket-guided recombination. We applied this procedure to about 2500 human, DFG-in, and non-covalent kinase-ligand complexes from the KLIFS database, whose 85 pocket residues are aligned and therefore easily comparable across the structurally resolved kinome.

### 4.3.1 KinFragLib: Fragmenting Kinase Inhibitors to Explore Subpockets

For KinFragLib’s subpocket-based fragmentation of structure-bound kinase inhibitors, we defined the following kinase-specific subpockets (Figure 1 in [143]): The adenine pocket (AP) lays next to the hinge region where ligands form crucial hydrogen bonds. Next to AP is the solvent-exposed (SE) subpocket and the partially solvent-exposed front pocket (FP). In the back cleft, next to the  $\alpha$ C-helix are the back pockets 1 and 2 (B1 and B2), which are connected to the front cleft via the narrow gate area (GA). Furthermore, we define certain subpocket connectivities based on observations in the KLIFS dataset.

The fragmentation algorithm is mostly based on the RDKit [131] toolkit (Figure 2 in [143]): We calculate the subpockets for a ligand at hand, which undergoes an initial (test) fragmentation based on the BRICS [57] algorithm. Each fragment is assigned to its closest subpocket center before the ligand undergoes a second (final) fragmentation where only those bonds are cut that connect two fragments from different subpockets. This results in fragments with subpocket labels and dummy atoms that link back to the subpocket that they were connected to, which is relevant for recombination.

We ran this fragmentation procedure for about 2500 complexes. This populated the subpocket pools with over 7000 fragments, of which about 60% are duplicates, since many PDB entries belong to structure-activity relationship (SAR) studies, resulting in about 3000 fragments after deduplication. All ligands occupy AP, followed by FP, SE, and GA, while only a few of them bind to the back cleft since most ligands are front cleft binders. Half of the ligands bind to three subpockets, followed by two and four. The KinFragLib fragment library is freely available on GitHub alongside all performed analyses in the form of Jupyter Notebooks [162]; this framework can be used to zoom from these statistics into the fragmentation of individual ligands (Figure 4 in [143]). More generally, an analysis of the most common fragments per subpocket showed typical hinge binders in the AP subpocket, small and lipophilic fragments in the narrow GA subpocket, while more soluble fragments dominate in the FP and SE subpockets (Figure 6 in [143]).

### 4.3.2 KinFragLib: Recombining Fragments for Novel Kinase Inhibitors

After having analyzed the subpocket fragment space, all matching fragment combinations were enumerated based on a structurally diverse set of "Rule of Three" (Ro3)-compliant [54] (and hinge-like AP) fragments. Recombination always started at AP, since all ligands bind here, and was only allowed if following the BRICS rules. The procedure was terminated if no open bonds were left or if four fragments were already combined to avoid large compounds.

To reduce computational cost, we selected 600 diverse fragments (cluster representatives from a subset of deduplicated, Ro3-compliant, and "hinge-like" fragments). Their recombination resulted in over 6 million molecules. (i) Over 60% of the recombined molecules comply with Lipinski's "Rule of Five" (Ro5). (ii) We were able to reconstruct 35 exact and 324 substructure matches in our original KLIFS ligands, confirming that we can correctly re-assemble our input ligands. (iii) We showed that we generated mostly novel molecules; a standardized InChI string comparison to ChEMBL25 [163, 164], with about 1.8 million molecules, found about 200 exact matches (excluding the matches in (ii)). (iv) Amongst the hits from (iii), we found 47 molecules with reported human targets (based on ChEMBL bioactivities  $\leq 500$  nM), including 3 non-kinase and 44 kinase targets, of which 10 kinase targets show bioactivities in the low nM range. In summary, we demonstrated KinFragLib's recombination power by generating over 4 million novel and Ro5-compliant molecules, re-assembling input ligands, and designing molecules with reported kinase inhibitors that were not part of the original ligand set (Figure 7 in [143]).

### 4.3.3 Addressing Limitations of the KinFragLib Approach

The KinFragLib fragment and recombined molecule datasets have been used and adapted since their publication in 2019 to address two open challenges that are highly relevant for real-world drug design campaigns:

- Which compounds are relevant in terms of synthesizability?
- Which compound subset should we extract to sample a specific chemical space or to sample a diverse set of molecules from the whole chemical space?

**Assessing molecule synthesizability.** In her master thesis, Sonja Leo refined and extended the fragment library filtering, which was supervised by Andrea Volkamer, Jérémie Mortier, and myself [165]. The developed Custom-KinFragLib pipeline allows for customizable filtering steps: (i) remove unwanted substructures that can cause mutagenic, reactive, or other unfavorable effects [166] or non-specific interactions with assays (PAINS) [167], (ii) keep only drug- and fragment-like molecules based on the Ro3 and the Quantitative Estimate of Druglikeness (QED) [168], (iii) check for synthesizability, on the one hand, by keeping only commercially available building blocks from the Enamine REAL Space [169] using the DataWarrior software [170] and, on the other hand, by avoiding hard-to-synthesize molecules with the Synthetic Bayesian Accessibility (SYBA) tool [171], and (iv) check for retrosynthetic pathways with the ASKCOS [172] model.

Applying all filters reduced the over 7000 fragments to about 400; however, thanks to the modular setup of the filtering pipeline, the user can decide which filtering steps to include. In the future, further feasibility scores could be included such as the Synthetic Accessibility score (SAscore) [173] to rate fragments by how often they are in PubChem, the Synthetic Complexity score (SCscore) [174] to compare molecules with reactants from Reaxys [175], and the Retrosynthetic Accessibility score (RAScore) [176] to indicate if a retrosynthetic route can be found or not.

**Visualizing and navigating chemical spaces.** The KinFragLib datasets come with Jupyter-Notebook-based visualization functionalities, however navigating interactively through the whole fragment and recombined ligand space was not included. The ChemInformatics Model Explorer (CIME) can serve as a solution to this problem. This tool was published by Humer et al. [177] in 2022 and is a freely available and interactive web-based system that allows users to inspect chemical data sets and more. The authors used the KinFragLib datasets to demonstrate how CIME can explore KinFragLib’s chemical space by creating a UMAP [178] projection of one property, while coloring and highlighting molecules by other properties.

In their first example, they showed that CIME can visualize that a latent space representation of the KinFragLib fragments can better predict the solubility of FP fragments than their ECFP [179] representation. In their second example, they showed how to detect recombined ligand space regions that are densely populated with compounds highly similar to existing compounds. In summary, CIME is an interesting visualization tool that can help to select fragments or recombined molecules with user-defined constraints.

#### 4.3.4 Addressing Future Applications of the KinFragLib Approach

The KinFragLib datasets could be used for the following future applications: (i) Some sub-pocket fragment pools have defined characteristics and could therefore be used as focused screening libraries not only for kinases but also for other target groups. For example, the AP sub-pocket pool could be used for bioisosteric replacements of hydrogen bond donor/acceptor patterns during hit optimization for any target. (ii) Instead of recombining all fragments, one or more fragments of interest could be defined as a starting point. For example, the user could start with one or more interesting fragments in the AP pocket and enumerate all connecting sub-pocket fragments. Such a setup is currently already possible using the method’s command-line interface (CLI). (iii) At this point, KinFragLib does not check explicitly for 3D compatibility of recombined fragments but implies that fragments from the neighboring subpockets should cover a similar space. To be more rigorous, the distance between to-be-combined dummy atoms could be checked and their combination only be allowed within a certain threshold. Furthermore, the recombined molecule could be transferred or docked to the target binding site to check for clashes and binding pose. This idea is similar to the BREED [59] algorithm, which might be adaptable to KinFragLib’s needs.

## 4.4 FAIR Pipelines and Tools in Kinase-Centric Drug Design

Computational pipelines and toolkits play a crucial role in modern drug discovery projects. The design-make-test-analyze (DMTA) cycle [180] is a dynamic and time-sensitive endeavor to progress with target campaigns and demands an interplay between many disciplines and approaches. Computational drug design supports this process ideally with a customized pipeline that often combines different methods from different toolkits and datasets from diverse resources.

The setup of such a complex pipeline can be difficult and time-consuming; feeding output data from one tool as input to another is often not straightforward, data curation and standardization is not trivial, tool documentation is not always user-friendly, and even finding a suitable tool is sometimes hampered by simply not knowing the correct terminology. Once a pipeline is set up, development shifts to maintenance: toolkits and databases change or sometimes deprecate causing broken pipelines, while users will always find another bug, requiring consistent pipeline support. Last but not least, the (very welcome) advent of FAIR principles and software best practices adds another layer of necessary skills.

### 4.4.1 TeachOpenCADD: Distributing a FAIR Platform for CADD Pipelines

In 2019, we launched the teaching platform TeachOpenCADD on GitHub to (i) provide Python code examples of common tasks in computer-aided drug design (CADD), which (ii) are set up as pipelines to answer frequent research questions and (iii) use exclusively open source resources to make the material accessible to everyone (with a computer and internet). Such a platform can help with many of the aforementioned challenges by providing domain-specific pipeline templates and teaching software best practices by example. The topics cover cheminformatics and structural bioinformatics tasks, as well as life-science-focused database queries. Each topic covers both theoretical background and practical programming in a single Jupyter Notebook called talktorials (talk + tutorial) because they can be used as a tutorial but also for presentations (denoted as T001 for the first talktorial). The material can be accessed via the read-only TeachOpenCADD website [181], executed online via the Binder integration [182, 183], or executed locally via the TeachOpenCADD conda package [146].

We published **Publication F** [144] (Section 3.3.1) in 2019 with an initial stack of 10 talktorials mainly focused on topics from cheminformatics. In 2022, we published **Publication G** [145] (Section 3.3.2) with another 12 talktorials extending on topics from structural bioinformatics and database queries. Also in 2022, we released a kinase similarity edition of 6 talktorials with **Publication D** [95] (Section 3.1.3) as already discussed in Section 4.2. The talktorials use the kinase EGFR as an example but they are adaptable to other kinases and protein groups (except for kinase-specific topics). As of September 2022, TeachOpenCADD covers 28 topics (Figure 4.1):

In terms of *database queries*, we show how to communicate with the ChEMBL [71] (T001), PDB [70] (T010), KLIFS [63] (T012), and PubChem [135, 184] (T013) databases and offer a general talktorial on online API webservices (T011). In a case study, we demonstrate the collection of EGFR kinase data from all these databases (Figure 2 in [145]).

In terms of *cheminformatics*, we show how to filter a compound dataset (retrieved from ChEMBL in T001) using Lipinski’s Ro5 (T002) and flag unwanted substructures that can cause mutagenic, reactive, or other unfavorable effects [166] or non-specific interactions with assays (PAINS) [167] (T003). We show how to perform a similarity search for ligand-based screening (T004), cluster compounds based on their similarity (T005), find the maximum common substructure within the largest set of molecules as clustered in T004 (T006), and build machine learning models to predict if ligands are active or not for a specific kinase (T007). We also introduce one-hot encoding to represent ligands (T021) and utilize neural networks for ligand-based screening (T022).

In terms of *structural bioinformatics*, we offer the following topics: binding site comparison (T010), binding site detection (T014), protein-ligand docking (T015), protein-ligand interaction detection (T016), molecular dynamics simulations and their analysis (T019 and T020), and ligand-based ensemble pharmacophore modeling (T009). Throughout most of our structure-based topics, we utilize the NGLview [80, 81] tool, whose features are introduced in a stand-alone talktorial (T017). Finally, we propose an end-to-end pipeline that optimizes an input lead molecule based on the best interaction profile from automated docking of similar molecules in PubChem (T018).

The latest edition to TeachOpenCADD comprises a set of talktorials covering different similarity measures as already discussed in Section 4.2. For a set of kinases (T023), we show how to calculate similarities based on the KLIFS pocket sequences (T024), the KiSSim pocket structure fingerprints (T025), the KLIFS interaction fingerprints (T026), and kinase ligand profiles (T027). We summarize the results in a final talktorial for in-depth comparison of these different

perspectives (T028). Such a pipeline is applicable to any set of kinases and could be a default task in the early stages of kinase projects. Note that the structure-based perspectives T025 and T026 are currently only applicable to structures deposited in the KLIFS database.

Topics T001–T008, i.e., the ChEMBL and PDB queries as well as the early cheminformatics topics, were also translated into KNIME workflows as described in **Publication I** [185] (Appendix 5.1.1). This format is optimal for users without coding experience, who seek to solve these tasks in a drag-and-drop mode. Workflows can be assembled by stringing together small pre-implemented code units (nodes) with predefined functionalities.

The TeachOpenCADD talktorials aim to reach novices to the field from all scientific disciplines; users with a computer science background might spend more time in the CADD theory section, while users with a life science background might spend more time in the programming section. The material can be used as teaching or learning material in different settings such as the general classroom, individual student projects, or self-study; we have outlined potential teaching setups in **Publication J** [186] (Section 5.1.2), a book chapter as part of the "Teaching Programming across the Chemistry Curriculum" series. Furthermore, the talktorials provide also a good starting point to support research questions, either by making use of parts of the pipeline as a whole or by reusing only selected code bits for smaller operations. This versatility of the platform makes it interesting to a wide audience within the community, as exemplified by frequently posted GitHub issues, about 18000 article views [187], over 380 GitHub repository stars (as of 2022-09-27) [188], and teaching feedback [186].

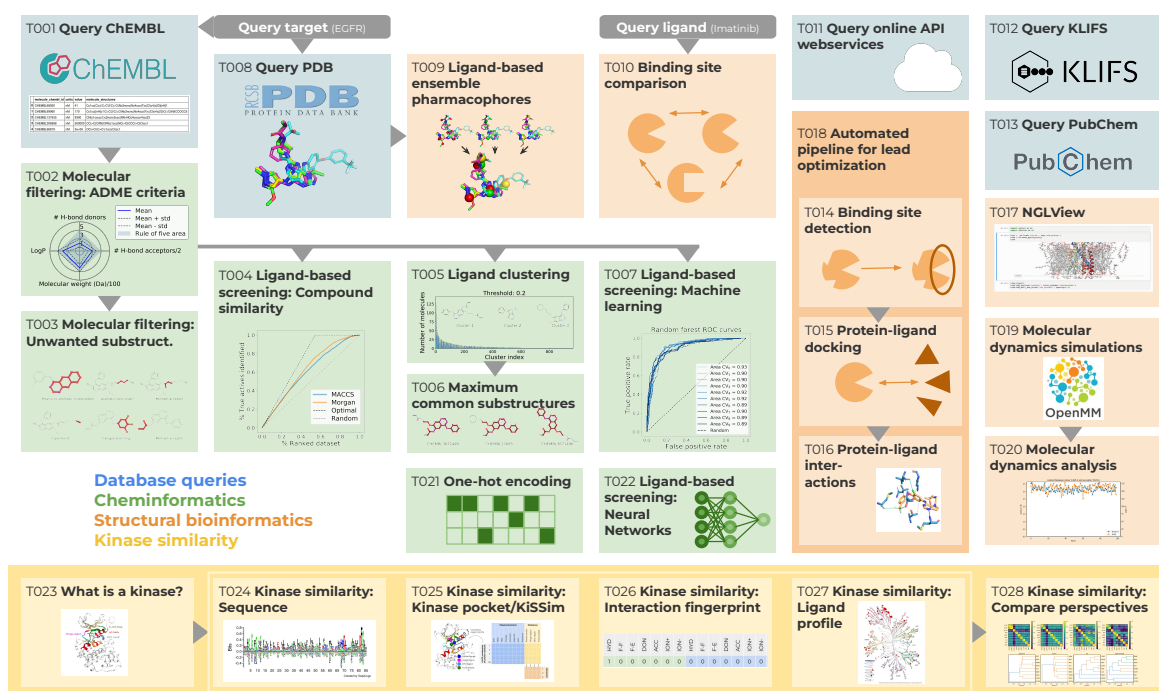


Figure 4.1: TeachOpenCADD topics T001–T028 as of September 2022 covering database queries (blue), cheminformatics (green), structural bioinformatics (orange), and the special edition on kinase similarities (yellow) [95, 144, 145].

### 4.4.2 OpenCADD: Building a FAIR Toolkit for Structural Cheminformatics

Several projects discussed in this thesis (including KiSSim [141], KinFragLib [143], and TeachOpenCADD [144, 145]) share some common building blocks. In such cases, moving these building blocks to an external toolkit avoids rewriting the same code and allows maintenance in one place. We achieved this as part of the open source OpenCADD project [189] that offers a framework for stand-alone tools for structural cheminformatics. For example, the KiSSim and KinFragLib projects both rely on the definition of subpockets within the KLIFS binding site (Figures 1 in [141] and [143], respectively). Since both methods are independent of each other—solving different research questions—we moved the code for the subpocket generation as a stand-alone functionality to the OpenCADD toolkit, OpenCADD-pocket. This functionality can be used now independently from the KiSSim and KinFragLib methods as well as kinases in general.

KLIFS data queries are another example, which reaches far beyond the two aforementioned projects. Instead of writing similar code bases that query the KLIFS database across different projects, a more sustainable and efficient solution is to implement a stand-alone tool within the OpenCADD toolkit, which has a well-designed application programming interface (API), is properly maintained, and can be deployed to users working on kinase-focused projects that build upon the KLIFS database. The OpenCADD toolkit holds to this date, the following main modules:

- `structure.pocket` (OpenCADD-pocket) defines and visualizes protein (sub)pockets with special functionalities for KLIFS structures.
- `structure.superposition` superimposes macromolecules using sequence and structural information (work by Jaime Rodríguez-Guerra, Julian Pipart, Corey Taylor, Dennis Köser, Annie Pham, and Enes Kurnaz) [190]; not discussed or used in this thesis.
- `databases.klifs` (OpenCADD-KLIFS) provides utilities to query the KLIFS database locally and remotely.

The latter module, OpenCADD-KLIFS, has been published in **Publication H** [92] (Section 3.3.3). OpenCADD-KLIFS offers a clean Python API to fetch kinase-focused data for different data types such as kinases, ligands, structures, bioactivities, interactions, and drugs by a variety of identifiers such as a kinase name, PDB ID, ligand expo ID [191] (or the equivalent KLIFS kinase, structure, and ligand IDs). The returned dataset is formatted as a Pandas DataFrame, a table-like data structure, that contains all the data type-associated annotations in a standardized manner. This allows out-of-the-box downstream manipulation such as filtering steps.

The Python API is identical between an online database query or an offline query within a KLIFS download folder; this feature has been often used to switch from smaller online queries during the development of a method to larger offline queries on a downloaded, time-stamped KLIFS dataset. This tool has made accessing KLIFS data extremely convenient within Python pipelines and has been used across multiple projects, including TeachOpenCADD [192], OpenCADD-pocket [193], KiSSim [152], KinoML (structure-informed machine learning for kinase modeling) [194], and PLIPify (protein-ligand interaction frequencies across structure ensembles) [195].



### 4.4.3 Advocating for Software Best Practices and FAIR Research

To maintain software projects and allow standards across multiple (old and new) contributors, we adhere to the following principles in the TeachOpenCADD project but also in the other projects discussed in this thesis. The following list of guidelines is taken from Publication G [145]:

- **Testing.** Reproducibility is ensured by testing, in the case of software, if unit tests pass, or in the case of Jupyter Notebooks, if they can run without errors and whether the output of specific operations can be reproduced. For this purpose, we use the tools `pytest` [196] and `nbval` [197].
- **Continuous integration.** We are testing the packages and TeachOpenCADD talktorials regularly for Linux, OSX, and Windows and different Python versions on GitHub Actions [198]. This ensures identical behavior across different operating systems and Python versions and also spots issues like conflicting dependency updates or changing outputs.
- **Repository structure.** The repository structure is based on the `cookiecutter-cms` template [199], which provides a Python-focused project scaffold with pre-configured settings for packaging, continuous integration, Sphinx-based documentation [200], and much more. We have adapted the template for our Jupyter-Notebook-focused projects.
- **Code style.** We aim to adhere to the PEP8 [201] style guide for Python code, which defines how to write idiomatic Python (Pythonic) code. Such rules are important so that new developers or—in TeachOpenCADD’s case—talktorial users can quickly read and understand the code. Furthermore, we use `black/black-nb` [202, 203] and linting tools such as `pylint` [204] or `flake8` [205] to format Python code and Jupyter Notebooks compliant with PEP8.

Pipelines such as those offered within the TeachOpenCADD platform and toolkits such as OpenCADD may in themselves not solve a real-world research question, however, they empower users to implement their use cases faster and more robustly and make the interaction with databases and toolkits smoother — thereby leaving more time and energy for the scientific questions to solve.



## Chapter 5

# Conclusion

Drug discovery is a complex, lengthy, and costly process with high failure rates. Computational methods try to mitigate these challenges in the early stages of drug discovery projects to make predictions on (i) missing data, (ii) what data to generate next, and (iii) how to generate such data. In this thesis, I presented two novel data-driven methods for kinase research, an important field to combat cancer, the world’s second leading disease. Both methods use kinase pocket information from the KLIFS database to read across the structural kinome, i.e., KiSSim [141] for computational target prediction (reviewed in **Publication A** [22]) and KinFragLib [143] for computational fragment-based drug design.

**Section 3.1** addressed the lack of a kinase-tailored method that can encode and compare the binding site accurately and facilitate its setup, maintenance, and incorporation into a larger pipeline consisting of multiple perspectives on kinase similarity. In **Publication B** [141], we discussed the KiSSim method that can explain and predict kinome-wide off-targets and polypharmacology. The method can flag targets with similar binding sites beyond the traditional sequence identity/similarity measures, which are usually applied during the target identification phase of drug design campaigns. In **Publication D** [95], the structure-based KiSSim similarity measure is embedded into a pipeline with other measures, such as the sequence-, interaction-, and ligand-based similarities; this automated process allows a production-ready off-target analysis for a user-defined set of kinases. These two studies provide a refined and automated procedure of the presented case study on kinase similarities in **Publication C** [142]. Beyond detecting off-targets, KiSSim can help to highlight structural differences between a set of targets to inform potential selectivity-driven ligand modifications during hit optimization. Given the availability of structural ensembles of well-studied kinases, KiSSim can also be used as an analysis and visualization tool for protein flexibility.

Using complementary information to tackle off-target prediction helps with KiSSim’s biggest limitation: only about 300 out of the 500 kinases are structurally resolved, hence off-target prediction covering the full kinome is not possible with KiSSim alone but can be complemented with the methods as discussed in Publications C and D. We applied KiSSim primarily to kinase structures in the DFG-in conformation because the majority ( $\sim 85\%$ ) of structures show this state, allowing us to cover a wide range of kinases. However, KiSSim can also be applied to DFG-out structures, which would be advisable, especially for projects that aim to target this state.

**Section 3.2** and **Publication E** [143] outlined how we contributed to kinase-focused fragment-based drug design with the KinFragLib method. Ligands in complex with kinase structures in the KLIFS database were decomposed with respect to the subpockets that they

occupy to generate fragment pools for each kinase subpocket. This dataset is useful to explore the chemical space of subpockets and guide subpocket-informed recombination. In our use case, we generated about 4 million novel and "Rule of Five" (Ro5)-compliant molecules based on about 600 fragments, which included known kinase inhibitors that were not in the original ligand set, underlining KinFragLib's potential to generate novel kinase-focused molecules.

Selecting a diverse set of fragments is the heart of the project; the use case was based on cluster representatives from a subset of deduplicated, Ro3-compliant, and "hinge-like" fragments. However, more elaborate filtering steps can be applied as shown in Sonja Leo's master thesis: molecular complexity can be reduced while supporting druglikeness and synthesizability using filters and tools such as unwanted substructures [166, 167], QED [168], SYBA [171], Enamine REAL Space [169] searches, and ASKCOS [172]. Although users can already choose a fragment or subpocket as a starting point for recombination, future development could facilitate this process with a user-friendly interface besides the current command-line interface (CLI) option. Another filtering step could include the consideration of spatial comparability of two fragments within a binding site. Furthermore, Humer et al. [177] showed how their CIME web interface helps to explore the KinFragLib fragments and recombined molecules interactively, which is a useful tool next to our Jupyter-Notebook-based KinFragLib platform. Last but not least, KinFragLib cannot only be useful in the context of kinase-focused fragment recombination but can also serve as a focused fragment library for bioisosteric replacement. For example, the typical hydrogen bond donor/acceptor patterns in AP fragments can be useful during hit optimization phases in target projects beyond kinases.

The resulting datasets from the KiSSim and KinFragLib studies are publicly available in [153] and [162], accompanied by Jupyter Notebooks showing how to read the datasets and documenting all the published analysis. Two considerations regarding the datasets are noteworthy for future development: (i) Access to the results would be even easier if they were available via a web application, optimally reachable via the KLIFS database itself, and (ii) integrating new structural kinase data is possible but, as of now, no automated procedure is in place to provide regular updates.

The subpocket-based exploration of the KLIFS dataset concerning pocket similarity (KiSSim) and fragment space (KinFragLib) could be transferred to other target classes, which have conserved binding sites and a decent amount of structural coverage, e.g., GPCRs or proteases. Such transfer would require (i) the definition and alignment of binding sites across the target class, and (ii) the definition of relevant subpockets.

**Section 3.3** emphasized computer-aided drug design (CADD) as an integral part of the iterative drug discovery process that has more and more data and data-driven methods at its disposal. Reproducible and reliable pipelines can help to make the design-make-test-analyze (DMTA) cycle faster and more efficient. To enable reproducible and reliable research, the software projects in this thesis have been developed following the principles of FAIR research, i.e., they are findable, accessible, interoperable, and reusable. Furthermore, the software adheres to modern Python software best practices and is modular, tested, and packaged to facilitate maintenance, contributions, and usage. This setup enables us the share CADD-relevant pipelines and tools with the scientific community.

In **Publications F and G** [144, 145], we presented TeachOpenCADD as a FAIR platform for the CADD community. TeachOpenCADD covers many common research questions in CADD, ranging from pipelines for cheminformatics, structural bioinformatics, and database queries that can be applied to a target of interest. Such pipelines can either be used to learn and teach domain-specific concepts or to start solving real-world research questions. We outlined how TeachOpenCADD can be used in a teaching setting in **Publication J** [186], as part of the

"Teaching Programming across the Chemistry Curriculum" series. Furthermore, TeachOpenCADD is not only a rich resource for CADD pipelines and teaching material but it is also a good example of how to set up websites, automated testing, and packaging for Jupyter-Notebook-centric repositories. To provide also non-coding solutions, we showed in **Publication I** [185] how some of the cheminformatics-based TeachOpenCADD topics were translated into KNIME workflows, which allow stringing together code units (nodes) with defined functionality to an easy-to-understand workflow.

The kinase-centric projects in this thesis were all based on structural data from the KLIFS database. To avoid similar code scripts across all these projects that allow fetching data from the KLIFS database, we introduced the tool OpenCADD-KLIFS in **Publication H** [92]. This tool offers a user-friendly and concise Python API to query kinase data online (KLIFS database) or offline (KLIFS download folder). Switching between these two modes is hassle-free thanks to identical APIs.

Tools like this offer less scientific insights in themselves, however, help to make projects more efficient, reliable, reproducible, and maintainable — this is what I enjoyed most throughout my doctoral studies.



# List of Publications

This list summarizes all publications that have been published as part of my doctoral studies between 2018–2022. Shared first authorship is denoted with the \* symbol.

**Publication A** Dominique Sydow\*, Lindsey Burggraaff\*, Angelika Szengel, Herman W. T. van Vlijmen, Adriaan P. IJzerman, Gerard J. P. van Westen, and Andrea Volkamer. Advances and Challenges in Computational Target Prediction. *Journal of Chemical Information and Modeling*. **2019**; 59(5):1728–1742.

[10.1021/acs.jcim.8b00832] [Section 1.2.1]

**Publication B** Dominique Sydow, Eva Aßmann, Albert J. Kooistra, Friedrich Rippmann, and Andrea Volkamer. KiSSim: Predicting Off-Targets from Structural Similarities in the Kinome. *Journal of Chemical Information and Modeling*. **2022**; 62(10):2600–2616.

[10.1021/acs.jcim.2c00050] [Section 3.1.1]

**Publication C** Denis Schmidt, Magdalena M. Scharf, Dominique Sydow, Eva Aßmann, Maria Martí-Solano, Marina Keul, Andrea Volkamer, and Peter Kolb. Analyzing Kinase Similarity in Small Molecule and Protein Structural Space to Explore the Limits of Multi-Target Screening. *Molecules*. **2021**; 26(3):629.

[10.3390/molecules26030629] [Section 3.1.2]

**Publication D** Talia B. Kimber\*, Dominique Sydow\*, and Andrea Volkamer. Kinase Similarity Assessment Pipeline for Off-Target Prediction [v1.0]. *Living Journal of Computational Molecular Science*. **2022**; 3(1):1599–1599.

[10.33011/livecoms.3.1.1599] [Section 3.1.3]

**Publication E** Dominique Sydow\*, Paula Schmiel\*, Jérémie Mortier, and Andrea Volkamer. KinFragLib: Exploring the Kinase Inhibitor Space Using Subpocket-Focused Fragmentation and Recombination. *Journal of Chemical Information and Modeling*. **2020**; 60(12):6081–6094.

[10.1021/acs.jcim.0c00839] [Section 3.2.1]

**Publication F** Dominique Sydow, Andrea Morger, Maximilian Driller, and Andrea Volkamer. TeachOpenCADD: A Teaching Platform for Computer-Aided Drug Design Using Open Source Packages and Data. *Journal of Cheminformatics*. **2019**; 11(1):29.

[10.1186/s13321-019-0351-x] [Section 3.3.1]

- Publication G** Dominique Sydow\*, Jaime Rodríguez-Guerra\*, Talia B Kimber, David Schaller, Corey J Taylor, Yonghui Chen, Mareike Leja, Sakshi Misra, Michele Wichmann, Armin Ariamajd, and Andrea Volkamer. TeachOpenCADD 2022: Open Source and FAIR Python Pipelines to Assist in Structural Bioinformatics and Cheminformatics Research. *Nucleic Acids Research*. **2022**; 50(W1):W753–W760.  
[10.1093/nar/gkac267] [Section 3.3.2]
- Publication H** Dominique Sydow, Jaime Rodríguez-Guerra, and Andrea Volkamer. OpenCADD-KLIFS: A Python Package to Fetch Kinase Data from the KLIFS Database. *Journal of Open Source Software*. **2022**; 7(70):3951.  
[10.21105/joss.03951] [Section 3.3.3]
- Publication I** Dominique Sydow\*, Michele Wichmann\*, Jaime Rodríguez-Guerra, Daria Goldmann, Gregory Landrum, and Andrea Volkamer. TeachOpenCADD-KNIME: A Teaching Platform for Computer-Aided Drug Design Using KNIME Workflows. *Journal of Chemical Information and Modeling*. **2019**; 59(10):4083–4086.  
[10.1021/acs.jcim.9b00662] [Section 5.1.1]
- Publication J** Dominique Sydow\*, Jaime Rodríguez-Guerra\*, and Andrea Volkamer. Teaching Computer-Aided Drug Design Using TeachOpenCADD, *Teaching Programming across the Chemistry Curriculum*. **2021**; ACS Symposium Series, Vol. 1387.  
[10.1021/bk-2021-1387.ch010] [Section 5.1.2]



# Appendix


This thesis appendix consists of the following published and unpublished projects:

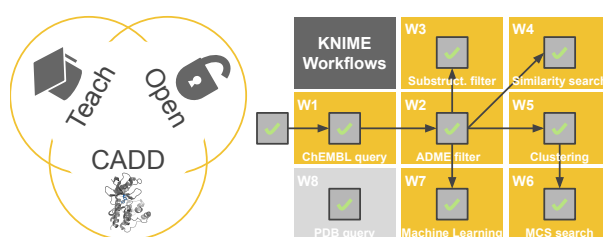
- Appendix 5.1.1: Published article [185] in the context of the TeachOpenCADD platform describing the implementation of cheminformatics tasks as KNIME workflows.
- Appendix 5.1.2: Published book chapter [186] in the context of the TeachOpenCADD platform outlining how TeachOpenCADD can be used in teaching.
- Appendix 5.2: Unpublished project called Ratar, a novel binding site comparison method.
- Appendix 5.3: Illustrations by Ferdinand Krupp on this thesis' projects.

## 5.1 Further Publications

### 5.1.1 TeachOpenCADD-KNIME: A Teaching Platform for Computer-Aided Drug Design Using KNIME Workflows Publication I

The TeachOpenCADD platform offers a variety of solutions to common questions in computer-aided drug design in the form of Jupyter Notebooks. While the platform is intended also for users new to programming, we publish with this article cheminformatics-related topics in the form of KNIME workflows, which require no programming. Such workflows are built up by connecting small pre-implemented code units (nodes) that have a defined and standardized functionality. This drag-and-drop mode makes workflows easy and intuitive to set up.

 <https://hub.knime.com/volkamerlab/spaces/Public/latest/TeachOpenCADD>



Contribution:

#### Co-first author

Conceptualization (50%)

Data Curation (40%)

Investigation (50%)

Methodology (40%)

Software (40%)

Visualization (50%)

Writing — Original Draft (90%)

Writing — Review & Editing (90%)

Reprinted with permission from Sydow D\*, Wichmann M\*, Rodríguez-Guerra J, Goldmann D, Landrum G, Volkamer A. TeachOpenCADD-KNIME: A Teaching Platform for Computer-Aided Drug Design Using KNIME Workflows. *Journal of Chemical Information and Modeling*. **2019**; 59(10):4083-4089. 10.1021/acs.jcim.9b00662 (\*contributed equally)

Copyright © 2019 American Chemical Society. Article licensed under the ACS Editors' Choice license; further permissions related to the material excerpted should be directed to the ACS.

## TeachOpenCADD-KNIME: A Teaching Platform for Computer-Aided Drug Design Using KNIME Workflows

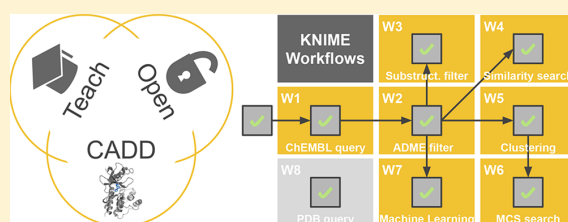
Dominique Sydow,<sup>†,||</sup> Michele Wichmann,<sup>†,||</sup> Jaime Rodríguez-Guerra,<sup>†</sup> Daria Goldmann,<sup>‡</sup> Gregory Landrum,<sup>§</sup> and Andrea Volkamer<sup>\*,†,||</sup>

<sup>†</sup>In Silico Toxicology, Institute of Physiology, Charité - Universitätsmedizin Berlin, Charitéplatz 1, 10117 Berlin, Germany

<sup>‡</sup>KNIME GmbH, Körtestr. 10, 10967 Berlin, Germany

<sup>§</sup>KNIME AG, Technoparkstr. 1, 8005 Zurich, Switzerland

**ABSTRACT:** Open-source workflows have become more and more an integral part of computer-aided drug design (CADD) projects since they allow reproducible and shareable research that can be easily transferred to other projects. Setting up, understanding, and applying such workflows involves either coding or using workflow managers that offer a graphical user interface. We previously reported the TeachOpenCADD teaching platform that provides interactive Jupyter Notebooks (talktorials) on central CADD topics using open-source data and Python packages. Here we present the



conversion of these talktorials to KNIME workflows that allow users to explore our teaching material without any line of code. TeachOpenCADD KNIME workflows are freely available on the KNIME Hub: <https://hub.knime.com/volkamerlab/space/TeachOpenCADD>.

### INTRODUCTION

In computer-aided drug design (CADD), computational tools are used to process and rationalize large and heterogeneous data sets involving small molecules and macromolecules. For this endeavor, open-access resources have gained momentum, especially for setting up complex workflows, since they enable modular, reproducible, and reusable research.

We recently reported the TeachOpenCADD<sup>1</sup> teaching platform (<https://github.com/volkamerlab/teachopencadd>) that provides learning material for CADD using open-source data and Python libraries. Central topics in CADD are covered in the form of interactive Jupyter Notebooks that contain both theory and code for each topic.

An alternative to code-based pipelines are workflow managers that allow the design of protocols via an intuitive drag-and-drop style graphical interface without the need for coding. KNIME<sup>2,3</sup> is a popular workflow manager for data science with several open-source modules for CADD,<sup>4</sup> while its usage ranges from small in-house applications such as compound library preparation to more complex workflow applications integrating chemical, pharmacological, and structural information. An example of the latter is 3D-e-Chem,<sup>5,6</sup> which allows, e.g., structure-based bioactivity data mapping of kinase inhibitors or structure-based GPCR–kinase cross-reactivity prediction.

Here we address users who aim to learn how to use KNIME for CADD applications as well as users who desire to study central CADD topics without necessarily learning how to code. We report the conversion of the TeachOpenCADD Python pipeline (talktorials T1–T8) to a KNIME workflow pipeline (workflows W1–W8). The KNIME pipeline is publicly

available on the KNIME Hub: <https://hub.knime.com/volkamerlab/space/TeachOpenCADD> (current release: <https://doi.org/10.5281/zenodo.3475086>).

### METHODS

KNIME (the Konstanz Information Miner) provides an open-source data analysis, reporting, and integration platform. KNIME enables users to create data workflows, execute selected analysis steps, and check intermediate and final results, models, and interactive views via a graphical user interface. Coding is not required, since the workflows are built up by stringing together small preimplemented code units (nodes) with defined, tested, and thus standardized functionalities, which can be configured with individual settings. In addition, KNIME offers functionalities to design complex workflows in a well-structured way via metanodes that encapsulate parts of a workflow.

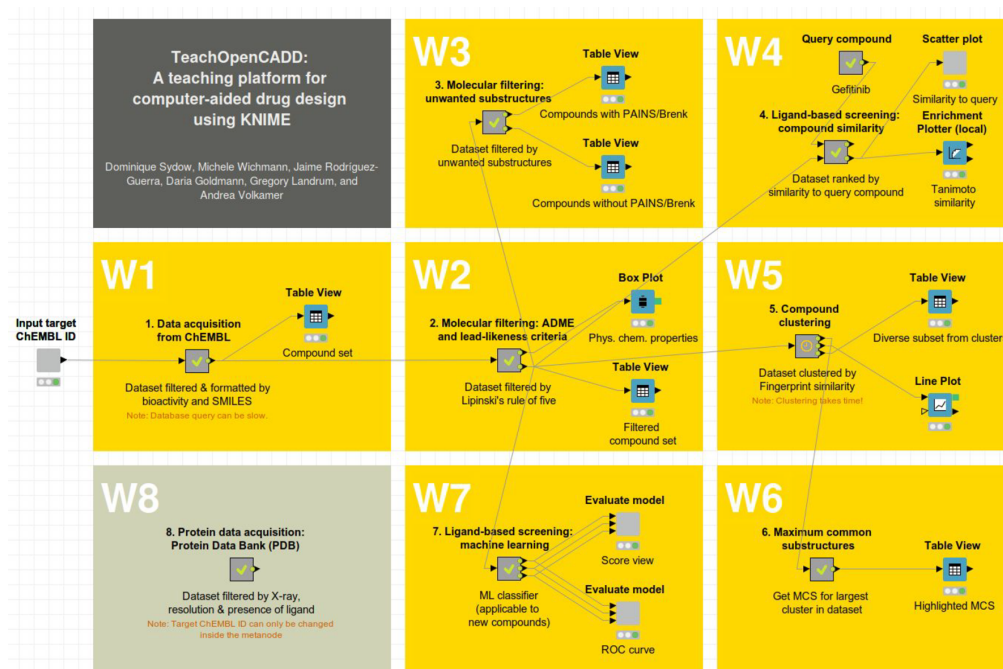
This work was developed using KNIME version 4.0.0 and uses nodes from the KNIME Analytics Platform, KNIME Extensions, and Community Extensions by RDKit<sup>3,7</sup> and Vernalis<sup>8</sup> (RSCB PDB Tools).

### RESULTS

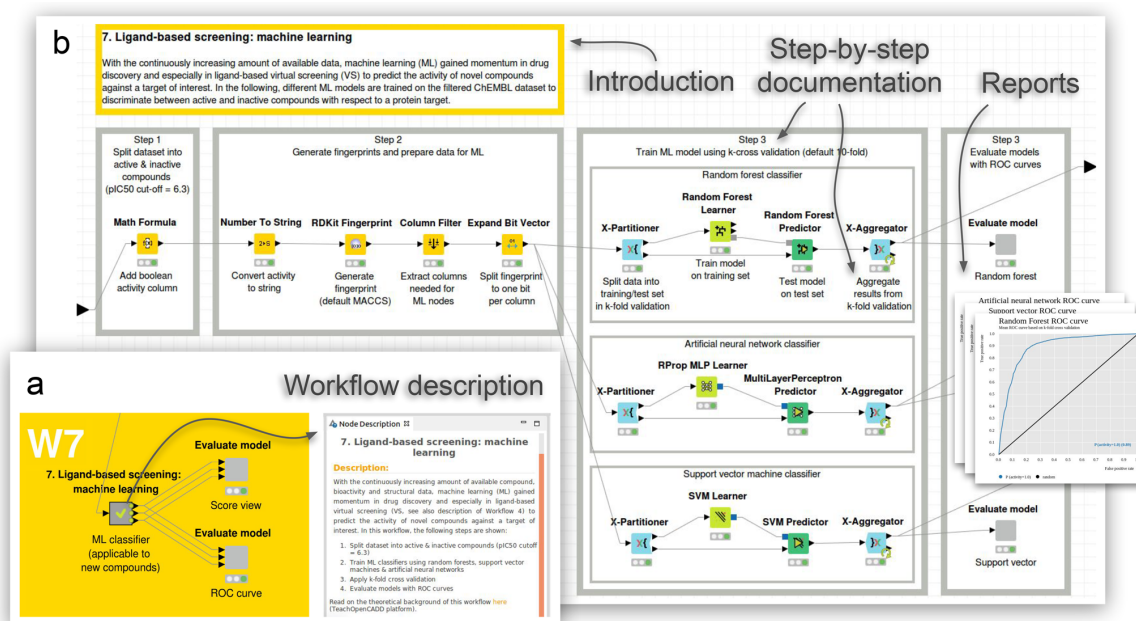
The TeachOpenCADD KNIME pipeline consists of eight interconnected workflows (W1–W8) in the form of metanodes, each containing one CADD topic. The pipeline is illustrated using the epidermal growth factor receptor

**Received:** August 8, 2019

**Published:** October 15, 2019



**Figure 1.** The TeachOpenCADD KNIME pipeline offers eight KNIME workflows covering central topics in CADD while using open-source data and KNIME nodes. This figure shows the graphical interface of KNIME, demonstrating the software's visual potential.



**Figure 2.** Workflow composition shown for workflow W7 (ligand-based screening: machine learning). (a) Each workflow metanode is labeled with a brief topic description and the main workflow steps. (b) The interior of each workflow metanode consists of an introduction, nodes organized in boxes per step, node documentation, and output reports.

(EGFR)<sup>9,10</sup> but can easily be applied to other targets of interest. Topics include how to fetch, filter, and analyze compound data associated with a query target and are briefly described in the following (Figure 1). For a detailed

description, we refer the reader to the initial TeachOpenCADD publication.<sup>1</sup>

First, compound data for the query target EGFR are acquired from the ChEMBL web services<sup>11</sup> (W1)<sup>12</sup> and

subsequently filtered for drug-likeness using Lipinski's rule of five (W2). This filtered data set forms the basis for the remaining workflows. Unwanted substructures that potentially cause toxicity or nonspecific assay interactions are detected (W3), and a similarity search for a ligand-based screen with the EGFR inhibitor gefitinib as the query<sup>13</sup> is conducted (W4). Compounds are grouped using a hierarchical clustering algorithm (W5),<sup>14</sup> whereupon the maximum common substructure is detected and visualized for the largest cluster (W6).<sup>15</sup> Additionally, machine learning approaches are employed to build models for active compound prediction (W7).<sup>16</sup> Lastly, ligand–EGFR complexes are fetched from the PDB web services<sup>17</sup> and filtered by criteria such as structure resolution (W8).<sup>18</sup> The last two previously reported talktutorials, T9 and T10, were not translated to workflows because of their extensive use of PyMOL, which is currently not supported in KNIME.

The workflows can be examined and executed independently from each other or as a pipeline. As shown in Figure 2 for W7, each workflow is introduced with a brief topic motivation and grouped into multiple steps using gray boxes that contain a step description and all step-associated nodes labeled with task descriptions. Results from intermediate steps (e.g., filtered compound tables) or from final plotting nodes can be viewed interactively and configured easily using the nodes' graphical interface.

## CONCLUSION

The TeachOpenCADD platform offers learning material on central topics of cheminformatics and structural bioinformatics. In the present work, teaching material was translated from code-based Jupyter Notebooks to KNIME workflows, which have several advantages. KNIME workflows (i) are knitted together from preimplemented nodes with standardized functionalities, (ii) are easy to understand because of the visual representation of their architecture, and (iii) permit a low-threshold entry for nonprogrammers to build customized pipelines.

The TeachOpenCADD KNIME pipeline is suitable for self-study training and classroom teaching but can also serve as a starting point for workflows in research projects. TeachOpenCADD is open for contributions and ideas from the community with regard to both Jupyter Notebooks and KNIME workflows.

## AUTHOR INFORMATION

### Corresponding Author

\*E-mail: [andrea.volkamer@charite.de](mailto:andrea.volkamer@charite.de).

### ORCID

Dominique Sydow: 0000-0003-4205-8705

Michele Wichmann: 0000-0002-7441-1561

Jaime Rodríguez-Guerra: 0000-0001-8974-1566

Daria Goldmann: 0000-0002-4793-8579

Gregory Landrum: 0000-0001-6279-4481

Andrea Volkamer: 0000-0002-3760-580X

### Author Contributions

<sup>||</sup>D.S. and M.W. share first authorship.

### Notes

The authors declare no competing financial interest.

## ACKNOWLEDGMENTS

A.V. and D.S. received funding from the Deutsche Forschungsgemeinschaft (Grant VO 2353/1-1). A.V. received funding from the Bundesministerium für Bildung und Forschung (Grant 031A262C). J.R.-G. received funding from the Stiftung Charité (Einstein BIH Visiting Fellow Project). M.W. received funding from the "SUPPORT für die Lehre" Program (Förderung innovativer Lehrvorhaben) of Freie Universität Berlin.

## REFERENCES

- (1) Sydow, D.; Morger, A.; Driller, M.; Volkamer, A. TeachOpenCADD: A Teaching Platform for Computer-Aided Drug Design Using Open Source Packages and Data. *J. Cheminf.* **2019**, *11*, 29.
- (2) Berthold, M. R.; Cebon, N.; Dill, F.; Gabriel, T. R.; Kötter, T.; Meil, T.; Ohl, P.; Sieb, C.; Thiel, K.; Wiswedel, B. KNIME: The Konstanz Information Miner. In *Data Analysis, Machine Learning and Applications*; Preisach, C., Burkhardt, H., Schmidt-Thieme, L., Decker, R., Eds.; Springer: Berlin, 2008; pp 319–326.
- (3) Fillbrunn, A.; Dietz, C.; Pfeuffer, J.; Rahn, R.; Landrum, G. A.; Berthold, M. R. KNIME for Reproducible Cross-Domain Analysis of Life Science Data. *J. Biotechnol.* **2017**, *261*, 149–156.
- (4) Mazanetz, M. P.; Goode, C. H. F.; Chudyk, E. I. Ligand- and Structure-Based Drug Design and Optimization Using KNIME. *Curr. Med. Chem.* **2019**, DOI: 10.2174/0929867326666190409141016.
- (5) McGuire, R.; Verhoeven, S.; Vass, M.; Vriend, G.; de Esch, I. J. P.; Lusher, S. J.; Leurs, R.; Ridder, L.; Kooistra, A. J.; Ritschel, T.; de Graaf, C. 3D-e-Chem-VM: Structural Cheminformatics Research Infrastructure in a Freely Available Virtual Machine. *J. Chem. Inf. Model.* **2017**, *57*, 115–121.
- (6) Kooistra, A. J.; Vass, M.; McGuire, R.; Leurs, R.; de Esch, I. J. P.; Vriend, G.; Verhoeven, S.; de Graaf, C. 3D-e-Chem: Structural Cheminformatics Workflows for Computer-Aided Drug Discovery. *ChemMedChem* **2018**, *13*, 614–626.
- (7) RDKit Nodes for KNIME. <https://www.knime.com/rdkit> (accessed May 15, 2019).
- (8) Roughley, S. Five Years of the KNIME Vernalis Cheminformatics Community Contribution. *Curr. Med. Chem.* **2018**, DOI: 10.2174/0929867325666180904113616.
- (9) UniProt Entry for EGFR. <https://www.uniprot.org/uniprot/P00533> (accessed May 16, 2019).
- (10) Chen, J.; Zeng, F.; Forrester, S. J.; Eguchi, S.; Zhang, M.-Z.; Harris, R. C. Expression and Function of the Epidermal Growth Factor Receptor in Physiology and Disease. *Physiol. Rev.* **2016**, *96*, 1025–1069.
- (11) Gaulton, A.; Bellis, L. J.; Bento, A. P.; Chambers, J.; Davies, M.; Hersey, A.; Light, Y.; McGlinchey, S.; Michalovich, D.; Al-Lazikani, B.; Overington, J. P. ChEMBL: A LargeScale Bioactivity Database for Drug Discovery. *Nucleic Acids Res.* **2012**, *40*, D1100–7.
- (12) Adapting KNIME Workflow Example to Extract Bioactivities for a Target ID. KNIME EXAMPLES Server under 50\_Applications/30\_RESTful\_ChEMBL/03\_ChEMBL\_Bioactivity\_Search (accessed May 18, 2019).
- (13) DrugBank Entry for Gefitinib. <https://www.drugbank.ca/drugs/DB00317> (accessed May 16, 2019).
- (14) Adapting KNIME Workflow Example to Cluster Molecules Using RDKit Nodes. KNIME EXAMPLES Server Under 99\_Community/03\_RDKit/01\_Clustering (accessed May 24, 2019).
- (15) Adapting KNIME Workflow Example Created by Daria Goldmann. KNIME Introduction and Training Session on 2019-01-21 at Volkamer Lab in Berlin: 0 × 1\_Maximum\_Common\_Substructure (accessed Jan 21, 2019).
- (16) Adapting KNIME Workflow Example Created by Daria Goldmann. KNIME Introduction and Training Session on 2019-01-21 at Volkamer Lab in Berlin: 0 × 2\_Machine\_Learning (accessed Jan 21, 2019).

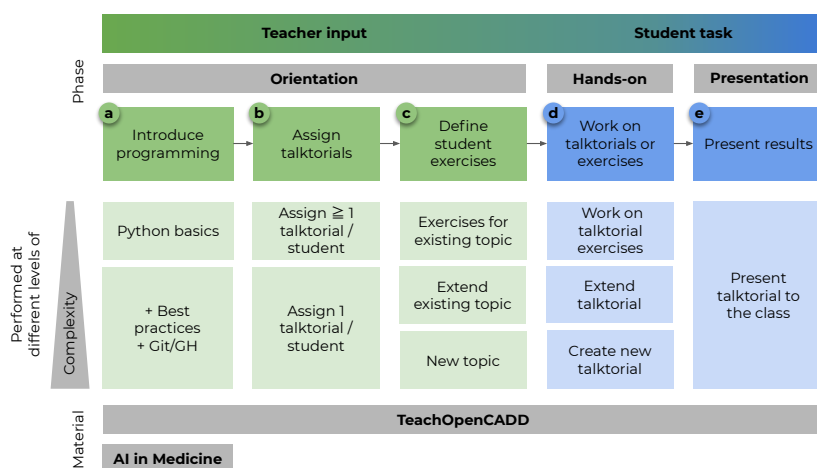
(17) Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. The Protein Data Bank. *Nucleic Acids Res.* **2000**, *28*, 235–42.

(18) Adapting KNIME Workflow Example to Download and Save PDB Queries using Vernalis Nodes. KNIME EXAMPLES Server Under 99\_Community/04\_Vernalis/01\_PDB\_Query\_Downloader\_and\_Save\_Locally (accessed May 24, 2019).



### 5.1.2 Teaching Computer-Aided Drug Design Using TeachOpenCADD Publication J

The TeachOpenCADD platform offers solutions to common tasks in computer-aided drug design, which are useful to novices in the field with all kinds of training backgrounds as well as to advanced users who need templates for their research questions. In this book chapter, we outline how the TeachOpenCADD material can be used in teaching settings such as classrooms, individual student projects, but also self-training.



Contribution:

#### Co-first author

Conceptualization (80%)

Visualization (80%)

Writing — Original Draft (75%)

Writing — Review & Editing (33%)

Reprinted with permission from Sydow D\*, Rodríguez-Guerra J\*, Volkamer A. Teaching Computer-Aided Drug Design Using TeachOpenCADD. *Teaching Programming across the Chemistry Curriculum*. 2021; ACS Symposium Series, Vol. 1387. 10.1021/bk-2021-1387.ch010 (\*contributed equally)

Copyright © 2021 American Chemical Society.



## Chapter 10

# Teaching Computer-Aided Drug Design Using TeachOpenCADD

Dominique Sydow,<sup>1,2</sup> Jaime Rodríguez-Guerra,<sup>1,2</sup> and Andrea Volkamer<sup>1,\*</sup>

<sup>1</sup>*In silico* Toxicology and Structural Bioinformatics, Institute of Physiology, Charité – Universitätsmedizin Berlin, corporate member of Freie Universität Berlin and Humboldt-Universität zu Berlin, Augustenburger Platz 1, 13353 Berlin, Germany

<sup>2</sup>These authors contributed equally to this work.

\*Email: andrea.volkamer@charite.de.

### Abstract

TeachOpenCADD is a teaching platform developed with students for students and researchers. The material teaches how to leverage open source cheminformatics and structural bioinformatics resources to explore key questions in computer-aided drug design (CADD). Both the theoretical and practical aspects of CADD concepts are covered in interactive Jupyter Notebooks using Python. This setup makes it easy for students from various fields of science to understand computational drug design techniques with hands-on programming examples. In this book chapter, we explain the motivation for putting the TeachOpenCADD material together, how this teaching material can be and has been used in different teaching formats, and what lessons we have learned so far.

### Introduction

Data has never been produced at such speed and in such amounts, while new technologies allow to digest this information and to put it to practical use. Thus, it is not surprising that data and computational sciences, along with a new wave of AI solutions, protruded different research areas including life sciences to analyze and maintain such large amounts of data (1–4). Needless to say that this requires a change in teaching the next generation of scientists. Processing this information demands knowledge of computational concepts and programming skills (see communications regarding medicinal chemists competencies (3, 5–7)).

In this chapter, we focus on teaching material that introduces basic ideas of computer-aided drug design (CADD). This research area uses techniques from cheminformatics and structural bioinformatics to support rational and data-driven design of novel drugs (1, 8, 9). Note that different but overlapping terms are used in this field when referring to exercises that involve computational

approaches. When the emphasis is directed to areas more invested in the programmatic aspects, terms such as cheminformatics and (*structural*) *bioinformatics* are commonly used. When the focus is shifted towards the atomistic details, terms in use are *computational chemistry*, *molecular modeling* or *structural biology*, depending on the molecular entities involved.

Drug discovery and development is a time and cost intensive process. Cost estimates range up to 2.8 billion US dollars spent and a duration of over 13 years before a new drug is approved (10–12). This is associated to the fact that a high number of drugs fail in late stages due to problems with safety and efficacy (13). Over the last decades, computers have become an integral part of the drug development pipeline with the aim of rationally driving the design of more effective and less toxic drugs. Computational methods have been shown to positively impact the drug design process (2, 8, 9, 14–16). *In silico* techniques support especially the early phases, e.g., target and hit identification, hit-to-lead optimization, as well as off-target and ADMETox predictions. This is often referred to as the "fail early, fail cheap" principle, signaling the impact of using computational tools to prioritize promising compounds early in the process.

CADD combines expertise at the intersection of chemistry, biology and pharmacology as well as mathematics, data and computer sciences. In fact, techniques from the latter fields are applied to data from the former areas to address questions in drug design and development. Training chemists in computational skills that enable them to understand and comfortably handle such information is becoming increasingly important in industrial as well as academic settings (17–19). Thus, especially for audiences coming from a less technical background, it might be hard to enter the field partly because focused and simple application examples are rare.

Our motivation for TeachOpenCADD (20, 21) has been to provide a starting point for students, teachers and researchers from different fields and at different entry levels to become aware of the CADD tools available. The material should enable them to study or teach the concepts of different CADD tasks, with small and easy to follow collections of theory combined with code examples. We use only open source software and data resources to remove any entry barrier. Thereby, we promote open science and embrace the FAIR principles, i.e., findability, accessibility, interoperability, and reuse of digital assets (22). Other prominent examples for such CADD or cheminformatics teaching collections include the Chemistry Development Kit (CDK) (23, 24) or the Teach-Discover-Treat (TDT) (25) initiative.

Throughout the chapter, more details about the TeachOpenCADD platform and the available training material are given, including computational concepts and resources and an excerpt on Python programming. Finally, different training settings and lessons learned from our own and other courses are covered.

### TeachOpenCADD Platform

The key idea behind the TeachOpenCADD platform is that the students work in an interactive environment where they can learn about a topic's theoretical background and perform practical programming tasks in the same place. Integrating these different objectives is possible with interactive Jupyter Notebooks (26). These are open source web applications to create and share documents that can contain narrative text alongside live code, visualizations and equations. This setup is widely used for exploring and communicating data science projects (27), and reflects what teaching is about: exploring a new topic, at best including a small sample project, and communicating ideas, questions and findings amongst students and teachers. This makes the application perfectly suited for TeachOpenCADD and teaching in general (Figure 1).

### T002 · Molecular filtering: ADME and lead-likeness criteria

**Authors:**

- Michael Volkamer, CADD seminar 2017, ChemSU/FU Berlin
- Martina Wiegand, CADD seminar 2018, ChemSU/FU Berlin
- Dominique Sjöberg, 2018-2020, Volkamer Lab, Charte
- Andrea Volkamer, 2018-2020, Volkamer Lab, Charte

**Talktorial T002:** This talktorial is part of the TeachOpenCADD pipeline described in the [3rd TeachOpenCADD paper](#), comprising of talktorials T001-T010.

#### ADME - absorption, distribution, metabolism, and excretion

Pharmacokinetics are usually divided into four steps: Absorption, Distribution, Metabolism, and Excretion. These are summarized as ADME. Often, ADME also includes Toxicology and is thus referred to as ADMET or ADMETox. Below, the ADME steps are discussed in more detail ([Wikipedia](#) and [MolPharm](#), (2012), 7(8), 1388-1405).

**Absorption:** The amount and the time of drug-uptake into the body depends on multiple factors which can vary between individuals and their conditions as well as on the properties of the substance. Factors such as (poor) compound solubility, gastric emptying time, intestinal transit time, chemical (in-)stability in the stomach, and (in-)ability to permeate the intestinal wall can all influence the extent to which a drug is absorbed after e.g. oral administration, inhalation, or contact to skin.

**Distribution:** The distribution of an absorbed substance, i.e. within the body, between blood and different tissues, and crossing the blood-brain barrier are affected by regional blood flow rates, molecular size and polarity of the compound, and binding to serum proteins and transporter enzymes. Critical effects in toxicology can be the accumulation of highly apolar substances in fatty tissue, or crossing of the blood-brain barrier.

**Metabolism:** After entering the body, the compound will be metabolized. This means that only part of this compound will actually reach its target. Mainly liver and kidney enzymes are responsible for the break-down of xenobiotics (substances that are extrinsic to the body).

**Excretion:** Compounds and their metabolites need to be removed from the body via excretion, usually through the kidneys (urine) or in the feces. Incomplete excretion can result in accumulation of foreign substances or adverse interference with normal metabolism.

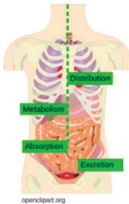


Figure 1: ADME processes in the human body (figure taken from [OpenStax](#)) and adapted.

#### Discussion

In this talktorial, we have learned about Lipinski's  $Ro5$  as a measure to estimate a compound's oral bioavailability and we have applied the rule on a dataset using `rskit`. Note that drugs can also be administered via alternative routes, i.e. Inhalation, skin penetration and injection.

In this talktorial, we have looked at only one of many more ADME properties. Webobservers such as [StasisADME](#) give a more comprehensive view on compound properties.

#### Quiz

- In what way can the chemical properties described by the  $Ro5$  affect ADME?
- Find or design a molecule which violates three or four rules.
- How can you plot information for an additional molecule in the radar charts that we have created in this talktorial?

#### Aim of this talktorial

In the context of drug design, it is important to filter candidate molecules by e.g. their physicochemical properties. In this talktorial, the compounds acquired from ChEMBL (Talktorial 001) will be filtered by Lipinski's rule of five to keep only orally bioavailable compounds.

#### Contents in Theory

- ADME - absorption, distribution, metabolism, and excretion
- Calculates and get molecular properties for  $Ro5$
- Investigate compliance with  $Ro5$
- Apply  $Ro5$  to the ChEMBL dataset
- Visualize  $Ro5$  properties (radar plot)

#### Contents in Practical

- Define and visualize example molecules
- Calculate and get molecular properties for  $Ro5$
- Investigate compliance with  $Ro5$
- Apply  $Ro5$  to the ChEMBL dataset
- Visualize  $Ro5$  properties (radar plot)

#### References

- ADME criteria ([OpenStax](#) and [Adv Pharm](#), (2012), 7(8), 1388-1405)
- [StasisADME](#) webserver
- What are lead compounds? ([OpenStax](#))
- What is the Lipinski rule? ([OpenStax](#))
- Lipinski et al. "Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings." [Adv. Drug Deliv. Rev.](#) (2001), 20, 3-20
- Ribbe et al. "Graphical representation of ADME-related molecule properties for medicinal chemists" [Comput. Toxicol.](#) (2011), 18, 85-92

#### Talktorial sections

1. Aim, content & references
2. Theory
3. Practical
4. Discussion
5. Quiz

```

In [16]: molecules["molecular_weight"] - molecules["Ro5"].apply(Descriptors.ExactMolWt)
molecules["n_hba"] - molecules["Ro5"].apply(Descriptors.NumHDonors)
molecules["logp"] - molecules["Ro5"].apply(Descriptors.NoLogP)
# colors are used for plotting the molecules later
molecules["color"] = ["red", "green", "blue", "cyan"]
# status check output
molecules[["molecular_weight", "n_hba", "n_hbd", "logp"]]

Out[16]:
  molecular_weight  n_hba  n_hbd  logp
0  1201.841368      12     5    3.26900
1  306.184447       4     1    1.68492
2  536.438202       0     0    12.60580
3  314.224580       2     2    5.84050

In [8]: # Full preview
molecules


Out[8]:
  name          smiles          ROMol  molecular_weight  n_hba  n_hbd  lc
0  cyclosporine  CCC1=C(O)N(CC1=O)N(C)C(=O)N(C)C(=O)N(C)C(=O)N(C)C...  1201.841368      12     5     3.
1  clozapine     CN1CCN(CC1)C2=C3C=CC(=CC3=NC4=C1N2)C=C4C  306.184447       4     1     1.
2  beta-carotene CC1=C(C)C(CCC1)(C)C=C(C)C(C)C=C(C)C(C)C=C...  536.438202       0     0    12.
3  cannabidiol  CCCC1=CC(=C(C=C1)O)C2=C(C)C(C2)C=C(C)C  314.224580       2     2     5.

```

#### Explanatory text

#### Executable code

#### Code output



All-in-one  
Jupyter Notebook

Figure 1. Structure of each lesson in TeachOpen CADD, exemplified by talktorial T002 that explores ADME and lead-likeness criteria for filtering molecule data sets. (1) Aim of the talktorial, including content and references, (2) Theory, (3) Practical with code examples, (4) Discussion, and (5) Quiz. Figure is adapted from [(20), Fig. 2] (published under a CC-BY-4.0 license) and contains screenshots of TeachOpenCADD talktorial T002 (published under a CC-BY-4.0 license) taken directly from the Jupyter Notebook (published under a 3-clause BSD license).

The lessons within TeachOpenCADD are called *talktorials*, a combination of talk and tutorial (inspired by a format at the RDKit User Group Meeting (28)). A talktorial is well suited both as reading and presentation material (talk) together with simple code examples (tutorial). Each talktorial follows the same structure, like a book chapter, covering the following sections: (1) Aim of the talktorial, including content and references, (2) Theory, (3) Practical with code examples, (4) Discussion, and (5) Quiz. An extract of an example talktorial is shown in Figure 1.

The TeachOpenCADD material is provided in different modes and is hosted on GitHub (<https://github.com/volkamerlab/teachopencadd>). The easiest access to the material is the read-only TeachOpenCADD website (<https://projects.volkamerlab.org/teachopencadd/>), which renders the content of the notebooks in a static version. Additionally, the website groups the lessons into collections, which allows the reader to focus on specific research questions. Furthermore, it holds the instructions on how to install and access the Jupyter Notebooks for the interactive mode. The latter allows to work with the material locally as well as collaboratively via GitHub (29). The structure of the lessons and the different availability modes allow (i) to target a large audience from beginners to advanced users in either programming and/or drug design, and (ii) to conduct different study settings such as self-studies or classroom teaching.

137

Ringer McDonald and Nash; Teaching Programming across the Chemistry Curriculum  
ACS Symposium Series; American Chemical Society: Washington, DC, 2021.

For cases where the lessons should not focus too much on programming, but on the drug design operations, the first 10 lessons of the TeachOpenCADD material (20) have also been made available as KNIME workflows (21) (not covered in this chapter). KNIME (30) is a workflow manager for data science, with several open source modules for life science applications (31–33). We also encourage people who have no or only little programming experiences to begin with introductory Python lessons. Starting points could be the AI in Medicine material (34), extracted from a course provided in the medical students curriculum at Charité, or other sources (35–37). For people interested in more cheminformatics-related training material, we refer to other collections, blogs, and books (38–40).

### Training Material

We pursue two main goals with the TeachOpenCADD platform. First, we introduce *computational concepts and resources* for common tasks in cheminformatics and structural bioinformatics. This ensures that the students understand CADD concepts and how these are implemented and applied. This also enables the students to interpret the results of a program with respect to its scope and potential pitfalls. Second, students are taught how to actively use, adapt and extend such concepts and resources in the context of *Python programming*. The TeachOpenCADD platform is designed to empower students to read, understand and eventually write code. Note that it is not a programming course, but it teaches programming by example while the students go through the material, inspect and execute the code.

If students have no prior programming experience, we refer to the AI in Medicine repository (34). Here, an entry level Python introduction paves the way to leverage the power of key Python libraries for data science such as NumPy (41), Pandas (42), Matplotlib (43), and Scikit-Learn (44). On the TeachOpenCADD platform, these libraries are used and combined with domain-specific libraries to address tasks in chem- and structural bioinformatics. Examples include RDKit (45), a cheminformatics library, or the ChEMBL and PDB web resource clients (46, 47) for data acquisition. In the following section, we show examples on how the TeachOpenCADD and AI in Medicine materials cover training in computational concepts and resources in CADD by using Python programming.

### Computational Concepts and Resources

TeachOpenCADD is organized in small lessons called talktorials. As shown in Figure 2, each talktorial covers one topic from the following three areas: data acquisition (blue tiles), cheminformatics (green tiles), and structural bioinformatics (orange tiles). Since this book focuses on the chemistry curriculum, we discuss in this chapter talktorials on cheminformatics topics in more detail together with a few examples from structural bioinformatics (see also the first TeachOpenCADD publication (20)). However, please note that over time more talktorials covering a broader range of applications have been added to the platform and the content is continuously growing (dotted boxes in Figure 2 indicate the status as of May 2021).

The first set of TeachOpenCADD talktorials (T001-T007) can be knitted together to form a typical CADD pipeline with the goal of finding active compounds against a query target. The pipeline is showcased for the epidermal growth factor receptor (EGFR) kinase (48). The talktorials introduce how to access compound and bioactivity data from ChEMBL (49) (T001), how to filter compounds based on physicochemical properties (T002), and how to highlight unwanted substructures (T003).

Molecular descriptors and measures for molecular similarity are explained and applied for a simple similarity-based virtual screening (T004) as well as for compound clustering (T005). Next, found clusters are inspected and rationalized using maximum common substructures (T006). Finally, different machine learning (ML) models are built for a more elaborated screening pipeline to predict active compounds against the chosen EGFR target (T007). On the structural bioinformatics side, protein-ligand complexes are fetched from the PDB (50) (T008) and used to create ligand-based ensemble pharmacophores (T009). Finally, geometry-based binding site comparison allows the exploration of potential off-targets (T010).

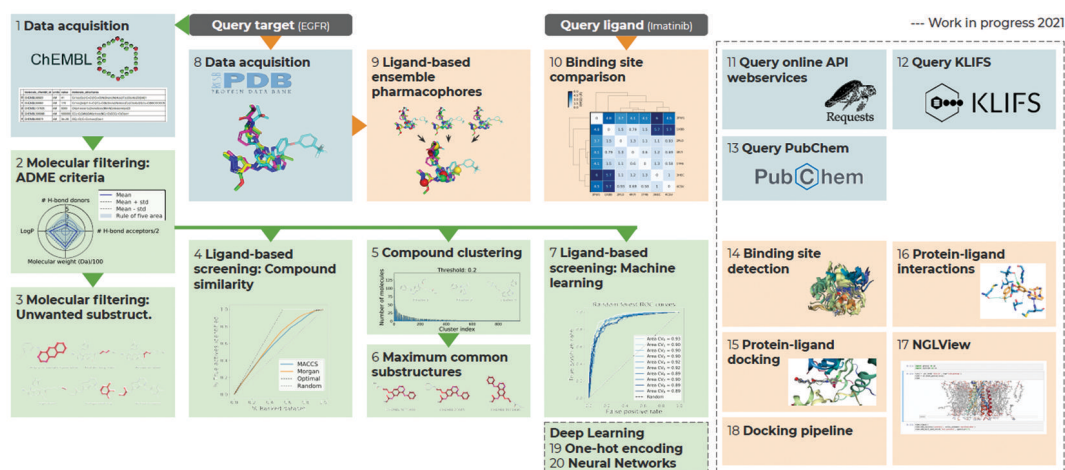


Figure 2. Overview of TeachOpenCADD talktorials, covering data acquisition (blue), cheminformatics (green) and structural bioinformatics (orange) topics. The first 10 talktorials have been published in 2019 (20), while another 10 talktorials are work in progress and will be released in 2021 (dotted boxes). Figure is adapted and extended from [(20), Fig. 1] (published under a CC-BY-4.0 license).

**Table 1. TeachOpenCADD Talktorial Indices with Short Description of the Topics from the First TeachOpenCADD Publication (20)**

Index	Content
T001	Fetch compound and bioactivity data from the ChEMBL database
T002	Filter compounds based on ADME criteria using Lipinski's Rule of Five
T003	Detect unwanted substructures related to toxicity, reactivity and PAINS by SMARTS patterns
T004	Encode molecules as MACCS and Morgan fingerprints and perform similarity search based on Tanimoto and Dice metrics
T005	Cluster compounds using the Butina algorithm and select a diverse compound subset
T006	Find maximum common substructures in a compound set using the FMCS algorithm
T007	Predict active compounds for a target of interest using ML models (RF, SVM, NN)
T008	Fetch structural data from the PDB database
T009	Identify common pharmacophoric features for a set of ligands
T010	Detect off-targets based on geometry-based binding site comparison

Thanks to the talktorials' modularity, it is possible to not only use them as a pipeline but also to work with them independently. In the following, the cheminformatics-centric talktorials are shortly discussed. They cover compound databases, descriptors, similarity, activity prediction and substructures. Talktorials on structural bioinformatics are commented on afterwards. A talktorial summary is given in Table 1.

### Compound Databases

Getting started with a CADD project requires to know where to find compound and bioactivity data and how to access it. Students are introduced to different compound resources with a focus on the ChEMBL database (49). ChEMBL is a curated open source chemical database containing over two million compounds and over 17 million bioactivities (version ChEMBL 28). In this context, computer-readable compound notations like SMILES (51) and measured bioactivity values like the  $IC_{50}$  (half maximum inhibitory concentration) are discussed. After becoming familiar with ChEMBL, the students learn how to access the database programmatically and how to filter the obtained chemical and bioactivity data using RDKit functionalities (T001). An overview of the talktorial goal and the programmatic tasks covered in T001 is given in Figure 3 with a few code examples, all taken directly from the respective Jupyter Notebook.

TeachOpenCADD introduces also other databases such as the PDB (50) (T010), a database for biological macromolecular structures. Furthermore, new talktorials are currently being included covering PubChem (52), the largest collection of freely accessible compounds, and KLIFS (53), which integrates structural data of kinases and their interactions with co-crystallized inhibitors.

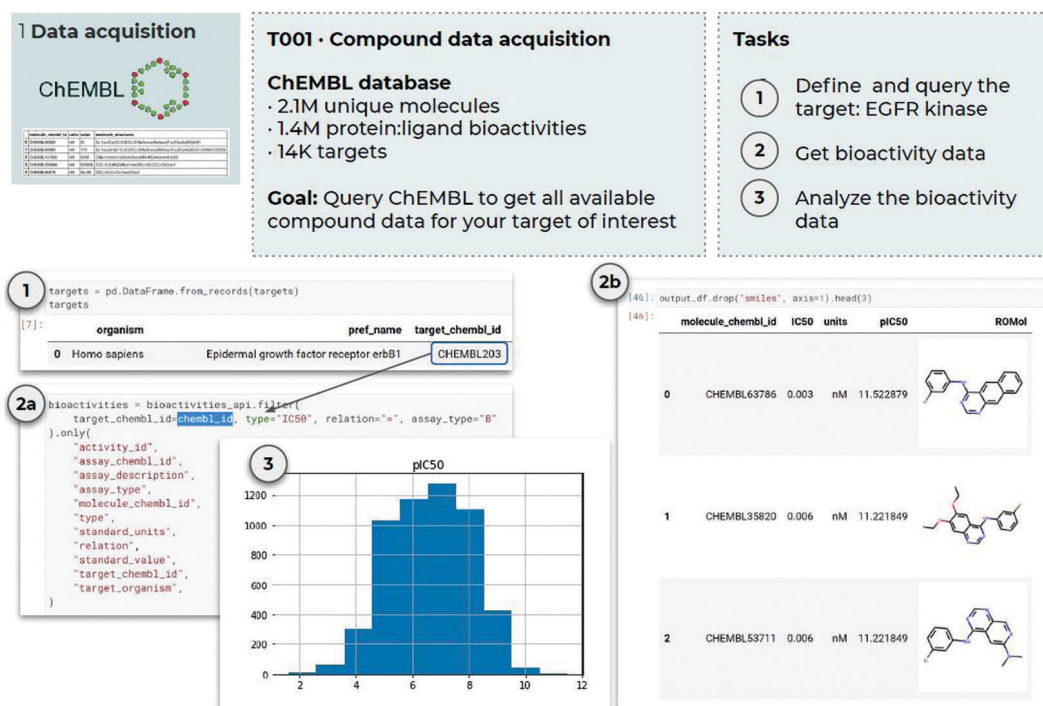


Figure 3. Overview of the goal and practical programming tasks of TeachOpenCADD talktorial T001 (published under a CC-BY-4.0 license) is displayed alongside coding examples taken directly from the Jupyter Notebook (published under a 3-clause BSD license).

### *Compound Descriptors*

When receiving a data set, usually basic questions arise such as: How many compounds are in the data set? What kind of physicochemical properties do they have (e.g., molecular weight, number of hydrogen bond donors/acceptors or logP)? Which compounds are potentially orally bioavailable, following Lipinski's Rule of Five (54)? These questions are addressed in talkorial T002. The composition of a data set, i.e., the chemical similarity or diversity of the compounds, is often investigated next. For many computations, molecules need to be transformed into computer-readable formats. Prominent examples for such encodings are molecular fingerprints (T004). Those fingerprints are often bit vectors that encode the presence or absence of pre-defined rule-based substructures as in MACCS fingerprints (55) or of circular atom environments as in Morgan fingerprints (56, 57).

### *Compound Similarity*

Many tasks in cheminformatics revolve around the assumption that similar compounds may bind to similar targets, and thereby exert similar biological effects. Thus, it is of interest to find similar compounds with respect to a query compounds as shown in talkorial T004 (virtual screening using similarity search). Similarity measures taught are the popular Tanimoto and Dice metrics (58), e.g., calculated based on the molecular fingerprints of the molecules. Furthermore, the composition of a data set can be analyzed by clustering the compounds based on distances between their molecular fingerprints. Representative compounds can then be extracted to build a more diverse subset. This is shown in talkorial T005 using the Butina clustering algorithm (59).

### *Compound Activity Prediction*

Relationships between the structure and physicochemical properties of a compound and its bioactivities are not necessarily linear, as assumed in the simplified similarity search. Thus, machine learning (ML) methods are applied to learn the non-linear patterns distinguishing active from inactive compounds in a labelled training data set. In talkorial T007, students are taught how to build standard ML classification models to predict if a new compound is active or inactive against a query target. Discussed supervised classification models are random forest (RF) (60), support vector (SVM) (61), and neural network (NN) classifiers (62).

### *Compound Substructures*

Medicinal chemists are well trained in finding and understanding important or critical substructures in a molecule by eye. Encoding such knowledge computationally can help to screen large data sets and to highlight important substructures for quick visual inspection. On the one hand, pre-defined substructures, e.g., encoded via SMARTS patterns (63), can be used to flag or filter molecules (T003). Such substructures can include knowledge from medicinal chemistry on toxicity and reactivity (64) or on pan-assay interference (PAINS) (65) of tested compounds. On the other hand, mutual substructures in a set of compounds can be used to assess chemical diversity or to define a common core fragment for structure-activity-relationship (SAR) studies. Thus, in talkorial T006, students are introduced to a maximum common substructure (MCS) search algorithm (66) to rationalize the commonalities within the clustered compounds.

### *Structural Bioinformatics – A Glimpse*

In the first set of talktorials (20), we cover three topics from structural bioinformatics, including data acquisition from the PDB (T008), ligand-based pharmacophores (T009), and off-target prediction (T010). Similarly to the ChEMBL query (T001), we introduce the Protein Data Bank (PDB) (50) and how to programmatically access structural data from it (T008). Exemplified by the EGFR target, ligand-bound structures are fetched from the PDB. Furthermore, structural alignments are performed to access and save a superposed set of ligands for further analysis. In T009, the concept of ligand-based pharmacophores (67) is introduced and the superposed ligands are reused. Pharmacophoric features, which describe potential interactions such as hydrogen bond donors, acceptors, and hydrophobic contacts, are detected for each of the ligands. Furthermore, ligand-based ensemble pharmacophores are generated by clustering the individual pharmacophores in 3D space. Such a pharmacophore model could be used for virtual screening against a compound database to find compounds that match the predefined pharmacophore features. The third talktorial (T010) covers simple geometry-based binding site comparison to predict off-targets (68). Off-targets are proteins that interact with a drug or (one of) its metabolite(s) without being the designated target, potentially causing unwanted side effects.

Further talktorials from the upcoming 2021 release include topics such as binding site detection (69), protein-ligand docking (70), protein-ligand interaction detection (71), and structure visualization in Jupyter Notebooks using NGLview (72).

### **Python Programming**

Introducing novices to programming, i.e., being able to read, understand and produce code, involves multiple layers. Depending on the students' background, some content may already be familiar but a refresher on the basics is usually beneficial. These layers of increasing complexity include the following ideas.

1. Introduce basic programming concepts and define corresponding terms: What is meant by variable, data structures, flow control, function, or module?
2. Exemplify the respective programming language syntax: How do we actually write all this as Python code?
3. Explore the scope of available libraries: How can we import external code? Which libraries are widely used in the community? What functionality do they provide?
4. Introduce best practices: With a bit of coding experience in mind, how do we write code that will be easier to understand and reusable by ourselves and others in the future?

The TeachOpenCADD platform itself is not meant as a Python programming course but demonstrates how to solve concrete tasks programmatically. Instead, our AI in Medicine repository (34) offers introductory talktorials on Python basics and important data science libraries, covering the previously described layers. This knowledge is then extended by domain-specific applications and libraries in eachOpenCADD. Furthermore, we lead by example and introduce Python best practices. We enforce them in all published talktorials and encourage them in all the students' hands-on exercises. In the following, we discuss a few of the Python programming talktorials from the AI in Medicine repository (34), as summarized in Table 2. Note that there are other excellent Python programming resources such as the "Python for Chemists" course set up by the GDCh/CIC team



(35), the "MolSSI Education Resources" assembled by the The Molecular Sciences Software Institute (MolSSI) (36), or the "Core Lessons" offered by the Software Carpentry (37).

#### *Python Introduction*

In the talktorial "Python Programming: Introduction to the Language" (73), we demonstrate how to use Jupyter Notebooks and we introduce Python data structures. This introduction covers how to assign variables and perform operations on them, to index and slice lists, and to create and alter dictionaries. We talk about flow controls, by taking decision with if-else conditions and by repeating actions with for-loops. Last but not least, we show how to reuse code by defining and calling functions. We end the lesson with a short teaser on the power of importing external libraries, which is discussed in detail in the material discussed below.

#### *Data Science Introduction*

NumPy (41), Pandas (42), Matplotlib (43), and Scikit-learn (44) are widely used versatile libraries for data science. They deserve an introduction regardless of whether they will be used in context of cheminformatics or not. After a short introduction to the scientific computing package NumPy, we dive right into Pandas for data manipulation and analysis with easy-to-navigate tabular rendering in Jupyter Notebook ("Python Programming: NumPy/Pandas" (74)). Data slicing is introduced, i.e., selecting data by columns and rows and grouping data by a category of interest. This is complemented by showcasing easy and quick plotting options in Pandas. For more advanced plotting scenarios, students can take a look at the detailed talktorial "Python Programming: Data Visualisation using Matplotlib" (75). Additionally, "Python Programming: Machine learning using Scikit-learn" (76) gives a brief overview on building predictive machine learning models.

#### *Best Practices Introduction*

In many cases, learning to write code that solves a task comes first and only after a considerable amount of time the somewhat advanced programmer gets in touch with coding best practices. These include idiomatic conventions for Python code (Pythonic code) (77), version control with Git (78), and collaborative work processes on platforms like GitHub (29). In order to make our students aware of such best practices, we provide some guideline materials (79), focusing on idiomatic Python style, version control with Git and collaboration on GitHub.

**Table 2. Selected Programming Resources and Keywords**

<i>Topic</i>	<i>Keywords</i>
Python introduction (73)	Variables, flow controls, functions, libraries
Data science introduction	
NumPy/Pandas introduction (74)	Data analysis and manipulation
Matplotlib introduction (75)	Data visualization
Scikit-learn introduction (76)	Machine learning
Best practices introduction (79)	Jupyter, Python, Git

## Training Settings

The TeachOpenCADD material has been developed with the intention to enable many different training settings with different levels of complexity and required skills. The material can be used for student courses at the university in an online or offline *general classroom setting*. Thereby, the material can easily be adapted to the course duration and number of participants. Likewise, the setting can be transferred to a student internship or rotation in research groups (*individual student project setting*). Furthermore, students and researchers can use the material in *self-study settings* to find their way into the field of CADD or to advance their knowledge in a certain area. Finally, the material can serve as a starting point for adaptations to the users' own research questions. Since the material covers introductions as well as advanced tasks, it can be used by both beginners and more advanced users. In the following, some of these settings are outlined with concrete examples for varying levels of complexity.

### General Classroom Setting

Proposed frameworks for using TeachOpenCADD in a classroom setting are summarized in Figure 4. In the following, we will discuss the individual training phases, levels of complexity depending on the students' background, and explain our own course setup while teaching bioinformatics master students. Please refer to the section Lessons learned when using the TeachOpenCADD material for suggestions on how to set up the work space based on the scope of the course.

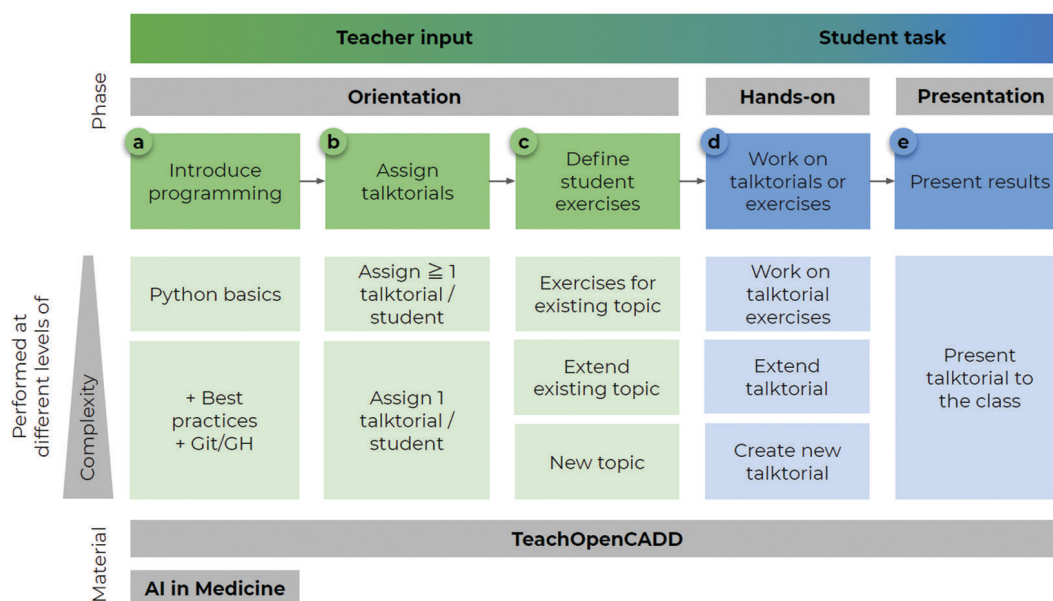


Figure 4. Proposed training settings for classrooms with different levels of complexity based on the students' background (low, intermediate and high complexity indicated by the light green/blue boxes): In the orientation phase, the teacher introduces the students to programming, assigns TeachOpenCADD talktorials and defines tasks (a-c). The students work on their tasks during the hands-on phase (d) and present their results in the presentation phase (e).

*Orientation, Hands-On, and Presentation Phases*

We propose to use TeachOpenCADD in a classroom setting divided into orientation, hands-on, and presentation phases.

*Orientation Phase*

The teacher gives a basic introduction into programming concepts, enabling the students to at least read and understand code. Then, the available talktorial(s) are assigned to the individuals or groups, alongside pre-defined exercises that the students are supposed to work on (Figure 4 a-c). Potential levels of complexity are discussed later in detail. This phase involves a fair amount of input by the teacher and can take a day or longer depending on the level of input detail.

*Hands-On Phase*

After the introduction and orientation, the students get the chance to dive into their topics. This starts with understanding and executing their assigned talktorials, followed by working on the given exercises (Figure 4 d). The teacher is available for questions and discussions, e.g., during Q&A sessions with the whole group, subgroups or individuals. The duration of this phase can be adapted to the level of the tasks' complexity and the students' background.

*Presentation Phase*

The course finishes with the student presentations, in which they exchange their newly gained knowledge. Every student has the opportunity to walk the group through their talktorial(s), explaining the topic's theoretical background and the aim of the exercise as well as demonstrating the code, if applicable (Figure 4 e). In order to facilitate a lively discussion during and after the presentations, it can be beneficial to promote a few students to be the respective session's chairs (rotating per presentation). This ensures that there will be questions to discuss.

Note that the teacher could also choose to walk the students through the available talktorials in a lecture style and to let the students work on small exercises in between. In this case, the orientation, hands-on, and presentation phases would be more intermingled.

*Levels of Complexity*

The complexity of the training settings, including the given exercises, can easily be adapted to the students' background.

*Low Complexity*

The teacher introduces basic Python programming concepts and assigns one or more talktorials to each student. The teacher hands out small coding exercises or questions of understanding, closely related to the practical part covered in the talktorial. Examples could be as follows:

- T001: Starting point is the retrieval of bioactivity data from ChEMBL for the kinase EGFR. The students' task could be to apply the notebook to another target such as the SARS-CoV-2 spike protein. This involves finding the respective accession ID in the UniProt database (80), rerunning the notebook, and analyzing the results.

- T003: Based on the substructure searches discussed in this talktorial, students could discuss how these substructures are encoded as SMARTS patterns. These patterns could then be interactively explored using the SMARTS Plus webserver (81).
- T004: This talktorial teaches different compound encodings and similarity search. Students could rerun the talktorial with a novel set of compounds to find the most similar compounds to a new SMILES query.

This complexity level with the proposed exercises would fit also nicely in cases where the teacher chooses to go through the talktorials together with the students.

#### *Intermediate Complexity*

Each student is assigned one talktorial to explore a topic and asked to extend the existing content with a related but independent task. In this intermediate state, students should produce their content, i.e., their first lines of code, more independently. If such an extension should be shared amongst the students or with the teacher for review on GitHub, programming best practices and version control tools should be added to the Python introduction (79). Note that this is optional depending on the students' and teacher's background and can be seen as a transition to the higher complexity setup. Examples for such related but independent tasks could be:

- T004: Starting point is the encoding of compounds as MACCS and Morgan fingerprints, with the aim of finding similar compounds to a query using the Tanimoto and Dice metrics. Students could research and apply alternative fingerprint encodings or similarity measures; and discuss their pros and cons.
- T007: Starting points are the introduced machine learning methods to prioritize potentially active compounds. Students could try to optimize the random forest (RF) models, for example fine-tuning the hyper-parameters, or investigate which features (i.e., fingerprint bits) were most important for the RF classification.

#### *High Complexity*

In this scenario, the setup is similar as in the level before, including the Git/GitHub and Python best practices introduction. However, the students do not extend an existing talktorial but create a new one from scratch. This includes composing the theory and programming part adhering to the TeachOpenCADD talktorial template (82). Examples could be as follows:

- Perform a principle component analysis (PCA) to visualize the chemical space of a given compound data set.
- Build a regression model that predicts compound bioactivities trained on a given compound data set with known bioactivity values. Discuss applications for classification (T007) vs. regression models.

#### *Bioinformatics Seminar Setup*

Once a year, we offer a CADD seminar for Bioinformatics Master students. Since they come from diverse Bachelor study programs, they exhibit mixed scientific backgrounds. Some training programs are more on the biology/chemistry side, while others are more on the computational side. Thus, the level of practical experience in Python programming can also differ largely. The course

stretches over one semester with (bi-)weekly sessions of 2-3 hours with roughly 10-12 students. Normally, all sessions are offered in person. Students are presenting or working on their notebooks with the option to ask questions when needed. During the SARS-CoV-19 pandemic, our sessions took place via video calls.

- (i) *Introduction*: On the first day, we start with an introduction to CADD, Python programming, best practices and Git/GitHub (79). Then, topics are assigned to the students. Note that we started from scratch in the first year, so naturally we offered a list of new topics. Beginning with the second round, the topics are often related to existing talktorials, however some can cover new terrain. Students pick their topics of interest. Next, they take a look at related TeachOpenCADD talktorials (if available) and study literature (distributed by us) to get familiar with the theoretical background of their topic.
- (ii) *Short topic presentations*: In the second seminar, everyone shortly pitches their topic to the class in a 15 minutes presentation using a medium of their choice.
- (iii) *Working on talktorials*: During the following three weeks, students work on their talktorials, which follow the same setup as the existing talktorials. They cover the aim of the talktorial including a table of contents and references, the theoretical background, the practical coding part, discussion and quiz (adhering to the TeachOpenCADD talktorial template (82)).
- (iv) *Q&A sessions*: We are available during the Q&A sessions (once a week for 2 to 3 hours) to discuss problems, questions and ideas. Students are asked to submit their talktorial progress to GitHub regularly using pull requests (see (79)). This way, we can review the content and code and they gain practical experience with version control and code reviews.
- (v) *Presentation of established talktorials*: The last three sessions are reserved for the student presentations where each student has 30 minutes to present the talktorial and take questions. Per presentation, three students are assigned as session chairs to ensure that there are questions to discuss. Note that the names of the students and other contributors, who worked on the published talktorials, are mentioned in the respective talktorials.

### Individual Student Projects

The setting described for classrooms is also applicable to individual student projects such as internships in a research group. Note we used this setup mostly as a follow-up of the seminar described above. If students were interested in continuing working on talktorials, they joined for a two month research internship. Nevertheless, the setup is equally suitable for interested students who are preferably familiar with installing software. These students begin with studying the available TeachOpenCADD material on their own. Depending on their background they might also start with an introduction to Python programming (see Table 2). After assigning a topic, similar as in the classroom setting, the students start getting familiar with the selected topic, studying the respective literature and composing the outline of their talktorial. In regular meetings, the supervisor is available for questions and checks if the students understand the material. Once the foundation is laid, the students can start working on their topics, while following the TeachOpenCADD talktorial template (82). The students share their work and progress continuously via GitHub allowing for code reviews by the supervisor. At the end of the internship, the students present their work to the group in the context of a regular group seminar to discuss their results and challenges.

### Self-Study Setting

Judging from individual feedback we received, the probably most frequent setting is using TeachOpenCADD for self-study. As mentioned before, the TeachOpenCADD material is available in three interaction modes. First, users can inspect the TeachOpenCADD material in a read-only mode via our website, which contains rendered versions of the talktorials. This allows everyone who is interested in learning more about the CADD concepts to have starting material, covering theory and practical examples in one place. Second, users can run the talktorials remotely in an executable environment called Binder (83) or in Google Colab (84), if they want to get first-hand experiences without any installation hurdles. Third, users can download the TeachOpenCADD material locally to execute and modify the Jupyter Notebooks. Besides the teaching character, each talktorial solves an important research question on its own, and/or can be stacked together to a drug design pipeline. Thus, it can be used as a starting point for individual research projects. In that regard, the talktorials also inspire the work of people in our group. This is true especially for those who just started because there is a central place, TeachOpenCADD, to look up common tasks in cheminformatics and structural bioinformatics.

### Lessons Learned When Using the TeachOpenCADD Material

In the previous sections, we have laid out the pedagogical foundations of our CADD courses, which ultimately led to the creation of TeachOpenCADD. Alongside the didactic challenges, we also found technical hurdles that might hinder the teaching and learning experience. This section will hopefully clarify some of these details. Furthermore, experiences from colleagues that used the material are summarized.

#### Installation and Setup: Reduce Entry Barriers

The first barrier the students face when trying to learn CADD is often simply getting started. Setting up the work space in their own computers, which includes installing and configuring certain pieces of unfamiliar software. This can be a daunting task that involves many new concepts, such as Python distributions, dependencies, versions or package managers. For some students, this might be their first exposure to programming or even a command-line interface. As a result, teachers need to be mindful and tailor the learning environment to their students' background.

Throughout the years, we taught CADD courses and seminars for bioinformatics students, the AI in Medicine (34) course for medical students and workshops in the BB3R graduate program (85) for pharmaceutical and other students. Thus, we have gained experience with different approaches suitable for different contexts: beginners, intermediate and advanced.

For the complete beginners, we recommend avoiding any kind of local installation altogether. To this end, we refer our users to our website, which includes all the content needed to *understand* the CADD techniques in theory, and peek at the code involved. However, it does not provide an interactive environment, which means that no code will be written or executed.

For the intermediate stage, interactivity can be provided installation-free through services like Binder (83) and Google Colab (84). Both tools allow to import the notebooks directly from the public repository via GitHub URLs. It is a good compromise for casual access to the lessons, but it might create some friction for the students. In Binder, the start-up time can be lengthy, while for Google Colab signing up for an account is required. If resources allow, it would be advisable to engage the IT team. They could set up a local installation on computers available in the facilities or configure a Jupyter Hub (86) instance on the university premises for remote access.

For more advanced users, we have decided on using conda packages to easily install TeachOpenCADD in local instances. These are available for Linux, Windows and MacOS, but require a pre-existing Miniconda or Anaconda (87) installation. That said, once that requirement is satisfied, the whole installation takes two commands (88).

### **Establish Conventions**

While developing the TeachOpenCADD content together with the students, we encountered that individuals have different narrative styles, which makes it hard(er) to follow a set of lessons. Besides, more work from us is needed to make them publication-ready and maintain them. Thus, over time, we enforced more and more that all of our content strictly follows the same structure. This is done deliberately to maintain homogeneity across lessons. Every new lesson added is created from our own template (82) to ensure the resulting table of contents is consistent with the existing content.

We also strive for consistency in the Python code we provide (79). We adhere to the established idiomatic conventions of the community (77, 89) and emphasize their importance early on the coursework. While we believe that discussing the full style guide for the chosen language might be excessive for introductory materials, we do think some key aspects are necessary. In particular, why style is important, and how to name variables, document functions and use whitespace adequately.

### **Review Student Work in a Programmatic Way**

One of the aspects we cover in our courses is how to collaborate (79) on programming projects through the GitHub (29) platform. This handles version control for the project, i.e., an annotated history of changes made in the lessons along time. The review mechanisms implemented on the site (named pull requests) are useful to provide both general and granular feedback on the student submissions. This can be done once (like a final graded evaluation) or –ideally– incrementally to guide the students through the different phases of the submission (define scope, cover theoretical background, implement code needed to solve exercises). Since we work with Jupyter Notebooks, we recommend using the ReviewNB add-on (90) for a better review experience. This can arguably be one more thing to learn in the coursework, but in our opinion the expertise obtained by following industry-standard processes can help in transitioning to other career paths and is worth the investment.

### **Experiences from Courses with Different Backgrounds**

Introducing CADD in different curricula involves dealing with very diverse backgrounds. Some of the students might have had exposure to more technical computing before, but others might be facing a command-line interface for the first time. Some students might be able to execute instructions in the terminal so they can install the dependencies needed for the coursework. Others might struggle with how to copy and paste commands in a text-based interface.

We include here feedback from colleagues who used the material in the context of a course for chemistry/chemical biology as well as pharmacology/pharmacy students. Both courses ran for one semester and had between 15 and 45 students enrolled. In both setups, the TeachOpenCADD material was used in parts, i.e., ideas that fit to the respective curriculum were extracted and adapted, if needed. The former course considered parts of the Jupyter Notebooks, which were provided via a central JupyterHub instance, running on a virtual server hosted by the respective IT center. The latter course covered parts of the KNIME workflows, which were installed locally by the students themselves. The teachers introduced the topics first, while the students followed on their own Jupyter

Notebooks or KNIME workflows, respectively. Then, the students were given individual exercises inspired by or identical to the given material (i.e., even running the whole KNIME workflow on their own target). While the overall feedback was positive, some reported challenges included the need for a basic Python introduction. We covered the latter already in more detail in this book chapter through the AI in Medicine material. Depending on the students' background, this could be considered an extra preceding course in the future. Other feedback was related to slow response times from web services that are queried, which unfortunately is not in our hands. Given no prior experience with KNIME, workflows with a lot of nodes can be overwhelming at first. Students benefited from getting a short demo and then building a small workflow themselves with only a few nodes. Subsequently, they worked on the more complex TeachOpenCADD KNIME workflows.

Given the feedback we also conclude that for the typical chemistry/pharmacy students' background, it could be advantageous to simplify the questions. We suggest that this can be smoothed by digesting the individual assignments into layered questions of increasing complexity. For example, instead of asking to write code to "search ChEMBL for compounds highly similar to a given query compound", one could create the following subtasks:

- What is the SMILES representation of a chemical compound?
- How can you encode it in a machine readable version (molecular fingerprints)?
- How would you compare two compounds that might be similar? Do you know of any metric to calculate this quantitatively?
- Are there any databases that allow you to search for similar compounds?
- What methods can you use to query databases programmatically?
- Define a function that, given a SMILES string, will query a database of your choice for chemical compounds with a similarity above a chosen threshold (e.g., 90%).

### Conclusion

The TeachOpenCADD platform is a rich resource for training material on common tasks in cheminformatics and structural biology. Jupyter Notebooks cover both computational concepts and resources as well as Python code in one place (talktorials). In case an entry level introduction to Python programming is needed, the TeachOpenCADD material can be supplemented by introductory talktorials from the AI in Medicine material. Due to the narrative and coding character of the talktorials, the material can be used in many different ways depending on the given training setting and the students' background. In this book chapter, we have outlined different possible training scenarios from low to high complexity and reported our own and our colleagues' experiences with the material. We are happy to help you if you consider using TeachOpenCADD in your teaching curriculum. TeachOpenCADD is a living resource. In case used packages or web services (91) change or get deprecated, we are notified thanks to automated notebook checks (continuous integration (92)) that run nightly or thanks to our users via GitHub issues. Both notification systems are public to everyone. We ourselves or external contributors have been and will continue working on fixing such issues to provide fully-functional teaching material. TeachOpenCADD is not only maintained but is also continuously updated, as part of our teaching as well as project related work. Contributions from the community are always very welcome.



### Acknowledgments

We thank Albert Kooistra (University of Copenhagen, Denmark) and Paul Czodowski (TU Dortmund, Germany) for their feedback on their experiences with the TeachOpenCADD material. Furthermore, we gratefully acknowledge all Volkamer Lab members who have been co-supervising students, polishing talktorials and assisting teaching in the broader context of TeachOpenCADD since 2017 as well as all students working with and on the TeachOpenCADD material. We also thank all Ritter Lab (Charité – Universitätsmedizin Berlin) members who have been working with us on the AI in Medicine material.

### References

1. Brown, N.; Ertl, P.; Lewis, R.; Luksch, T.; Reker, D.; Schneider, N. Artificial Intelligence In Chemistry And Drug Design. *Journal of Computer-Aided Molecular Design* **2020**, *34* (7), 709–715. DOI: <https://doi.org/10.1007/s10822-020-00317-x>.
2. Bender, A.; Cortés-Ciriano, I. Artificial Intelligence In Drug Discovery: What Is Realistic, What Are Illusions? Part 1: Ways To Make An Impact, Why We Are Not There Yet. *Drug Discovery Today* **2021**, *26* (2), 511–524. DOI: <https://doi.org/10.1016/j.drudis.2020.12.009>.
3. Griffen, E. J.; Dossetter, A. G.; Leach, A. G. Chemists: AI Is Here; Unite To Get The Benefits. *Journal of Medicinal Chemistry* **2020**, *63* (16), 8695–8704. DOI: <https://doi.org/10.1021/acs.jmedchem.0c00163>.
4. Struble, T. J.; Alvarez, J. C.; Brown, S. P.; Chytil, M.; Cisar, J.; DesJarlais, R. L.; Engkvist, O.; Frank, S. A.; Greve, D. R.; Griffin, D. J.; Hou, X.; Johannes, J. W.; Kreatsoulas, C.; Lahue, B.; Mathea, M.; Mogk, G.; Nicolaou, C. A.; Palmer, A. D.; Price, D. J.; Robinson, R. I.; Salentin, S.; Xing, L.; Jaakkola, T.; Green, W. H.; Barzilay, R.; Coley, C. W.; Jensen, K. F. Current And Future Roles Of Artificial Intelligence In Medicinal Chemistry Synthesis. *Journal of Medicinal Chemistry* **2020**, *63* (16), 8667–8682. DOI: <https://doi.org/10.1021/acs.jmedchem.9b02120>.
5. Lusher, S. J.; McGuire, R.; van Schaik, R. C.; Nicholson, C. D.; de Vlieg, J. Data-Driven Medicinal Chemistry In The Era Of Big Data. *Drug Discovery Today* **2014**, *19* (7), 859–868. DOI: <https://doi.org/10.1016/j.drudis.2013.12.004>.
6. Bajorath, J. Foundations Of Data-Driven Medicinal Chemistry. *Future Science OA* **2018**, *4* (8), FSO320. DOI: <https://doi.org/10.4155/fsoa-2018-0057>.
7. Ritchie, T. J.; McLay, I. M. Should Medicinal Chemists Do Molecular Modelling? *Drug Discovery Today* **2012**, *17* (11-12), 534–537. DOI: <https://doi.org/10.1016/j.drudis.2012.01.005>.
8. Sliwoski, G.; Kothiwale, S.; Meiler, J.; Lowe, E. W. Computational Methods In Drug Discovery. *Pharmacological Reviews* **2013**, *66* (1), 334–395. DOI: <https://doi.org/10.1124/pr.112.007336>.
9. Drie, J. H. V. Computer-Aided Drug Design: The Next 20 Years. *Journal of Computer-Aided Molecular Design* **2007**, *21* (10-11), 591–601. DOI: <https://doi.org/10.1007/s10822-007-9142-y>.

10. Wouters, O. J.; McKee, M.; Luyten, J. Estimated Research And Development Investment Needed To Bring A New Medicine To Market, 2009-2018. *JAMA* **2020**, *323* (9), 844–853. DOI: <https://doi.org/10.1001/jama.2020.1166>.
11. Leelananda, S. P.; Lindert, S. Computational Methods In Drug Discovery. *Beilstein journal of organic chemistry* **2016**, *12*, 2694–2718. DOI: <https://doi.org/10.3762/bjoc.12.267>.
12. Paul, S. M.; Mytelka, D. S.; Dunwiddie, C. T.; Persinger, C. C.; Munos, B. H.; Lindborg, S. R.; Schacht, A. L. How To Improve R&D Productivity: The Pharmaceutical Industry's Grand Challenge. *Nature Reviews Drug Discovery* **2010**, *9* (3), 203–214. DOI: <https://doi.org/10.1038/nrd3078>.
13. Waring, M. J.; Arrowsmith, J.; Leach, A. R.; Leeson, P. D.; Mandrell, S.; Owen, R. M.; Pairaudeau, G.; Pennie, W. D.; Pickett, S. D.; Wang, J.; Wallace, O.; Weir, A. An Analysis Of The Attrition Of Drug Candidates From Four Major Pharmaceutical Companies. *Nature Reviews Drug Discovery* **2015**, *14* (7), 475–486. DOI: <https://doi.org/10.1038/nrd4609>.
14. Talele, T.; Khedkar, S.; Rigby, A. Successful Applications Of Computer Aided Drug Discovery: Moving Drugs From Concept To The Clinic. *Current Topics in Medicinal Chemistry* **2010**, *10* (1), 127–141. DOI: <https://doi.org/10.2174/156802610790232251>.
15. Macalino, S. J. Y.; Gosu, V.; Hong, S.; Choi, S. Role Of Computer-Aided Drug Design In Modern Drug Discovery. *Archives of Pharmacal Research* **2015**, *38* (9), 1686–1701. DOI: <https://doi.org/10.1007/s12272-015-0640-5>.
16. Jorgensen, W. L. The Many Roles Of Computation In Drug Discovery. *Science* **2004**, *303* (5665), 1813–1818. DOI: <https://doi.org/10.1126/science.1096361>.
17. Tantillo, D. J.; Siegel, J. B.; Saunders, C. M.; Palazzo, T. A.; Painter, P. P.; O'Brien, T. E.; Nuñez, N. N.; Nouri, D. H.; Lodewyk, M. W.; Hudson, B. M.; Hare, S. R.; Davis, R. L. Computer-Aided Drug Design For Undergraduates. *Journal of Chemical Education* **2019**, *96* (5), 920–925. DOI: <https://doi.org/10.1021/acs.jchemed.8b00712>.
18. Rafferty, M. F. No Denying It: Medicinal Chemistry Training Is In Big Trouble. *Journal of Medicinal Chemistry* **2016**, *59* (24), 10859–10864. DOI: <https://doi.org/10.1021/acs.jmedchem.6b00741>.
19. Ainsworth, S. J. Universities Tailor Programs To Meet The Pharmaceutical Industry's Needs. *Chem. Eng. News* **2014**, *92* (21), 65–69.
20. Sydow, D.; Morger, A.; Driller, M.; Volkamer, A. TeachOpenCADD: A Teaching Platform For Computer-Aided Drug Design Using Open Source Packages And Data. *J. Cheminform.* **2019**, *11* (1), 29. DOI: <https://doi.org/10.1186/s13321-019-0351-x>.
21. Sydow, D.; Wichmann, M.; Rodríguez-Guerra, J.; Goldmann, D.; Landrum, G.; Volkamer, A. TeachOpenCADD-KNIME: A Teaching Platform For Computer-Aided Drug Design Using KNIME Workflows. *Journal of Chemical Information and Modeling* **2019**, *59* (10), 4083–4086. DOI: <https://doi.org/10.1021/acs.jcim.9b00662>.
22. Wilkinson, M. D.; Dumontier, M.; Aalbersberg, I. J.; Appleton, G.; Axton, M.; Baak, A.; Blomberg, N.; Boiten, J.-W.; da Silva Santos, L. B.; Bourne, P. E.; Bouwman, J.; Brookes, A. J.; Clark, T.; Crosas, M.; Dillo, I.; Dumon, O.; Edmunds, S.; Evelo, C. T.; Finkers, R.; Gonzalez-Beltran, A.; Gray, A. J.; Groth, P.; Goble, C.; Grethe, J. S.; Heringa, J.; 't Hoen, P. A.; Hooft, R.; Kuhn, T.; Kok, R.; Kok, J.; Lusher, S. J.; Martone, M. E.; Mons, A.; Packer, A. L.; Persson, B.; Rocca-Serra, P.; Roos, M.; van Schaik, R.; Sansone, S.-A.; Schultes, E.; Sengstag, T.;

- Slater, T.; Strawn, G.; Swertz, M. A.; Thompson, M.; van der Lei, J.; van Mulligen, E.; Velterop, J.; Waagmeester, A.; Wittenburg, P.; Wolstencroft, K.; Zhao, J.; Mons, B. The FAIR Guiding Principles For Scientific Data Management And Stewardship. *Scientific Data* **2016**, *3* (1). DOI: <https://doi.org/10.1038/sdata.2016.18>.
23. Steinbeck, C.; Han, Y.; Kuhn, S.; Horlacher, O.; Luttmann, E.; Willighagen, E. The Chemistry Development Kit (CDK): An Open-Source Java Library For Chemo- And Bioinformatics. *Journal of Chemical Information and Computer Sciences* **2003**, *43* (2), 493–500. DOI: <https://doi.org/10.1021/ci025584y>.
24. Willighagen, E. L.; Mayfield, J. W.; Alvarsson, J.; Berg, A.; Carlsson, L.; Jeliaskova, N.; Kuhn, S.; Pluskal, T.; Rojas-Chertó, M.; Spjuth, O.; Torrance, G.; Evelo, C. T.; Guha, R.; Steinbeck, C. The Chemistry Development Kit (CDK) V2.0: Atom Typing, Depiction, Molecular Formulas, Substructure Searching. *Journal of Cheminformatics* **2017**, *9* (1). DOI: <https://doi.org/10.1186/s13321-017-0220-4>.
25. Jansen, J. M.; Cornell, W.; Tseng, Y. J.; Amaro, R. E. Teach-Discover-Treat (TDT): Collaborative Computational Drug Discovery For Neglected Diseases. *Journal of Molecular Graphics and Modelling* **2012**, *38*, 360–362. DOI: <https://doi.org/10.1016/j.jmgm.2012.07.007>.
26. T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, C. Willing, and J. development team, “Jupyter Notebooks - A Publishing Format For Reproducible Computational Workflows,” in *Positioning and Power in Academic Publishing: Players, Agents and Agendas* (F. Loizides and B. Schmidt, eds.), (Netherlands), pp. 87–90, IOS Press, 2016. DOI: <https://doi.org/10.3233/978-1-61499-649-1-87>.
27. Jupyter community, “A Gallery Of Interesting Jupyter Notebooks.” <https://github.com/jupyter/jupyter/wiki/A-gallery-of-interesting-Jupyter-Notebooks>. [Online; accessed 17-March-2021].
28. RDKit, “9th RDKit UGM.” [https://github.com/rdkit/UGM\\_2020](https://github.com/rdkit/UGM_2020), Oct. 2020. [Online; accessed 23-March-2021].
29. GitHub, “GitHub.” <https://github.com>. [Online; accessed 17-March-2021].
30. Berthold, M. R.; Cebron, N.; Dill, F.; Gabriel, T. R.; Kötter, T.; Meinel, T.; Ohl, P.; Sieb, C.; Thiel, K.; Wiswedel, B. “KNIME: The Konstanz Information Miner,” in *Data Analysis, Machine Learning and Applications*, pp. 319–326, Springer Berlin Heidelberg, 2008. DOI: [https://doi.org/10.1007/978-3-540-78246-9\\_38](https://doi.org/10.1007/978-3-540-78246-9_38).
31. Fillbrunn, A.; Dietz, C.; Pfeuffer, J.; Rahn, R.; Landrum, G. A.; Berthold, M. R. KNIME For Reproducible Cross-Domain Analysis Of Life Science Data. *Journal of Biotechnology* **2017**, *261*, 149–156. DOI: <https://doi.org/10.1016/j.jbiotec.2017.07.028>.
32. Mazanetz, M. P.; Goode, C. H.; Chudyk, E. I. Ligand- And Structure-Based Drug Design And Optimization Using KNIME. *Current Medicinal Chemistry* **2020**, *27* (38), 6458–6479. DOI: <https://doi.org/10.2174/0929867326666190409141016>.
33. Kooistra, A. J.; Vass, M.; McGuire, R.; Leurs, R.; de Esch, I. J. P.; Vriend, G.; Verhoeven, S.; de Graaf, C. 3d-e-chem: Structural cheminformatics workflows for computer-aided drug discovery. *ChemMedChem* **2018**Feb, *13*, 614–626. DOI: <https://doi.org/10.1002/cmdc.201700754>.

34. Volkamer Lab and Ritter Lab, “AI In Medicine.” [https://github.com/volkamerlab/ai\\_in\\_medicine/tree/2021.02](https://github.com/volkamerlab/ai_in_medicine/tree/2021.02). [Online; accessed 20-March-2021].
35. GDCh / CIC Team, “Python For Chemists.” <https://github.com/GDChCICTeam/python-for-chemists>. [Online; accessed 23-March-2021].
36. The Molecular Sciences Software Institute (MolSSI), “MolSSI Education Resources.” <http://education.molssi.org/resources.html>. [Online; accessed 23-March-2021].
37. Software Carpentry, “Software Carpentry Core Lessons.” <https://software-carpentry.org/lessons/>. [Online; accessed 23-March-2021].
38. Rodriguez-Guerra, J.; Landrum, G., “Community-Curated List Of Resources From The RDKit UGM 2020.” [https://github.com/rdkit/UGM\\_2020/blob/master/info/curated\\_list\\_of\\_resources.md](https://github.com/rdkit/UGM_2020/blob/master/info/curated_list_of_resources.md). [Online; accessed 19-March-2021].
39. Walters, P., “A Highly Opinionated List Of Open Source Cheminformatics Resources.” [https://github.com/PatWalters/resources/blob/main/cheminformatics\\_resources.md](https://github.com/PatWalters/resources/blob/main/cheminformatics_resources.md). [Online; accessed 19-March-2021].
40. Hsiao, Y., “Awesome Cheminformatics Resources.” <https://github.com/hsiaoyi0504/awesome-cheminformatics#resources>. [Online; accessed 19-March-2021].
41. Harris, C. R.; Millman, K. J.; van der Walt, S. J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N. J.; Kern, R.; Picus, M.; Hoyer, S.; van Kerkwijk, M. H.; Brett, M.; Haldane, A.; del Río, J. F.; Wiebe, M.; Peterson, P.; Gérard-Marchant, P.; Sheppard, K.; Reddy, T.; Weckesser, W.; Abbasi, H.; Gohlke, C.; Oliphant, T. E. Array programming with Numpy. *Nature* **2020**, 585 (7825), 357–362. DOI: <https://doi.org/10.1038/s41586-020-2649-2>.
42. McKinney, W., “Data Structures For Statistical Computing In Python,” in *Proceedings of the 9th Python in Science Conference* (van der Walt, S.; Millman, J., eds.), pp. 56–61, 2010. DOI: <https://doi.org/10.25080/Majora-92bf1922-00a>.
43. Hunter, J. D. Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering* **2007**, 9 (3), 90–95. DOI: <https://doi.org/10.1109/MCSE.2007.55>.
44. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. Scikit-Learn: Machine Learning In Python. *Journal of Machine Learning Research* **2011**, 12, 2825–2830.
45. Landrum, G., “RDKit: Open-Source Cheminformatics.” <http://www.rdkit.org>. [Online; accessed 03-March-2021].
46. Davies, M.; Nowotka, M.; Papadatos, G.; Dedman, N.; Gaulton, A.; Atkinson, F.; Bellis, L.; Overington, J. P. ChEMBL Web Services: Streamlining Access To Drug Discovery Data And Utilities. *Nucleic Acids Research* **2015**, 43, W612–W620. DOI: <https://doi.org/10.1093/nar/gkv352>.
47. Gilpin, W. PyPDB: A Python API For The Protein Data Bank. *Bioinformatics* **2015**, 32 (9), 159–60. DOI: <https://doi.org/10.1093/bioinformatics/btv543>.
48. UniProtKB, “Human EGFR (Epidermal Growth Factor Receptor).” <https://www.uniprot.org/uniprot/P00533>. [Online; accessed 04-March-2021].
49. Gaulton, A.; Hersey, A.; Nowotka, M.; Bento, A. P.; Chambers, J.; Mendez, D.; Mutowo, P.; Atkinson, F.; Bellis, L. J.; Cibrián-Uhalte, E.; Davies, M.; Dedman, N.; Karlsson, A.;

- Magariños, M. P.; Overington, J. P.; Papadatos, G.; Smit, I.; Leach, A. R. The ChEMBL Database In 2017. *Nucleic Acids Research* **2016**, *45* (D1), D945–D954. DOI: <https://doi.org/10.1093/nar/gkw1074>.
50. Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. The Protein Data Bank. *Nucleic Acids Research* **2000**, *28* (1), 235–242. DOI: <https://doi.org/10.1093/nar/28.1.235>.
51. Weininger, D. SMILES, A Chemical Language And Information System. 1. Introduction To Methodology And Encoding Rules. *Journal of Chemical Information and Modeling* **1988**, *28* (1), 31–36. DOI: <https://doi.org/10.1021/ci00057a005>.
52. Kim, S.; Chen, J.; Cheng, T.; Gindulyte, A.; He, J.; He, S.; Li, Q.; Shoemaker, B. A.; Thiessen, P. A.; Yu, B.; Zaslavsky, L.; Zhang, J.; Bolton, E. E. PubChem In 2021: New Data Content And Improved Web Interfaces. *Nucleic Acids Research* **2020**, *49* (D1), D1388–D1395. DOI: <https://doi.org/10.1093/nar/gkaa971>.
53. Kanev, G. K.; de Graaf, C.; Westerman, B. A.; de Esch, I. J. P.; Kooistra, A. J. KLIFS: An Overhaul After The First 5 Years Of Supporting Kinase Research. *Nucleic Acids Research* **2020**, *49* (D1), D562–D569. DOI: <https://doi.org/10.1093/nar/gkaa895>.
54. Lipinski, C. A.; Lombardo, F.; Dominy, B. W.; Feeney, P. J. Experimental And Computational Approaches To Estimate Solubility And Permeability In Drug Discovery And Development Settings. *Advanced Drug Delivery Reviews* **1997**, *23* (1), 3–25. DOI: [https://doi.org/10.1016/S0169-409X\(00\)00129-0](https://doi.org/10.1016/S0169-409X(00)00129-0).
55. Accelrys Inc., San Diego, CA, USA, “MACCS Structural Keys,” 2011.
56. Morgan, H. L. The Generation Of A Unique Machine Description For Chemical Structures-A Technique Developed At Chemical Abstracts Service. *Journal of Chemical Documentation* **1965**, *5* (5), 107–113. DOI: <https://doi.org/10.1021/c160017a018>.
57. Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *Journal of Chemical Information and Modeling* **2010**, *50*, 742–754. DOI: <https://doi.org/10.1021/ci100050t>.
58. Maggiora, G.; Vogt, M.; Stumpfe, D.; Bajorath, J. Molecular Similarity In Medicinal Chemistry. *Journal of Medicinal Chemistry* **2014**, *57* (4), 3186–3204. DOI: <https://doi.org/10.1021/jm401411z>.
59. Butina, D. Unsupervised Data Base Clustering Based On Daylight’s Fingerprint And Tanimoto Similarity: A Fast And Automated Way To Cluster Small And Large Data Sets. *Journal of Chemical Information and Modeling* **1999**, *39*, 747–750. DOI: <https://doi.org/10.1021/ci9803381>.
60. Breiman, L. Random Forests. *Machine Learning* **2001**, *45* (1), 5–32. DOI: <https://doi.org/10.1023/A:1010933404324>.
61. Cortes, C.; Vapnik, V. Support-Vector Networks. *Machine Learning* **1995**, *20* (9), 273–297. DOI: <https://doi.org/10.1007/BF00994018>.
62. Rosenblatt, F., *Principles Of Neurodynamics; Perceptrons And The Theory Of Brain Mechanisms*. Spartan Books, 1962. DOI: [https://doi.org/10.1007/978-3-642-70911-1\\_20](https://doi.org/10.1007/978-3-642-70911-1_20).
63. Daylight Chemical Information Systems Inc., “SMARTS.” <https://www.daylight.com/dayhtml/doc/theory/theory.smarts.html>. [Online; accessed 03-March-2021].
64. Brenk, R.; Schipani, A.; James, D.; Krasowski, A.; Gilbert, I. H.; Frearson, J.; Wyatt, P. G. Lessons Learnt From Assembling Screening Libraries For Drug Discovery For Neglected

- Diseases. *ChemMedChem* **2008**, 3 (3), 435–444. DOI: <https://doi.org/10.1002/cmdc.200700139>.
65. Baell, J. B.; Holloway, G. A. New Substructure Filters For Removal Of Pan Assay Interference Compounds(PAINS) From Screening Libraries And For Their Exclusion In Bioassays. *Journal of Medicinal Chemistry* **2010**, 53 (4), 2719–2740. DOI: <https://doi.org/10.1021/jm901137j>.
66. Raymond, J. W.; Willett, P. Maximum Common Subgraph Isomorphism Algorithms For The Matching Of Chemical Structures. *Journal of Computer-Aided Molecular Design* **2002**, 16 (7), 521–33. DOI: <https://doi.org/10.1023/A:1021271615909>.
67. Seidel, T.; Wolber, G.; Murgueitio, M. S., “Pharmacophore Perception And Applications,” in *Applied Chemoinformatics*, pp. 259–282, Wiley-VCH Verlag GmbH & Co. KGaA, 2018. DOI: <https://doi.org/10.1002/9783527806539.ch6f>.
68. Sydow, D.; Burggraaff, L.; Szengel, A.; van Vlijmen, H. W. T.; IJzerman, A. P.; van Westen, G. J. P.; Volkamer, A. Advances And Challenges In Computational Target Prediction. *Journal of Chemical Information and Modeling* **2019**, 59 (5), 1728–1742. DOI: <https://doi.org/10.1021/acs.jcim.8b00832>.
69. Volkamer, A.; von Behren, M. M.; Bietz, S.; Rarey, M., “Prediction, Analysis, And Comparison Of Active Sites,” in *Applied Chemoinformatics*, pp. 283–311, Wiley-VCH Verlag GmbH & Co. KGaA, 2018. DOI: <https://doi.org/10.1002/9783527806539.ch6g>.
70. Meng, X.-Y.; Zhang, H.-X.; Mezei, M.; Cui, M. Molecular Docking: A Powerful Approach For Structure-Based Drug Discovery. *Current Computer Aided-Drug Design* **2011**, 7 (2), 146–157. DOI: <https://doi.org/10.2174/157340911795677602>.
71. Du, X.; Li, Y.; Xia, Y.-L.; Ai, S.-M.; Liang, J.; Sang, P.; Ji, X.-L.; Liu, S.-Q. Insights Into Protein–Ligand Interactions: Mechanisms, Models, Methods. *International Journal of Molecular Sciences* **2016**, 17 (2), 144. DOI: <https://doi.org/10.3390/ijms17020144>.
72. Nguyen, H.; Case, D. A.; Rose, A. S. NGLView - Interactive Molecular Graphics For Jupyter Notebooks. *Bioinformatics* **2017**, 34 (7), 1241–1242. DOI: <https://doi.org/10.1093/bioinformatics/btx789>.
73. Volkamer Lab and Ritter Lab, “AI In Medicine Talktorial "Python Programming: Introduction To The Language".” [https://github.com/volkamerlab/ai\\_in\\_medicine/blob/2021.02/week1\\_session1\\_grundkonzepte.ipynb](https://github.com/volkamerlab/ai_in_medicine/blob/2021.02/week1_session1_grundkonzepte.ipynb). [Online; accessed 20-March-2021].
74. Volkamer Lab and Ritter Lab, “AI In Medicine Talktorial "Python Programming: Numpy/Pandas”.” [https://github.com/volkamerlab/ai\\_in\\_medicine/blob/2021.02/week1\\_session2\\_numpy\\_pandas.ipynb](https://github.com/volkamerlab/ai_in_medicine/blob/2021.02/week1_session2_numpy_pandas.ipynb). [Online; accessed 20-March-2021].
75. Volkamer Lab and Ritter Lab, “AI In Medicine Talktorial "Python Programming: Data Visualisation Using Matplotlib”.” [https://github.com/volkamerlab/ai\\_in\\_medicine/blob/2021.02/week1\\_session3\\_matplotlib.ipynb](https://github.com/volkamerlab/ai_in_medicine/blob/2021.02/week1_session3_matplotlib.ipynb). [Online; accessed 20-March-2021].
76. Volkamer Lab and Ritter Lab, “AI In Medicine Talktorial "Python Programming: Machine Learning Using Scikit-Learn”.” [https://github.com/volkamerlab/ai\\_in\\_medicine/blob/2021.02/week1\\_session4\\_intro\\_to\\_ml\\_and\\_scikit\\_learn.ipynb](https://github.com/volkamerlab/ai_in_medicine/blob/2021.02/week1_session4_intro_to_ml_and_scikit_learn.ipynb). [Online; accessed 20-March-2021].
77. Python Software Foundation, “Python Enhancement Proposal 8.” <https://www.python.org/dev/peps/pep-0008/>. [Online; accessed 17-March-2021].
78. Software Freedom Conservancy, “The Git Project.” <https://git-scm.com/>. [Online; accessed 23-March-2021].

79. Rodríguez-Guerra, J., “Intro To Best Practices In Jupyter, Python & Git.” <https://doi.org/10.5281/zenodo.4630714>. [Online; accessed 25-March-2021].
80. Bateman, A.; Martin, M.-J.; Orchard, S.; Magrane, M.; Agivetova, R.; Ahmad, S.; Alpi, E.; Bowler-Barnett, E. H.; Britto, R.; Bursteinas, B.; Bye-A-Jee, H.; Coetzee, R.; Cukura, A.; Silva, A. D.; Denny, P.; Dogan, T.; Ebenezer, T.; Fan, J.; Castro, L. G.; Garmiri, P.; Georghiou, G.; Gonzales, L.; Hatton-Ellis, E.; Hussein, A.; Ignatchenko, A.; Insana, G.; Ishtiaq, R.; Jokinen, P.; Joshi, V.; Jyothi, D.; Lock, A.; Lopez, R.; Luciani, A.; Luo, J.; Lussi, Y.; MacDougall, A.; Madeira, F.; Mahmoudy, M.; Menchi, M.; Mishra, A.; Moulang, K.; Nightingale, A.; Oliveira, C. S.; Pundir, S.; Qi, G.; Raj, S.; Rice, D.; Lopez, M. R.; Saidi, R.; Sampson, J.; Sawford, T.; Speretta, E.; Turner, E.; Tyagi, N.; Vasudev, P.; Volynkin, V.; Warner, K.; Watkins, X.; Zaru, R.; Zellner, H.; Bridge, A.; Poux, S.; Redaschi, N.; Aimo, L.; Argoud-Puy, G.; Auchincloss, A.; Axelsen, K.; Bansal, P.; Baratin, D.; Blatter, M.-C.; Bolleman, J.; Boutet, E.; Breuza, L.; Casals-Casas, C.; de Castro, E.; Echioukh, K. C.; Coudert, E.; Cuche, B.; Doche, M.; Dornevil, D.; Estreicher, A.; Famiglietti, M. L.; Feuermann, M.; Gasteiger, E.; Gehant, S.; Gerritsen, V.; Gos, A.; Gruaz-Gumowski, N.; Hinz, U.; Hulo, C.; Hyka-Nouspikel, N.; Jungo, F.; Keller, G.; Kerhornou, A.; Lara, V.; Mercier, P. L.; Lieberherr, D.; Lombardot, T.; Martin, X.; Masson, P.; Morgat, A.; Neto, T. B.; Paesano, S.; Pedruzzi, I.; Pilbout, S.; Pourcel, L.; Pozzato, M.; Pruess, M.; Rivoire, C.; Sigrist, C.; Sonesson, K.; Stutz, A.; Sundaram, S.; Tognolli, M.; Verbregue, L.; Wu, C. H.; Arighi, C. N.; Arminski, L.; Chen, C.; Chen, Y.; Garavelli, J. S.; Huang, H.; Laiho, K.; McGarvey, P.; Natale, D. A.; Ross, K.; Vinayaka, C. R.; Wang, Q.; Wang, Y.; Yeh, L.-S.; Zhang, J.; Ruch, P.; Teodoro, D. UniProt: The Universal Protein Knowledgebase In 2021. *Nucleic Acids Research* **2020**, 49 (D1), D480–D489. DOI: <https://doi.org/10.1093/nar/gkaa1100>.
81. Schomburg, K.; Ehrlich, H.-C.; Stierand, K.; Rarey, M. From Structure Diagrams To Visual Chemical Patterns. *Journal of Chemical Information and Modeling* **2010**, 50 (9), 1529–1535. DOI: <https://doi.org/10.1021/ci100209a>.
82. Volkamer Lab, “TeachOpenCADD Talktorial Template.” [https://github.com/volkamerlab/teachopencadd/blob/master/teachopencadd/talktorials/T000\\_template/talktorial.ipynb](https://github.com/volkamerlab/teachopencadd/blob/master/teachopencadd/talktorials/T000_template/talktorial.ipynb). [Online; accessed 11-March-2021].
83. Project Jupyter, Bussonnier, M.; Forde, J.; Freeman, J.; Granger, B.; Head, T.; Holdgraf, C.; Kelley, K.; Nalvarte, G.; Osheroff, A.; Pacer, M.; Panda, Y.; Perez, F.; Kelley, B. R.; Willing, C., “Binder 2.0 - Reproducible, Interactive, Sharable Environments For Science At Scale ,” in *Proceedings of the 17th Python in Science Conference* (Akici, F.; Lippa, D.; Niederhut, D.; Pacer, M., eds.), pp. 113 – 120, 2018. DOI: <https://doi.org/10.25080/Majora-4af1f417-011>.
84. Google Research, “Google Colab.” <https://colab.research.google.com/>. [Online; accessed 17-March-2021].
85. Volkamer Lab, “In Silico Toxicity/3R Workshop.” <https://www.bb3r.de/en/graduierntenkolleg/index.html>. [Online; accessed 18-March-2021].
86. Project Jupyter, “Jupyter Hub.” <https://jupyter.org/hub>. [Online; accessed 17-March-2021].
87. Anaconda, “Anaconda Individual Edition.” <https://www.anaconda.com/products/individual>. [Online; accessed 17-March-2021].
88. Volkamer Lab, “How To Install TeachOpenCADD.” <https://projects.volkamerlab.org/teachopencadd/installing.html>. [Online; accessed 25-March-2021].

89. Python Software Foundation, “*Black: The Uncompromising Python Code Formatter.*” <https://github.com/psf/black>. [Online; accessed 17-March-2021].
90. Rathi, A., “*ReviewNB: Rich Diffs & Commenting For Jupyter Notebooks.*” <https://www.reviewnb.com/>. [Online; accessed 17-March-2021].
91. Volkamer Lab, “*TeachOpenCADD’s External Resources.*” <https://github.com/volkamerlab/teachopencadd#acknowledgments>. [Online; accessed 14-May-2021].
92. GitHub, “*About Continuous Integration.*” <https://docs.github.com/en/actions/guides/about-continuous-integration>. [Online; accessed 17-May-2021].



## 5.2 Further Projects

### 5.2.1 Ratar: Read-Across the Targetome

The unpublished work on "Read-Across the Targetome" (Ratar) was conducted as part of the DFG project 391684253 [206] alongside the aforementioned published work, i.e., the review on computational target prediction (Section 1.2.1) and projects such as KiSSim (Sections 3.1.1 and 3.1.3), KinFragLib (Section 3.2.1), TeachOpenCADD (Sections 3.3.1, 3.3.2, 5.1.1, and 5.1.2), and OpenCADD (Section 3.3.3).

The preliminary results from this early stage development of Ratar will be presented in this section. The associated software described here is open-sourced on GitHub in the context of the `ratar` Python package [161].

#### Introduction

How to probe and validate a potential pathway or target remains one of the key questions in basic research in life sciences. Often these investigations lack suitable chemical tool compounds for the elucidation of the function of a specific protein. Platforms such as Guide to Pharmacology [90, 207] and Chemical Probes Portal [208] summarize known tool compounds, while consortia such as the Structural Genomics Consortium [147] and Target 2035 [149, 209–211] have formed to generate novel tool compounds for the validation of biological targets.

While these efforts will continue to summarize and generate experimental results, computational solutions can offer a fast and cheap alternative for the generation of a comprehensive set of tool compounds for novel targets. Tools for computational target prediction have been discussed in Section 1.2.1 and reviewed in detail in Sydow and Burggraaff et al. [22] (Publication A). While ligand-based target prediction methods focus only on the similarity between small molecules, structure-based methods take into account information from protein binding sites. Shortcomings of the latter methods include either long runtime or non-compliance with FAIR principles; i.e., the tools are not available at all or for free usage, the tools are only available via a webserver but not as a stand-alone tool to be incorporated into pipelines, or the setup and maintenance is difficult. As part of the "Read-Across the Targetome" (Ratar) project, we aim to overcome these challenges and deliver a fast and FAIR target prediction tool. The project's *hypothesis* is based on the similarity principle, i.e., similar pockets bind similar compounds. The *goal* of using protein pocket similarity is to extrapolate compound information from one target to another.

The query target's binding site is encoded and compared to a dataset of proteins with pre-calculated binding sites. The most similar proteins are proposed as potential off-targets; tool compounds that act upon these top-ranked proteins can be suggested as tool compounds for the query target (Figure 5.1). Hence, this approach performs two steps, i.e., (i) using the query binding site to screen all structurally known binding sites for similarities (protein-to-protein relationship), and (ii) extracting all tool compounds reported to bind to the top-ranked binding sites (protein-to-ligand relationship).

Such an approach can be used to assist in central life science questions such as:

1. Which proteins are most similar to my target of interest? Such investigations can give insights into undesired off-targets, open opportunities for polypharmacology, or provide ideas on the function of understudied targets.
2. Are there chemical probes/tool compounds available for my query target? Any drug

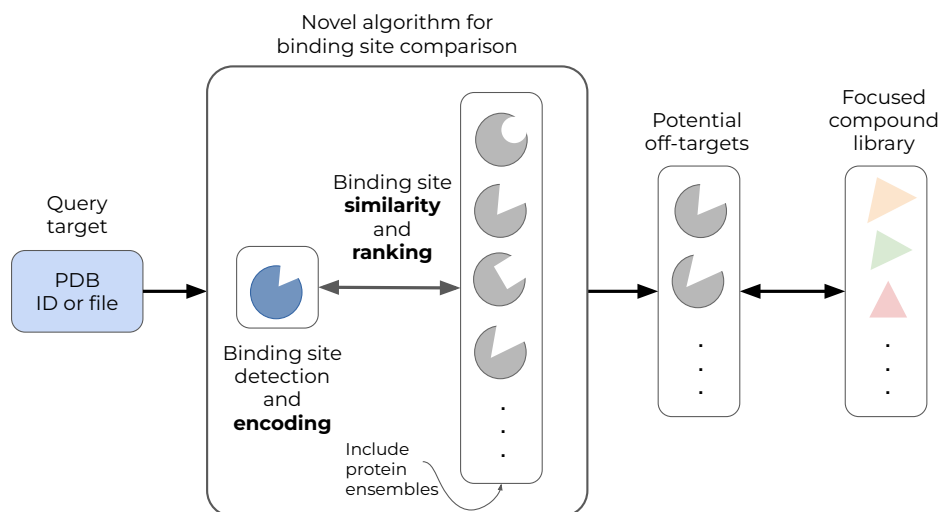


Figure 5.1: Schematic depiction of the "Read-Across the Targetome" (Ratar) objectives: Reading across the pocketome of structurally resolved proteins provides a list of potential off-targets, i.e., the targets that are most similar to a query target with respect to their binding site. Known binders to these top-ranked targets can build a focused compound screening library to study the query target.

discovery project needs suitable tool compounds to establish affinity assays and to provide a starting point for the drug candidate's desired chemistry.

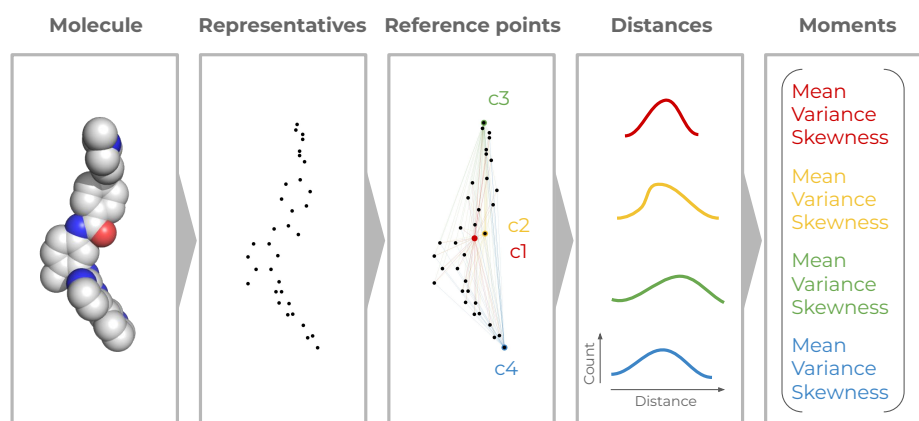
3. How can I set up a focused screening library for my target of interest that has no known ligands? High-throughput screens can be expensive regarding time and money; both can be saved if the screening library effectively covers the chemical space of related targets.

As part of this thesis, the methodology to read across the proteome — to help answer question 1— is outlined and discussed in the context of some preliminary results.

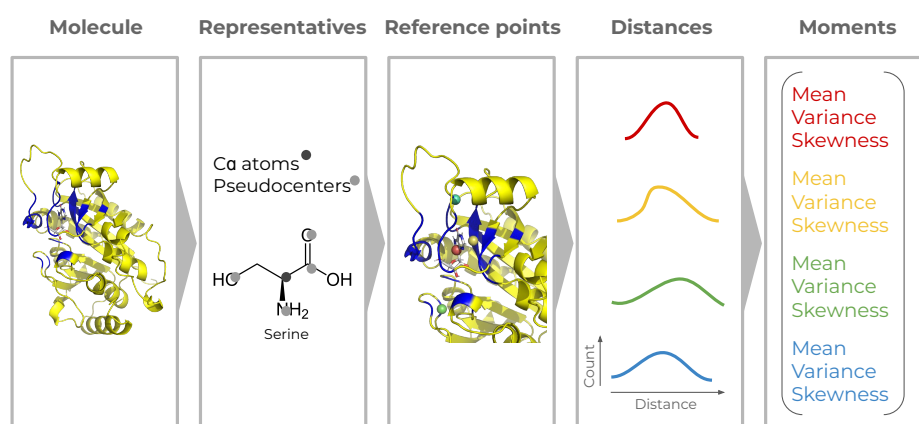
## Methods

To enable fast and efficient pocket comparison, a new vector-based approach for protein structure comparison was implemented, which is inspired by the Ultra-Fast Shape Recognition (USR) method [154] for fast and transformation-invariant small molecule shape comparison. The USR method is moment-based and uses relative atom positions. Four *reference points* are calculated with respect to all ligand atoms, which are called here *representatives*: the centroid (c1), the atoms closest to and farthest from the centroid (c2 and c3) as well as the atom farthest from the farthest atom (c4). Next, the first three moments —the mean distance, the standard deviation, and (the cube root of) the skewness— of the distances from all atoms towards the four reference points are calculated, resulting in a 12-bit fingerprint. The USR-based molecule encoding procedure is outlined in Figure 5.2a. Finally, the similarity between two molecules  $S_{qi}$  is described by an inverse Manhattan distance between the three moments of inertia of the four reference points  $M_l^q$  and  $M_l^i$ :

$$S_{qi} = \frac{1}{1 + \frac{1}{12} \sum_{l=1}^{12} |M_l^q - M_l^i|}$$



(a) Molecule encoding using the USR method.



(b) Binding site encoding using the USR-inspired Ratar method.

Figure 5.2: Translation of the (a) ligand-encoding method USR [154] to the (b) pocket-encoding method Ratar: The *molecule* (ligand or binding site) has *representatives* (ligand atoms or pocket  $C\alpha$  atoms or pseudocenters), which are the basis for defining *reference points*: centroid ( $c1$ ), closest and farthest atoms from  $c1$  ( $c2$  and  $c3$ ), and farthest atom from  $c3$  ( $c4$ ). Distances from each reference point to all representatives are calculated and each reference point's distance distribution is reduced to the first three moments, resulting in a 12-bit fingerprint. The molecule representations in (a) are adapted from Figure 1 in Ballester and Richards [154].

The USR method has two modifications: (i) The Chiral Shape Recognition (CSR) [212] method replaces the USR's reference point  $c2$  by the cross product of the two vectors  $c3 - c1$  and  $c4 - c1$  to distinguish enantiomers. (ii) The ElectroShape [213] method incorporates electrostatic properties of the molecule using charge information as a fourth dimension. Reference points are defined as in the CSR method carrying three spatial and one charge dimension. The CSR reference point resulting from the cross product does not represent an atom and therefore carries no inherent charge; ElectroShape defines this reference point twice with the same spatial coordinates and its fourth dimension a positive and negative charge each. This procedure results in five four-dimensional (4D) reference points in total.

The concepts of the USR, CSR, and ElectroShape methods—collectively termed *USR methods* from hereon—are translated from molecules to binding sites and extended with the Ratar method as depicted in Figure 5.2b. In the following, the *Ratar method* refers to the novel bind-

ing site comparison tool that is inspired by the USR methods, and the *Ratar framework* refers to the collection of (re-)implementations of the USR methods and the novel Ratar method.

**Molecule.** While the USR method and its derivatives use all ligand atoms, the Ratar framework uses all binding site atoms. In this study, binding sites (pockets) are defined as in the scPDB, i.e., by all residues with at least one atom within 6.5 Å of any atom of the co-crystallized ligand [214]. Ratar’s evaluation is based on a benchmark set of similar and dissimilar binding sites, containing 769 pairs of nonredundant similar binding sites and 769 pairs of nonredundant dissimilar binding sites as defined by Weill and Rognan [160] to evaluate the binding site comparison tool FuzCav.

**Representatives.** The Ratar framework offers different options to define the binding site representatives: (i) the pocket’s  $C\alpha$  atoms (*ca*), (ii) the residues so-called pseudocenter atoms (*pca*), which carry physicochemical importance for binding, or (iii) aggregated pseudocenters, i.e., aggregate multiple atoms belonging to one pseudocenter (*pc*), e.g., aromatic ring center for *pc* instead of six aromatic ring atoms for *pca*. The concept of pseudocenters was introduced by Schmitt et al. [215] to condense the physicochemical properties of residues to five essential features, i.e., hydrogen bond donor and acceptor, mixed donors/acceptors, as well as hydrophobic aliphatic and aromatic contacts. The assigned features per amino acid including all feature-related atoms are summarized in Table 1 in [215].

The *representatives’ dimensions* range from 3D (3 spatial coordinates) as used in the USR and CSR methods, 4D (3 spatial coordinates and 1 charge coordinate) as used in the ElectroShape method, and 6D (3 spatial coordinates and 3 physicochemical coordinates) as defined for the Ratar method. The physicochemical properties are represented in the Ratar method by Z-scales [159, 216], which are the first principal components of a multivariate characterization of the amino acids and showed good performance in a descriptor benchmark study by van Westen et al. [217].

**Reference points.** To find the exact position of a point in  $R^n$ , distances to  $n + 1$  fixed reference points are needed [213]. The Ratar framework offers different options to define reference points: Reimplementations of the reference points used in the (i) USR, (ii) CSR, and (iii) ElectroShape methods as discussed earlier, which serve as baseline methods. In addition, (iv) the Ratar method with its six-dimensional atoms uses seven reference points, i.e., the representatives’ centroid (*c1*), closest point to *c1* (*c2*), furthest point to *c1* (*c3*), furthest point to *c3* (*c4*) and the normalized cross products between vectors between atoms *c1* – *c4* as described in Figure 5.3.

**Distances.** For each reference point, distances are calculated for all representatives as described for the USR methods. This results in one distance distribution per reference point with as many values as representatives. For example, a binding site with 30  $C\alpha$  atoms (representatives) and seven reference points is described with seven distance distributions; each distance distribution consists of 30 distances between one reference point to all representatives.

**Moments.** Each distance distribution is condensed to the first three moments, i.e., the mean distance, the standard deviation, and (the cube root of) the skewness. The moments are concatenated to a single fingerprint: The USR, CSR, ElectroShape, and Ratar methods result in a binding site fingerprint of 12, 12, 15, and 21 bits, see examples in Figure 5.3.

All binding sites of the FuzCav data set were encoded with these different encoding schemes: the USR (3D), CSR (3D), ElectroShape (4D), and Ratar (6D) methods using different representatives as a starting point, i.e.,  $C\alpha$  atoms, pseudocenter atoms, and pseudocenters, resulting in 12 different fingerprint setups. Fingerprints are compared pairwise using the inverse Manhattan distance as reported for the original USR method.

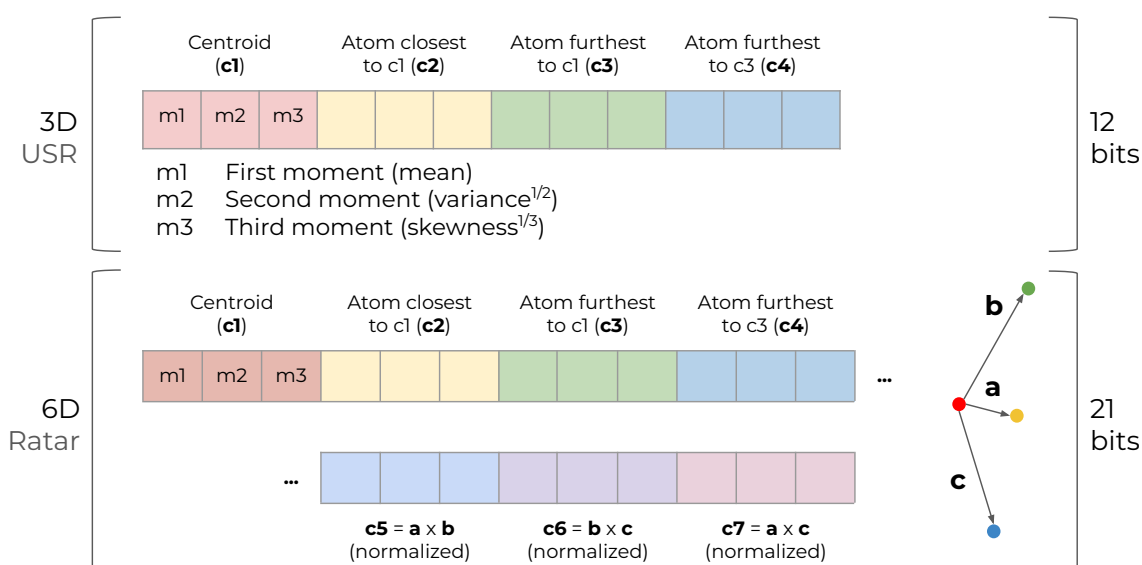


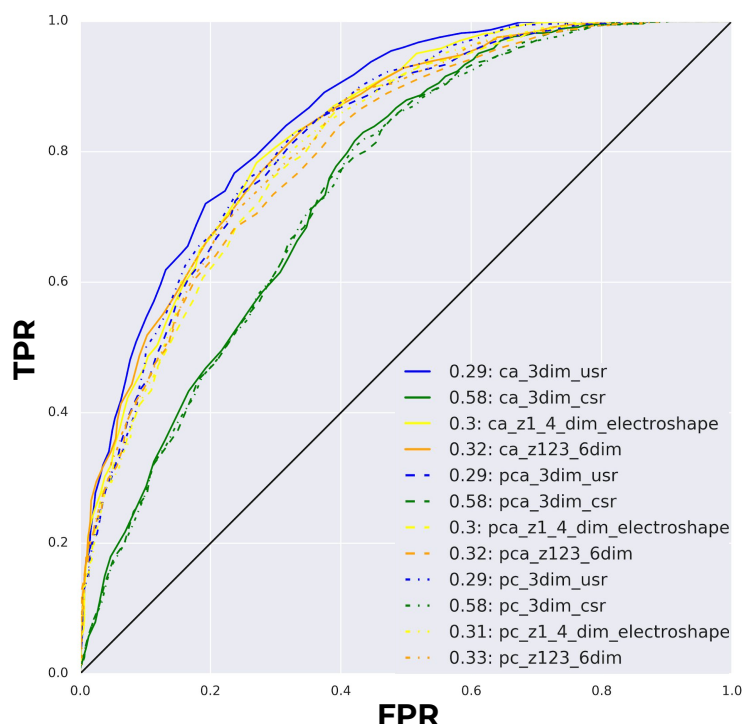
Figure 5.3: Fingerprint composition used for the USR method [154] (originally developed for ligand encoding and here applied to binding sites) and the Ratar method (here proposed for binding site comparison).

## Results

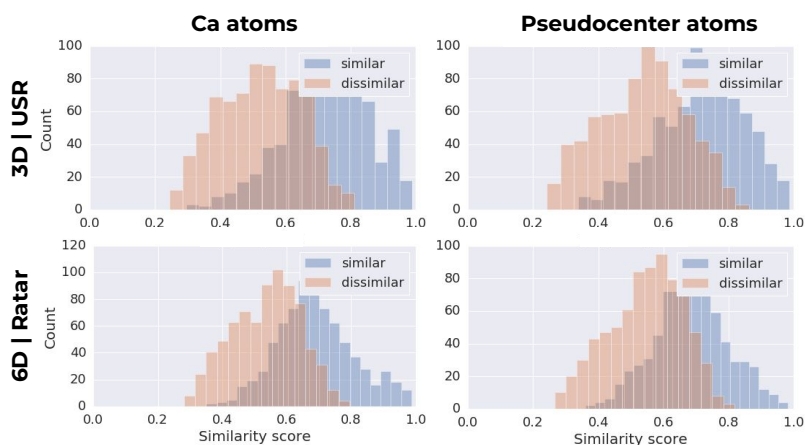
The pairwise comparisons between pairs of similar and dissimilar binding sites as published by Weill and Rognan [160] (called here FuzCav dataset) show the following performance: As shown in Figure 5.4a, the Ratar method performs with an area under the curve (AUC) of about 0.61. It performs as well as the pocket-adapted methods USR, CSR, and ElectroShape based on receiver operating characteristic (ROC) curves showing the rank performance of similar pairs. At this point, the binding site representation method — $C\alpha$  atoms, pseudocenters, or pseudocenter atoms— had no significant influence on performance. Figure 5.4b shows selected setups as distance histograms, i.e., the pocket-adapted USR and Ratar method for  $C\alpha$  atoms and pseudocenters: (i) The dissimilar pairs show a distribution shifted to lower similarity scores compared to the similar pairs, which is the desired behavior. (ii) The binding site representations show no significant impact on the distance distributions. (iii) The similar pair distribution is broader using the USR than the Ratar method (less pronounced also observed for the dissimilar pair distribution), which shows that some similar pairs are more difficult to detect than dissimilar pairs. (iv) The Ratar distributions are narrower than their USR counterparts (with a stronger effect for similar than dissimilar pairs).

## Discussion and Conclusion

The current status of the Ratar framework contains the following baseline methods: re-implementations of the USR, CSR, and ElectroShape methods, which were adapted to encode pockets instead of ligands, and the novel Ratar method, which extends the pocket-adapted USR methods with Z-scales. We used the FuzCav dataset to assess the performance of these different setups; even though we can see that dissimilar pairs overall are assigned to lower similarity scores than similar pairs, the dataset cannot be separated properly, yet, as reflected in an AUC of about 0.61.



(a) ROC curves showing the performance to distinguish similar from dissimilar structure pairs based on different comparison methods: Binding sites represented as C $\alpha$  atoms (ca), pseudocenters (pc) Schmitt et al. [215], or pseudocenter atoms (pca) and encoded by applying the ligand-based USR, CSR, ElectroShape methods to pockets as well as the USR-extension Ratar.



(b) Histograms for selected comparison setups showing similarity scores between pairs of similar (blue) and dissimilar (orange) pairs: Binding sites represented as C $\alpha$  atoms and pseudocenters and encoded with the USR and Ratar methods.

Figure 5.4: (a) Performance of pocket-adapted USR [154], CSR [212], ElectroShape [213] methods as well as (b) the novel Ratar [161] method, evaluated using the similar and dissimilar structure pairs as published by Weill and Rognan [160].

Encoding full binding sites with only one fingerprint might be too coarse-grained. The next step in this project should be to perform the encoding on overlapping binding site patches. This would potentially help to (i) represent the binding sites more accurately, and (ii) allow rationalizing which regions within a binding site pair showed the highest similarities (if any). Furthermore, using the FuzCav dataset should only be the beginning. To ensure proper benchmarking against existing binding site comparison tools, Ratar should be tested on the ProSPECCTS [218] benchmark study, which provides comprehensive benchmark datasets and reports performances of published tools.

### 5.3 Project Illustrations

In the following, we append Ferdinand Krupp's illustrations that were not included in the main part of this thesis (Figures 5.5–5.10).

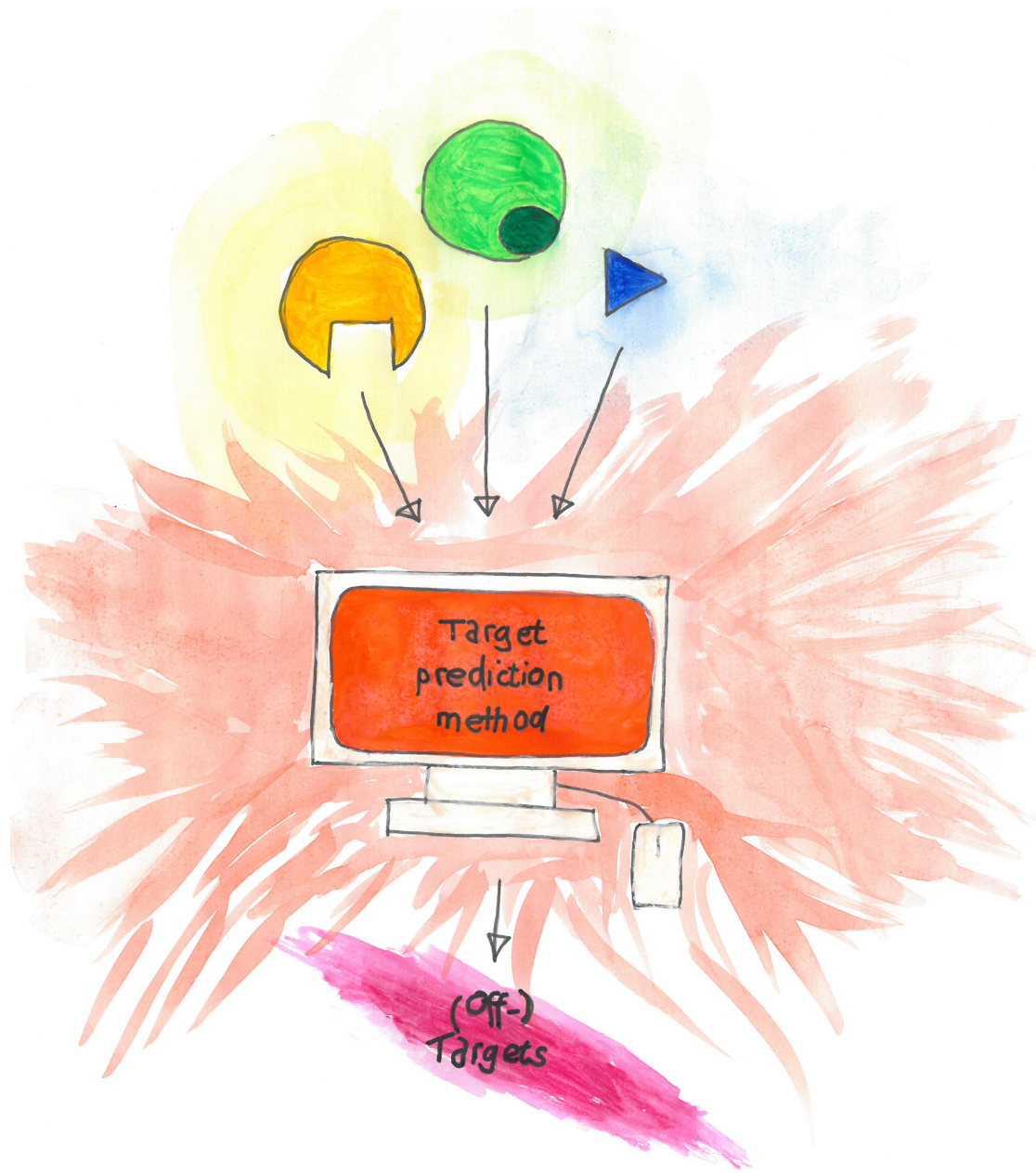


Figure 5.5: Computational target prediction as illustrated by Ferdinand Krupp, adapted from TOC figure in Sydow et al. [22].



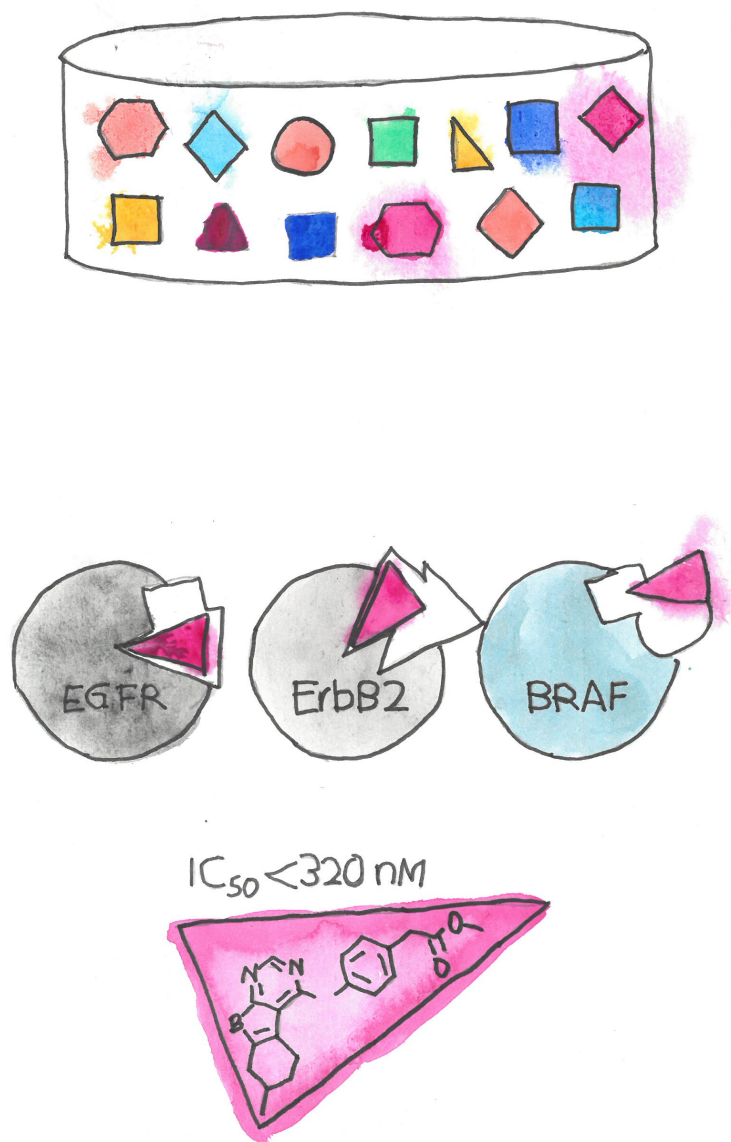


Figure 5.6: Multi-target screening as illustrated by Ferdinand Krupp, adapted from TOC figure in Schmidt et al. [142].



Figure 5.7: Kinase similarity pipeline as illustrated by Ferdinand Krupp, adapted from TOC figure in Kimber et al. [95].



Figure 5.8: TeachOpenCADD as illustrated by Ferdinand Krupp, adapted from TOC figure in Sydow et al. [144].

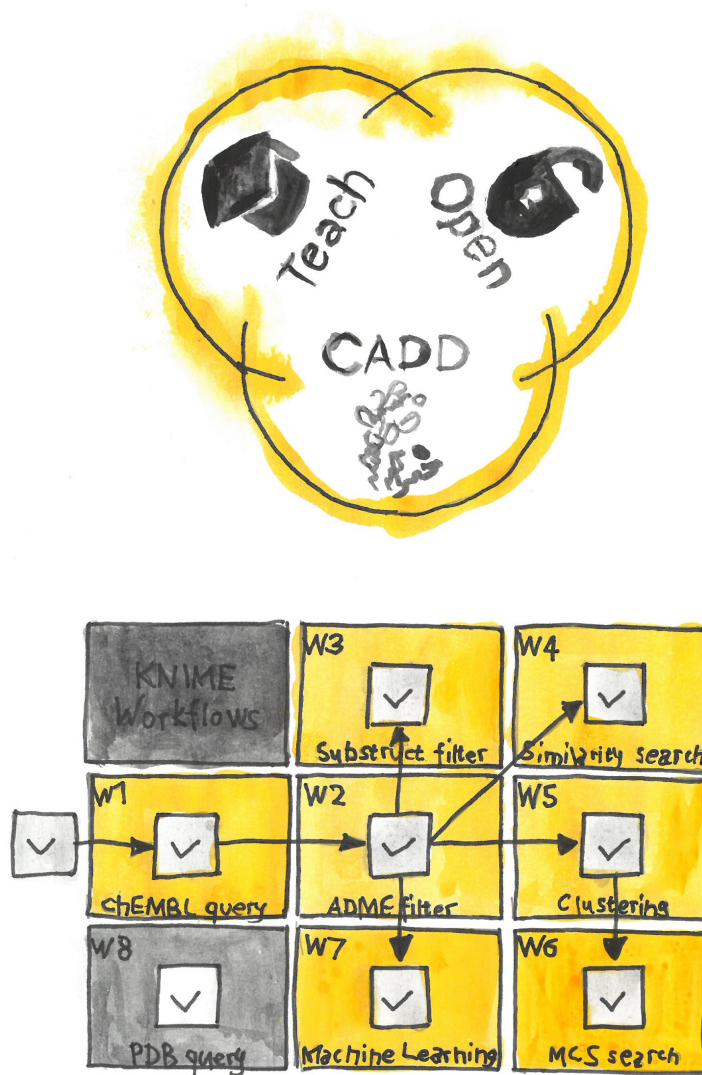


Figure 5.9: TeachOpenCADD-KNIME as illustrated by Ferdinand Krupp, adapted from TOC figure in Sydow et al. [185].

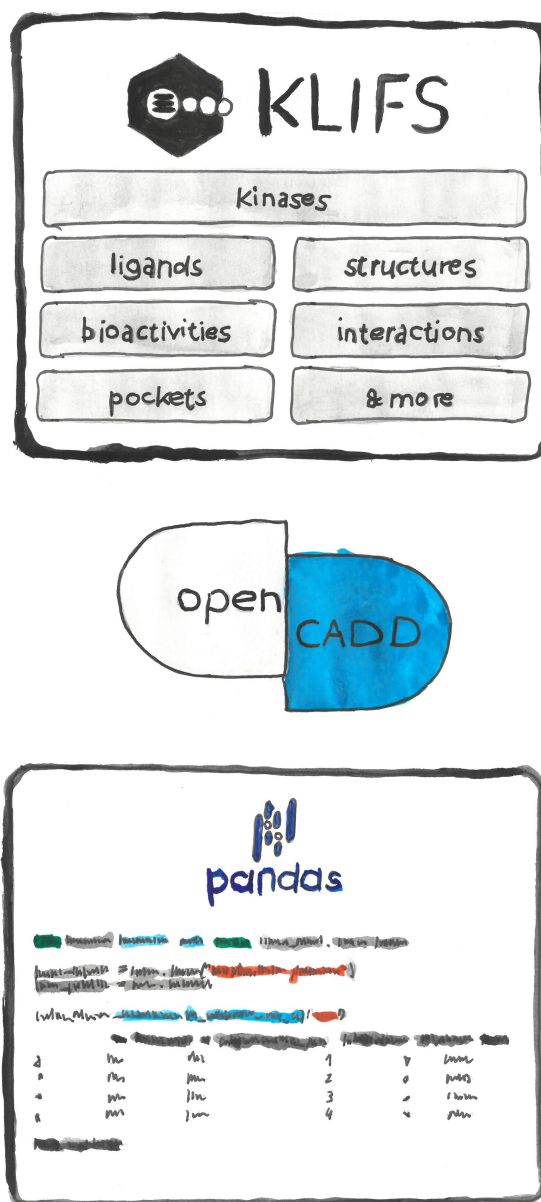


Figure 5.10: OpenCADD-CLIFS as illustrated by Ferdinand Krupp, adapted from Figure 1 in Sydow et al. [92].



# List of Figures

1.1	Binding site detected with ProteinsPlus for CDK2 bound to ATP. . . . .	6
1.2	Sequence-based kinome tree annotated with structural and profiling data . . . . .	29
1.3	Kinase binding site residues and regions as defined by KLIFS in 1D . . . . .	30
1.4	Kinase binding site regions, motifs, and conformations in 3D . . . . .	31
1.5	Statistics about kinase inhibitors in clinical trials . . . . .	32
1.6	Binding modes of ATP, gefitinib, and imatinib in 2D . . . . .	33
1.7	KLIFS interaction fingerprint of kinase binding sites . . . . .	36
3.1	Schematic overview of publications that are included in this doctoral thesis . . . .	41
3.2	Predicting kinome-wide (sub)pocket-based off-targets as illustrated by F. Krupp .	43
3.3	Exploring kinome-wide subpocket fragment spaces as illustrated by F. Krupp . .	133
3.4	FAIR pipelines and tools in kinase-centric drug design as illustrated by F. Krupp	169
4.1	TeachOpenCADD topics as of September 2022 . . . . .	205
5.1	Schematic depiction of the Ratar method objective . . . . .	248
5.2	Translation of the ligand-encoding USR method to binding sites . . . . .	249
5.3	USR- and Ratar-based fingerprints . . . . .	251
5.4	Performance of pocket-adapted USR methods and novel Ratar method . . . . .	252
5.5	Computational target prediction as illustrated by F. Krupp . . . . .	254
5.6	Multi-target screening as illustrated by F. Krupp . . . . .	255
5.7	Kinase similarity pipeline as illustrated by F. Krupp . . . . .	256
5.8	TeachOpenCADD as illustrated by F. Krupp . . . . .	257
5.9	TeachOpenCADD-KNIME as illustrated by F. Krupp . . . . .	258
5.10	OpenCADD-KLIFS as illustrated by F. Krupp . . . . .	259





# List of Tables

1.1	Overview of eukaryotic kinase groups . . . . .	29
-----	--	----



# Acronyms

<b>ABPP</b>	Activity-based proteome profiling
<b>AUC</b>	Area under the curve
<b>AP</b>	Adenine pocket (KinFragLib)
<b>API</b>	Application programming interface
<b>ATP</b>	Adenosine triphosphate
<b>B1</b>	Back pocket 1 (KinFragLib)
<b>B2</b>	Back pocket 2 (KinFragLib)
<b>CADD</b>	Computer-aided drug design
<b>CDK</b>	Chemistry Development Kit
<b>CIME</b>	ChemInformatics Model Explorer
<b>CLI</b>	Command-line interface
<b>DFG</b>	Aspartate-phenylalanine-glycine motif in kinases
<b>DMTA</b>	Design-Make-Test-Analyze
<b>DoG</b>	Difference of Gaussian
<b>EGFR</b>	Epidermal growth factor receptor
<b>ECFP</b>	Extended-connectivity fingerprints
<b>FAIR</b>	Findable, accessible, interoperable, and reusable
<b>FAIR4RS</b>	FAIR for Research Software
<b>FBDD</b>	Fragment-based drug design
<b>FDA</b>	Food and Drug Administration
<b>FP</b>	Front pocket (KinFragLib)
<b>GA</b>	Gate area (KinFragLib)
<b>GPCR</b>	G-protein coupled receptor
<b>GtoPdb</b>	Guide to Pharmacology database
<b>HGNC</b>	HUGO Gene Nomenclature Committee
<b>HTML</b>	HyperText Markup Language
<b>HTS</b>	High-throughput screening
<b>IFP</b>	Interaction fingerprint
<b>IP</b>	Intellectual property
<b>IRE</b>	Gefitinib (ligand expo ID)
<b>KCGS</b>	Kinase Chemogenomics set
<b>KLIFS</b>	Kinase-Ligand Interaction Fingerprint and Structures
<b>NME</b>	New molecular entity
<b>ODOSOS</b>	Open Data, Open Standards, and Open Source
<b>PDB</b>	Protein Data Bank
<b>PEP</b>	Python Enhancement Proposals
<b>PCM</b>	Proteochemometrics
<b>QED</b>	Quantitative estimate of druglikeness

<b>RA</b> score	Retrosynthetic accessibility score
<b>RDF</b>	Resource Description Framework
<b>Ro3</b>	Rule of Three
<b>Ro5</b>	Rule of Five
<b>SA</b> score	Synthetic accessibility score
<b>SAR</b>	Structure-activity relationship
<b>SC</b> score	Synthetic complexity score
<b>SE</b>	Solvent-exposed pocket (KinFragLib)
<b>STI</b>	Imatinib (ligand expo ID)
<b>STPK</b>	Serine/threonine-specific kinase
<b>SYBA</b>	Synthetic baysian accessibility
<b>TK</b>	Tyrosine kinase
<b>TPK</b>	Tyrosine-specific kinase
<b>UMAP</b>	Uniform Manifold Approximation and Projection

# Bibliography

- [1] Mark A. Lindsay. Target discovery. *Nat. Rev. Drug Discovery*, 2(10):831–838, 2003. doi:10.1038/nrd1202.
- [2] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. *Molecular Biology of the Cell*. Garland Science, 5th edition, 2008.
- [3] Francesca Spyrakis, Axel BidonChanal, Xavier Barril, and F. Javier Luque. Protein flexibility and ligand recognition: Challenges for molecular modeling. *Curr. Top. Med. Chem.*, 11(2):192–210, 2011. doi:10.2174/156802611794863571.
- [4] Xing Du, Yi Li, Yuan-Ling Xia, Shi-Meng Ai, Jing Liang, Peng Sang, Xing-Lai Ji, and Shu-Qun Liu. Insights into protein-ligand interactions: Mechanisms, models, and methods. *Int. J. Mol. Sci.*, 17(2):144, 2016. doi:10.3390/ijms17020144.
- [5] Daniel E. Koshland. Application of a theory of enzyme specificity to protein synthesis. *Proc. Natl. Acad. Sci.*, 44(2):98–104, 1958. doi:10.1073/pnas.44.2.98.
- [6] Jacques Monod, Jeffries Wyman, and Jean-Pierre Changeux. On the nature of allosteric transitions: A plausible model. *J. Mol. Biol.*, 12(1):88–118, 1965. doi:10.1016/S0022-2836(65)80285-6.
- [7] Jérémy Desaphy, Guillaume Bret, Didier Rognan, and Esther Kellenberger. sc-PDB: A 3D-database of ligandable binding sites — 10 years on. *Nucleic Acids Res.*, 43(D1):D399–D404, 2015. doi:10.1093/nar/gku928.
- [8] Andrea Volkamer, Mathias M. von Behren, Stefan Bietz, and Matthias Rarey. Prediction, analysis, and comparison of active sites. In *Applied Chemoinformatics*, pages 283–311. Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, Germany, 2018. doi:10.1002/9783527806539.ch6g.
- [9] Andrea Volkamer, Daniel Kuhn, Friedrich Rippmann, and Matthias Rarey. DoGSiteScorer: A web server for automatic binding site prediction, analysis and druggability assessment. *Bioinformatics*, 28(15):2074–2075, 2012. doi:10.1093/bioinformatics/bts310.
- [10] Katrin Stierand, Patrick C. Maaß, and Matthias Rarey. Molecular complexes at a glance: Automated generation of two-dimensional complex diagrams. *Bioinformatics*, 22(14):1710–1716, 2006. doi:10.1093/bioinformatics/btl150.
- [11] Patrick C. Fricker, Marcus Gastreich, and Matthias Rarey. Automated drawing of structural molecular formulas under constraints. *J. Chem. Inf. Model.*, 44(3):1065–1078, 2004. doi:10.1021/ci049958u.

- [12] Raghu Bhagavat, Santhosh Sankar, Narayanaswamy Srinivasan, and Nagasuma Chandra. An augmented pocketome: Detection and analysis of small-molecule binding pockets in proteins of known 3D structure. *Structure*, 26(3):499–512.e2, 2018. doi:10.1016/j.str.2018.02.001.
- [13] Jianghong An, Maxim Totrov, and Ruben Abagyan. Pocketome via comprehensive identification and classification of ligand binding envelopes. *Mol. Cell. Proteom.*, 4(6):752–61, 2005. doi:10.1074/mcp.M400159-MCP200.
- [14] Irina Kufareva, Andrey V. Ilatovskiy, and Ruben Abagyan. Pocketome: An encyclopedia of small-molecule binding sites in 4D. *Nucleic Acids Res.*, 40(D1):D535–D540, 2012. doi:10.1093/nar/gkr825.
- [15] Andrea Volkamer, Sameh Eid, Samo Turk, Sabrina Jaeger, Friedrich Rippmann, and Simone Fulle. Pocketome of human kinases: Prioritizing the ATP binding sites of (yet) untapped protein kinases for drug discovery. *J. Chem. Inf. Model.*, 55(3):538–549, 2015. doi:10.1021/ci500624s.
- [16] Janik B. Hedderich, Margherita Persechino, Katharina Becker, Franziska M. Heydenreich, Torben Gutermuth, Michel Bouvier, Moritz Bünemann, and Peter Kolb. The pocketome of G-protein-coupled receptors reveals previously untargeted allosteric sites. *Nat. Commun.*, 13(1):2567, 2022. doi:10.1038/s41467-022-29609-6.
- [17] Tommaso Palomba, Massimo Baroni, Simon Cross, Gabriele Cruciani, and Lydia Siragusa. ELIOT: A platform to navigate the E3 pocketome and aid the design of new PROTACs. *Chem. Biol. Drug. Des.*, 2022. doi:10.1111/cbdd.14123.
- [18] Georgi K. Kanev, Albert J. Kooistra, Iwan J. P. de Esch, and Chris de Graaf. Structural chemogenomics databases to navigate protein-ligand interaction space. In *Comprehensive Medicinal Chemistry III*, pages 444–471. Elsevier, Oxford, 2017. doi:10.1016/B978-0-12-409547-2.12298-X.
- [19] Babs Briels, Chris de Graaf, and Andreas Bender. Structural chemogenomics. In *Structural Biology in Drug Discovery*, pages 53–77. John Wiley & Sons, Ltd, 2020. doi:10.1002/9781118681121.ch3.
- [20] Albert J. Kooistra, Márton Vass, Ross McGuire, Rob Leurs, Iwan J. P. de Esch, Gert Vriend, Stefan Verhoeven, and Chris de Graaf. 3D-e-Chem: Structural cheminformatics workflows for computer-aided drug discovery. *ChemMedChem*, 13(6):614–626, 2018. doi:10.1002/cmdc.201700754.
- [21] KLIFS team. KLIFS structure entry 4367 for 1FIN, 2022. URL [https://klifs.net/details.php?structure\\_id=4367](https://klifs.net/details.php?structure_id=4367). [accessed 2022-08-10].
- [22] Dominique Sydow, Lindsey Burggraaff, Angelika Szengel, Herman W. T. van Vlijmen, Adriaan P. IJzerman, Gerard J. P. van Westen, and Andrea Volkamer. Advances and challenges in computational target prediction. *J. Chem. Inf. Model.*, 59(5):1728–1742, 2019. doi:10.1021/acs.jcim.8b00832.
- [23] Albert J. Kooistra, Georgi K. Kanev, Oscar P. J. van Linden, Rob Leurs, Iwan J. P. de Esch, and Chris de Graaf. KLIFS: A structural kinase-ligand interaction database. *Nucleic Acids Res.*, 44(D1):D365–D371, 2015. doi:10.1093/nar/gkv1082.

- [24] Márton Vass, Albert J. Kooistra, Dehua Yang, Raymond C. Stevens, Ming-Wei Wang, and Chris de Graaf. Chemical diversity in the G protein-coupled receptor superfamily. *Trends Pharmacol. Sci.*, 39(5):494–512, 2018. doi:10.1016/j.tips.2018.02.004.
- [25] Márton Vass, Sabina Podlewska, Iwan J.P. De Esch, Andrzej J. Bojarski, Rob Leurs, Albert J. Kooistra, and Chris De Graaf. Aminergic GPCR-ligand interactions: A chemical and structural map of receptor mutation data. *J. Med. Chem.*, 62(8):3784–3839, 2019. doi:10.1021/acs.jmedchem.8b00836.
- [26] Paul N. Mortenson, Valerio Berdini, and Marc O'Reilly. Fragment-based approaches to the discovery of kinase inhibitors. *Meth. Enzymol.*, 548:69–92, 2014. doi:10.1016/B978-0-12-397918-6.00003-3.
- [27] Slava Ziegler, Verena Pries, Christian Hedberg, and Herbert Waldmann. Target identification for small bioactive molecules: Finding the needle in the haystack. *Angew. Chem., Int. Ed.*, 52(10):2744–2792, 2013. doi:10.1002/anie.201208749.
- [28] Mohamed Diwan M. AbdulHameed, Sidhartha Chaudhury, Narender Singh, Hongmao Sun, Anders Wallqvist, and Gregory J. Tawa. Exploring polypharmacology using a ROCS-based target fishing approach. *J. Chem. Inf. Model.*, 52(2):492–505, 2012. doi:10.1021/ci2003544.
- [29] Andrew Anighoro, Jürgen Bajorath, and Giulio Rastelli. Polypharmacology: Challenges and opportunities in drug discovery. *J. Med. Chem.*, 57(19):7874–7887, 2014. doi:10.1021/jm5006463.
- [30] Richard Morphy, Corinne Kay, and Zoran Rankovic. From magic bullets to designed multiple ligands. *Drug Discovery Today*, 9(15):641–651, 2004. doi:10.1016/S1359-6446(04)03163-0.
- [31] Andreas Bender, Josef Scheiber, Meir Glick, John W. Davies, Kamal Azzaoui, Jacques Hamon, Laszlo Urban, Steven Whitebread, and Jeremy L. Jenkins. Analysis of pharmacology data and the prediction of adverse drug reactions and off-target effects from chemical structure. *ChemMedChem*, 2(6):861–873, 2007. doi:10.1002/cmdc.200700026.
- [32] Tudor I. Oprea, Julie E. Bauman, Cristian G. Bologa, Tione Buranda, Alexandre Chigaev, Bruce S. Edwards, Jonathan W. Jarvik, Hattie D. Gresham, Mark K. Haynes, Brian Hjelle, Robert Hromas, Laurie Hudson, Debra A. Mackenzie, Carolyn Y. Muller, John C. Reed, Peter C. Simons, Yelena Smagley, Juan Strouse, Zurab Surviladze, Todd Thompson, Oleg Ursu, Anna Waller, Angela Wandinger-Ness, Stuart S. Winter, Yang Wu, Susan M. Young, Richard S. Larson, Cheryl Willman, and Larry A. Sklar. Drug repurposing from an academic perspective. *Drug Discovery Today: Ther. Strategies*, 8(3-4):61–69, 2011. doi:10.1016/j.ddstr.2011.10.002.
- [33] Michael J. Keiser, Vincent Setola, John J. Irwin, Christian Laggner, Atheir I. Abbas, Sandra J. Hufeisen, Niels H. Jensen, Michael B. Kuijer, Roberto C. Matos, Thuy B. Tran, Ryan Whaley, Richard A. Glennon, Jérôme Hert, Kelan L. H. Thomas, Douglas D. Edwards, Brian K. Shoichet, and Bryan L. Roth. Predicting new molecular targets for known drugs. *Nature*, 462(7270):175–181, 2009. doi:10.1038/nature08506.

- [34] Ted T. Ashburn and Karl B. Thor. Drug repositioning: Identifying and developing new uses for existing drugs. *Nat. Rev. Drug Discovery*, 3(8):673–683, 2004. doi:10.1038/nrd1468.
- [35] Alicia B. Berger, Phillip M. Vitorino, and Matthew Bogyo. Activity-based protein profiling: Applications to biomarker discovery, in vivo imaging and drug discovery. *Am. J. Pharmacogenomics*, 4(6):371–381, 2004. doi:10.2165/00129785-200404060-00004.
- [36] Markus Schirle, Marcus Bantscheff, and Bernhard Kuster. Mass spectrometry-based proteomics in preclinical drug discovery. *Cell Chem. Biol.*, 19(1):72–84, 2012. doi:10.1016/J.CHEMBIOL.2012.01.002.
- [37] Annelot C. M. van Esbroeck, Antonius P. A. Janssen, Armand B. Cognetta, Daisuke Ogasawara, Guy Shpak, Mark van der Kroeg, Vasudev Kantae, Marc P. Baggelaar, Femke M. S. de Vrij, Hui Deng, Marco Allarà, Filomena Fezza, Zhanmin Lin, Tom van der Wel, Marjolein Soethoudt, Elliot D. Mock, Hans den Dulk, Ilse L. Baak, Bogdan I. Florea, Giel Hendriks, Luciano De Petrocellis, Herman S. Overkleeft, Thomas Hanke-meier, Chris I. De Zeeuw, Vincenzo Di Marzo, Mauro Maccarrone, Benjamin F. Cravatt, Steven A. Kushner, and Mario van der Stelt. Activity-based protein profiling reveals off-target proteins of the FAAH inhibitor BIA 10-2474. *Science*, 356(6342):1084–1087, 2017. doi:10.1126/science.aaf7497.
- [38] Gregory Sliwoski, Sandeepkumar Kothiwale, Jens Meiler, and Edward W. Lowe. Computational methods in drug discovery. *Pharmacol. Rev.*, 66(1):334–395, 2014. doi:10.1124/pr.112.007336.
- [39] Andreas Bender and Robert C. Glen. Molecular similarity: A key technique in molecular informatics. *Org. Biomol. Chem.*, 2(22):3204, 2004. doi:10.1039/b409813g.
- [40] David Gfeller, Aurélien Grosdidier, Matthias Wirth, Antoine Daina, Olivier Michielin, and Vincent Zoete. SwissTargetPrediction: A web server for target prediction of bioactive small molecules. *Nucleic Acids Res.*, 42(W1):W32–W38, 2014. doi:10.1093/nar/gku293.
- [41] Michael J. Keiser, Bryan L. Roth, Blaine N. Armbruster, Paul Ernsberger, John J. Irwin, and Brian K. Shoichet. Relating protein pharmacology by ligand chemistry. *Nat. Biotechnol.*, 25:197, 2007. doi:10.1038/nbt1284.
- [42] Yoshihiro Yamanishi, Masaaki Kotera, Yuki Moriya, Ryusuke Sawada, Minoru Kanehisa, and Susumu Goto. DINIES: Drug-target interaction network inference engine based on supervised analysis. *Nucleic Acids Res.*, 42(W1):W39–W45, 2014. doi:10.1093/nar/gku337.
- [43] Janez Konc and Dušanka Janežič. ProBiS algorithm for detection of structurally similar protein binding sites by local structural alignment. *Bioinformatics*, 26(9):1160–1168, 2010. doi:10.1093/bioinformatics/btq100.
- [44] Jérémy Desaphy, Eric Raimbaud, Pierre Ducrot, and Didier Rognan. Encoding protein-ligand interaction patterns in fingerprints and graphs. *J. Chem. Inf. Model.*, 53(3):623–637, 2013. doi:10.1021/ci300566n.
- [45] Xia Wang, Yihang Shen, Shiwei Wang, Shiliang Li, Weilin Zhang, Xiaofeng Liu, Luhua Lai, Jianfeng Pei, and Honglin Li. PharmMapper 2017 update: A web server for potential drug target identification with a comprehensive target pharmacophore database. *Nucleic Acids Res.*, 45(W1):W356–W360, 2017. doi:10.1093/nar/gkx374.



- [46] Karen T. Schomburg, Stefan Bietz, Hans Briem, Angela M. Henzler, Sascha Urbaczek, and Matthias Rarey. Facing the challenges of structure-based target prediction by inverse virtual screening. *J. Chem. Inf. Model.*, 54(6):1676–1686, 2014. doi:10.1021/ci500130e.
- [47] Shiliang Li, Chaoqian Cai, Jiayu Gong, Xiaofeng Liu, and Honglin Li. A fast protein binding site comparison algorithm for proteome-wide protein function prediction and drug repurposing. *Proteins: Struct., Funct., Bioinf.*, 89(11):1541–1556, 2021. doi:10.1002/prot.26176.
- [48] Merveille Eguida and Didier Rognan. A computer vision approach to align and compare protein cavities: Application to fragment-based drug design. *J. Med. Chem.*, 63(13):7127–7142, 2020. doi:10.1021/acs.jmedchem.0c00422.
- [49] Martin Simonovsky and Joshua Meyers. DeeplyTough: Learning structural comparison of protein binding sites. *J. Chem. Inf. Model.*, 60(4):2356–2366, 2020. doi:10.1021/acs.jcim.9b00554.
- [50] Limeng Pu, Rajiv Gandhi Govindaraj, Jeffrey Mitchell Lemoine, Hsiao-Chun Wu, and Michal Brylinski. DeepDrug3D: Classification of ligand-binding pockets in proteins with a convolutional neural network. *PLoS Comput. Biol.*, 15(2):e1006718, 2019. doi:10.1371/journal.pcbi.1006718.
- [51] DrugBank. Vemurafenib, 2022. URL <https://go.drugbank.com/drugs/DB08881>. [accessed 2022-08-22].
- [52] Paul D. Leeson and Brian Springthorpe. The influence of drug-like concepts on decision-making in medicinal chemistry. *Nat. Rev. Drug Discovery*, 6(11):881–890, 2007. doi:10.1038/nrd2445.
- [53] Anna Carbery, Rachael Skyner, Frank von Delft, and Charlotte M. Deane. Fragment libraries designed to be functionally diverse recover protein binding information more efficiently than standard structurally diverse libraries. *J. Med. Chem.*, 65(12):11404–11413, 2022. doi:10.1021/acs.jmedchem.2c01004.
- [54] Miles Congreve, Robin Carr, Chris Murray, and Harren Jhoti. A ‘Rule of Three’ for fragment-based lead discovery? *Drug Discovery Today*, 8(19):876–877, 2003. doi:10.1016/S1359-6446(03)02831-9.
- [55] Lauro Ribeiro de Souza Neto, José Teófilo Moreira-Filho, Bruno Junior Neves, Rocío Lucía Beatriz Riveros Maidana, Ana Carolina Ramos Guimarães, Nicholas Furnham, Carolina Horta Andrade, and Floriano Paes Silva Jr. In silico strategies to support fragment-to-lead optimization in drug discovery. *Front. Chem.*, 8:93, 2020. doi:10.3389/fchem.2020.00093.
- [56] Xiao Qing Lewell, Duncan B. Judd, Stephen P. Watson, and Michael M. Hann. RECAP - Retrosynthetic Combinatorial Analysis Procedure: A powerful new technique for identifying privileged molecular fragments with useful applications in combinatorial chemistry. *J. Chem. Inf. Comput. Sci.*, 38(3):511–522, 1998. doi:10.1021/ci970429i.
- [57] Jörg Degen, Christof Wegscheid-Gerlach, Andrea Zaliani, and Matthias Rarey. On the art of compiling and using ‘drug-like’ chemical fragment spaces. *ChemMedChem*, 3(10):1503–1507, 2008. doi:10.1002/cmdc.200800178.

- [58] Tairan Liu, Misagh Naderi, Chris Alvin, Supratik Mukhopadhyay, and Michal Brylinski. Break down in order to build up: Decomposing small molecules for fragment-based drug design with eMolFrag. *J. Chem. Inf. Model.*, 57(4):627–631, 2017. doi:10.1021/acs.jcim.6b00596.
- [59] Albert C. Pierce, Govinda Rao, and Guy W. Bemis. BREED: Generating novel inhibitors through hybridization of known ligands. Application to CDK2, P38, and HIV protease. *J. Med. Chem.*, 47(11):2768–2775, 2004. doi:10.1021/jm030543u.
- [60] Philip Cohen and Dario R. Alessi. Kinase drug discovery — What’s next in the field? *ACS Chem. Biol.*, 8(1):96–104, 2013. doi:10.1021/cb300610s.
- [61] Rita Santos, Oleg Ursu, Anna Gaulton, A. Patrícia Bento, Ramesh S. Donadi, Cristian G. Bologa, Anneli Karlsson, Bissan Al-Lazikani, Anne Hersey, Tudor I. Oprea, and John P. Overington. A comprehensive map of molecular drug targets. *Nat. Rev. Drug Discovery*, 16(1):19–34, 2017. doi:10.1038/nrd.2016.230.
- [62] Albert J. Kooistra and Andrea Volkamer. Kinase-centric computational drug development. *Annu. Rep. Med. Chem.*, 50:197–236, 2017. doi:10.1016/BS.ARM.C.2017.08.001.
- [63] Georgi K. Kanev, Chris de Graaf, Bart A. Westerman, Iwan J. P. de Esch, and Albert J. Kooistra. KLIFS: An overhaul after the first 5 years of supporting kinase research. *Nucleic Acids Res.*, 49(D1):D562–D569, 2021. doi:10.1093/NAR/GKAA895.
- [64] Philip Cohen, Darren Cross, and Pasi A. Jänne. Kinase drug discovery 20 years after imatinib: Progress and future directions. *Nat. Rev. Drug Discovery*, 20(7):551–569, 2021. doi:10.1038/s41573-021-00195-4.
- [65] Richard Morphy. Selectively nonselective kinase inhibition: Striking the right balance. *J. Med. Chem.*, 53(4):1413–1437, 2009. doi:10.1021/JM901132V.
- [66] OpenKinome team. Kinodata: Human kinases listed across different resources, 2022. URL [https://github.com/openkinome/kinodata/blob/master/human-kinases/human\\_kinases.ipynb](https://github.com/openkinome/kinodata/blob/master/human-kinases/human_kinases.ipynb). [accessed 2022-08-06].
- [67] Gerard Manning, David B. Whyte, Ricardo Martinez, Tony Hunter, and Sucha Sudarsanam. The protein kinase complement of the human genome. *Science*, 298(5600):1912–1934, 2002. doi:10.1126/science.1075762.
- [68] Sugunadevi Sakkiah, Guang Ping Cao, Staya P. Gupta, and Keun Woo Lee. Overview of the structure and function of protein kinases. *Curr. Enzym. Inhib.*, 13(2):81–88, 2017. doi:10.2174/1573408013666161226155608.
- [69] Sameh Eid, Samo Turk, Andrea Volkamer, Friedrich Rippmann, and Simone Fulle. Kin-Map: A web-based tool for interactive navigation through human kinome data. *BMC Bioinform.*, 18(1):16, 2017. doi:10.1186/s12859-016-1433-7.
- [70] Helen M. Berman, Gerard J. Kleywegt, Haruki Nakamura, and John L. Markley. The Protein Data Bank at 40: Reflecting on the past to prepare for the future. *Structure*, 20(3):391–396, 2012. doi:10.1016/j.str.2012.01.010.

- [71] Anna Gaulton, Anne Hersey, Michał Nowotka, A. Patrícia Bento, Jon Chambers, David Mendez, Prudence Mutowo, Francis Atkinson, Louisa J. Bellis, Elena Cibrián-Uhalte, Mark Davies, Nathan Dedman, Anneli Karlsson, María Paula Magariños, John P. Overington, George Papadatos, Ines Smit, and Andrew R. Leach. The ChEMBL database in 2017. *Nucleic Acids Res.*, 45(D1):D945–D954, 2017. doi:10.1093/nar/gkw1074.
- [72] OpenKinome team. Kinodata: ChEMBL29 kinase dataset, 2022. URL <https://github.com/openkinome/kinodata/releases/tag/v0.3>. [accessed 2022-08-10].
- [73] Dominique Sydow. Kinome tree with PDB and ChEMBL data, 2022. URL [https://github.com/dominiquesydow/dissertation\\_notebooks/blob/v1.1.0/notebooks/kinome\\_stats/kinome\\_stats.ipynb](https://github.com/dominiquesydow/dissertation_notebooks/blob/v1.1.0/notebooks/kinome_stats/kinome_stats.ipynb). [accessed 2022-08-11].
- [74] Oscar P. J. van Linden, Albert J. Kooistra, Rob Leurs, Iwan J. P. De Esch, and Chris De Graaf. KLIFS: A knowledge-based structural database to navigate kinase-ligand interaction space. *J. Med. Chem.*, 57(2):249–277, 2014. doi:10.1021/jm400378w.
- [75] Thomas. D. Schneider and R. Michael Stephens. Sequence logos: A new way to display consensus sequences. *Nucleic Acids Res.*, 18(20):6097–6100, 1990. doi:10.1093/nar/18.20.6097.
- [76] Jeffrey Jie-Lou Liao. Molecular recognition of protein kinase binding pockets for design of potent and selective kinase inhibitors. *J. Med. Chem.*, 50(3):409–424, 2007. doi:10.1021/jm0608107.
- [77] KLIFS team. KLIFS structure entry 823 for 4I22, 2022. URL [https://klifs.net/details.php?structure\\_id=823](https://klifs.net/details.php?structure_id=823). [accessed 2022-08-10].
- [78] KLIFS team. KLIFS structure entry 1092 for 2HYY, 2022. URL [https://klifs.net/details.php?structure\\_id=1092](https://klifs.net/details.php?structure_id=1092). [accessed 2022-08-10].
- [79] Dominique Sydow. Kinase-ligand structures visualized with NGLview and OpenCADD, 2022. URL [https://github.com/dominiquesydow/dissertation\\_notebooks/blob/v1.1.0/notebooks/intro\\_kinase\\_structure/intro\\_kinase\\_structure.ipynb](https://github.com/dominiquesydow/dissertation_notebooks/blob/v1.1.0/notebooks/intro_kinase_structure/intro_kinase_structure.ipynb). [accessed 2022-08-11].
- [80] Alexander S. Rose and Peter W. Hildebrand. NGL Viewer: A web application for molecular visualization. *Nucleic Acids Res.*, 43(W1):W576–W579, 2015. doi:10.1093/nar/gkv402.
- [81] Hai Nguyen, David A. Case, and Alexander S. Rose. NGLView — interactive molecular graphics for Jupyter Notebooks. *Bioinformatics*, 34(7):1241–1242, 2017. doi:10.1093/bioinformatics/btx789.
- [82] Volkamer Lab. OpenCADD, 2022. URL <https://github.com/volkamerlab/opencadd>. [accessed 2022-08-15].
- [83] Fabrice Carles, Stéphane Bourg, Christophe Meyer, and Pascal Bonnet. PKIDB: A curated, annotated and updated database of protein kinase inhibitors in clinical trials. *Molecules*, 23(4):908, 2018. doi:10.3390/molecules23040908.
- [84] Dominique Sydow. Kinase inhibitor statistics based on PKIDB data, 2022. URL [https://github.com/dominiquesydow/dissertation\\_notebooks/blob/v1.1.0/notebooks/pkidb\\_stats/pkidb\\_stats.ipynb](https://github.com/dominiquesydow/dissertation_notebooks/blob/v1.1.0/notebooks/pkidb_stats/pkidb_stats.ipynb). [accessed 2022-08-11].

- [85] Gail A. Van Norman. Drugs, devices, and the FDA: Part 1: An overview of approval processes for drugs. *J. Am. Coll. Cardiol. Basic Trans. Science.*, 1(3):170–179, 2016. doi:10.1016/j.jacbts.2016.03.002.
- [86] Dorian Fabbro, Sandra W. Cowan-Jacob, and Henrik Moebitz. Ten things you should know about protein kinases: IUPHAR Review 14. *Br. J. Pharmacol.*, 172(11):2675–2700, 2015. doi:10.1111/bph.13096.
- [87] Robert Roskoski Jr. Classification of small molecule protein kinase inhibitors based upon the structures of their drug-enzyme complexes. *Pharmacol. Res.*, 103:26–48, 2016. doi:10.1016/j.phrs.2015.10.021.
- [88] Susan Tweedie, Bryony Braschi, Kristian Gray, Tamsin E. M. Jones, Ruth L. Seal, Bethan Yates, and Elspeth A. Bruford. Genenames.org: The HGNC and VGNC resources in 2021. *Nucleic Acids Res.*, 49(D1):D939–D946, 2021. doi:10.1093/nar/gkaa980.
- [89] Alex Bateman, Maria-Jesus Martin, Sandra Orchard, Michele Magrane, Rahat Agivetova, Shadab Ahmad, Emanuele Alpi, Emily H. Bowler-Barnett, Ramona Britto, Borisas Bursteinas, Hema Bye-A-Jee, Ray Coetzee, Austra Cukura, Alan Da Silva, Paul Denny, Tunca Dogan, ThankGod Ebenezer, Jun Fan, Leyla Garcia Castro, Penelope Garmiri, George Georghiou, Leonardo Gonzales, Emma Hatton-Ellis, Abdulrahman Hussein, Alexandr Ignatchenko, Giuseppe Insana, Rizwan Ishtiaq, Petteri Jokinen, Vishal Joshi, Dushyanth Jyothi, Antonia Lock, Rodrigo Lopez, Aurelien Luciani, Jie Luo, Yvonne Lussi, Alistair MacDougall, Fabio Madeira, Mahdi Mahmoudy, Manuela Menchi, Alok Mishra, Katie Moulang, Andrew Nightingale, Carla Susana Oliveira, Sangya Pundir, Guoying Qi, Shriya Raj, Daniel Rice, Milagros Rodriguez Lopez, Rabie Saidi, Joseph Sampson, Tony Sawford, Elena Speretta, Edward Turner, Nidhi Tyagi, Preethi Vasudev, Vladimir Volynkin, Kate Warner, Xavier Watkins, Rossana Zaru, Hermann Zellner, Alan Bridge, Sylvain Poux, Nicole Redaschi, Lucila Aimò, Ghislaine Argoud-Puy, Andrea Auchincloss, Kristian Axelsen, Parit Bansal, Delphine Baratin, Marie-Claude Blatter, Jerven Bolleman, Emmanuel Boutet, Lionel Breuza, Cristina Casals-Casas, Edouard de Castro, Kamal Chikh Echioukh, Elisabeth Coudert, Beatrice Cuche, Mikael Doche, Dolnide Dornevil, Anne Estreicher, Maria Livia Famiglietti, Marc Feuermann, Elisabeth Gasteiger, Sebastien Gehant, Vivienne Gerritsen, Arnaud Gos, Nadine Gruaz-Gumowski, Ursula Hinz, Chantal Hulo, Nevila Hyka-Nouspikel, Florence Jungo, Guillaume Keller, Arnaud Kerhornou, Vicente Lara, Philippe Le Mercier, Damien Lieberherr, Thierry Lombardot, Xavier Martin, Patrick Masson, Anne Morgat, Teresa Batista Neto, Salvo Paesano, Ivo Pedruzzi, Sandrine Pilbout, Lucille Pourcel, Monica Pozzato, Manuela Pruess, Catherine Rivoire, Christian Sigrist, Karin Sonesson, Andre Stutz, Shyamala Sundaram, Michael Tognolli, Laure Verbregue, Cathy H. Wu, Cecilia N. Arighi, Leslie Arminski, Chuming Chen, Yongxing Chen, John S. Garavelli, Hongzhan Huang, Kati Laiho, Peter McGarvey, Darren A. Natale, Karen Ross, C. R. Vinayaka, Qinghua Wang, Yuqi Wang, Lai-Su Yeh, Jian Zhang, Patrick Ruch, and Douglas Teodoro. UniProt: The universal protein knowledgebase in 2021. *Nucleic Acids Res.*, 49(D1):D480–D489, 2020. doi:10.1093/nar/gkaa1100.
- [90] Joanna L. Sharman, Simon D. Harding, Christopher Southan, Elena Faccenda, Adam J. Pawson, Jamie A. Davies, and NC-IUPHAR. Accessing expert-curated pharmacological data in the IUPHAR/BPS Guide to PHARMACOLOGY. *Curr. Protoc. Bioinf.*, 61(1):1.34.1–1.34.46, 2018. doi:10.1002/cpbi.46.

- [91] Ross McGuire, Stefan Verhoeven, Márton Vass, Gerrit Vriend, Iwan J. P. de Esch, Scott J. Lusher, Rob Leurs, Lars Ridder, Albert J. Kooistra, Tina Ritschel, and Chris de Graaf. 3D-e-Chem-VM: Structural cheminformatics research infrastructure in a freely available virtual machine. *J. Chem. Inf. Model.*, 57(2):115–121, 2017. doi:10.1021/acs.jcim.6b00686.
- [92] Dominique Sydow, Jaime Rodríguez-Guerra, and Andrea Volkamer. OpenCADD-KLIFS: A Python package to fetch kinase data from the KLIFS database. *J. Open Source Softw.*, 7(70):3951, 2022. doi:10.21105/joss.03951.
- [93] The pandas development team. pandas-dev/pandas: Pandas, 2020.
- [94] Gilles Marcou and Didier Rognan. Optimizing fragment and scaffold docking by use of molecular interaction fingerprints. *J. Chem. Inf. Model.*, 47(1):195–207, 2006. doi:10.1021/CI600342E.
- [95] Talia B. Kimber, Dominique Sydow, and Andrea Volkamer. Kinase similarity assessment pipeline for off-target prediction [v1.0]. *LiveCoMS*, 3(1):1599–1599, 2022. doi:10.33011/livecoms.3.1.1599.
- [96] OpenKinome team. Kinodata: ChEMBL30 kinase dataset, 2022. URL <https://github.com/openkinome/kinodata/releases/tag/v0.4>. [accessed 2022-09-22].
- [97] Mazen W. Karaman, Sanna Herrgard, Daniel K. Treiber, Paul Gallant, Corey E. Atteridge, Brian T. Campbell, Katrina W. Chan, Pietro Cicceri, Mindy I. Davis, Philip T. Edeen, Raffaella Faraoni, Mark Floyd, Jeremy P. Hunt, Daniel J. Lockhart, Zdravko V. Milanov, Michael J. Morrison, Gabriel Pallares, Hitesh K. Patel, Stephanie Pritchard, Lisa M. Wodicka, and Patrick P. Zarrinkar. A quantitative analysis of kinase inhibitor selectivity. *Nature Biotechnology*, 26(1):127–132, 2008. doi:10.1038/nbt1358.
- [98] Mindy I. Davis, Jeremy P. Hunt, Sanna Herrgard, Pietro Cicceri, Lisa M. Wodicka, Gabriel Pallares, Michael Hocker, Daniel K. Treiber, and Patrick P. Zarrinkar. Comprehensive analysis of kinase inhibitor selectivity. *Nat. Biotechnol.*, 29(11):1046–1051, 2011. doi:10.1038/nbt.1990.
- [99] David H. Drewry, Carrow I. Wells, David M. Andrews, Richard Angell, Hassan Al-Ali, Alison D. Axtman, Stephen J. Capuzzi, Jonathan M. Elkins, Peter Ettmayer, Mathias Frederiksen, Opher Gileadi, Nathanael Gray, Alice Hooper, Stefan Knapp, Stefan Laufer, Ulrich Luecking, Michael Michaelides, Susanne Müller, Eugene Muratov, R. Aldrin Denny, Kumar S. Saikatendu, Daniel K. Treiber, William J. Zuercher, and Timothy M. Willson. Progress towards a public chemogenomic set for protein kinases and a call for contributions. *PLoS ONE*, 12(8):e0181585, 2017. doi:10.1371/journal.pone.0181585.
- [100] Jing Tang, Agnieszka Szwejda, Sushil Shakyawar, Tao Xu, Petteri Hintsanen, Krister Wennerberg, and Tero Aittokallio. Making sense of large-scale kinase inhibitor bioactivity data sets: A comparative and integrative analysis. *J. Chem. Inf. Model.*, 54(3):735–743, 2014. doi:10.1021/ci400709d.
- [101] Nienke Moret, Nicholas A. Clark, Marc Hafner, Yuan Wang, Eugen Lounkine, Mario Medvedovic, Jinhua Wang, Nathanael Gray, Jeremy Jenkins, and Peter K. Sorger. Cheminformatics tools for analyzing and designing optimized small-molecule collections and libraries. *Cell Chem. Biol.*, 26(5):765–777.e3, 2019. doi:10.1016/j.chembiol.2019.02.018.

- [102] Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, Jildau Bouwman, Anthony J. Brookes, Tim Clark, Mercè Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T. Evelo, Richard Finkers, Alejandra Gonzalez-Beltran, Alasdair J. G. Gray, Paul Groth, Carole Goble, Jeffrey S. Grethe, Jaap Heringa, Peter A. C. 't Hoen, Rob Hooft, Tobias Kuhn, Ruben Kok, Joost Kok, Scott J. Lusher, Maryann E. Martone, Albert Mons, Abel L. Packer, Bengt Persson, Philippe Rocca-Serra, Marco Roos, Rene van Schaik, Susanna-Assunta Sansone, Erik Schultes, Thierry Sengstag, Ted Slater, George Strawn, Morris A. Swertz, Mark Thompson, Johan van der Lei, Erik van Mulligen, Jan Velterop, Andra Waagmeester, Peter Wittenburg, Katherine Wolstencroft, Jun Zhao, and Barend Mons. The FAIR guiding principles for scientific data management and stewardship. *Sci. Data*, 3(1):160018, 2016. doi:10.1038/sdata.2016.18.
- [103] GO FAIR initiative. GO FAIR initiative aiming to implement the FAIR data principles, 2022. URL <https://www.go-fair.org/fair-principles>. [accessed 2022-08-15].
- [104] Neil P. Chue Hong, Daniel S. Katz, Michelle Barker, Anna-Lena Lamprecht, Carlos Martinez, Fotis E. Psomopoulos, Jen Harrow, Leyla Jael Castro, Morane Gruenpeter, Paula Andrea Martinez, Tom Honeyman, Alexander Struck, Allen Lee, Axel Loewe, Ben van Werkhoven, Catherine Jones, Daniel Garijo, Esther Plomp, Françoise Genova, Hugh Shanahan, Joanna Leng, Maggie Hellström, Malin Sandström, Manodeep Sinha, Mateusz Kuzak, Patricia Herterich, Qian Zhang, Sharif Islam, Susanna-Assunta Sansone, Tom Pollard, Udayanto Dwi Atmojo, Alan Williams, Andreas Czerniak, Anna Niehues, Anne Claire Fouilloux, Bala Desinghu, Carole Goble, Céline Richard, Charles Gray, Chris Erdmann, Daniel Nüst, Daniele Tartarini, Elena Ranguelova, Hartwig Anzt, Ilian Todorov, James McNally, Javier Moldon, Jessica Burnett, Julián Garrido-Sánchez, Khalid Belhajjame, Laurents Sesink, Lorraine Hwang, Marcos Roberto Tovani-Palone, Mark D. Wilkinson, Mathieu Servillat, Matthias Liffers, Merc Fox, Nadica Miljković, Nick Lynch, Paula Martinez Lavanchy, Sandra Gesing, Sarah Stevens, Sergio Martinez Cuesta, Silvio Peroni, Stian Soiland-Reyes, Tom Bakker, Tovo Rabemanantsoa, Vanessa Sochat, Yo Yehudi, and RDA FAIR4RS WG. FAIR principles for research software (FAIR4RS principles), 2022. URL <https://doi.org/10.15497/RDA00068>.
- [105] Anna-Lena Lamprecht, Leyla Garcia, Mateusz Kuzak, Carlos Martinez, Ricardo Arcila, Eva Martin Del Pico, Victoria Dominguez Del Angel, Stephanie van de Sandt, Jon Ison, Paula Andrea Martinez, Peter McQuilton, Alfonso Valencia, Jennifer Harrow, Fotis Psomopoulos, Josep Ll Gelpi, Neil Chue Hong, Carole Goble, and Salvador Capella-Gutierrez. Towards FAIR principles for research software. *Data Sci. J.*, 3(1):37–59, 2020. doi:10.3233/DS-190026.
- [106] Zenodo team. Zenodo website, 2022. URL <https://zenodo.org>. [accessed 2022-08-16].
- [107] Dustin Ingram. PEP 566 - Metadata for Python software packages 2.1, 2022. URL <https://peps.python.org/pep-0566>. [accessed 2022-08-16].
- [108] GitHub team. GitHub website, 2022. URL <https://github.com>. [accessed 2022-08-16].
- [109] GitLab team. GitLab website, 2022. URL <https://about.gitlab.com>. [accessed 2022-08-16].

- [110] BitBucket team. BitBucket website, 2022. URL <https://bitbucket.org>. [accessed 2022-08-16].
- [111] OpenAPI team. OpenAPI specifications, 2022. URL <https://swagger.io/specification>. [accessed 2022-08-16].
- [112] Docker team. Docker website, 2022. URL <https://www.docker.com>. [accessed 2022-08-16].
- [113] Guido van Rossum, Barry Warsaw, and Nick Coghlan. PEP 8 — Style guide for Python code, 2022. URL <https://peps.python.org/pep-0008>. [accessed 2022-08-16].
- [114] Rajarshi Guha, Michael T. Howard, Geoffrey R. Hutchison, Peter Murray-Rust, Henry Rzepa, Christoph Steinbeck, Jörg Wegner, and Egon L. Willighagen. The Blue Obelisk—interoperability in chemical informatics. *J. Chem. Inf. Model.*, 46(3):991–998, 2006. doi:10.1021/ci050400b.
- [115] Noel M. O’Boyle, Michael Banck, Craig A. James, Chris Morley, Tim Vandermeersch, and Geoffrey R. Hutchison. Open Babel: An open chemical toolbox. *J. Cheminform.*, 3(1):33, 2011. doi:10.1186/1758-2946-3-33.
- [116] Christoph Steinbeck, Yongquan Han, Stefan Kuhn, Oliver Horlacher, Edgar Luttmann, and Egon Willighagen. The Chemistry Development Kit (CDK): An open-source Java library for chemo- and bioinformatics. *J. Chem. Inf. Model.*, 43(2):493–500, 2003. doi:10.1021/ci025584y.
- [117] Christoph Steinbeck, Christian Hoppe, Stefan Kuhn, Matteo Floris, Rajarshi Guha, and Egon L. Willighagen. Recent developments of the Chemistry Development Kit (CDK) - An open-source Java library for chemo- and bioinformatics. *Curr. Pharm. Des.*, 12(17):2111–2120, 2006. doi:10.2174/138161206777585274.
- [118] John W. May and Christoph Steinbeck. Efficient ring perception for the Chemistry Development Kit. *J. Cheminform.*, 6(1):3, 2014. doi:10.1186/1758-2946-6-3.
- [119] Egon L. Willighagen, John W. Mayfield, Jonathan Alvarsson, Arvid Berg, Lars Carlsson, Nina Jeliakova, Stefan Kuhn, Tomáš Pluskal, Miquel Rojas-Chertó, Ola Spjuth, Gilleain Torrance, Chris T. Evelo, Rajarshi Guha, and Christoph Steinbeck. The Chemistry Development Kit (CDK) v2.0: Atom typing, depiction, molecular formulas, and substructure searching. *J. Cheminform.*, 9(1):33, 2017. doi:10.1186/s13321-017-0220-4.
- [120] Noel M. O’Boyle, Rajarshi Guha, Egon L. Willighagen, Samuel E. Adams, Jonathan Alvarsson, Jean-Claude Bradley, Igor V. Filippov, Robert M. Hanson, Marcus D. Hanwell, Geoffrey R. Hutchison, Craig A. James, Nina Jeliakova, Andrew S. I. D. Lang, Karol M. Langner, David C. Lonie, Daniel M. Lowe, Jérôme Pansanel, Dmitry Pavlov, Ola Spjuth, Christoph Steinbeck, Adam L. Tenderholt, Kevin J. Theisen, and Peter Murray-Rust. Open data, open source and open standards in chemistry: The Blue Obelisk five years on. *J. Cheminform.*, 3(1):37, 2011. doi:10.1186/1758-2946-3-37.
- [121] Open Molecular Software Foundation. OMSF website, 2022. URL <https://omsf.io>. [accessed 2022-08-16].
- [122] Google. Google Summer of Code website, 2022. URL <https://summerofcode.withgoogle.com>. [accessed 2022-08-16].

- [123] Chan Zuckerberg Initiative. The CZI's Open Science program, 2022. URL <https://chanzuckerberg.com/science/programs-resources/open-science>. [accessed 2022-08-16].
- [124] Quansight Labs. Quansight Labs' supported projects, 2022. URL <https://labs.quansight.org>. [accessed 2022-08-16].
- [125] NFDI4Chem. Chemistry Consortium in the NFDI website, 2022. URL <https://www.nfdi4chem.de>. [accessed 2022-08-16].
- [126] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020. doi:10.1038/s41586-020-2649-2.
- [127] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.*, 12:2825–2830, 2011. URL <http://jmlr.org/papers/v12/pedregosa11a.html>.
- [128] John D. Hunter. Matplotlib: A 2D graphics environment. *Comput. Sci. Eng.*, 9(3):90–95, 2007. doi:10.1109/MCSE.2007.55.
- [129] Michael L. Waskom. seaborn: Statistical data visualization. *J. Open Source Softw.*, 6(60):3021, 2021. doi:10.21105/joss.03021.
- [130] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, Carol Willing, and Jupyter development team. Jupyter notebooks - a publishing format for reproducible computational workflows. In Fernando Loizides and Birgit Schmidt, editors, *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87–90, Netherlands, 2016. IOS Press. doi:10.3233/978-1-61499-649-1-87.
- [131] RDKit. RDKit: Open-source cheminformatics, 2022. URL <http://www.rdkit.org>. [accessed 2022-08-29].
- [132] Patrick Kunzmann and Kay Hamacher. Biotite: A unifying open source computational biology framework in Python. *BMC Bioinformatics*, 19(1):346, 2018. doi:10.1186/s12859-018-2367-z.
- [133] William Gilpin. PyPDB: A Python API for the Protein Data Bank. *Bioinformatics*, 32(1):159–60, 2015. doi:10.1093/bioinformatics/btv543.
- [134] Mark Davies, Michał Nowotka, George Papadatos, Nathan Dedman, Anna Gaulton, Francis Atkinson, Louisa Bellis, and John P. Overington. ChEMBL Web Services: Streamlining access to drug discovery data and utilities. *Nucleic Acids Res.*, 43(W1):W612–W620, 2015. doi:10.1093/nar/gkv352.



- [135] Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A. Shoemaker, Paul A. Thiessen, Bo Yu, Leonid Zaslavsky, Jian Zhang, and Evan E. Bolton. PubChem in 2021: New data content and improved web interfaces. *Nucleic Acids Res.*, 49(D1):D1388–D1395, 2020. doi:10.1093/nar/gkaa971.
- [136] Rainer Fährrolfes, Stefan Bietz, Florian Flachsenberg, Agnes Meyder, Eva Nittinger, Thomas Otto, Andrea Volkamer, and Matthias Rarey. ProteinsPlus: A web portal for structure analysis of macromolecules. *Nucleic Acids Res.*, 45(W1):W337–W343, 2017. doi:10.1093/nar/gkx333.
- [137] Steven M. Paul, Daniel S. Mytelka, Christopher T. Dunwiddie, Charles C. Persinger, Bernard H. Munos, Stacy R. Lindborg, and Aaron L. Schacht. How to improve R&D productivity: The pharmaceutical industry’s grand challenge. *Nat. Rev. Drug Discovery*, 9(3):203–214, 2010. doi:10.1038/nrd3078.
- [138] Stephani Joy Y. Macalino, Vijayakumar Gosu, Sunhye Hong, and Sun Choi. Role of computer-aided drug design in modern drug discovery. *Arch. Pharmacol Res.*, 38(9):1686–1701, 2015. doi:10.1007/s12272-015-0640-5.
- [139] Bernd Beck and Tim Geppert. Industrial applications of in silico ADMET. *J. Mol. Model.*, 20(7):2322, 2014. doi:10.1007/s00894-014-2322-5.
- [140] Our World in Data. Cancer, 2022. URL <https://ourworldindata.org/cancer>. [accessed 2022-08-28].
- [141] Dominique Sydow, Eva Aßmann, Albert J. Kooistra, Friedrich Rippmann, and Andrea Volkamer. KiSSim: Predicting off-targets from structural similarities in the kinome. *J. Chem. Inf. Model.*, 62(10):2600–2616, 2022. doi:10.1021/acs.jcim.2c00050.
- [142] Denis Schmidt, Magdalena M. Scharf, Dominique Sydow, Eva Aßmann, Maria Martí-Solano, Marina Keul, Andrea Volkamer, and Peter Kolb. Analyzing kinase similarity in small molecule and protein structural space to explore the limits of multi-target screening. *Molecules*, 26(3):629, 2021. doi:10.3390/molecules26030629.
- [143] Dominique Sydow, Paula Schmiel, Jérémie Mortier, and Andrea Volkamer. KinFragLib: Exploring the kinase inhibitor space using subpocket-focused fragmentation and recombination. *J. Chem. Inf. Model.*, 60(12):6081–6094, 2020. doi:10.1021/acs.jcim.0c00839.
- [144] Dominique Sydow, Andrea Morger, Maximilian Driller, and Andrea Volkamer. TeachOpenCADD: A teaching platform for computer-aided drug design using open source packages and data. *J. Cheminf.*, 11(1):29, 2019. doi:10.1186/s13321-019-0351-x.
- [145] Dominique Sydow, Jaime Rodríguez-Guerra, Talia B Kimber, David Schaller, Corey J Taylor, Yonghui Chen, Mareike Leja, Sakshi Misra, Michele Wichmann, Armin Ariamajd, and Andrea Volkamer. TeachOpenCADD 2022: Open source and FAIR Python pipelines to assist in structural bioinformatics and cheminformatics research. *Nucleic Acids Res.*, 50(W1):W753–W760, 2022. doi:10.1093/nar/gkac267.
- [146] The conda-forge community. The conda-forge project: Community-based software distribution built on the conda package format and ecosystem, 2015. URL <https://doi.org/10.5281/zenodo.4774216>. [accessed 2022-08-29].

- [147] Structural Genomics Consortium. SGC website, 2022. URL <https://www.thesgc.org>. [accessed 2022-08-24].
- [148] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstern, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021. doi:10.1038/s41586-021-03819-2.
- [149] Asher Mullard. A probe for every protein. *Nat. Rev. Drug Discovery*, 18(10):733–736, 2019. doi:10.1038/d41573-019-00159-9.
- [150] Yuqi Zhang, Marton Vass, Da Shi, Esam Abualrous, Jenny Chambers, Nikita Chopra, Chris Higgs, Koushik Kasavajhala, Hubert Li, Prajwal Nandekar, Hideyuki Sato, Edward Miller, Matt Repasky, and Steven Jerome. Benchmarking refined and unrefined AlphaFold2 structures for hit discovery, 2022. URL <https://doi.org/10.26434/chemrxiv-2022-kcn0d>. [accessed 2022-08-29].
- [151] Jürgen Bajorath. Representation and identification of activity cliffs. *Expert Opin. Drug Discovery*, 12(9):879–883, 2017. doi:10.1080/17460441.2017.1353494.
- [152] KiSSim team. KiSSim method, 2022. URL <https://github.com/volkamerlab/kissim>. [accessed 2022-08-29].
- [153] KiSSim team. KiSSim applications, 2022. URL [https://github.com/volkamerlab/kissim\\_app](https://github.com/volkamerlab/kissim_app). [accessed 2022-08-30].
- [154] Pedro J. Ballester and W. Graham Richards. Ultrafast shape recognition to search compound databases for similar molecular shapes. *J. Comput. Chem.*, 28(10):1711–1723, 2007. doi:10.1002/jcc.20681.
- [155] Structural Genomics Consortium. SGC-STK17B-1: A chemical probe for STK17B/DRAK2 kinase, 2017. URL <https://www.thesgc.org/chemical-probes/SGC-STK17B-1>. [accessed 2022-09-21].
- [156] Manfred Hendlich, Friedrich Rippmann, and Gerhard Barnickel. LIGSITE: Automatic and efficient detection of potential small molecule-binding sites in proteins. *J. Mol. Graph. Model.*, 15(6):359–363, 1997. doi:10.1016/S1093-3263(98)00002-3.
- [157] Stefan Schmitt, Daniel Kuhn, and Gerhard Klebe. A new method to detect related function among proteins independent of sequence and fold homology. *J. Mol. Biol.*, 323(2):387–406, 2002. doi:10.1016/S0022-2836(02)00811-2.
- [158] Timo Krotzky, Thomas Fober, Eyke Hüllermeier, and Gerhard Klebe. Extended graph-based models for enhanced similarity search in Cavbase. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, 11(5):878–890, 2014. doi:10.1109/TCBB.2014.2325020.

- [159] M. Sandberg, L. Eriksson, J. Jonsson, M. Sjöström, and S. Wold. New chemical descriptors relevant for the design of biologically active peptides. A multivariate characterization of 87 amino acids. *J. Med. Chem.*, 41(14):2481–2491, 1998. doi:10.1021/jm9700575.
- [160] Nathanaël Weill and Didier Rognan. Alignment-free ultra-high-throughput comparison of druggable protein-ligand binding sites. *J. Chem. Inf. Model.*, 50(1):123–135, 2010. doi:10.1021/ci900349y.
- [161] Dominique Sydow. volkamerlab/ratar: v.0.1.0, 2022.
- [162] Volkamer Lab. KinFragLib GitHub repository, 2022. URL <https://github.com/volkamerlab/KinFragLib>. [accessed 2022-08-29].
- [163] Anna Gaulton, Anne Hersey, Michał Nowotka, A. Patrícia Bento, Jon Chambers, David Mendez, Prudence Mutowo, Francis Atkinson, Louisa J. Bellis, Elena Cibrián-Uhalte, Mark Davies, Nathan Dedman, Anneli Karlsson, María Paula Magariños, John P. Overington, George Papadatos, Ines Smit, and Andrew R. Leach. The ChEMBL database in 2017. *Nucleic Acids Res.*, 45(D1):D945–D954, 2017. doi:10.1093/nar/gkw1074.
- [164] ChEMBL. ChEMBL25 download, 2019. URL <https://doi.org/10.6019/CHEMBL.database.25>. [accessed 2022-08-29].
- [165] Sonja Leo. Custom-KinFragLib: Exploring filter strategies to reduce the library size while increasing the feasibility of the recombined compounds. Master’s thesis, Freie Universität Berlin, Berlin, Germany, 2021.
- [166] Ruth Brenk, Alessandro Schipani, Daniel James, Agata Krasowski, Ian Hugh Gilbert, Julie Frearson, and Paul Graham Wyatt. Lessons learnt from assembling screening libraries for drug discovery for neglected diseases. *ChemMedChem*, 3(3):435–444, 2008. doi:10.1002/cmde.200700139.
- [167] Jonathan B. Baell and Georgina A. Holloway. New substructure filters for removal of pan assay interference compounds (PAINS) from screening libraries and for their exclusion in bioassays. *J. Med. Chem.*, 53(7):2719–2740, 2010. doi:10.1021/jm901137j.
- [168] G. Richard Bickerton, Gaia V. Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L. Hopkins. Quantifying the chemical beauty of drugs. *Nat. Chem.*, 4(2):90–98, 2012. doi:10.1038/nchem.1243.
- [169] Enamine. Enamine REAL Space, 2022. URL <https://enamine.net/compound-collections/real-compounds/real-space-navigator>. [accessed 2022-08-28].
- [170] Thomas Sander, Joel Freyss, Modest von Korff, and Christian Rufener. DataWarrior: An open-source program for chemistry aware data visualization and analysis. *J. Chem. Inf. Model.*, 55(2):460–473, 2015. doi:10.1021/ci500588j.
- [171] Milan Voršilák, Michal Kolář, Ivan Čmelo, and Daniel Svozil. SYBA: Bayesian estimation of synthetic accessibility of organic compounds. *J. Cheminform.*, 12(1):35, 2020. doi:10.1186/s13321-020-00439-2.
- [172] Connor W. Coley, Dale A. Thomas, Justin A. M. Lummiss, Jonathan N. Jaworski, Christopher P. Breen, Victor Schultz, Travis Hart, Joshua S. Fishman, Luke Rogers, Hanyu Gao,

- Robert W. Hicklin, Pieter P. Plehiers, Joshua Byington, John S. Piotti, William H. Green, A. John Hart, Timothy F. Jamison, and Klavs F. Jensen. A robotic platform for flow synthesis of organic compounds informed by AI planning. *Science*, 365(6453):eaax1566, 2019. doi:10.1126/science.aax1566.
- [173] Peter Ertl and Ansgar Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *J. Cheminform.*, 1(1):8, 2009. doi:10.1186/1758-2946-1-8.
- [174] Connor W. Coley, Luke Rogers, William H. Green, and Klavs F. Jensen. SCScore: Synthetic complexity learned from a reaction corpus. *J. Chem. Inf. Model.*, 58(2):252–261, 2018. doi:10.1021/acs.jcim.7b00622.
- [175] Elsevier Life Sciences IP Limited. Reaxys database, 2022. URL <https://www.reaxys.com>. [accessed 2022-09-22].
- [176] Amol Thakkar, Veronika Chadimová, Esben Jannik Bjerrum, Ola Engkvist, and Jean-Louis Reymond. Retrosynthetic accessibility score (RAscore) — rapid machine learned synthesizability classification from AI driven retrosynthetic planning. *Chem. Sci. J.*, 12(9):3339–3349, 2021. doi:10.1039/D0SC05401A.
- [177] Christina Humer, Henry Heberle, Floriane Montanari, Thomas Wolf, Florian Huber, Ryan Henderson, Julian Heinrich, and Marc Streit. ChemInformatics Model Explorer (CIME): Exploratory analysis of chemical model explanations. *J. Cheminform.*, 14(1):21, 2022. doi:10.1186/s13321-022-00600-z.
- [178] Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction, 2018. URL <https://arxiv.org/abs/1802.03426>.
- [179] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *J. Chem. Inf. Model.*, 50(5):742–754, 2010. doi:10.1021/ci100050t.
- [180] Petra Schneider, W. Patrick Walters, Alleyn T. Plowright, Norman Sieroka, Jennifer Listgarten, Robert A. Goodnow, Jasmin Fisher, Johanna M. Jansen, José S. Duca, Thomas S. Rush, Matthias Zentgraf, John Edward Hill, Elizabeth Krutoholow, Matthias Kohler, Jeff Blaney, Kimito Funatsu, Chris Luebke, and Gisbert Schneider. Rethinking drug design in the artificial intelligence era. *Nat. Rev. Drug Discovery*, 19(5):353–364, 2020. doi:10.1038/s41573-019-0050-3.
- [181] Volkamer Lab. TeachOpenCADD website, 2022. URL <https://projects.volkamerlab.org/teachopencadd>. [accessed 2022-08-29].
- [182] Binder. Binder, 2022. URL <https://mybinder.org>. [accessed 2022-08-29].
- [183] Binder. TeachOpenCADD on Binder, 2022. URL <https://mybinder.org/v2/gh/volkamerlab/TeachOpenCADD/master>. [accessed 2022-08-29].
- [184] Sunghwan Kim, Paul A. Thiessen, Evan E. Bolton, Jie Chen, Gang Fu, Asta Gindulyte, Lianyi Han, Jane He, Siqian He, Benjamin A. Shoemaker, Jiyao Wang, Bo Yu, Jian Zhang, and Stephen H. Bryant. PubChem substance and compound databases. *Nucleic Acids Res.*, 44(D1):D1202–D1213, 2016. doi:10.1093/nar/gkv951.

- [185] Dominique Sydow, Michele Wichmann, Jaime Rodríguez-Guerra, Daria Goldmann, Gregory Landrum, and Andrea Volkamer. TeachOpenCADD-KNIME: A teaching platform for computer-aided drug design using knime workflows. *J. Chem. Inf. Model.*, 59(10): 4083–4086, 2019. doi:10.1021/acs.jcim.9b00662.
- [186] Dominique Sydow, Jaime Rodríguez-Guerra, and Andrea Volkamer. *Teaching Computer-Aided Drug Design Using TeachOpenCADD*, chapter 10, pages 135–158. ACS Symposium Series, 2021. doi:10.1021/bk-2021-1387.ch010.
- [187] J. Cheminform. TeachOpenCADD paper’s access and citations, 2022. URL <https://jcheminf.biomedcentral.com/articles/10.1186/s13321-019-0351-x/metrics>. [accessed 2022-09-27].
- [188] GitHub. TeachOpenCADD’s GitHub stars, 2022. URL <https://github.com/volkamerlab/teachopencadd/stargazers>. [accessed 2022-09-27].
- [189] Dominique Sydow, Jaime Rodríguez-Guerra, , Dennis Köser, Annie Pham, Enes Kurnaz, Julian Pipart, and Corey Taylor. volkamerlab/opencadd: v1.0.1, 2022. URL <https://doi.org/10.5281/zenodo.6065555>. [accessed 2022-08-29].
- [190] Volkamer Lab. OpenCADD-superposition, 2022. URL <https://github.com/volkamerlab/opencadd/tree/ds-move-paper/papers/opencadd-superposition>. [accessed 2022-08-29].
- [191] Zukang Feng, Li Chen, Himabindu Maddula, Ozgur Akcan, Rose Oughtred, Helen M. Berman, and John Westbrook. Ligand Depot: A data warehouse for ligands bound to macromolecules. *Bioinformatics*, 20(13):2153–2155, 2004. doi:10.1093/bioinformatics/bth214.
- [192] TeachOpenCADD. TeachOpenCADD: A teaching platform for computer-aided drug design (CADD) using open source packages and data, 2022. URL <https://github.com/volkamerlab/teachopencadd>. [accessed 2022-08-29].
- [193] OpenCADD. OpenCADD-pocket: Identification and analysis of protein (sub)pockets, 2022. URL <https://github.com/volkamerlab/opencadd>. [accessed 2022-08-29].
- [194] OpenKinome team. KinoML: Structure-informed machine learning for kinase modeling, 2022. URL <https://github.com/openkinome/kinoml>. [accessed 2022-08-29].
- [195] PLIPify. PLIPify: Protein-ligand interaction frequencies across multiple structures, 2022. URL <https://github.com/volkamerlab/plipify>. [accessed 2022-08-29].
- [196] pytest. pytest, 2022. URL <https://docs.pytest.org>. [accessed 2022-08-29].
- [197] nbval. nbval, 2022. URL <https://nbval.readthedocs.io/en/latest>. [accessed 2022-08-29].
- [198] GitHub. GitHub Actions, 2022. URL <https://docs.github.com/en/actions>. [accessed 2022-08-29].
- [199] MolSSI. cookiecutter-cms, 2022. URL <https://github.com/MolSSI/cookiecutter-cms>. [accessed 2022-08-29].
- [200] Sphinx. Sphinx - Python documentation generator, 2022. URL <https://www.sphinx-doc.org>. [accessed 2022-08-29].

- [201] Python Software Foundation. Python Enhancement Proposal 8, 2022. URL <https://www.python.org/dev/peps/pep-0008>. [accessed 2022-08-29].
- [202] Python Software Foundation. Black: The uncompromising Python code formatter, 2022. URL <https://github.com/psf/black>. [accessed 2022-08-29].
- [203] Black-nb. Black-nb: The uncompromising code formatter, for Jupyter notebooks, 2022. URL <https://github.com/tomcatling/black-nb>. [accessed 2022-08-29].
- [204] Pylint. Pylint: Static code analyser, 2022. URL <https://github.com/PyCQA/pylint>. [accessed 2022-08-29].
- [205] Flake8. Flake8: Your tool for style guide enforcement, 2022. URL <https://github.com/PyCQA/flake8>. [accessed 2022-08-29].
- [206] DFG project by Andrea Volkamer. Read-across the targetome — An integrated structure- and ligand-based workbench for computational design of novel tool compounds, 2017–2022. URL <https://gepris.dfg.de/gepris/projekt/391684253?language=en>. [accessed 2022-08-24].
- [207] Guide to Pharmacology team. Guide to Pharmacology, 2022. URL <https://www.guidetopharmacology.org>. [accessed 2022-08-24].
- [208] Chemical Probes Portal team. Chemical Probes Portal, 2022. URL <https://chemicalprobes.org>. [accessed 2022-08-24].
- [209] Target 2035 team. Target 2035, 2022. URL <https://www.target2035.net>. [accessed 2022-08-24].
- [210] Adrian J. Carter, Oliver Kraemer, Matthias Zwick, Anke Mueller-Fahrnow, Cheryl H. Arrowsmith, and Aled M. Edwards. Target 2035: Probing the human proteome. *Drug Discovery Today*, 24(11):2111–2115, 2019. doi:10.1016/j.drudis.2019.06.020.
- [211] Susanne Müller, Suzanne Ackloo, Arij Al Chawaf, Bissan Al-Lazikani, Albert Antolin, Jonathan B. Baell, Hartmut Beck, Shaunna Beedie, Ulrich A. K. Betz, Gustavo Arruda Bezerra, Paul E. Brennan, David Brown, Peter J. Brown, Alex N. Bullock, Adrian J. Carter, Apirat Chaikuad, Mathilde Chaineau, Alessio Ciulli, Ian Collins, Jan Dreher, David Drewry, Kristina Edfeldt, Aled M. Edwards, Ursula Egnér, Stephen V. Frye, Stephen M. Fuchs, Matthew D. Hall, Ingo V. Hartung, Alexander Hillisch, Stephen H. Hitchcock, Evert Homan, Natarajan Kannan, James R. Kiefer, Stefan Knapp, Milka Kostic, Stefan Kubicek, Andrew R. Leach, Sven Lindemann, Brian D. Marsden, Hisanori Matsui, Jordan L. Meier, Daniel Merk, Maurice Michel, Maxwell R. Morgan, Anke Mueller-Fahrnow, Dafydd R. Owen, Benjamin G. Perry, Saul H. Rosenberg, Kumar Singh Saikatendu, Matthieu Schapira, Cora Scholten, Sujata Sharma, Anton Simeonov, Michael Sundström, Giulio Superti-Furga, Matthew H. Todd, Claudia Tredup, Masoud Vedadi, Frank von Delft, Timothy M. Willson, Georg E. Winter, Paul Workman, and Cheryl H. Arrowsmith. Target 2035 — update on the quest for a probe for every protein. *RSC Med. Chem.*, 13(1):13–21, 2022. doi:10.1039/D1MD00228G.
- [212] M. Stuart Armstrong, Garrett M. Morris, Paul W. Finn, Raman Sharma, and W. Graham Richards. Molecular similarity including chirality. *J. Mol. Graph. Model.*, 28(4):368–370, 2009. doi:10.1016/J.JMGM.2009.09.002.

- [213] M. Stuart Armstrong, Garrett M. Morris, Paul W. Finn, Raman Sharma, Loris Moretti, Richard I. Cooper, and W. Graham Richards. ElectroShape: Fast molecular similarity calculations incorporating shape, chirality and electrostatics. *J. Comput. Aided Mol.*, 24(9):789–801, 2010. doi:10.1007/s10822-010-9374-0.
- [214] Esther Kellenberger, Pascal Muller, Claire Schalon, Guillaume Bret, Nicolas Foata, and Didier Rognan. sc-PDB: An annotated database of druggable binding sites from the Protein Data Bank. *J. Chem. Inf. Model.*, 46(2):717–727, 2006. doi:10.1021/ci050372x.
- [215] Stefan Schmitt, Daniel Kuhn, and Gerhard Klebe. A new method to detect related function among proteins independent of sequence and fold homology. *J. Mol. Biol.*, 323(2):387–406, 2002. doi:10.1016/S0022-2836(02)00811-2.
- [216] Jörgen Jonsson, Lennart Eriksson, Sven Hellberg, Michael Sjöström, and Svante Wold. Multivariate parametrization of 55 coded and non-coded amino acids. *Quant. Struct.-Act. Relat.*, 8(3):204–209, 1989. doi:10.1002/qsar.19890080303.
- [217] Gerard J. P. van Westen, Remco F. Swier, Isidro Cortes-Ciriano, Jörg K. Wegner, John P. Overington, Adriaan P. IJzerman, Herman W. T. van Vlijmen, and Andreas Bender. Benchmarking of protein descriptor sets in proteochemometric modeling (part 2): Modeling performance of 13 amino acid descriptor sets. *J. Cheminform.*, 5(1):42, 2013. doi:10.1186/1758-2946-5-42.
- [218] Christiane Ehrt, Tobias Brinkjost, and Oliver Koch. A benchmark driven guide to binding site comparison: An exhaustive evaluation using tailor-made data sets (ProSPECCTs). *PLoS Comput. Biol.*, 14(11):e1006483, 2018. doi:10.1371/journal.pcbi.1006483.