Masther Thesis

Markov-chain modelling of dynamic interaction patterns in supramolecular complexes

Freie Universität Berlin

Julian M. Kleber, Matr.-Nr.: 4965334
Institut für Mathematik und Informatik,
Freie Universität Berlin

Supervisors:
Prof. Dr. Gerhard Wolber
Institut für Pharmazie,
Freie Universität Berlin

Dr. Szymon Pach
Institut für Pharmazie,
Freie Universität Berlin

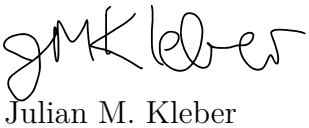
April 22, 2022

Declaration of Authorship

Ich versichere hiermit an Eides Statt, dass diese Arbeit von niemand anderem als meiner Person verfasst worden ist. Alle verwendeten Hilfsmittel wie Berichte, Bücher, Internetseiten oder ähnliches sind im Literaturverzeichnis angegeben, Zitate aus fremden Arbeiten sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

I hereby affirm on oath that this work will not be done by anyone else than my person. All tools used, such as reports, books, websites, or the like are listed in the bibliography. Quotations from third-party works are identified as such. The work has not yet been submitted in the same or similar form to any other examination board submitted and was also not published.

April 22, 2022



Julian M. Kleber

Acknowledgement

I want to say thank you to everyone supporting me and my work up to now. Thank you, Gerhard, for introducing me to your group and providing the computational resources needed to complete this work. Thank you, Szymon, for being such a decent sparring partner during the development of the software presented in this work. Thank you, Katrin, for providing critical questions and test systems outside of this work to verify the new developments.

I thank my family for supporting me. I thank my parents, Jutta, and Arno, as well as my grandmother Margarete for their everlasting support. Thank you, Joshua, for trying out even the craziest ideas from lean drug development with me. Thank you, C., for sharing wonderful moments with me. Thank you, Nina, for always holding to me. May you never be in danger of a disease again. Thank you Anke, thank you, Ulrich, for all the family gatherings. They were always wonderful and strengthening. I want to say thank you to my supporters who were providing valuable sparring, coaching, time, emotions and resources.

Abstract

For decades, the drug design community was concerned with deriving insights for their targets from a static picture using X-ray crystallography and pharmacophores. Likewise, computational methods for *in silico* studies of ligand binding rely heavily on docking, a static method. Still, computational methods such as molecular dynamics (MD) simulations were gaining prominence in the field of computational drug design. In consequence, as the computational power increased and the simulation technology matured, MD simulations became more and more critical for the drug design process. Thus, innately, there have to be deep conflicts between the current pool of models derived from a static mindset and insights derived from dynamic modelling. The recent introduction of structure-based ensemble-QSAR methods[1] and dynamic pharmacophores, termed *Dynophores*[2] served as a primer to bridge the gap between the two ways of modelling supramolecular complexes common in computational drug design. Recently, *Dynophores* replaced classical pharmacophores in a virtual screening study on the DNA topoisomerase IIa[3].

The subsequent work focuses on the development and the application of a new methodology deriving models from MD simulations using lean project management techniques, *Dynophores* and Markov modelling. The newly developed methodology abandons the static picture entirely, and investigates three prominent drug targets. The first target is the ZIKA virus protease and serves as a benchmark against domain expert knowledge for designing *de novo* inhibitors[4]. The second target is the human cyclin dependent kinase 2 (CDK2) complexed with three different ligands[5,6]. The system serves as a benchmark to derive new insights from experimental data, as the mechanisms of action and inhibition of CDK2 are up to now not genuinely understood. The third system is the hepatitis C virus (HCV) NS3/4A protease shall serve as a test if the method can explain resistance mechanisms in prominent pathogens against data derived from standard experimental protocols. Four different supramolecular complexes of the NS3/4A protease containing the wild type and three different mutations, each complexed with *vaniprevir*[7,8] were investigated. For all three test systems, new insights were generated.

Contents

List of Figures

Acronyms

Mathematical Symbols

x_t Observed random variable at time t. Used as the state variable in Markov (state) models.

$X^{(t)}$ Random variable at time t.

$\mathbf{o}$ Observation sequence

L Length of observation sequence

t Discrete time point

$\gamma_{i,j}$ Matrix element of the transition matrix Γ

Γ Transition matrix

$u(t)$ Vectors of all probabilities of observing the possible states $z \in Z$ at time t

$\mathbf{1}$ Row vector of ones

$\mathbf{1}'$ Column vector of ones (transpose of $\mathbf{1}$)

π Stationary Distribution

z (Hidden) Markov state

Z Set of distinct (hidden) states

M Number of distinct states z. It is the cardinality of the set Z in Markov chains and hidden Markov models.

V Set of distinct observations (is similar to Z in ordinary Markov chains)Symbols

Y Arbitrary statistical model, e.g., a Hidden Markov Model

α Forward parameter for computing probabilities of observation sequences

β Backward parameter for computing probabilities of observation sequences

k Boltzmann's constant

T Absolute temperature

θ Posterior distribution

w Weigthing function needed for calculating the Markov chain during umbrella sampling

q^N Cartesian coordinate configuration

ψ Wave function in the hidden space

χ Wave function of featurised input

b_{ij} Coefficient in wave funtion expansion

$\mathcal{G}$ Operator resembling a feedforward pass of some featurized input through a layer in a neural network

$\mathcal{M}$ Operator encoding featurised input into the hidden space

c_i Inverse normalizing constant

Abbreviations

MC Markov chain

HMM Hidden Markov model

EM Expectation maximization

BWA Baum-Welch algorithm

CDLL Complete-data log-likelihood

MCMC Markov Chain Monte Carlo

MSM Markov state model

MD Molecular dynamics

MHC Major histocompatibility complexes

US Umbrella sampling

TICA Time independent component analysis

TAE Time lagged autoencoder

PCA Principal component analysis

NLPCA Non linear pricipal component analysis

CV Slow collective variables

CADD Computer aided drug design

SBDD Structure based drug design

LBDD Ligand based drug design

SAR Structure activity relationship

QSAR Quantitative structure activity relationship

API Application programming interface

CDK Cyclin dependent kinase

CDK2 Cyclin dependent kinase 2

HCV Hepatitis C virus

RMSD Root-mean-square deviation of atomic positions

RMSF Root-mean-square fluctuation of atomic positions

ZIKV Zika virus

1 Introduction

1.1 Motivation

The aim of developing new software tools for drug design is to accelerate the development of novel and highly potent drug candidates. Traditionally, drug designers are using *Molecular dynamics* simulations to analyse dynamic behaviour of their targets. By developing new methods and a more profound understanding of the mechanisms of diseases and their countermeasures, this goal can be achieved on the technical level. To understand the needs of drug designers, a user survey was conducted to find out the main reasons why drug designers use dynamic modelling. The survey was prepared from a total of five questions. The questions were

1. Why do you use MD simulations in your workflow?
2. How long do you typically simulate?
3. Why do you use *Dynophores* as an analysis tool?
4. Do you use Umbrella-Sampling or other advanced sampling techniques? If so, why?
5. What do you hope for in new analysis tools?

From all the answers to all of these questions, the word cloud in Figure 1.1.1 was generated. The word cloud was generated with the Python package *wordcloud*[9].

From the word cloud in Figure 1.1.1 the following goal of building new software for drug design was conducted

> *Drug designers applying dynamic modelling applications are driven by the quest for finding models of binding interactions between proteins and ligands on the nanosecond scale.*

Finally, the mission statement for the project can now be formulated as

> *Computationally empowering drug designers.*

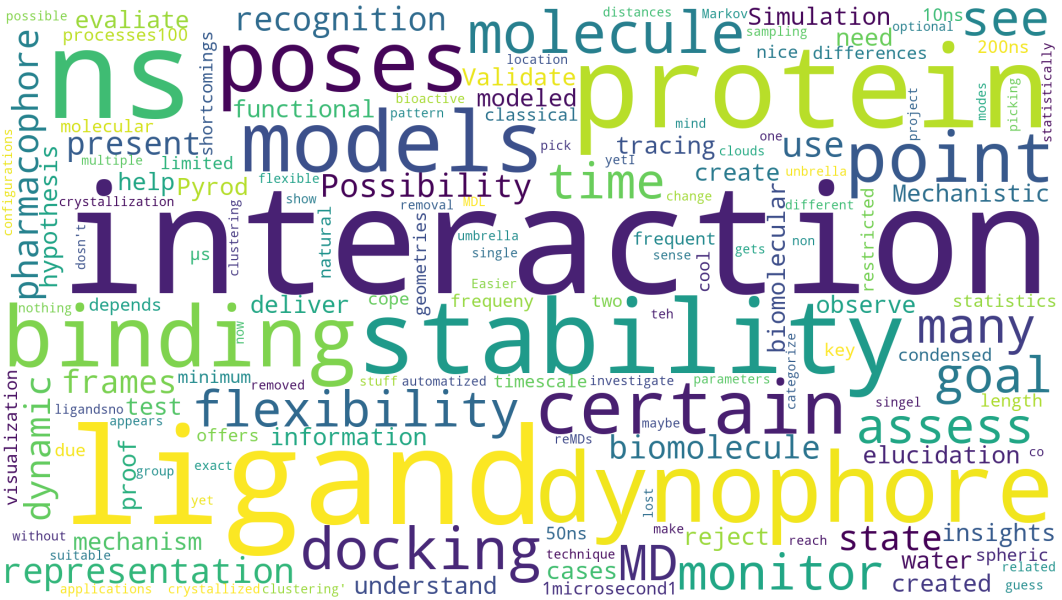


Figure 1.1.1: Word cloud of all answers given to the user survey ($N = 8$)[9].

1.2 Modelling Interactions

As *interaction* was the most prominent word from the user survey (with counting synonyms and closely related words such as pharmacophores, binding, dynophore, stability). The interactions between a protein and a ligand are by definition of quantum chemical nature. On the contrary, however, a MD simulation is usually not considered to be able to describe quantum chemical features of (weak) chemical bonds. As outlined in 1.7.3, interactions between a biomolecule and a ligand are typically deducted from the type, distances, and angles of the respective interaction partners. This is of course part of the overall Schrödinger equation of the system. However, certain features of the Schrödinger equation necessary to describe the quantum mechanical nature of the systems are missing in the calculation and the effects are at far most observed indirectly, e.g., through the Lennard-Jones and Coulomb potentials used in force fields to describe non-covalent interactions. However, even though the system is not described quantum-mechanically, the mathematical concepts behind the finding CV are strikingly similar to the wave mechanics used in quantum chemistry[10].

From the user survey analysed in Figure 1.1.1, it is evident that the need for simplifying the analysis of ligand binding modes is something the community wants and lacks sufficient tools to do so. To reduce the complexity of the model, searching for the CV of the system on top of the dynophore analysis is a promising route. Firstly, dynophores are the dynamical interaction patterns between two parts of the protein-ligand supramolecular

complex. And secondly, starting from the dynophore analysis essentially eliminates the parts *featurisation* and *dimensionality* reduction from the software pattern introduced by *MSMBuilder*[11].

Therefore, this work aims to design a prototype software that enables the drug designer to understand dynamical interactions of supramolecular complexes relevant to drug design via the Markovian analysis of molecular simulations on the ns scale by using a dynophore as the featurised input. The prototype software should then be able to extract qualitative dynamical knowledge and patterns of supramolecular protein-ligand complexes. The software is built on the assumption that the extraction of CV via hidden Markov modelling of dynophores gives meaningful and interpretable results for the drug design community. Therefore, the present work aims to verify the theory around the CV extracted by HMM[10,12–14].

Moreover, together with drug designers, the top three of the most important tasks for the drug designer were identified as:

1. *De novo* drug design

2. Theoretical validation of experimental ligand activities in quantitative and qualtitative manner

3. Understanding resistance mechanism of known pathogens

Thus, the newly developed software should support the drug designer in at least one of the aforementioned tasks. The performance against these key activities shall form as the predominant metric of success during the project.

In the subsequent section, the necessary concepts used to design the software *dylightful* able to use dynamic modelling to describe binding interactions between biomolecules and ligands as supramolecular complexes in the context of computer aided drug design (CADD) are introduced. The part *Methods* focuses on the project-management, software engineering, test system selection and simulation techniques used to design the software package *dylightful*. Each test system is introduced sharply and set in context to their purpose on testing the software package *dylightful*. The part *Results* focuses on the performance of the software *dylightful* against the test systems HCV NS3/4A complexed with *vaniprevir*, zika virus (ZIKV) complexed with a *de novo* ligand designed by *S. Pach et al.*[4] and CDK2 two systems complexed with different ligands.

1.3 Markov Models

1.3.1 Markov Property

A sequence of random variables $\{x_t : t \in \mathbf{N}\}$ is a discrete-time *Markov Chain* (MC) if it satisfies the *Markov property*

$$P(x_{t+1}|x_t, \ldots, x_0) = P(x_{t+1}|x_t) \tag{1.3.1}$$

Therefore, if the Markov property is satisfied, the random variable x_t at the time t depends solely on the random variable at the previous time–step x_{t-1}. The set of all possible random variables (states) is called state space E. The cardinality of the state space is denoted as M. The time dependence of the random variables $\{x_t\}$ can be formalized as a directed graph.

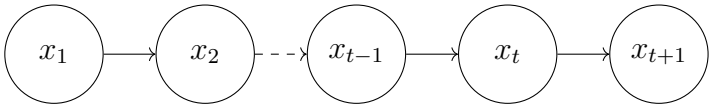


Figure 1.3.1: Visualization of a Markov chain of first order as a directed graph.

As MD simulations are done with a propagator that uses numerical integration techniques to integrate Newtons equations of motion, any given MD simulation has to be dependent of previous time step(s). The most common and by the drug design community most frequently applied integrator is the leap frog integration scheme. The integration scheme is of second order, and thus demanding the strict Markov property. Therefore, resulting simulations are Markov chains of first order.

1.3.2 Terminology

The sequence of all observations is called the observation sequence $\mathbf{o}$ of length L. The probability of transitioning from a certain state $x_t = i$ to another state $x_{t+1} = j$ is given by

$$P(x_{n+1} = j|x_n = i) \tag{1.3.2}$$

If the transition probability is independent of n, the Markov chain is called *homogeneous*. The homogeneous Markov chain is then given by

$$\gamma_{ij}(t) = P(x_{n+1} = j | x_n = i) \tag{1.3.3}$$

where $\gamma_{ij}(t)$ denotes the matrix element for the transition of state $x_t = i$ to state $x_t = j$. Finite space Markov chains satisfy the *Champman-Kolmogorov equations*. The Champman-Kolmogorov equation describes the probability of for being at state j after $t + u$ steps starting from state k

$$P(x_{n+1} = j | x_0 = k) = \sum_{z \in E} P(x_{n+1} = j | x_m = z) P(x_m = z | x_0 = k) \tag{1.3.4}$$

To prove the equation one The Champman-Kolmogorov equation can be written in the short form as follows

$$\Gamma(t + u) = \Gamma(t)\Gamma(u) \tag{1.3.5}$$

Thus, the transition probability matrix at the time t is given from the transition probability matrix $\Gamma(1)$ as

$$\Gamma(t) = \Gamma(1)^t \tag{1.3.6}$$

The transition state matrix Γ can also be vizualized as a directed graph (Figure 1.3.2).

Sometimes, the *unconditional probabilities* of observing a certain state at time t, $P(x_t = i)$ is of interest. The probabilities can be summarized as a vector

$$u(t) = (P(x_t = 1) \dots P(x_t = M)) \tag{1.3.7}$$

In some cases, the initial distribution $u(1)$ is denoted as π. In order to calculate the distribution at $t + 1$, $u(t + 1)$ on may postmultiply $u(t)$ with Γ

$$u(t + 1) = u(t)\Gamma \tag{1.3.8}$$

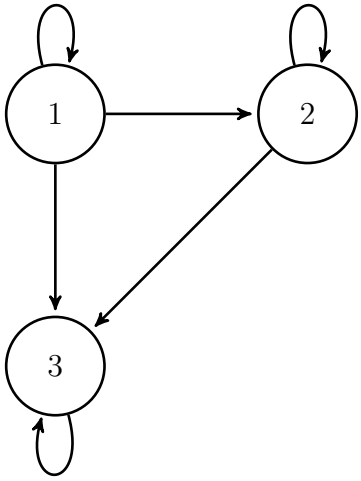


Figure 1.3.2: Transition state matrix of three different states visualized as a directed graph.

If

$$u(t) = u(t)\Gamma \tag{1.3.9}$$

and

$$u(t+1) = u(t)\mathbf{1}' \tag{1.3.10}$$

then $u(t)$ is called the stationary distribution δ. A markov chain (MC) is called *irreducible* if it is *homogenous*, *discrete in time*, and has a *finite state space*. An irreducible MC has a unique, strictly positive stationary distribution. If the MC is irreducible and aperiodic then a unique limiting distribution exists. The limiting distribution is then equal to the stationary distribution.

1.3.3 Higher Order Markov Chains

There are cases, wich require a weakened Markov property because the studied data does not satisfy the stringent Markov property. In such cases a model might be constructed for some $l \geq 2$

$$P(x_t|x_{t-1}, \ldots, x_0) = P(x_t|x_{t-1} \ldots x_{t-l}) \tag{1.3.11}$$

The Markov chain is said to be of order l. The fundamental properties of Markov chains of first order can be applied to Markov chains of higher order. The state space is then of cardinality M^l. Thus, no special theories must be derived.

1.4 Hidden Markov Models

In contrast to a simple Markov Model, in a *Hidden Markov Model* (HMM) we do not observe the states directly or fully. Therefore, also data that represents only part of the studied time series can be used for HMM modelling. Unlike in plain Markov models, in HMMs the states are rather the underlying cause for the observations. The states are thus calles *hidden states*. To formalize the theory, let L be the length of the observation sequence, $\mathbf{o}$, with $V = \{0, 1 \ldots, M-1\}$ denoting the set of distinct observations of length N. The associated probability matrix is called *emission matrix* E. Furthermore, there are M hidden states z, with Z denoting the set of distinct hidden states. The associated transition probability matrix is again referred to as, Γ and the initial state distribution is called π.

1.4.1 Problems of an HMM

Therefore, HMMs are associated with three different questions, each in need for different problem-solving techniques

- Evaluation Problem
- Decoding Problem
- Learning Problem

The *evaluation problem* asks for a given Model Y and observation sequence $\mathbf{o}$, how do we efficiently compute the probability $P(\mathbf{o}|Y)$ of observing the sequence? The *decoding problem* is about finding for a given observation sequence o the best sequence of *hidden* states $\mathbf{z}$. The *learning problem* is about how to efficiently compute the model parameters given the observation sequence $\mathbf{o}$. Therefore, we are trying to maximize $P(\mathbf{o}|Y)$.

1.4.2 Forward–Backward Algorithm

The forward–backward algorithm solves the evaluation problem. The algorithm needs to compute the forward parameter $\alpha(t)$ and the backward parameter $\beta(t)$.

Given an observation sequence $\mathbf{o}$ at time t consider the probability $P_{\mathbf{o},t}$ of observing that sequence given an HMM Y. The HMM is a function of M, δ, Γ, and E, such that Y(M, δ, Γ, E). $P_{\mathbf{o},t}$ may then be calculated as

$$P_{\mathbf{o},t} = \pi P(x_1)\Gamma P(x_2)\dots P(x_t)\mathbf{1}' \tag{1.4.1}$$

For a proof, see *Zucchini et al.*[15]. To derive the Forward algorithm, we calculate a_t for $t = 1, 2 \dots t$ as follows,

$$\alpha_t = \delta P(x_1) \prod_{i=2}^{t} \Gamma P(x_i) \tag{1.4.2}$$

The forward probabilities are joint probabilities for all time steps at the time t and the states $z \in Z$.

$$a_t(z) = Pr(X^(t) = x^(t), Z_t = z) \tag{1.4.3}$$

For the backwards probabilities β_t at times $s = 1, 2 \dots t$ follows therefore in a similar manner

$$\beta_t' = \left(\prod_{i=s+1}^{t} \Gamma P(x_i) \right) \mathbf{1}' \tag{1.4.4}$$

As opposed to the forward probability $a_t(z)$, $\beta_t(z)$ is a conditional probability because $\beta_t(z)$ identifies the probability of observing the prefix $x_i \dots x_t$ given that the MC is in state z at time $t = i$.

$$\beta_i(z) = P(X_{i+1} = x_{i+1}, X_{i+2} = x_{i+2}, \dots, x_t = x_t | Z_i = z) \tag{1.4.5}$$

$$\beta_i(z) = P(\mathbf{o}_t | Z_i = z) \tag{1.4.6}$$

Therefore, b_i' can be written as

$$\beta_i' = \Gamma P(x_{i+1}) \beta_{t+1}' \tag{1.4.7}$$

Now, given the forward probabilities a_i and β_i, the procedure for calculating the probabilities $P(X^{(t)} = x^{(t)}, Z_t = z)$ follows immediately

$$\alpha_i \beta_i' = P(\mathbf{X}^(t) = \mathbf{o}^{(t)}) = L \quad \forall i \tag{1.4.8}$$

With L_t denoting the likelihood of observing $\mathbf{o}^{(t)}$. Note, however, that L_t might also be computed via

$$L_t = \alpha_t \mathbf{1}' \tag{1.4.9}$$

The algorithm introduced in 1.4.9 would only need a single pass and the forward algorithm.

1.4.3 Viterbi Algorithm

The Viterbi algorithm solves the decoding problem. The Viterbi algorithm asks for the maximum posteriori estimate for the most likely sequence of hidden states $\mathbf{z}$, called *Viterbi path*. Thus, the Viterbi algorithm aims to find

$$\text{argmax}_{\mathbf{o}} = P(z|\mathbf{o}) \tag{1.4.10}$$

The probability $P(z|\mathbf{o})$ denotes the probability of the state sequence z given the observations $\mathbf{o}$. The Viterbi algorithm makes use of dynamical programming and makes use of log probabilities to avoid numerical instability of multiplications with regard to over- and underflows. The Viterbi uses the HMM as an input. Thus, the Viterbi algorithm needs the *emission probalilities*, *transition probabilites*, and the *initial probabilites* as an input.

1.4.4 Baum–Welch algorithm

The Baum-Welch algorithm (BWA) algorithm solves the learning problem. The BWA algorithm belongs to the class of expectation–maximization (EM) algorithms. The expectation maximization (EM) algorithm is an iterative method for performing maximum likelihood estimations when part of the data is missing. The EM algorithm makes use of the assumption that the complete-data log-likelihood (CDLL) may be straightforward to maximize, even if the likelihood of the observed data is not possible to compute. The complete-data log-likelihood (CDLL) refers to the log-likelihood of all parameters of interest. Therefore, the CDLL refers to non-observed states of the HMM.

The BWA consists of two major steps. After initialization of the parameters of interest, the BWA performs the following steps

- **E step**: Computing of the conditional expectations of the missing data that appear

in the CDLL

- **M step**: Maximize CDLL regarding conditional expectations of the parameters of interest

Therefore, the BWA combines the Forward-Backward–Algorithm and the Viterbi algorithm. According to modern implementations, the BWA algorithm scales exponentially with the number of hidden states[16].

1.5 Markov–Chain–Monte–Carlo

To solve the EM problem more efficiently, and numerically more stable, one might apply markov Chain Monte Carlo (MCMC) methods. The MCMC relies on Monte Carlo methods to solve the EM problem and is thus more of statistical nature. However, in contrast to the BWA, the MCMC method relies on sufficient random sampling methods. An algorithm that learns the transition probability distribution (matrix) Γ where n denotes the steps taken by the algorithm

$$\lim_{n\to\infty} \pi = \Gamma \tag{1.5.1}$$

is thus at the heart of MCMC methods.

1.5.1 Metropolis–Hastings algorithm

One popular algorithm to learn different probability distributions is the Metropolis–Hastings algorithm[17]. The Metropolis–Hastings algorithm is based on two assumptions:

1. There exists a stationary distribution
2. The stationary distribution is unique (ergodicity of the Markov chain)

The algorithm needs a sampling distribution g that could be the normal distribution and an estimate of the targeted probability distribution Γ. The sample distribution g is often called proposal distribution. The algorithm works best if $\Gamma \approx g$, such that the shapes of both distributions are similar. An example for similar shapes could be that both, Γ and g are normal distributions, but the normalizing constant for Γ is not known. The algorithm is then divided into two parts

1. Initialization

2. Iteration

During the initialization, the initial value(s) are picked, e.g., for Markov modelling it could be from the initial probability distribution. In Markov modelling, we would pick an initial *state* and thus subsequently, the variables of interest are referred to as the *states*.

During the iteration procedure, first a new candidate x_t is picked from the sampling distribution g. Then subsequently, the acceptance parameter α is calculated from the target and the proposal distribution given the most sampled value x_n as follows

$$\alpha = a_1 a_2 \tag{1.5.2}$$

Where x' is the most recent sampled value an a_1 is given as

$$a_1 = \frac{\Gamma(x')}{\Gamma(x_t)} \tag{1.5.3}$$

On the contrary, a_2 is given as

$$a_2 = \frac{g(x_t|x')}{g(x'|x_t)} \tag{1.5.4}$$

The new state is then chosen according to the user-defined acceptance criteria:

1. If $\alpha >= 1$

$$x_{n+1} = x' \tag{1.5.5}$$

2. Else x_{n+1} is sampled from a uniform distribution with the probabilities

$$x_{n+1} = \begin{cases} x', & \alpha \\ x_n, & 1-\alpha \end{cases} \tag{1.5.6}$$

The sampling is Markovian, as the acceptance criterion is based on the previous sample x_n. The Markov criterion holds for many derived MC, too.

1.6 Markov State Models

The term markov state model (MSM) is used frequently in the MD community to describe the Markov analysis of MD trajectories. Thus, MSM are applied to analyse time series of dynamical systems. For example, in the analysis of peptide unbinding from the major histocompatibility complexes (MHC) complex, MSM models are applied. Another approach would utilize coarse-grained models, which are a multiscale simulation approach. The advantage of using MSM modelling is to provide for an atomistic simulation picture. The protocol included cutting-off the MHC complex to 190 residues, Langevin integration, increasing the hydrogen masses to 4 atomic mass units, simulating at a time step of 4 fs, and equilibrating the system for a time of 500 ns. Therefore, to deduct relevant states during the unbinding process of the peptide, many simplifications are derived to be able to perform enough molecular simulation to have enough samples to calculate the *MSM* model. The need of sufficient sampling is also true for other systems. Moreover, the example shows that the trajectory is not important, but rather the sufficient sampling of relevant states. Thus, MC or hybrid methods should be the preferred modelling for MSM analysis.

1.6.1 Umbrella sampling

A common technique to solve the simulation time problem for MD simulations, is to use advanced sampling techniques. The most common one is umbrella sampling (US). The method was first described by *G. M. Torrie* and *J.P Valleau*[18] and is based on *Metropolis*[17] simulations. The system is not sampled equally, but rather energies important to the properties of the system are favoured during the MC sampling. For example, in the original publication, the authors evaluate the free energy difference as follows

$$\frac{A}{kT} - \frac{A_0}{kT} = -\ln \int f_0(\Delta U^*) \exp(-\Delta U^*) d\Delta U^* \tag{1.6.1}$$

To evaluate the integral it is necessary to introduce a MC with the limiting distribution of a configuration q'^N is calculated according to

$$\Gamma(q'n) = \frac{w(q'^N) \exp\left(-\frac{U_0(q'^N)}{kT_0}\right)}{w(q'^N) \exp\left(-\frac{U_0(q'^N)}{kT_0}\right)} \tag{1.6.2}$$

The weighting function w favours only the most relevant configurations for evaluating the integral in equation 1.6.1.

1.6.2 Time Independent Component Analysis

The time independent component analysis (TICA) refers to a method of dimensionality reduction in statistical modelling of time series. TICA is similar to principal component analysis (PCA). PCA focuses on finding the most significant linear combination of the input parameters. However, in contrast to PCA, TICA searches high autocorrelation features in the input degrees of freedom. If applied to a simulation, TICA thus finds the eigenvalues of the applied propagator. TICA may be stacked with clustering algorithms such as the *kMeans* algorithm to construct an MSM[19,20]. Thus, the aim is to learn or find the slow collective variables of the dynamical system. Yet, in practice, according to *Wehmeyer et al.* the input space has to be featurised[14]. Therefore, proper features have to be defined by the user, e.g., atom distances, or torsion angles between residues of high interest. As for the dynamic analysis of interaction patterns in supramolecular complexes, the natural choice for the featurised input, is to choose a time resolved interaction pattern parsed in bit vectors.

1.6.3 Time Lagged Autoencoder

In contrast to TICA, the time lagged autoencoder (TAE) learns the hidden feature space according to the classical encoder-decoder pattern known from artificial intelligence. The encoder-decoder pattern aims to perform non-linear PCA and is, therefore, termed as non linear pricipal component analysis (NLPCA). The NLPCA method makes use of artificial neural networks with three hidden layers. The central hidden layer is the so-called *bottleneck* layer. By training the neural network to first encode, and then decode the provided data through the *bottleneck* layer, it is assumed that the *bottleneck* layer learns the hidden compact representation of the provided data[21]. The encoder is usually optimized with respect to a linear regression goal[22]. Applied to time series in general, or MD simulations in particular, such an encoder-decoder pattern is known as TAE[14]. Therefore, TAE is doing non–linear variational principle. Moreover, by training a TAE one aims to find the CV of the system without having to think about proper input features to perform NLPCA. The approach of obtaining the CVs via HMM is essentially similar to the realization of the *variational principle* of observables[12]. To prove the equivalence of TAE and doing variational principle, it was essentially hypothesized by *Wehmeyer et al.* that TAE is equivalent to non-linear variational principle[14]. It was shown by *Wehmeyer at al*[14] TAE is finding the eigenvalues of the *Hamilton* operator H of the dynamic system through the observables $\mathbf{o}$. As TAE can find the hidden representation of the time-series for non–linear

cases, too, it must by analogy perform non-linear variational principle. Still, *Wehmeyer et al.* did not provide any proof[14]. According to *Clementi et al.* in the linear variational principle, a CV is defined as

$$\psi_i = \sum_j b_{ij}\chi_j(\mathbf{r}) \tag{1.6.3}$$

Whereas ψ_i denotes the eigenfunctions of the Transfer operator $\mathcal{T}$ of the system which are equivalent to the CV and the time-independent Hamilton operator[12]. Moreover, b_{ij} denotes the coefficient of the linear combination of input parameters, $\chi_j(\mathbf{r})$. Let $\mathcal{G}$ be the operator of passing input through an arbitrary neural network layer. If TAE was doing linear variational principle, then for $\mathcal{G}_a$ denoting a single neuron it must always be true that

$$\mathcal{G}_a(b_1\chi_1 + b_2\chi_2) = b_1\mathcal{G}_a\psi_1 + b_2\mathcal{G}_a\psi_2 \tag{1.6.4}$$

However, as $\mathcal{G}_a$ is using the RelU activation function, a simple example demonstrates that $\mathcal{G}_a$ must be a non-linear operator. Consider $\psi_1 = \psi_2 = \mathbf{1}$, $b_1 = -1$ and $b_2 = 1$ such that

$$\mathcal{G}_a(-1\psi_1 + 1\psi_2) = \mathcal{G}_a(-1) + \mathcal{G}_a(1) = 0 + 1 = 1 \tag{1.6.5}$$

But if $\mathcal{G}_\infty$ was linear, it would also be true that the following expression is equivalent to 1.6.5, but

$$\mathcal{G}_a(-3\psi_1 + 3\psi_2) = -\mathcal{G}_a\mathbf{1} + \mathcal{G}_a\mathbf{1} = -1 + 1 = 0 \neq 1 \tag{1.6.6}$$

Thus because any layer consists of many $\mathcal{G}_a$ that interplay in a linear combination to every neuron in the next layer, $\mathcal{G}$ must be a non–linear operator. ■

Now, as the encoding into the hidden space of the HMM respectively TAE could be expressed as the operator $\mathcal{M}$

$$\mathcal{M} = \prod_i \mathcal{G}_i \tag{1.6.7}$$

the operator of $\mathcal{M}$ encoding into the hidden space must be non–linear, too. However, the formulation for a featurised observation $\sum_j b_{ij}\chi_j$ encoded in the hidden space is then

essentially

$$\mathcal{M} \sum_j b_{ij}\chi_j = \mathcal{M}\psi_i = m_i\psi_i \tag{1.6.8}$$

With m_i denoting the probability of obtaining the hidden representation ψ_i. The further argumentation lends from standard quantum mechanical reasoning. Assuming χ_j are orthogonal, multiplying from left with the complex-conjugate of the wave functions results

$$\langle \sum_j b_{ij}\chi_{ij} | \mathcal{M} | \sum_j b_{ij}\chi_j \rangle = \langle \psi_i | \mathcal{M} | \psi_i \rangle = c_i m_i \tag{1.6.9}$$

with c_i as the inverse normalizing constant. Normalizing and minimizing according to the variational method known from quantum mechanics results then

$$\frac{\langle \sum_j b_{ij}\chi_j | \mathcal{M} | \sum_j b_{ij}\chi_j \rangle}{\langle \sum_j b_{ij}\chi_j | \sum_j b_{ij}\chi_j \rangle} = \frac{\langle \psi_i | \mathcal{M} | \psi_i \rangle}{\langle \psi_i | \psi_i \rangle} \geq m_{i,0} \tag{1.6.10}$$

with $m_{i,0}$ denoting the true probability of observing the hidden state ψ_i from the input $\sum_j b_{ij}\chi_j$ given the encoding operator $\mathcal{M}$. Thereby, on χ_j, any neural network that is minimizing weights to find some hidden representation of the input data, and especially TAE is doing non-linear variational principle. Deep learning and in particular TAE are thus generalizing TICA. ■

As elaborated in section 2.1, *Dynophores* can provide χ_j that are orthogonal. The hypothesis about the property of TAE performing non–linear variational principle is supported by the fact that TAE is the abstraction and improvement of the NLPCA algorithm. Throughout the *test systems*, the hypothesis shall be further validated on different examples relevant to the drug design community. Another advantage of TAE against classical EM methods lies in the algorithmic details of TAE. Because of the general architecture of an autoencoder shown in Figure 1.6.1, the autoencoder structure shown does at worst scale polynomially with the number of input features[23] (for the N–2–N case). The polynomial scaling is due to the necessity of training the neural network. A trained network would be expected to have a linear time complexity. Moreover, the TAE infrastructure is expected to converge to the true solution for very large autoencoder sizes[24]. The good time complexity of TAE thus makes the algorithm more suitable for the analysis of biological systems, as the dynamics of a protein–ligand complex is expected to be diverse and complicated.

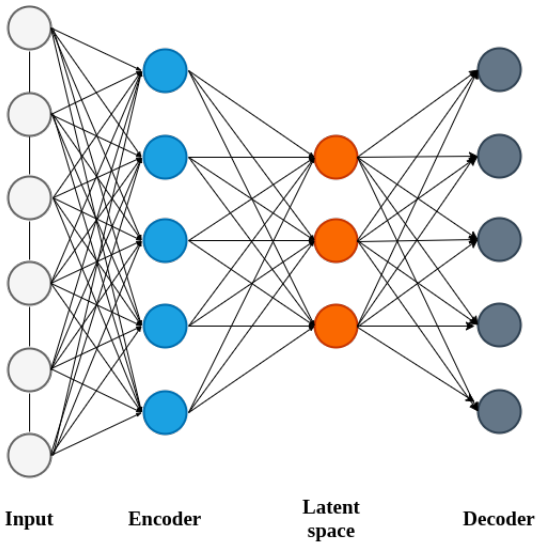


Figure 1.6.1: General encoder-decoder pattern applied to time series found in TAE architectures

1.7 Computer Aided Drug Design

CADD is a valuable tool in drug development. The major goal of CADD is always to save time and money during the drug development process. Therefore, the development of novel concepts and computational methods for reducing the experimental workload during the drug design. CADD thus decreases the costs and time needed to get a drug to the market. Moreover, also the quality of a given drug is increased for the individuals consuming the drugs by applying computational methods during drug development. In the design of small molecules, CADD differentiates into ligand based drug design (LBDD) and structure based drug design (SBDD). LBDD focuses on methods and knowledge derived from known active compounds, e.g., structures of antivirals. From a set of known active ligands to a given target, structure activity relationship (SAR) relations can be derived. SAR are statistical relations between structures and physicochemical properties that. SAR models help to design more active compounds given the knowledge extracted from the statistical model. On the contrary, SBDD focuses on the underlying biological macromolecule and does not require knowledge of ligands.

1.7.1 Pharmacophores

In CADD the pharmacophore concept is essential. The generation of pharmacophores may either be from the pool of LBDD, e.g, feature based[25] or from the pool of SBDD methods,

e.g, docking[26]. The pharmacophore was introduced by *Paul Ehrlich* in the early 20th century. *Ehrlich* describes the quest for finding novel chemical compounds to fight diseases of any kind as a quest to learn how to chemically aim ("[...]chemisch zielen lernen")[27]. *Paul Ehrlich* was building his argument on *neurotropic* or *parasitotropic* substances, to describe the property of a chemical compound to specifically target and exhibit its properties at neurons or parasites. A neurotropic substance could for example specifically colour neurons. Then, after building up his argument, *Ehrlich* introduces the *toxicophores* to describe the chemical features that give rise to toxicity of a chemical compound. By analogy, a *pharmacophore* carries the features that are responsible for the biological activity of a compound. The IUPAC definition reads "A pharmacophore is the ensemble of steric and electronic features that is necessary to ensure the optimal supramolecular interactions with a specific biological target structure and to trigger (or to block) its biological response"[28].

The pharmacophore concept is especially important to virtual screening methods. Virtual screening on a pharmacophores can give rise to novel structures with different chemical core structures but similar biological activities. The effect is known as "scaffold hopping", and was introduced in 1999 by *Schneider et al.*[29] The term explains the phenomenon that most biological active compounds exhibit their activity mostly through attached functional groups rather than their core *scaffold*. The analysis of the scaffolds and the computational quest to perform scaffold "hoping, leaping, or crawling" as termed by *Bajorath et al.* can give rise to novel structures and is a powerful tool in CADD[30].

In Figure 1.7.1 a pharmacophore of the $ZIKV^{Prot}$ is shown with the corresponding ligand that was used to generate the pharmacophore during a MD simulation[4]. The yellow blobs correspond to *hydrophobic* interactions, the red blob with the red arrows to a hydrogen acceptor and the blue coordination spheres show a positive ionizable area.

To make predictive use of pharmacophores, recent work from *Kohlbacher et al.*[31] and *Rahman et al.*[32] shows that quantitative structure activity relationship (QSAR) modelling on pharmacophores is a useful tool in CADD.

1.7.2 Data structure

The data structure of pharmacophores from LigandScout 4.4 is typically saved in the PML/PMZ format. The PML/PMZ format is containing information about the ligand interactions with the biomolecule. The PML/PMZ file format is a derivation from the XML file format, fitted for the chemoinformatic workloads experienced in the CADD field. Each

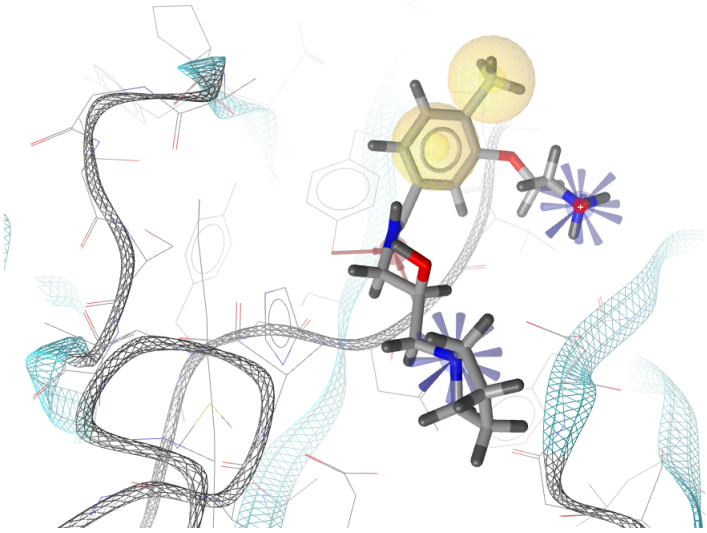


Figure 1.7.1: Pharmacophore of the ZIKV^{Prot} using LigandScout.

feature is described by the interacting atoms of the ligand, as well as the interaction type. The PML/PMZ file format is predominantly used in the LigandScout software[33], but is easy to be parsed because of its derivation from the XML file format.

1.7.3 Dynophores

Dynophores are generated via the LigandScout application programming interface (API)[26] by considering an MD simulation as an input and performing the pharmacophore analysis on each frame. Thus, over the simulation, an interaction pattern is created. Therefore, the shortcoming of pharmacophores only observing a single time point or a static state is overcome by including the time domain and dynophores can thus give a more complete picture of the interaction in supramolecular protein–ligand complexes. The data structure of the dynophore is on the abstract level similar to the data structure of a pharmacophore. However, the data structure of the dynophore contains more detailed information about the features of the pharmacophore that make up the dynamic interaction pattern. The important data features for the interaction analysis performed by *dylightful* are the interaction partners termed *envpartners*, the frame indices (time points in the simulation), and the respective *superfeatures*. The *software design* of the dynophore workflow is shown in Figure 1.7.2. The *Dynophore* software first generates pharmacophores for each time frame after reading in the ligand definition and the MD-trajectory. The output is then stored in

a *.json* and a *.pml* file. The visualisation is done by the *DynophoreApp* using the generated output files.

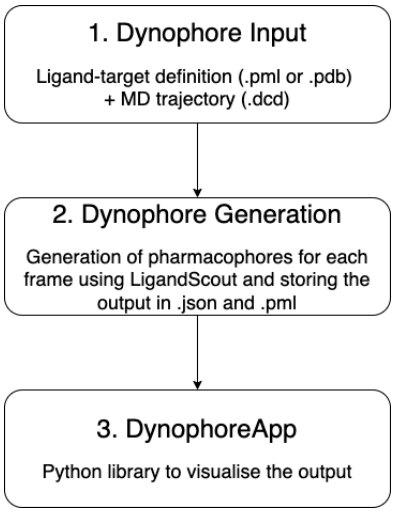


Figure 1.7.2: Software pattern of the dynophore workflow[2].

The dynophore is then visualized as a 3D-histogram inside the binding site. The dynophore generated from the $ZIKV^{Prot}$ simulated with the ligand **427__1**[4] is shown in Figure 1.7.3

The generated dynophore shown in Figure 1.7.3 depicts that the protein–ligand complex changed conformations during the simulation as the point cloud is concentrated across different regions in the euclidean space, but the point clouds are not all near the ligand.

As TAE requires featurised inputs to make extraction of useful CVs[14], time resolved dynophores could offer a valuable featurisation as the input is:

1. Focusing on the intramolecular interaction
2. Can be represented as orthogonal bit vectors that are necessary for TAE to work optimally (compare 1.6.3

Therefore, *Dynophores* are used as the featurised inputs for the further work.

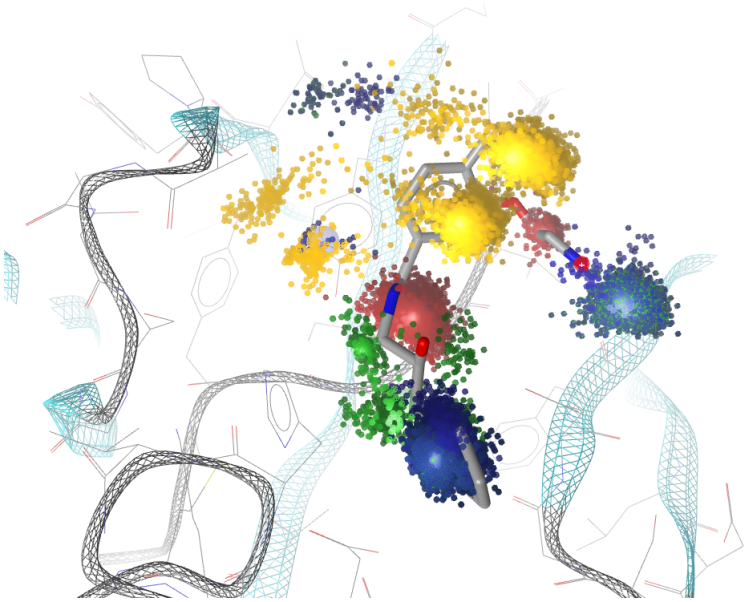


Figure 1.7.3: Dynophore of the ZIKV^{Prot} using the Dynophore software[2].

2 Methods

2.1 Software Engineering

As outlined in section 1.1, the CVs are most likely hidden from the simulation input, applying HMMs instead of plain Markov models to uncover the eigenfunctions of the system is the better choice. As outlined in 1.6.3 finding the collective variables is essentially the same as learning a hidden representation of smaller dimension with TAE. On the contrary, the eigenfunctions may work on different timescales and therefore, the Markov property is violated[12]. Thus, instead of classical HMM modelling, TAE is used. The choice for using TAE is further motivated by the equivalence of TICA and HMM in suitable cases[13]. Moreover, it was shown that TAE and TICA are equivalent for the linear case[14]. Finally, it was shown in 1.6.3, that TAE is essentially doing non-linear variational principle on orthogonal featurised input that is provided by *Dynophores*.

To make the output usable for subsequent virtual screening experiments, regression, and classification, constructing a Markov graph was identified as a usable data output from the software. Alongside the Markov graph, the software produces plots visualizing the interaction analysis.

2.1.1 Software Design

From the user survey (compare section 1.1) the goal of designing new methods for CADD is to make the elucidation of the interaction (binding modes) of a given protein-ligand complex easier. Further, as shown with *Dynophores*[2], the interactions between ligands and proteins in supramolecular complexes are more than single snapshots but are only valid throughout a time series. Thus, any new tool has to achieve the synthesis of a dynamic interaction perspective within the whole protein–ligand complex and CADD. To achieve the goal, the software shown in Figure 2.1.1 was designed.

In the software workflow, the central part is extracting interactions from all perspectives of the interaction partners. There is thus the perspective from the ligand, the mixed

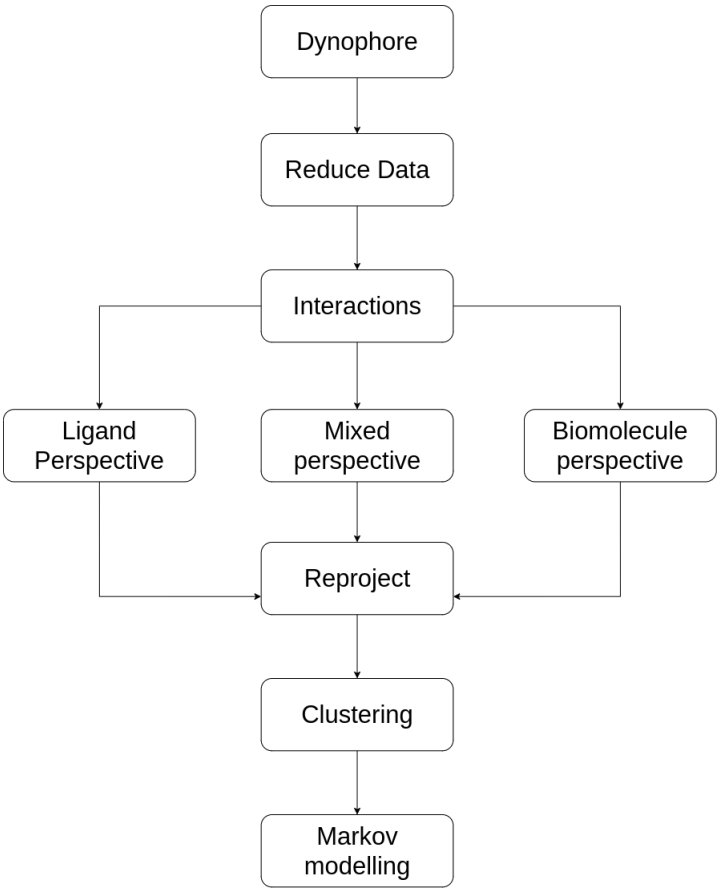


Figure 2.1.1: Software design based on the user survey in Figure 1.1.1.

perspective, and the perspective from the biomolecule. All can be analysed to gain different levels of understanding from different aspects of the interaction, whereas the only complete picture would come from the mixed perspective. The separation into the three perspectives of the ligand, the protein, and the complex are inspired by the LigandScout software[26,33]. Analysing the mixed picture is a necessity directly derived from the fact that in reality there is no system at rest and each interaction of the protein-ligand complex is on the same timescale of their intramolecular vibrations. Moreover, because of the mixed interaction terms in the force fields used in MD simulations demand the mixed picture for completeness, too.

The new software design thus profoundly builds on the concepts introduced by *Dynophores*. Therefore, choosing a dynophore as an input seems natural, but is not mandatory. After extracting the corresponding interactions, one can perform the reprojection into the Hilbert space of the CV via TAE and cluster the obtained CVs to construct a Markov model. The aforementioned pipeline is a standard procedure for producing MSM and is described, for example, in the documentation of *MSMBuilder*[11]. However, the software pattern differs from *MSMBuilder*, as the novel pipeline does not branch out HMM analyses from the workflow but rather uses the HMM modelling via TAE as the central part of the workflow. Moreover, different perspectives for extracting the CV are considered as useful but not as mandatory for automatizing the feature extraction. Automatizing the feature extraction was the main purpose of the study behind the usability of TAE by *Wehmeyer et al.*[14] The other parts of the pipeline are indeed equal to *MSMBuilder*[11].

Furthermore, by abstracting almost all algorithmic work up to selecting the number of hidden states that are obtained from the analysis, the user is given algorithmic control without having to think about writing algorithms and do machine learning and statistical modelling themselves. Therefore, the software abstracts the package *deeptime* further because *deeptime* aims to make the time series analysis for machine learning engineers easier[34]. To systematically build reliable tools for the drug design community, project management techniques from the *Lean* movement were applied. These included *customer development* and feedback loops throughout the whole process of developing the software, not only through surveys but also through *close contact* with the users, *test-driven development*, *work-as-it-progresses kanbans*, and *continuous integration*[35].

2.1.2 Software Implementation

The software was developed in pure Python with the aim of abstracting the process shown in Figure 2.1.1, which is already a simplification but at the same time, also a derivation of the process used in *MSMBuilder*[11]. Therefore, the patterns shown in Figure 2.1.2 were derived The abstracted software pattern was then implemented for protein–ligand

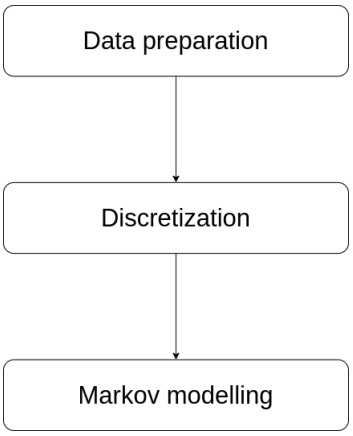


Figure 2.1.2: Abstracted software pattern derived from the pattern shown in Figure 2.1.1.

interactions as shown in Figure Figure 2.1.3 and implemented as Python modules in the

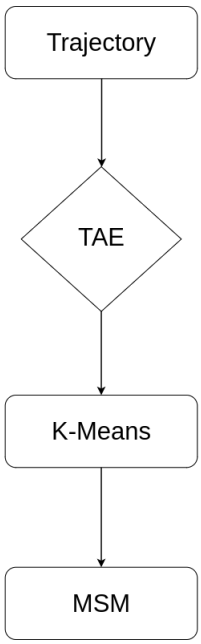


Figure 2.1.3: Implementation of the software pattern shown in Figure Figure 2.1.2 from the patterns shown in Figure 2.1.1.

dylightful package. The package is hosted on GitHub and the coverage according to the Python package code-cov[36] is 100 %. The package contains the modules

1. *parser.py*
2. *msm.py*
3. *discretizer.py*
4. *utilities.py*
5. *bar_plot.py*
6. *mdanalysis.py*
7. *metrics.py*
8. *postprocess.py*
9. *preanalysis.py*

The package is fully extendable, and the object patterns and the implemented CI-pipeline allows for rapid prototyping of new algorithms. The interface is easy to use and can be easily included into other packages or workflows. The code and especially the modules are documented using the Python package *sphinx* and the documention is hosted on https://dylightful.readthedocs.io/en/latest/. The documentation is corroborated with examples provided as Jupyter notebooks. Jupyter notebooks were chosen as the main user interface because all users participated in the interviews and feedback loops were familiar and regular users of Jupyter notebooks. The dependencies of the package are

1. deeptime[34]
2. MDAnalysis[37–39]
3. scikit-learn[40,41]
4. numpy[42,43]
5. pandas[44,45]

Furthermore, even though not necessary for the build, during the development, *hmmlearn*[46] was used.

2.1.3 Test Systems

Alongside with the drug designers, the three most important problems for the drug design community were identified as

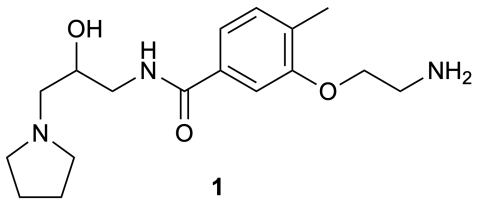
1. *De novo* drug design
2. Theoretical validation of experimental ligand activities in quantitative and qualitative manner
3. Resistance mechanism in pathogens

To test and enhance the performance of *dylightful* on these key tasks, the following test systems were chosen for each task respectively

1. ZIKV$^{\text{Prot}}$
2. CDK2
3. HCV NS3/4A protease

For the software package *dylightful* it is thus mandatory to be able to support the drug designer in at least one of these issues. Thus, to test the main functionalities and capabilities of the software package, three test systems were chosen. The first test consists of a simulation of the ZIKV$^{\text{Prot}}$ simulated with the ligand **427_1** shown in Scheme 2.1.1. The second test system investigates three different ligands complexed with human CDK2. The third test is about mutations of HCV NS3/4A protease that was mutated at specific residues.

2.1.4 Zika Virus Protease



Scheme 2.1.1: Ligand **427_1** for the ZIKV$^{\text{Prot}}$ used in the MD simulation[4].

The aim of the test of *dylightful* with ZIKV$^{\text{Prot}}$ complexed with compound **1**, is to see if the implemented software pattern shown in Figure 2.1.3 reproduces the visual inspection of domain experts.

2.1.5 Cyclin-dependent Kinase 2

Another test-system consists of the cyclin dependent kinase (CDK)2. The CDK2 is a kinase that has a critical role of the late M-Phase in the cell cycle[47–49]. CDK2 is a controversial drug target for cancer therapies, and the full mechanism of inhibition is not known. Moreover, CDKs show high structural similarity upon activation in their binding pocket and are thus difficult to target selectively[50]. As shown by *Jianzhong Chen et al.* from a dynamic picture, more insights might be derived[51] and better inhibitors can be designed. The publication shows that the mechanism of inhibition controls the slow collective variables of the dynamic motion of CDK2[51].

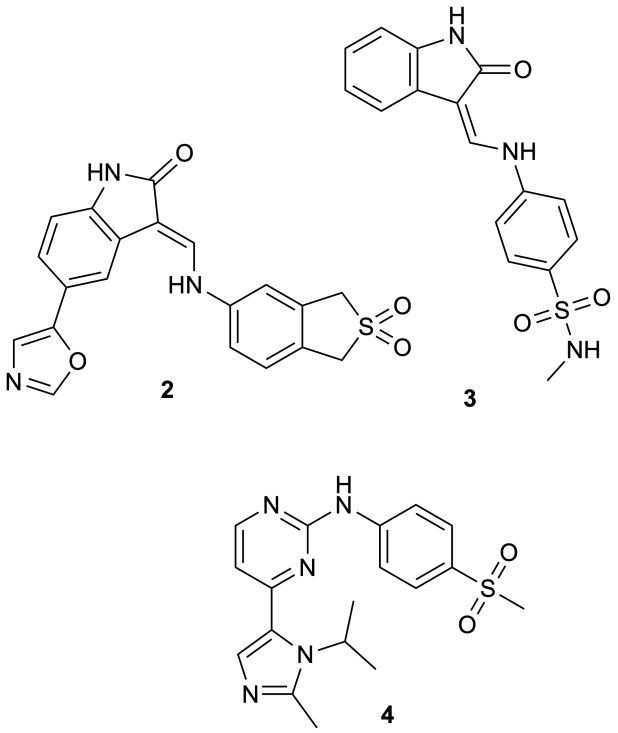
Moreover, it was shown that the more holistic study of CDK2 complexed with Cyclin A can lead to selective inhibitors[6]. For the test of the *dylightful* package, the crystal structure 1KE7 of CDK2 cocrystalized with an oxoindol based inhibitor was chosen[5] as a reference study.

The simulation protocol for the CDK2 system consisted of

1. Structure preparation in MOE[52] with remodelling
2. Structure preparation in Maestro[53]
3. Preparation of the minimal simulation box under physiological conditions with the TIP4P water model
4. Five replicas in the NPT ensemble for 100 ns using Desmond parametrized with the OPLS2005 force field[54,55] on Nvidia GTX 2080Ti graphic cards
5. First post-processing step using lab intern scripts to extract *.dcd* trajectories and start frames of the different replicas as *.pdb*
6. General analysis using the Python package *mdtraj*[56]
7. Dynophore generation using the *Dynophores* software in the unoficcial version 0.88[2]

The structure was prepared using the MOE[52] and Maestro[53] software packages according to physiological conditions. To test the *dylightful* software package against the newest finding of the literature regarding the dynamic mechanism of inhibition[6,51], the structures of 1KE5, and 6GUH were studied. Homology modelling for incomplete structures was done in MOE[52] using the standard settings from the structure preparation and the loop modeller. For each missing sequence, the best fit was chosen based on structure similarity and visual

inspection. However, each sequence was modelled equally for all studied systems. The crystal structures may have differed significantly in around the regions of the not modelled flexible loops 37-43 and 153–162. To make the simulations somewhat comparable, the influence of the homology modelling on the residues 37-43 and 153–162 was examined. However, up to now, a systematic tool to perform a dynamic analysis of the CDK2 system is lacking. Thus, to further test the capabilities of the software package *dylightful*, SAR is performed on the most prominent CV with three different ligands from the crystal structures 1KE7, 1KE5, and 6GUH[6].

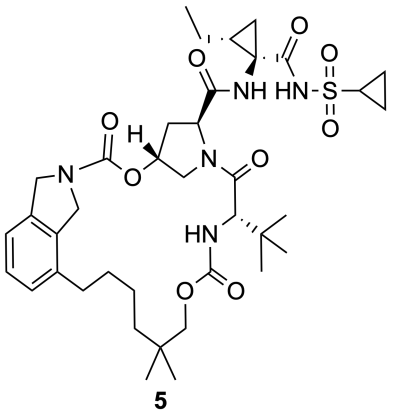


Scheme 2.1.2: Inhibitors for different CDK2 systems (from left to right 1KE7, 1KE5, 6GUH).

2.1.6 Hepatitis C Virus Protease

To test, if the behaviour of a system totally different from CDK2 can be analysed using the *dylightful* package, the HCV protease N3/4A was chosen as another test system. The test system HCV N3/4A protease complexed with *vaniprevir* investigates if the software package can distinguish patterns introduced by mutating the protein. The structures used to perform, and the data used to verify the modelling came from a study conducted by *Schiffer et al.*[7] and as an application, the software *dylightful* using the data made available

through the resistance study by *Schiffer et al.*[7] could give insights on how different mutations affect the dynamics of the protein–ligand system. The knowledge about the effect of target mutations on the dynamic protein–ligand interaction might then give insights into resistance mechanisms of the HCV pathogen.

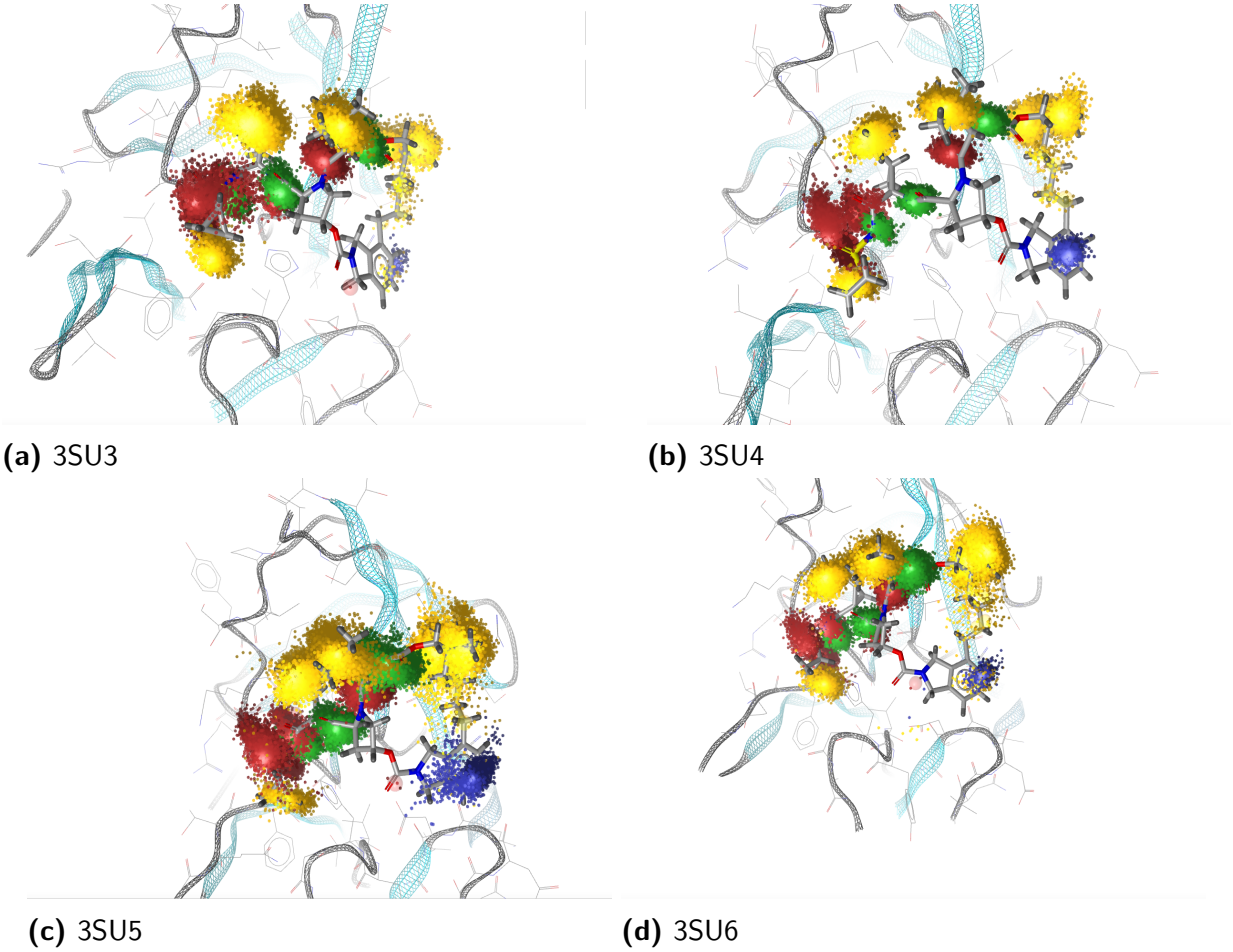


5

Scheme 2.1.3: Lewis structure of vaniprevir.

Knowledge about the resistance mechanism of the HCV protease is critical for the development of new antivirals as well as the sufficient treatment of infected hosts, as the virus mutates rapidly[57]. The high frequency of mutations in HCV is due to an error-prone RNA-polymerase, which additionally lacks proofreading capabilities. Therefore, the genetic variety of HCV in an infected host is increased during the infection[58]. The study on the atomistic resistance mechanisms by *Schiffer et al.*[7,8] revealed that firstly, the dynamics of the ligand plays an important role for the susceptibility of a mutation to the ligand activity, and secondly that there are traceable mutations that may interrupt the key cation–π–interaction at the R155 residue. According to *Schiffer et al.*[7], the first mutation, R155K, inhibits the interaction directly. The second mutation, D168A, disrupts the interaction with R155, by shifting the position of R155 for optimal cation–π–intreraction with vaniprevir. The A146T mutation shifts the P2 moiety of vaniprevir to the catalytic triad of the NS3/4A protease. Thus, according to *Schiffer et al.*, all mutations manifest their disruption of the cation–π–interaction from vaniprevir at the R155 residue in different ways. And as expected, from the atomistic analysis[7,8], the activity of vaniprevir gets reduced significantly. The activities for the different complexes 3SU3, 3SU4, 3SU5, and 3SU6 are 0.75 nm, 554 nm, 2635 nm, and 958 nm, respectively. Yet, from the experimental values, there are severe differences between the mutations. Hence, from running the test with the *dylightful* software package, it is expected that the dynamic analysis is entirely different in some, but at the same time very similar in other features. The MD simulations were run

in a similar fashion to the CDK2 (2.1.5) and ZIKV system[4] for 100 ns per replica.



(a) 3SU3 **(b)** 3SU4

(c) 3SU5 **(d)** 3SU6

Figure 2.1.4: Comparison of the dynophores of 3SU3, 3SU4, and 3SU6 generated with the *Dynophore* software[2].

From the histogram shown in Figure 2.1.4, no clear pattern might be derived. The interaction described by the authors is shown in the bottom right between the aromatic isoindoline moiety and R155 (blueish clouds). In fact, some mutations seem to have a more pronounced interaction between vaniprevir and the HCV NS3/4A protease than the wild type and the isoindoline residue. It is evident, however, that spatial distribution of the interactions are slightly different for each mutation compared to the wild type 3SU3. Surprisingly, though, the interaction of 3SU3 and 3SU4 seem very similar, even though the authors claimed the interaction between R155 and the isoindoline moiety would be completely disrupted[7]. Moreover, the authors hypothesized based on a static picture, the interaction between *vaniprevir* and R155 as one of the most significant for activity of the ligand *vaniprevir*.

3 Results

3.1 Zika virus protease

The results of the test system ZIKV$^{\text{Prot}}$ are introduced via the workflow of the software *dylightful*. After extracting the necessary data out of the dynophore generated by *Pach et al.*[4], the TAE is trained. As an example, the reprojected time series and the training, as well as validation accuracy, are given as plots for the mixed perspective as the mixed perspective gave the most insightful results. However, the workflow is the same for the protein and the ligand perspective. The training output for the ZIKV$^{\text{Prot}}$ system in the mixed interaction perspective is shown in Figure 3.1.1. The loss against the validation set is almost equal to the training set and appears to be converged. Thus, the training of the TAE model is considered to be successful.

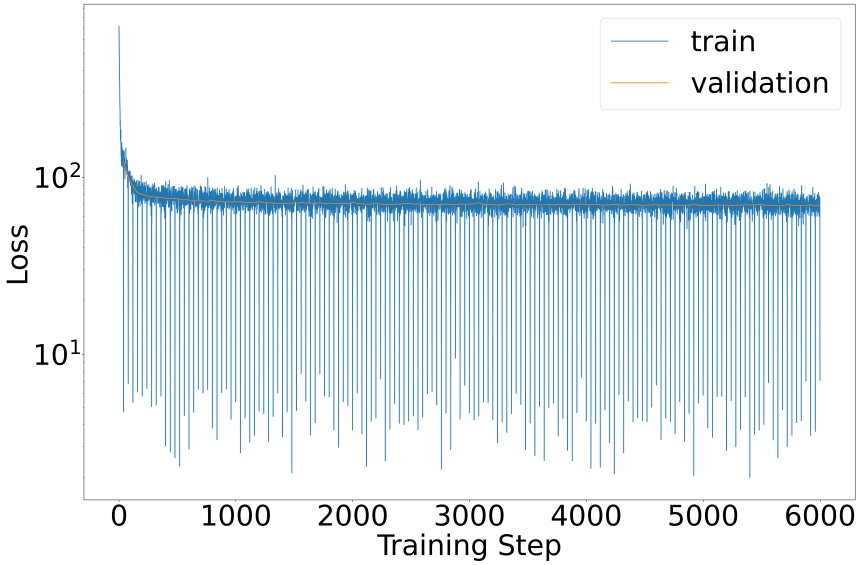


Figure 3.1.1: Convergence of the training done with TAE.

In addition, the reprojection of the original trajectory of the ZIKV$^{\text{Prot}}$ complexed with the ligand shown in Figure 3.1.2 demonstrates the discretisation of the original trajectory through TAE.

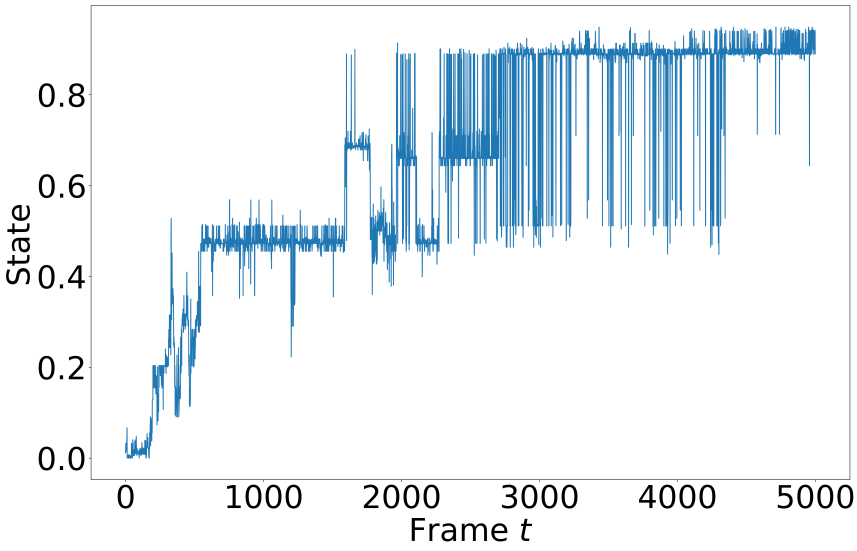


Figure 3.1.2: Mixed interaction trajectory transformed with TAE.

Clustering the trajectory with the *k-means* clustering algorithm gives the elbow plot shown in Figure 3.1.3. Indicating convergence at six states. To have a reasonable automated loss function at hand, for five repetitions, the standard deviation was calculated on the count matrix of the Markov analysis. Then, the mean of each element of the count matrix was calculated and the relative error. The resulting loss function is shown in Figure 3.1.4. According to the loss function, the number of states with the lowest variance was chosen as six Markov states.

Next, the MSM matrices were extracted and are shown in Figure 3.1.5. The MSM matrices show that there are two predominant and four rare states. All states seem to be stable in themselves, as the transition probability is the highest on the diagonal for all states.

The Markov states were extracted and the most and second most stable state are shown in Figure 3.1.6. Furthermore, the different conformation state described by *S. Pach et al.*[4] was extracted by the software, too, and is also shown in Figure 3.1.6. The states show different conformations of the protein. The most stable state based on the absolute number of occurrences, e.g., is showing a 3-10 helix to the bottom left, which is lacking in the second most stable state. The rare ligand orientation in the trajectory corresponds to the state mentioned by authors. The two orientations of the ligand in the binding pocket separated out by the MSM analysis are hypothesized to explain the higher binding affinity of the ligand **1** compared to other ligands[4]. Moreover, the predominant configuration seems to induce several protein conformations, whereas the rare spatial orientation of the

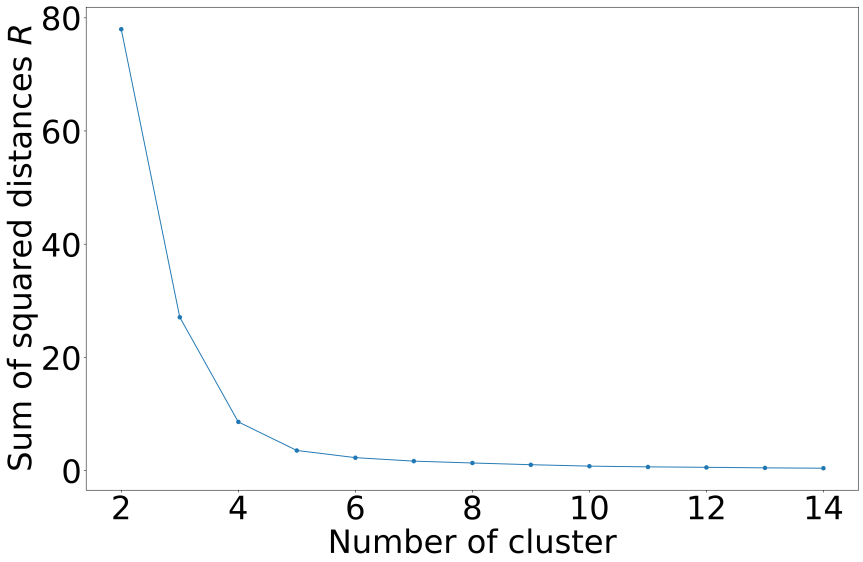


Figure 3.1.3: Convergence of the k-means clustering applied to the TAE transformation of the mixed interaction perspective.

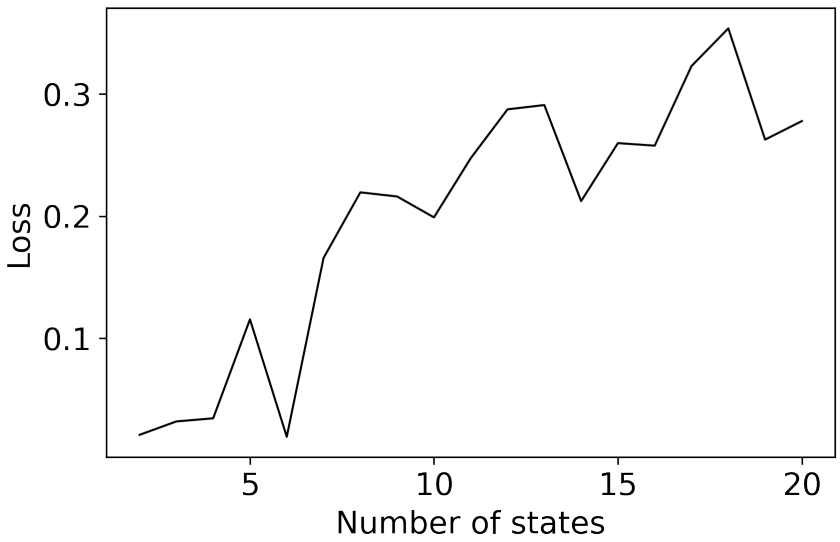
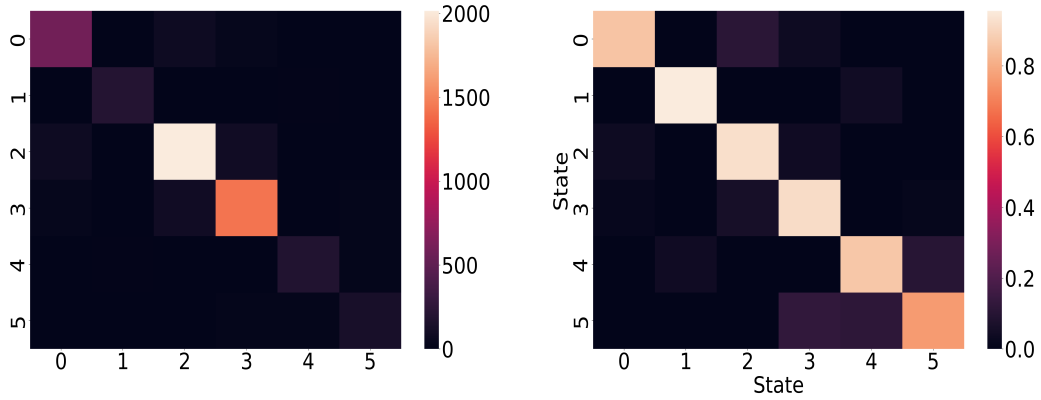


Figure 3.1.4: Convergence of the Markov analysis of ZIKV$^{\text{Prot}}$ for different number of states.



(a) Count matrix **(b)** Transition state matrix

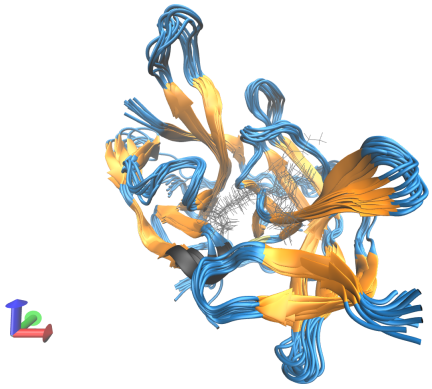
Figure 3.1.5: Markov matrices for the ZIKV^{Prot} complexed with **427_1**.

ligand **1** is inducing only a single conformation of the protein–ligand–complex.

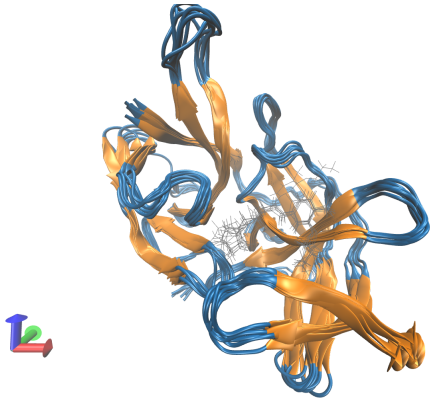
To see, whether different ligand binding modes are responsible for the stabilization of the different CVs shown in Figure 3.1.6, dynophores for each state were generated (Figure 3.1.7). Indeed, the rare conformation mentioned by *Pach et al.* could be extracted using the aforementioned workflow. Moreover, the two most prominent CVs show a slightly different *Dynophore* histograms as well. In contrast to the most stable state, the second most stable state shows an interaction with ASN152 through the left part of the ligand **1**. Moreover, the two most stable states show slightly different spatial orientations of the ligand **1**, too.

3.2 CDK2

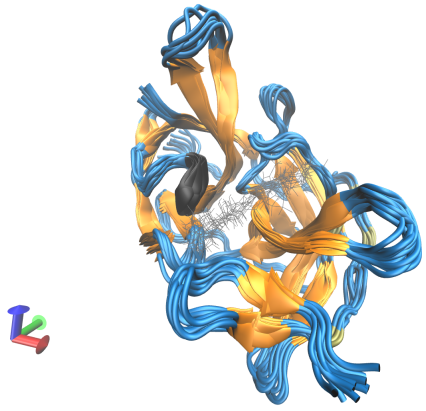
As a first step, the evaluation of modelling decisions regarding the 1KE7 supramolecular complex are evaluated. The original structure was capped at the residues 37-43 and 153-162. Thus, a homology model was assembled using MOE[52] (compare Section 2.1.5). The loop shows significant influence on the ligand binding behaviour. In the case where the loop was not remodelled but the capped structure was simulated, the system shows larger binding pocket during the course of the simulation. Moreover, the interactions between the remodelled loops of the residues 37–43 and 153–162 were measured. The height of the binding pocket may be a measure for the induced-fit initiated by the ligand, and is thus a measure for the inhibition ability of the ligand. To monitor the binding pocket-size, the distance between ILE10 and LEU83 was measured. The two different histograms shown in



(a) Most stable conformation

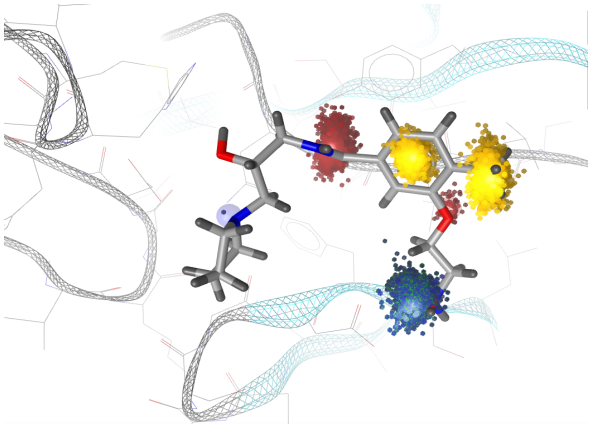


(b) Second most stable conformation of 1KE5

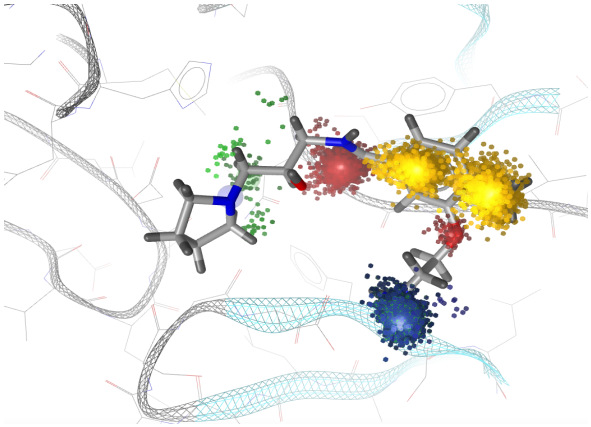


(c) Rare ligand conformations

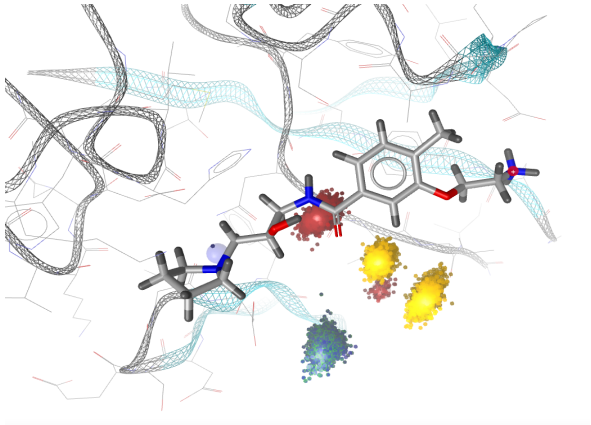
Figure 3.1.6: Different Markov states extracted from the MD simulation of the ZIKV$^{\text{Prot}}$[4] showing the two most stable conformations, and the described rare ligand conformation.



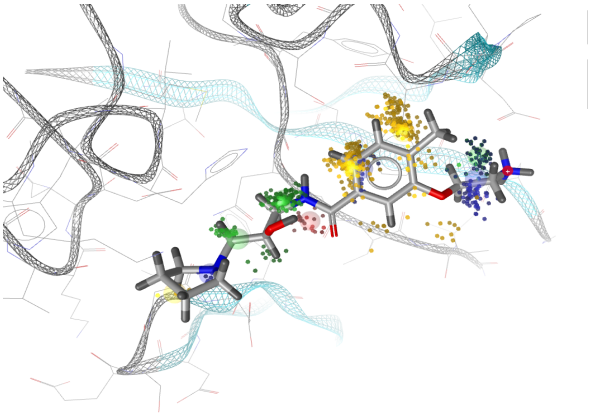
(a) Most stable state



(b) Second most stable state



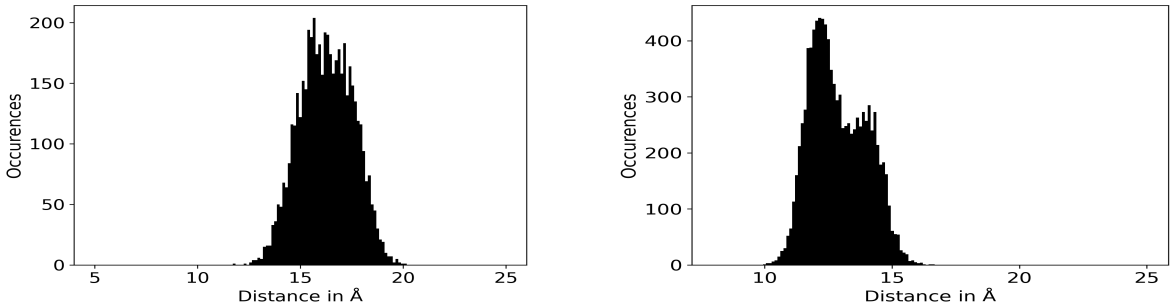
(c) Most stable stat overlaid with the rare ligand conformation



(d) Rare ligand conformation

Figure 3.1.7: Comparison of the classical *Dynophores* of compound **1** for the most stable, second most stable state and the state showing the rare ligand conformation described by *S. Pach et al.*[4]

Figure 3.2.1 clearly demonstrate the big influence of the remodelled loops of the residues 37–43 and 153–162 on the structure of the binding pocket.



(a) Plain structure of 1KE7 **(b)** Remodelled structure of 1KE7

Figure 3.2.1: Comparison of the plain structure of 1KE7 and the homology model obtained from MOE[52].

Therefore, the structures of 1KE5 and 6GUH were modelled according to the modelled structure of 1KE7 such that GLU40 and ARG147 can form hydrogen bonds. In Figure Figure 3.2.3–Figure 3.2.5, the time-resolved dynophores of the different structures are shown. The time resolved dynophores differ significantly, however no real trend can be derived from these pictures alone.

The time resolved dynophore is shown for both the not remodelled (Figure 3.2.2) and the remodelled protein (Figure 3.2.2), and one can see differences in both dynophores. The remodelled protein shows a more stable binding of the ligand to protein in the binding pocket compared to the capped protein.

The histograms of the binding pocket-size for all the three systems 1KE7, 1KE5, and 6GUH are compared in Figure 3.2.6. For 1KE7, the system seems to be in states where the binding pocket-size seems to be stabilized around 11 and 15 Å. In 6GUH, the binding pocket seems to be stabilized at around 11 Å, too. However, for 1KE5, the binding pocket seems to be predominantly stabilized at 10 Å and below. Therefore, as expected from the activities, 6GUH and 1KE7 are similar, whereas 1KE5 should show different behaviour. Mostly, the binding pocket-size does not seem to correlate with the ligand activity in a straightforward manner, but can be related to the ligand activity in the examples studied.

The finding of the different binding pocket sizes during the simulation can be rationalized with the interaction of the remodelled loops between each other. To compare the different protein–ligand systems 1KE7, 1KE5, and 6GUH, the number of hydrogen bonds between

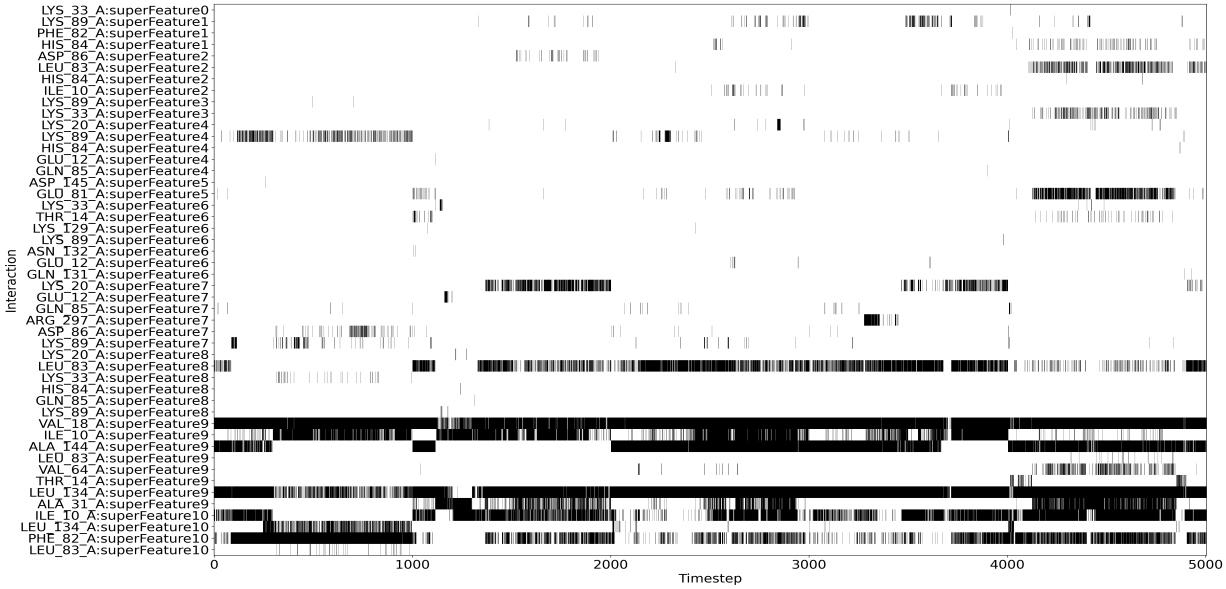


Figure 3.2.2: Time resolved dynophore of the plain 1KE7 structure showing the mixed interaction perspective.

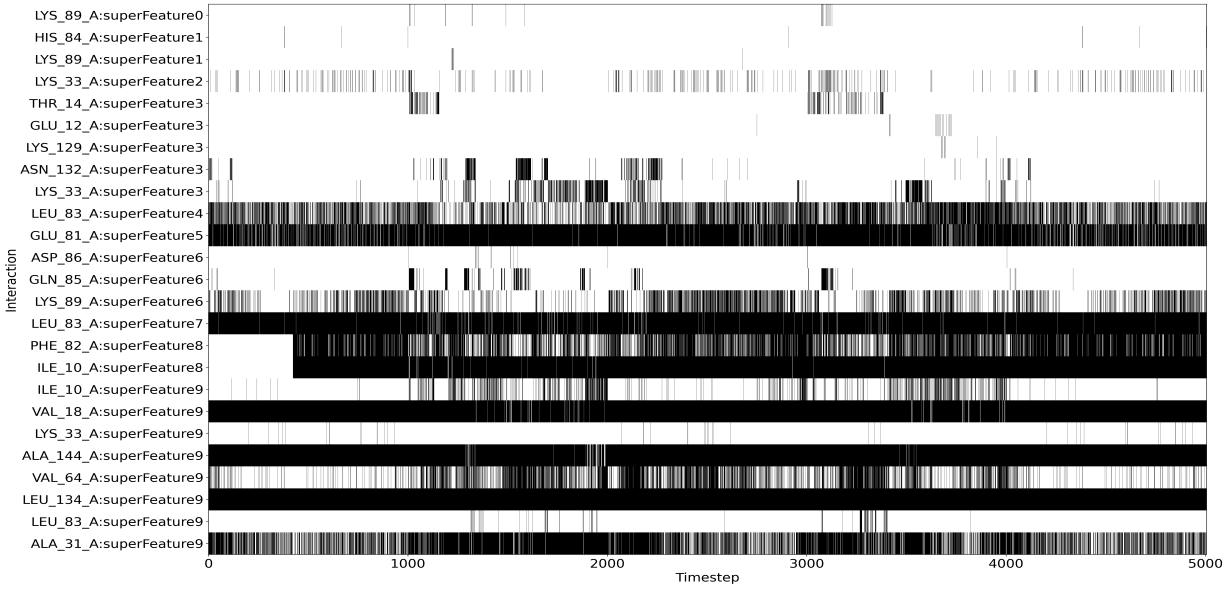


Figure 3.2.3: Time resolved dynophore of remodelled 1KE7 showing the mixed interaction perspective.

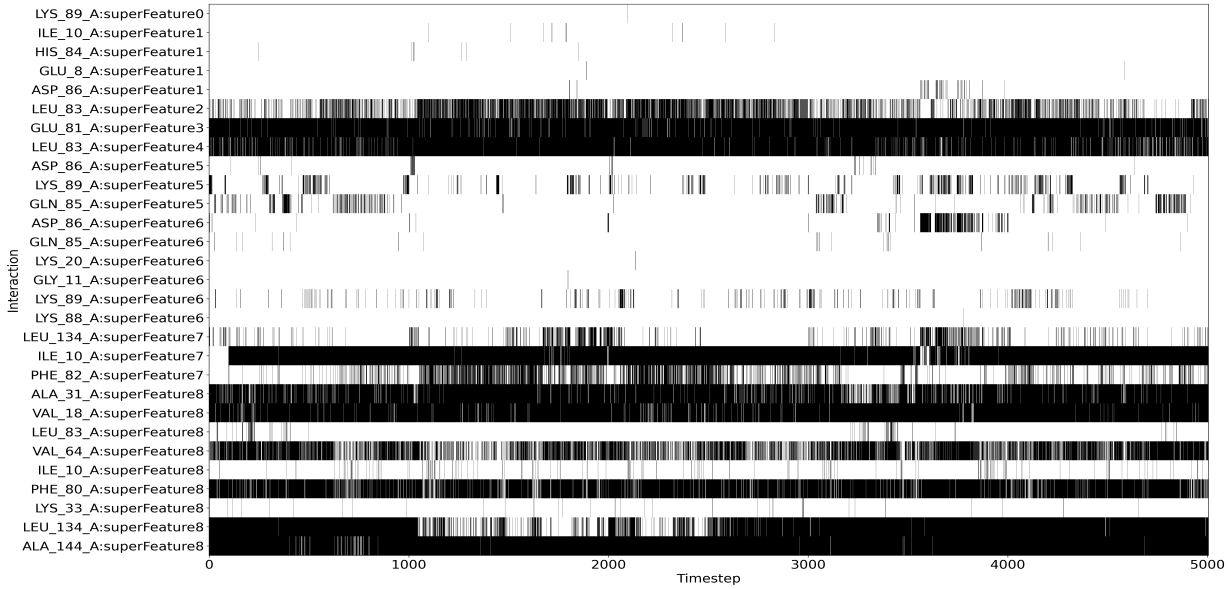


Figure 3.2.4: Time resolved dynophore of remodelled 1KE5 showing the mixed interaction perspective.

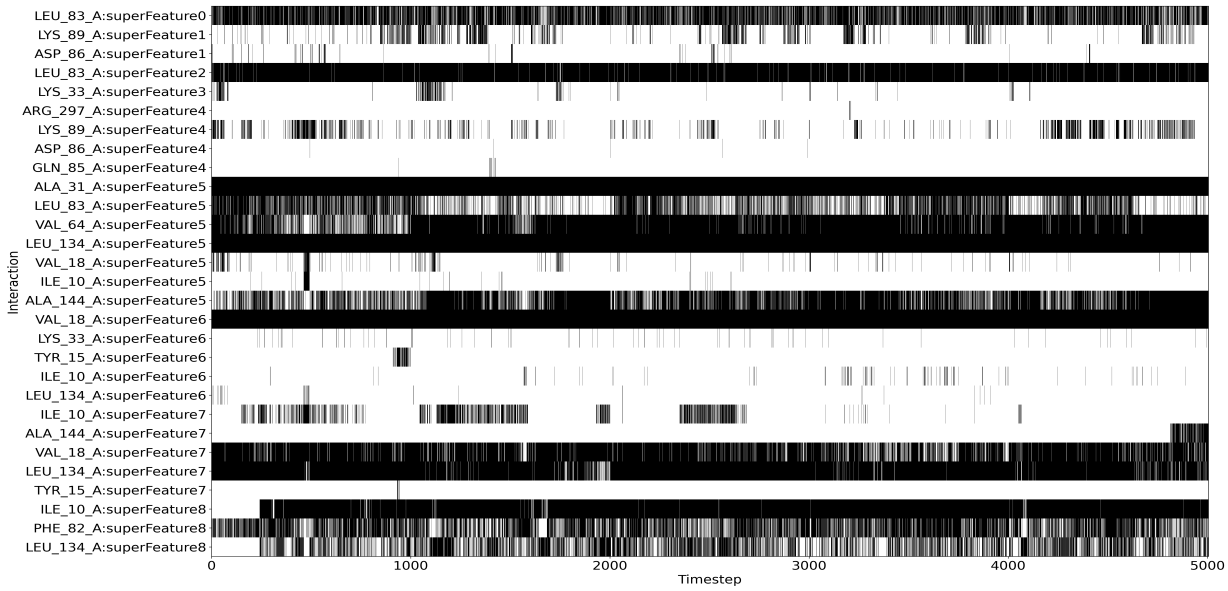
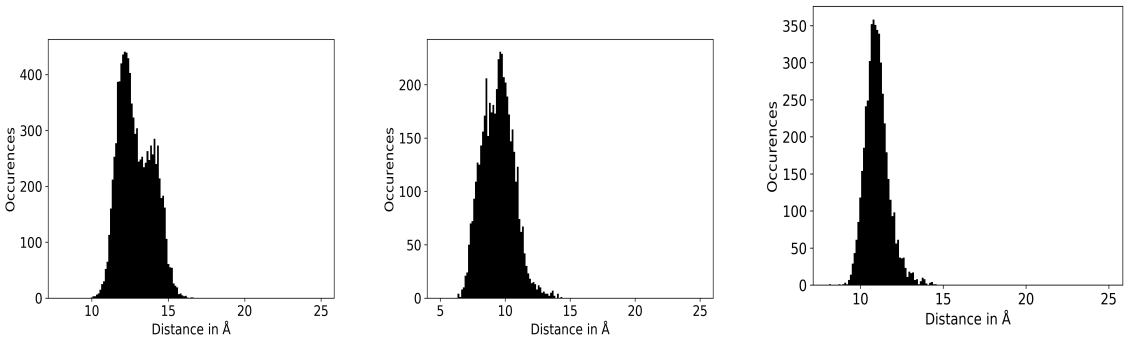


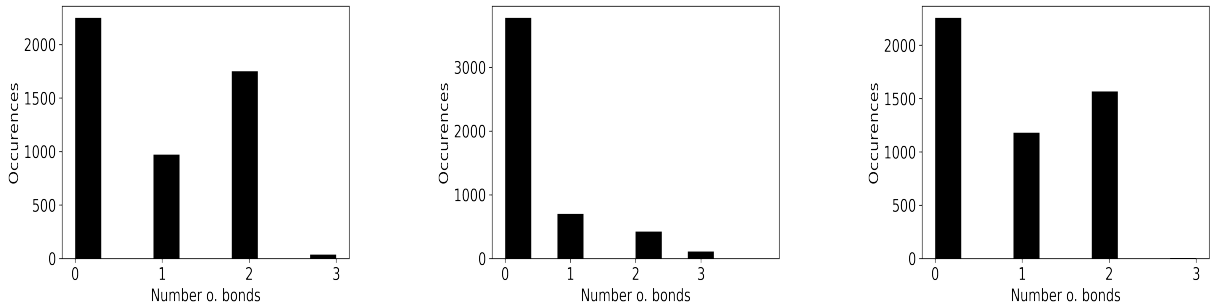
Figure 3.2.5: Time resolved dynophore of remodelled 6GUH showing the mixed interaction perspective.



(a) 1KE7 **(b)** 1KE5 **(c)** 6GUH

Figure 3.2.6: Histogram of the binding pocket size measured between LEU10 and ILE83 during the course of the simulation for 100 ns.

the remodelled loops were monitored. The histograms presented in Figure 3.2.7 underline that the ligands have different ability to induce stabilizing protein conformations. Yet, again, 1KE7 and 6GUH seem to show a similar behaviour.

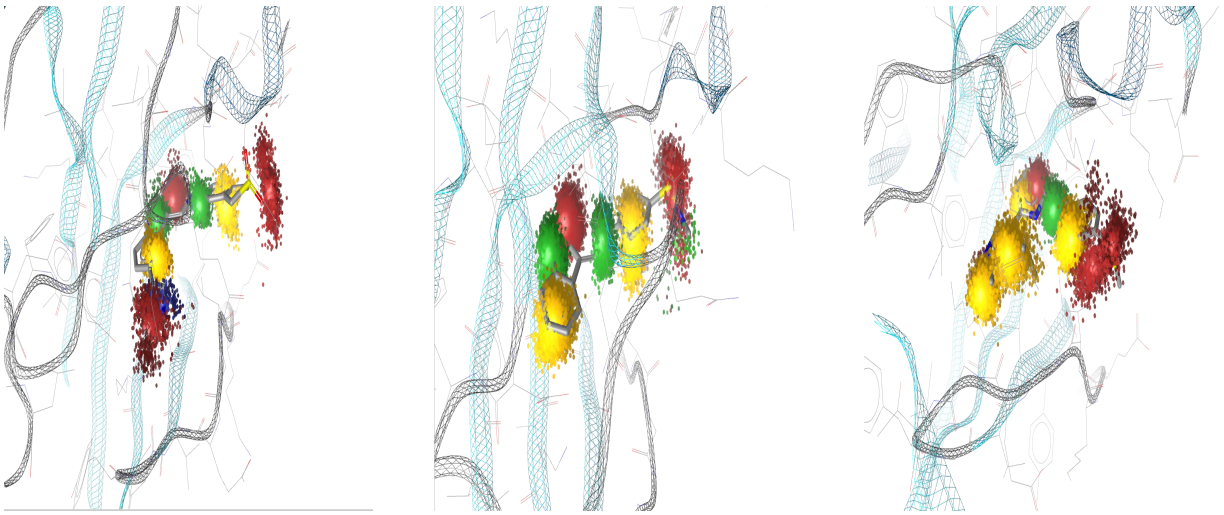


(a) 1KE7 **(b)** 1KE5 **(c)** 6GUH

Figure 3.2.7: Histogram of the number of hydrogen bonds between the remodelled loops during the course of the simulation.

The different activity can also be seen from the classical dynophore visualization shown in Figure 3.2.8. The pictures show a clear difference in the dynophore patterns and the structures show different number of interactions. The different dynamic behaviour of the ligands is thus not only present in the binding pocket, but manifests itself throughout the whole protein–ligand complex. Moreover, it most likely that the dynamic picture of the whole protein–ligand complex gives insights into the binding mechanism of different ligands, as there seem to be different important features of the ligands, that stabilize the binding. One or more conformations must be responsible for tiny binding pocket sizes (less than 10 Å) and other conformations must be responsible for the stabilization at around (11 Å).

For example, 6GUH and 1KE7 share similar superfeatures at the tail (ASP145, LYS33) whereas 1KE7 and 1KE5 share the same interaction pattern on the hinge region (near LEU83). It is thus likely, that all ligands show a different markophore. Still, 1KE7 and 6GUH should share the same dominant, CV as the histograms of the binding pocket-size shown in Figure 3.2.6 suggest a similar stabilized conformation.



(a) 1KE7 **(b)** 1KE5 **(c)** 6GUH

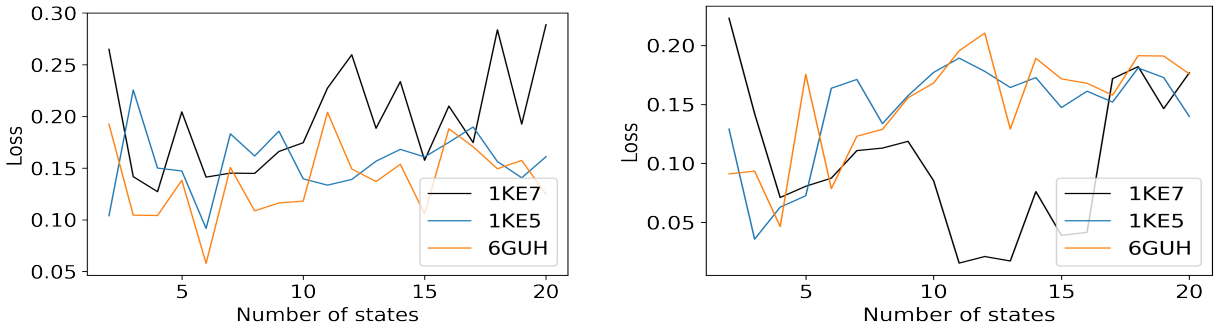
Figure 3.2.8: Comparison of different the dynophores obtained from the MD simulation of 1KE7, 1KE5, and 6GUH.

To do any qualitative or quantitative SAR based on the analysis, first the robustness of the simulation is tested by simulating longer and determining the states according to the method introduced in 3.1. The losses for 100 ns and 200 ns simulations were compared. The loss functions shown in Figure 3.2.9 demonstrate that the longer simulation time of 200 ns introduced reduced the overall variance of the analysis. Especially, the 1KE7 structure, seems to be framed as more complex than derived from the 100 ns simulation.

The 1KE7 structure most likely visited a different conformation. The new conformation can be seen from the RMSD plot shown in Figure 3.2.10

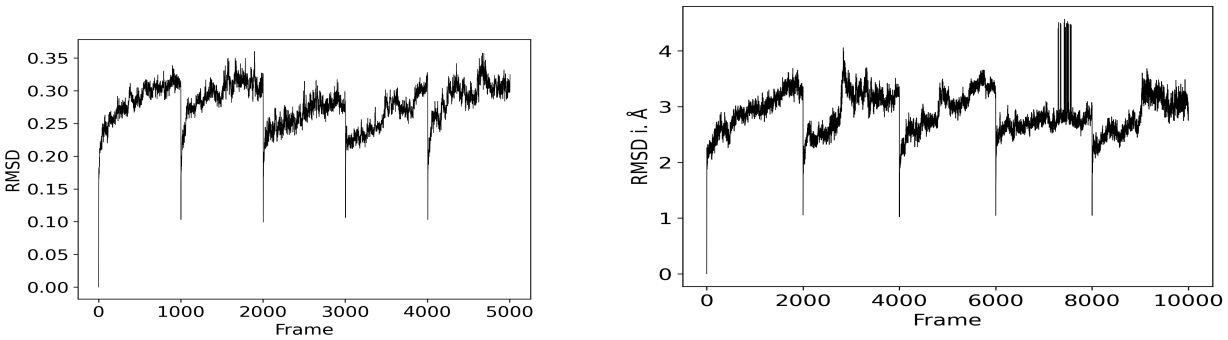
Next, as a sanity check, the histograms are compared again (Figure 3.2.11). However, the histograms shown in Figure 3.2.11 show no difference to the histograms from the shorter simulation for 100 ns shown in Figure 3.2.6 visually. Therefore, the in-depth analysis was needed to uncover the rare conformation seen in Figure 3.2.10.

To make the simulations comparable, the count and transition state matrices were extracted. The count matrices for the long simulation time of 200 ns for the three different



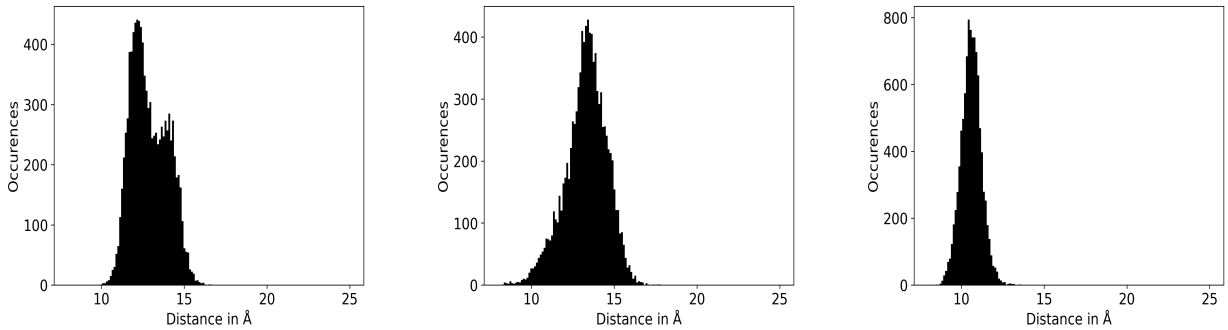
(a) 100 ns **(b)** 200 ns

Figure 3.2.9: Comparison of the loss functions for the Markov state determination for different simulation times.



(a) 100 ns **(b)** 200 ns

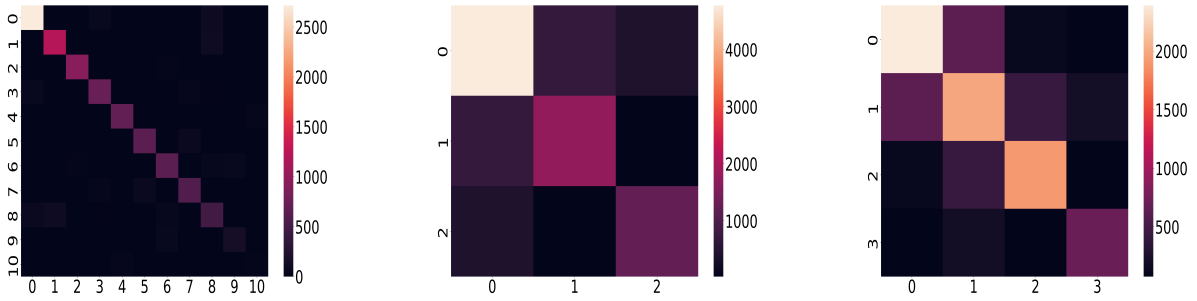
Figure 3.2.10: Comparison of the RMSD trajectory for the 1KE7 structure for different simulation times.



(a) 1KE7 **(b)** 1KE5 **(c)** 6GUH

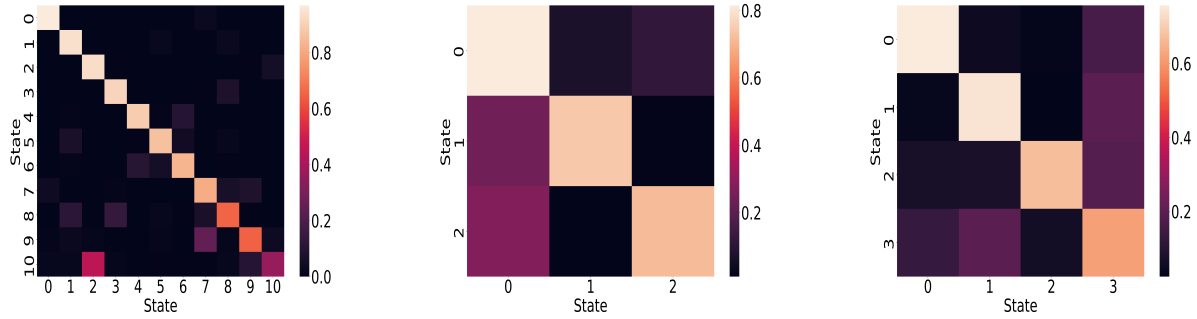
Figure 3.2.11: Histogram of the binding pocket size measured between LEU10 and ILE83 during the course of the simulation.

systems 1kE7, 1KE5, and 6GUH are shown in Figure 3.2.12. The count matrices already give a qualitative measure of the ligand activity.



(a) 1KE7 **(b)** 1KE5 **(c)** 6GUH

Figure 3.2.12: Transition probability matrices of the Markov model for the three different supramolecular complexes 1KE7, 1KE5, 6GUH.



(a) 1KE7 **(b)** 1KE5 **(c)** 6GUH

Figure 3.2.13: Count matrices of the Markov model for the three different supramolecular complexes 1KE7, 1KE5, 6GUH.

The activities for the systems 1KE7[5], 1KE5, and 6GUH[6] are 8.9 nM, 560 nM, and 26 (6) nM[59], respectively. As the experimental error for the activity determination of 6GUH seems large, and the experiments are from different labs and separated by almost 10 years, one might consider 1KE7, and 6GUH similarly active, whereas 1KE7 seems to be a little bit more active. In the plot shown in Figure 3.2.14 an attempt to do QSAR based on the Markov model is shown. To establish a linear relationship, the diagonal value of the most prominent state in the count matrix (Figure 3.2.12) were extracted. From the plot shown in Figure 3.2.14 it is evident that the counts relate to the stability in a non–linear manner.

Comparing the most prominent conformations for each supramolecular complex 1KE7,

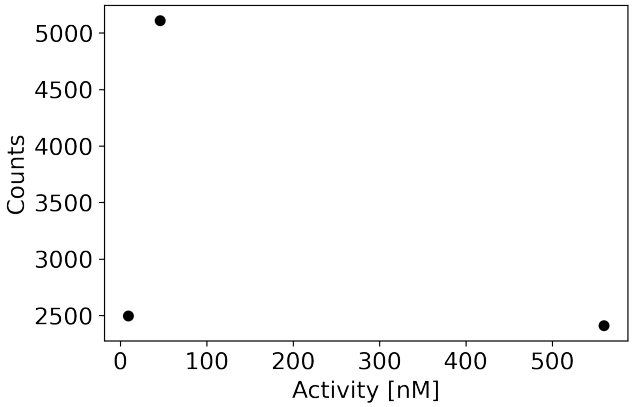
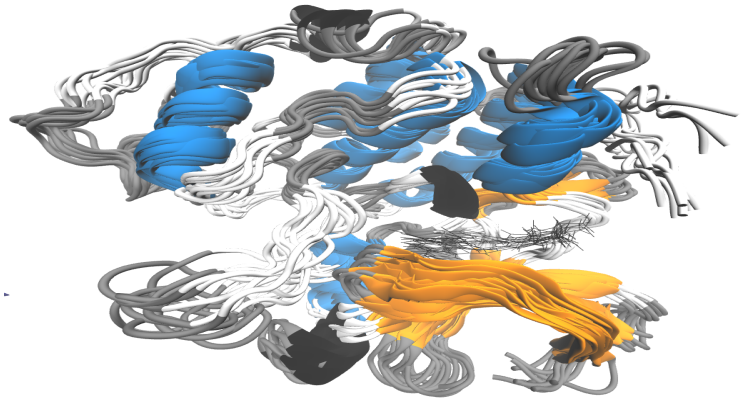


Figure 3.2.14: Attempt to model QSAR on different CDK2 inhibitors based on the diagonal of the count matrix obtained from the Markov model.

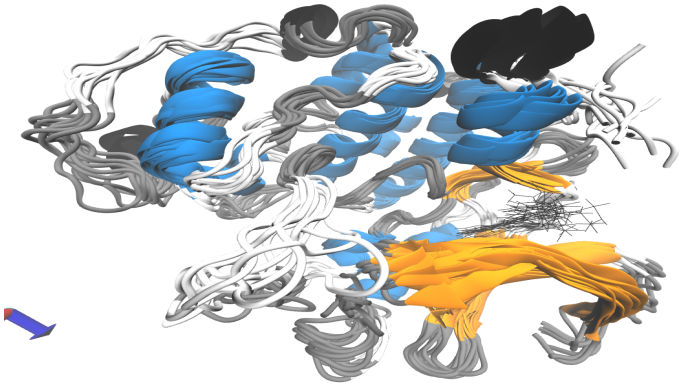
1KE5, and 6GUH shown in Figure 3.2.15 it becomes evident that slightly different conformations of the protein are stabilized by interaction with their respective ligand. When visually inspecting the most stable CV for each system, it becomes evident that 6GUH stabilizes a conformation in between the conformations of the most stable CV of 1KE7 and 1KE6. At the top, 6GUH is similar to 1KE7 and at the bottom more similar to 1KE5. The bottom-left of the protein-ligand complex is wider. The bottom-left is the area of the protein where cyclin A and cyclin B would form the supramolecular complex. As the binding site for cyclin A and B is the widest for 1KE5 in the most stable conformation, it could be an explanation why the experimental inhibition activity of the ligand complexed with CDK2 in 1KE5 is the lowest, as cyclin A binding is critical to the activity of CDK2[60]. On the contrary, CDK2 binds to the second subunit at the top of the protein, where 6GUH and 1KE7 are very similar[6]. The analysis aligns with the other studies, emphasizing the complexity of the ligand interaction that manifests different activities based on the inhibition of CDK2 of different potency depending on the supramolecular complex CDK2 is bound to (cyclin B, cyclin A or apo structure)[6,51].

3.3 HCV NS3/4A protease

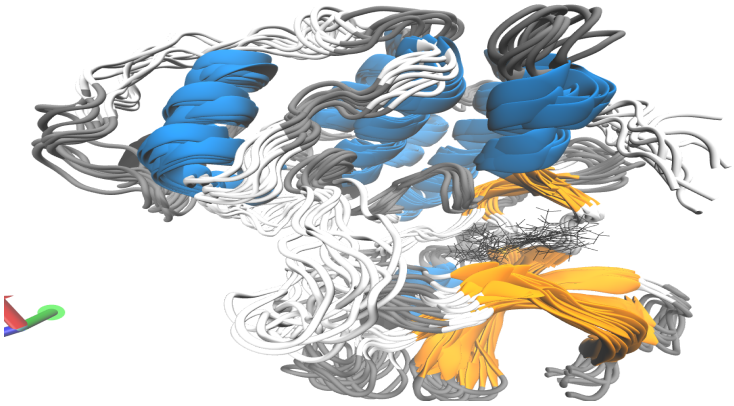
To investigate the contradiction between the hypothesis of the relation of R155 to the ligand activity of *vaniprevir* derived from crystal structure experiments[7,8] elaborated in 2.1.6 first the time resolved dynophores are reviewed similarly to 3.2. The time resolved dynophores shown in Figure 3.3.1 to Figure 3.3.4 seem to prove the hypothesis from *Schiffer et al.*[7,20] in the sense that the interaction between R155 (ARG1155 in the figures) is disrupted for the



(a) Most stable conformation of 1KE7



(b) Most stable conformation of 1KE5



(c) Most stable conformation of 6GUH

Figure 3.2.15: Different Markov states extracted from the MD simulation of the CDK2 complexes 1KE7, 1KE5, and 6GUH.

R155K mutation. Still, the new interaction between the ligand through the superfeatures 9, 12, and 13 are very prominent in the case of the interaction through superfeature 9 and 12 with LYS1155. However, in the mutations D168A A156T that are meant to disrupt the interaction of R155, the interaction of vaniprevir with R155 is even more pronounced after introducing the mutations. For 3SU5, interaction with superfeature 3 is seldom, whereas the interaction via superfeature 11, and 12 is very prominent. For 3SU6, the interaction of vaniprevir via superfeature 3 is prominent, via superfeature 9 seldom, and via superfeature 12 very prominent.

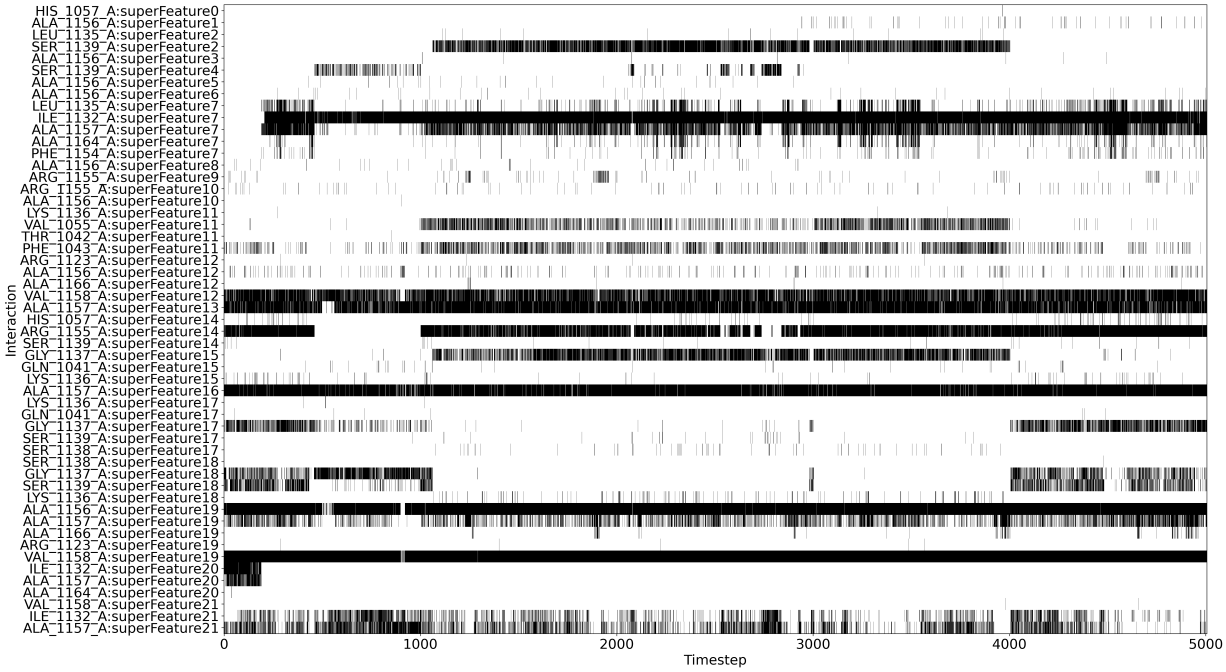


Figure 3.3.1: Time resolved dynophore of remodelled 3SU3 showing the mixed interaction perspective.

Apart from the interaction frequency, the number of interactions in 3SU5 and 3SU6 are different. In 3SU5 and 3SU6 there seem to be more interactions present indicating novel CV as compared to 3SU4 and 3SU3. Thus, the CV analysis similar to CDK2 was performed. Firstly, the RMSD of the protein–ligand complex was analysed to see if the trajectories might give some hints about the large activity difference of the four different protein–ligand complexes. The comparison of the RMSD histograms of the trajectories for the four supramolecular complexes is shown in .

The different histograms suggest several conclusions. Firstly, the histogram of 3SU4 is entirely different from all other histograms and suggest that the protein is stabilized in a

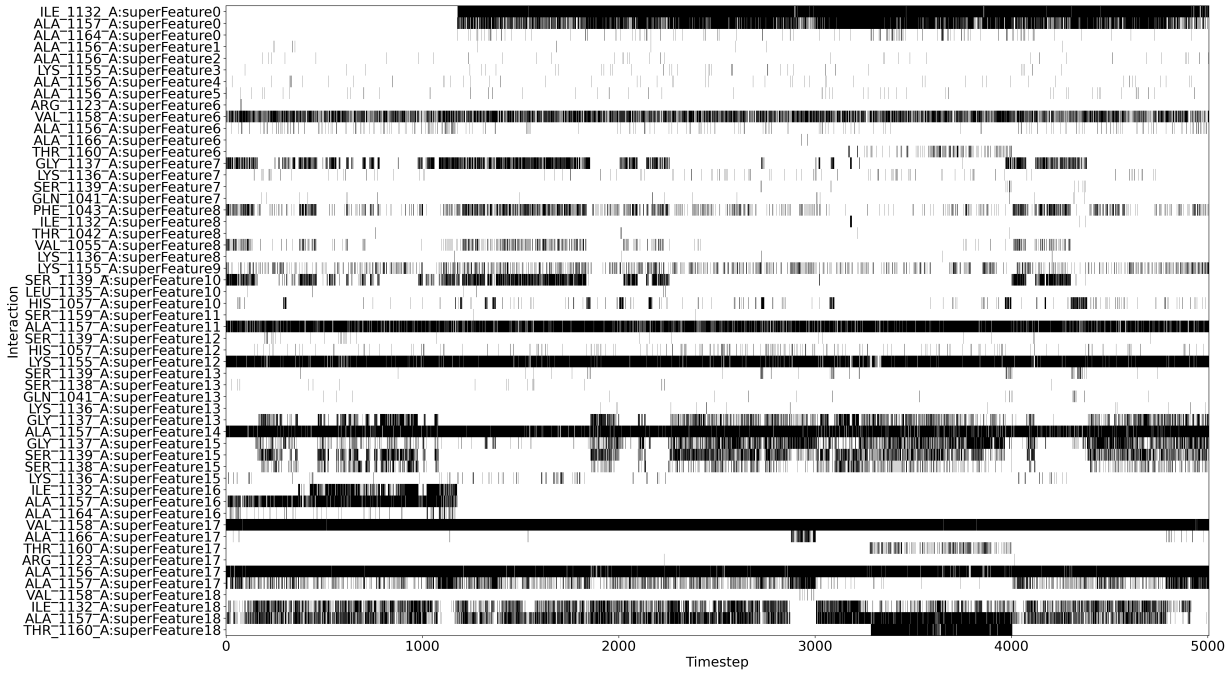


Figure 3.3.2: Time resolved dynophore of remodelled 3SU4 showing the mixed interaction perspective.

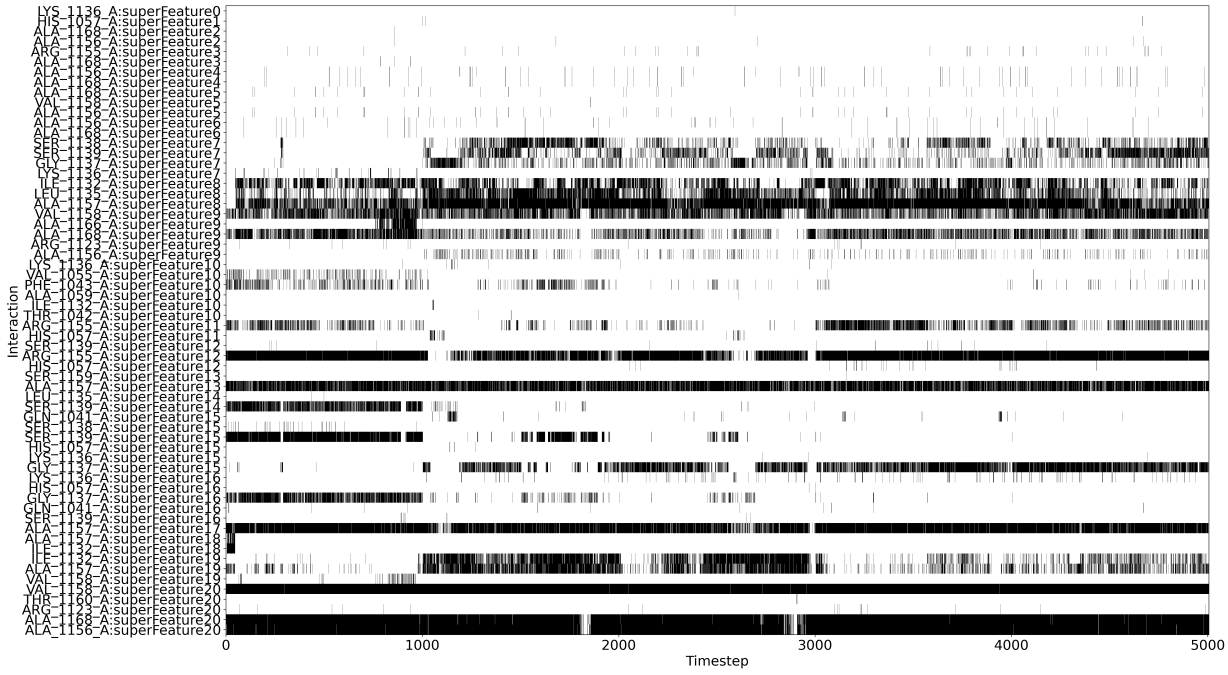


Figure 3.3.3: Time resolved dynophore of remodelled 3SU5 showing the mixed interaction perspective.

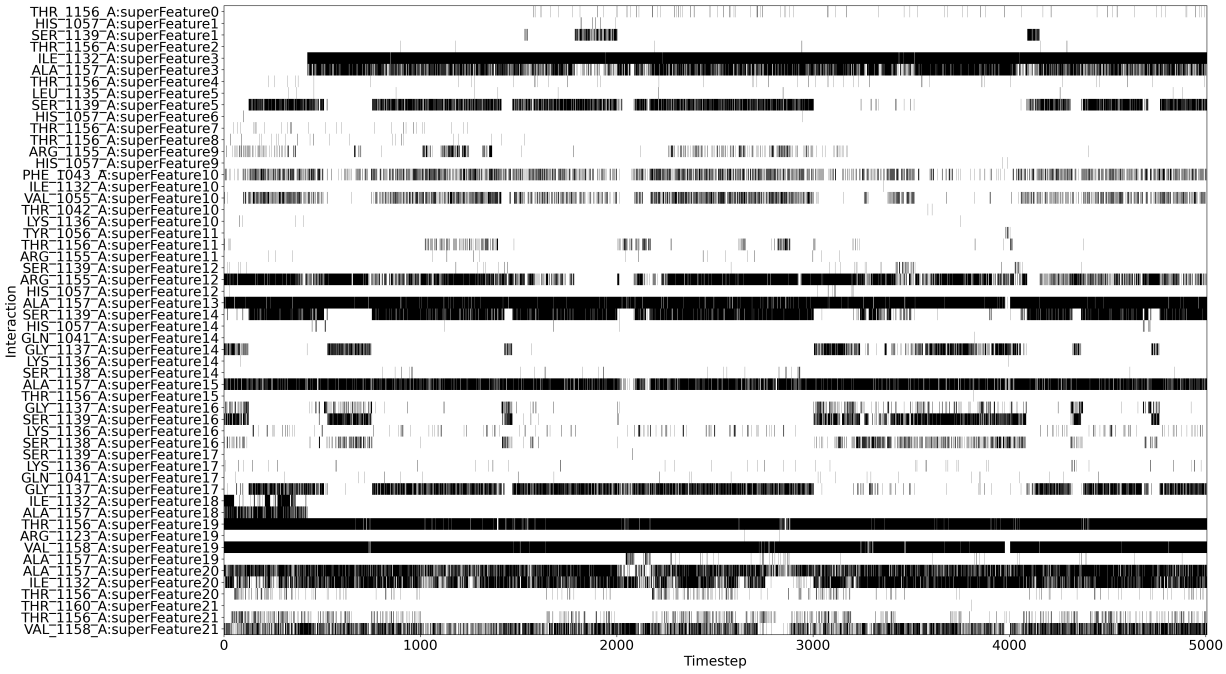
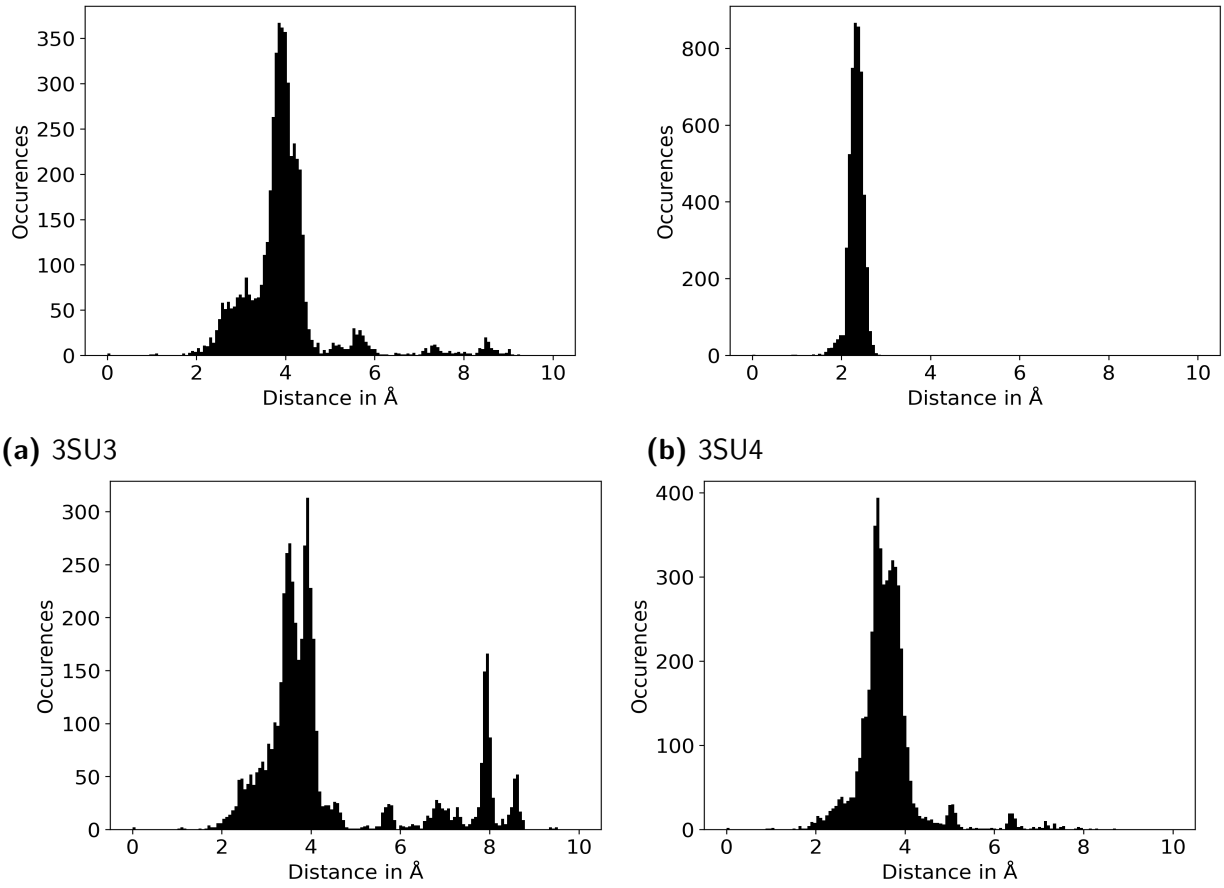


Figure 3.3.4: Time resolved dynophore of remodelled 3SU6 showing the mixed interaction perspective.

certain conformation. This conformation would then according to the histograms shown in 3.3 be present in all of the other complexes, too (3SU3, 3SU4, 3SU6). However, the differences between 3SU3 and the 3SU5 and 3SU6 are less clear. 3SU5 and 3SU6 show stable conformations between 2 Å and 4 Å. Whereas, the probabilities of observing different RMSD values in the interval are different for 3SU5 and 3SU6. Nevertheless, 3SU5 and 3SU6 do not show a prominent peak between 4 Å and 5 Å. Yet, 3SU5 seems to show conformations with high probability that occur at RMSD values bigger than 6 Å. Therefore, each of these systems seems to show according to classical RMSD analysis a different mechanism of altering the activity of the inhibitor *vaniprevir*. It remains elusive, however, if the simulation of the systems are sufficient to explain the activity difference.

Next, to see if the Markovian analysis gives further insights, the number of distinct CV plots for the different complexes are compared. First, five repetitions of the whole workflow described in 3.2 were performed.

From the loss functions shown in Figure 3.3.6, the states corresponding to the minimal loss were determined. Yet, the matrices showed comparably high variance when repeating the workflow five times. Sometimes, the clustering converged to a local minimum. The high variance can be explained by looking into the reprojections obtained from TAE (Fig-



(a) 3SU3 **(b)** 3SU4

(c) 3SU5 **(d)** 3SU6

Figure 3.3.5: Comparison of the the histrograms of the RMSD values of the MD trajectories of 3SU3, 3SU4, 3SU5 and 3SU6.

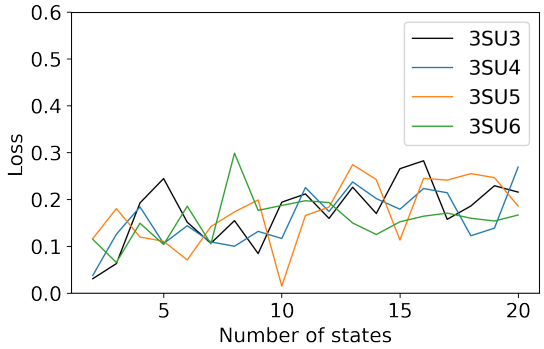
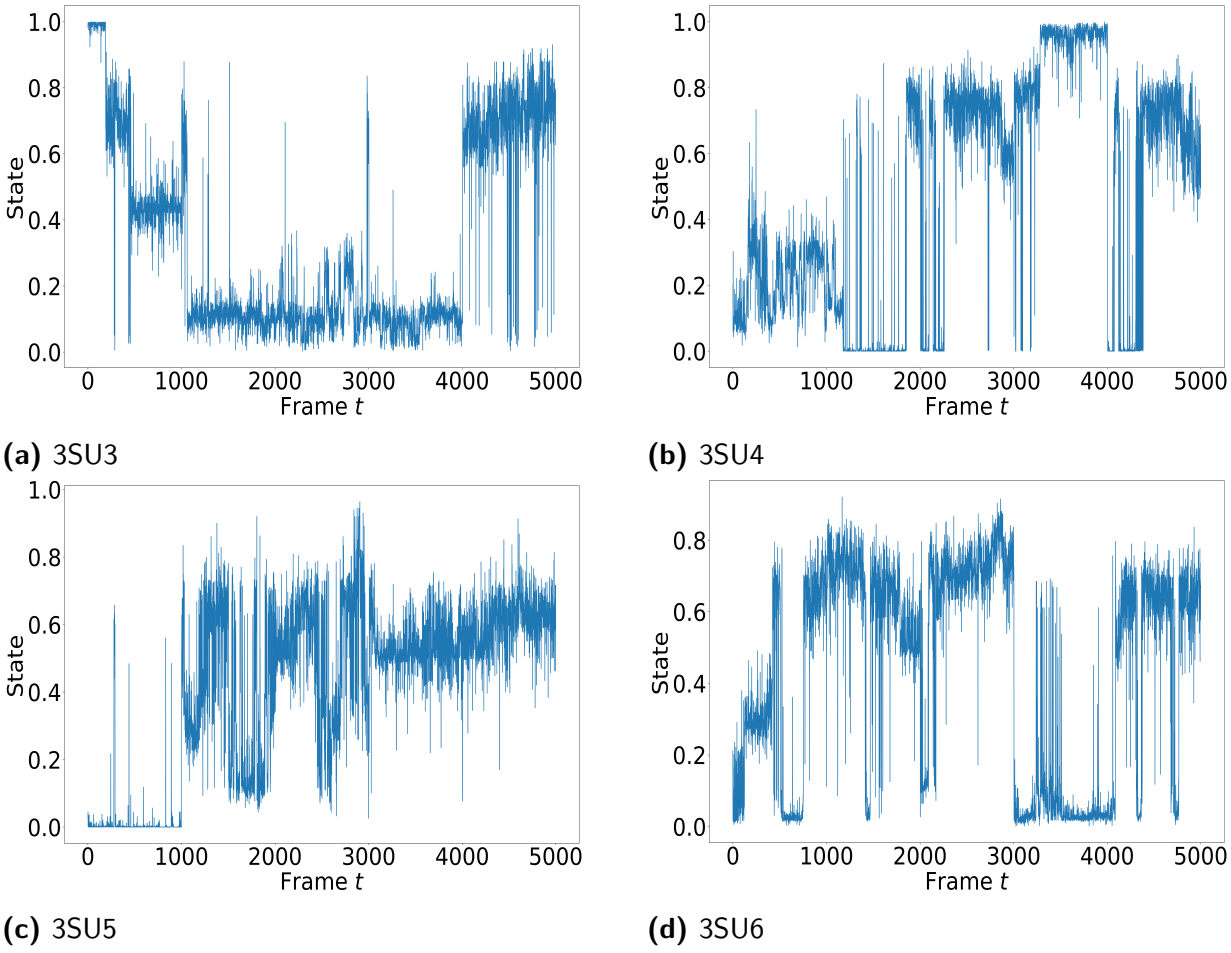


Figure 3.3.6: Loss for the determination of the number of CV for the supramolecular complexes 3SU3, 3SU4, 3SU5 and 3SU6.

ure 3.3.7). They do not show a very different picture. In fact, compared to the ZIKV$^{\text{Prot}}$ (Figure 3.1.2, the TAE reprojection is quite noisy. Thus, the workflow was repeated with twice the number of frames but the same simulation time, obtaining a new loss function shown in Figure 3.3.8. The count obtained count matrices with lower variance are shown in Figure 3.3.9.



(a) 3SU3 **(b)** 3SU4

(c) 3SU5 **(d)** 3SU6

Figure 3.3.7: Comparison of the count matrices from the HMM obtained trough *dylightful* for the systems 3SU3, 3SU4, 3SU5 and 3SU6.

Moreover, the ligand seems to stay in the same conformations in the states, such that no insight might be derived from the pure ligand interaction. It is surprising because the ligands exhibit different activities according to the enzyme inhibition assays conducted by the authors[7,8]. Still, the authors only monitored the enzyme cleavage activity. According to the analysis of the data obtained from 100 ns MD simulations done with *dylightful*, the different supramolecular complexes show slightly different induced dynamical configurations. Thus, according to the count matrices shown in Figure 3.3.9 one could imagine that the

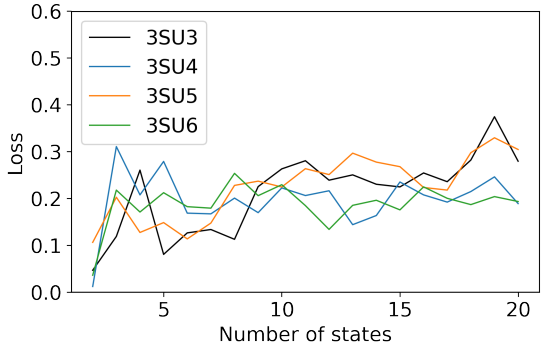
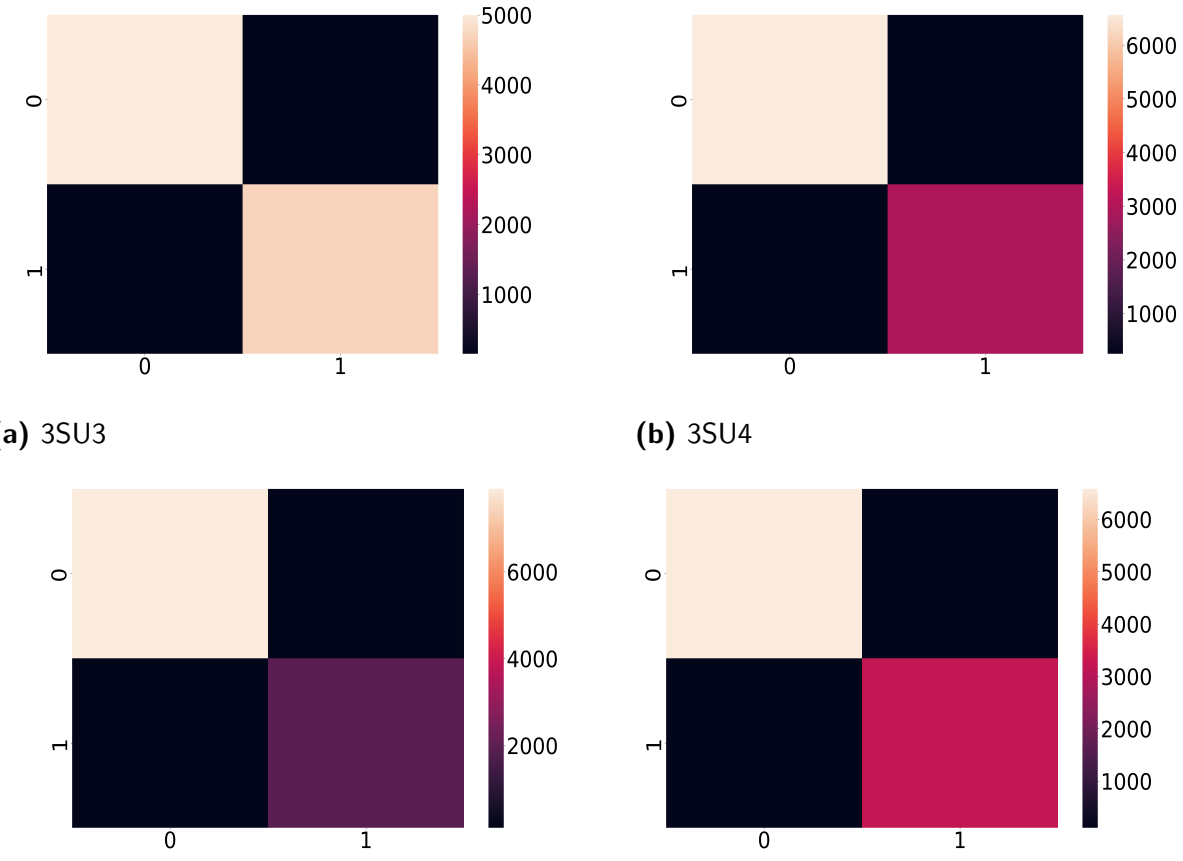


Figure 3.3.8: Loss for the determination of the number of CV for the supramolecular complexes 3SU3, 3SU4, 3SU5 and 3SU6 with approximately 10,000 frames.



(a) 3SU3

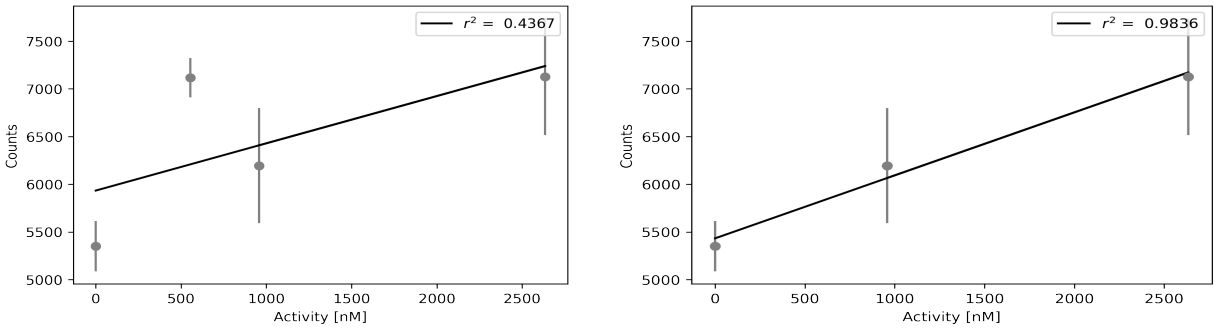
(b) 3SU4

(c) 3SU5

(d) 3SU6

Figure 3.3.9: Comparison of the count matrices from the HMM of approximately 10,000 frames obtained trough *dylightful* for the systems 3SU3, 3SU4, 3SU5 and 3SU6 .

mutation may not restrict ligand binding to the NS3/4A protease, but in turn the protease may change its conformation due to the mutation such that it can exhibit cleavage activity even though the protease was bound to the inhibitor *vaniprevir*. Therefore, the competitive binding might be lowered as different conformations of the system were induced. An experimental study using time-resolved Förster resonance energy transfer microscopy could give further insights. To verify the aforementioned hypothesis, QSAR based on the row sums of the count matrices in Figure 3.3.9 was performed resulting in an almost linear fit, when omitting 3SU4 as an outlier (Figure 3.3.10). The different behaviour of 3SU4 can't be explained in a straightforward manner with the linear modelling done. The framing of 3SU4 as an outlier points in the direction of a different mechanism of modifying the activity of *vaniprevir* as compared to 3SU3, 3SU5, and 3SU6. However, the outlying behaviour of 3SU4 could also be due to experimental uncertainty. The reason of the different behaviour of 3SU4 compared to the other complexes can't be investigated further in this study and would require additional experimental studies.



(a) All structures **(b)** Omitted 3SU4

Figure 3.3.10: Comparison of QSAR on the HCV NS3/4A protease with and without outlier based on row sums of the count matrix obtained from the Markov model based on approximately 10,000 frames and 100 ns simulation time.

4 Discussion

4.1 Application

Firstly, it was shown that the drug design community desperately needs new software tools, to enhance their abilities. The *lean* method proved as a valuable method throughout the development of *dylightful*. Mostly, the understanding of the drug design community regarding the dynamic behaviours of their targets was limited due to the lack of sufficient models to explain phenomenons they observe. Even though simulations are becoming more powerful, the drug design community is stuck to derive models and make decisions based on a static mindeset separating ligands and proteins (compare SBDD vs. LBDD) when judging protein–ligand interactions. The *dylightful* software makes use of the dynamic pharmacophores introduced by *D. Sydow*[2] and enhances the theory behind *Dynophores*. In each test-system, the software gave an unique understanding not described in the literature of each protein–ligand complexes studied. It is likely that the software *dylightful* will do so in other supramolecular complexes as well.

Additionally, it was demonstrated that by projecting the input to the hidden space of the trained TAE model, the *Viterbi* path of the hidden CV is obtained and that the method utilizing TAE finds eigenfunctions of the Hamiltonian of the systems doing non–linear variational principle. The analysis offers a valuable tool for judging modelling decisions in the context of protein–ligand interactions. Decisions with a profound effect on the system are differentiated immediately and with the understanding of the system as a whole rationalized easily. Moreover, sufficient sampling rates and simulation times for a given system can be judged based on the modelling done by *dylightful*.

Besides, the $\mathrm{ZIKV}^{\mathrm{Prot}}$ test system demonstrated that without observing the conformational changes of a protein–ligand complex directly, it is possible to obtain the different conformations as CVs through the HMM modelling via TAE from interaction patterns derived from dynophores.

Testing the *dylightful* software package on the CDK2 structures 1KE5, 1KE7, and 6GUH could not derive a linear quantitative SAR model, however qualitative understanding of

the inhibition activities of different ligands against CDK2 could be derived. The findings align well with the published literature[5,6,51].

For the HCV system, the software gave new insights on the mechanism of inhibition. The main insight was, that the inhibition mechanism was again only rationalized within a dynamic induced fit picture of the whole supramolecular complex. Thus, the test shows that some understanding of the system could be gained that is inaccessible with methods applying static models. However, to derive more profound insights, there must be more data provided such that the algorithms can discretize the trajectories with higher confidence. Still, a new method of understanding resistance mechanism is the HCV NS3/4A quantitatively was developed using the *dylightful* software package. Therefore, *dylightful* takes the idea of structure-based ensemble-QSAR[1] methods further to an entirely dynamic SAR method.

Because the test systems were chosen according to the key activities of the drug design community, and it can be concluded that *dylightful* empowers drug designers by making

1. *De novo* drug design
2. Deriving insights on activity mechanism from biological assays
3. Understanding resistance mechanisms of known pathogens

easier.

4.2 Limitations

As shown, the HMM modelling via TAE is good for qualitative analysis, but quantitative SAR is difficult in a low data regime. Thus, MD analysis of larger datasets is necessary. However, the analysis time would then also rise. Likewise, it is not clear, how more frames and bigger sample sizes correlate to the TAE analysis and the behaviour of the software package must be investigated further in subsequent more general research. Again, it remains elusive whether quantitative SAR is at all possible, or if the tool has only strengths as a purely qualitative tool. As each supramolecular system is different, the number of states representing the eigenvalues of the Hamiltonian of the simulated system is not necessarily the same throughout a series of ligands. Moreover, the number of determined states does not necessarily correlate with the activity in a straightforward or naive manner.

In the current development stage, the tool is in a prototype stage. The tool might be applied

by trained users to novel systems easily. However, the tool is not yet production ready in a fire-and-forget application. Besides, again, it remains elusive if the tool can eliminate the need of skilled and trained users to assess the results of *in silico* experiments.

The quantitative SAR analysis, on the other hand, is limited to the experimental accuracy of the system. Furthermore, the SAR analysis using dylightful can't rely on easy heuristics. Neither in the case of HCV, nor in the case of, CDK2 it was easy to derive easy heuristics. Likewise, doing quantitative SAR seems to result in non-linear relationships in the case of CDK2.

The *dylightful* software demonstrates the need for the routine application of dynamic models in drug design and demands for dynamic instead of static experimental methods, too. To synthesize the computational and experimental modelling, time resolved methods are needed. The necessary time-resolved methods could include NMR, and Förster resonance energy transfer microscopy.

4.3 Outlook

Improvements of the software in later work should include better usability through GUIs and where possible reduction of the state contamination observed during the testing of the systems $ZIKV^{Prot}$, CDK2, and HCV NS3/4A. More detailed studies regarding resistance mechanisms of viral and bacterial pathogens would be very interesting.

Moreover, it has been demonstrated that eigenvalues of the Hamiltonian of the dynamic system were extracted. It would be interesting if the software *dylightful* can do similar modelling on coarse-grained simulations, such as QM-MM simulations. Still, a study on plain quantum chemical simulations, e.g., time dependent DFT would be interesting. Studies on further applications in time-independent quantum chemistry are mandatory.

The software is not limited to classical ligands and could also model protein–protein or protein–biomolecule interactions. Modelling complexes of biomolecules is indeed a very intriguing topic because many biological complexes involve orchestrated biomolecules. Mostly, it has been demonstrated that in the case of CDK2 the understanding of the mechanisms of inhibition is limited when studying isolated protein-ligand complexes and a deepened study with CDK complexed with, e.g., cyclin A could give further insights.

To improve the sampling and ease the determination of a meaningful amount of CV, the sampling techniques during the simulation could be altered and compared to the origi-

nal test systems. For example, replica exchange or simulated tempering could be applied. Likewise, shifting from the Desmond[54,55] software to open algorithms for improved transparency, and reproducibility is an interesting path to explore.

Besides, quantitative SAR pipelines would require more complicated modelling approaches, than linear regression. A promising approach might be QSAR through graph-based neural networks.

Finally, it would be interesting if the software can be integrated in artificial intelligence pipelines to analyse or perform (predict) simulations. There are several approaches of predicting force fields[61–64]. Predicting quantum mechanical properties through deep learning has become popular, too, such that the gold standard of coupled cluster level of theory was reached in 2019[65]. On top, simulations based on graph-deep learning estimators were realised[66]. The software *dylightful* could help to accelerate research and enable innovations in the field of performing graph-based simulations, as the central representation of the new dynamic interaction representation (termed markophore) is consisting of two graphs (however a state graphs), too.

Bibliography

[1] X. Q. Sun, L. Chen, Y. Z. Li, W. H. Li, G. X. Liu, Y. Q. Tu, Y. Tang, *Acta Pharmacologica Sinica* **2014**, *35*, 301–310.

[2] D. Sydow, *Dynophores: Novel Dynamic Pharmacophores*, **2015**.

[3] M. Janežič, K. Valjavec, K. B. Loboda, B. Herlah, I. Ogris, M. Kozorog, M. Podobnik, S. G. Grdadolnik, G. Wolber, A. Perdih, *International Journal of Molecular Sciences* **2021**, *22*, DOI 10.3390/ijms222413474.

[4] S. Pach, T. M. Sarter, R. Yousef, D. Schaller, S. Bergemann, C. Arkona, J. Rademann, C. Nitsche, G. Wolber, *ACS Medicinal Chemistry Letters* **2020**, *11*, 514–520.

[5] H. N. Bramson, W. D. Holmes, R. N. Hunter, K. E. Lackey, B. Lovejoy, M. J. Luzzio, V. Montana, W. J. Rocque, D. Rusnak, L. Shewchuk, J. M. Veal, J. Corona, D. H. Walker, L. F. Kuyper, S. T. Davis, S. H. Dickerson, M. Edelstein, S. V. Frye, R. T. Gampe, P. A. Harris, A. Hassell, *Journal of Medicinal Chemistry* **2001**, *44*, 4339–4358.

[6] D. J. Wood, S. Korolchuk, N. J. Tatum, L. Z. Wang, J. A. Endicott, M. E. Noble, M. P. Martin, *Cell Chemical Biology* **2019**, *26*, 121–130.

[7] K. P. Romano, A. Ali, C. Aydin, D. Soumana, A. Özen, L. M. Deveau, C. Silver, H. Cao, A. Newton, C. J. Petropoulos, W. Huang, C. A. Schiffer, *PLoS Pathogens* **2012**, *8*, 22.

[8] D. Soumana, A. Newton, C. J. Petropoulos, W. Huang, C. A. Schi, *ACS Chemical Biology* **2013**, *8*, 1469–1478.

[9] A. Mueller, *wordcloud*, version 2018.06, **2021**.

[10] F. Noé, C. Clementi, *Current Opinion in Structural Biology* **2017**, *43*, 141–147.

[11] Stanford University, *MSMBuilder*, version 3.8.0, **2017**.

[12] F. Noé, H. Wu, J. H. Prinz, N. Plattner, *Journal of Chemical Physics* **2013**, *139*, DOI 10.1063/1.4828816.

[13] F. Noé, C. Clementi, *Journal of Chemical Theory and Computation* **2015**, *11*, 5002–5011.

[14] C. Wehmeyer, F. Noé, *Journal of Chemical Physics* **2018**, *148*, DOI 10.1063/1.5011399.

[15] W. Zucchini, I. MacDonald, *Hidden Markov Models for Time Series: An Introduction Using R*, Taylor & Francis, **2009**.

[16] I. Miklós, I. M. Meyer, *BMC Bioinformatics* **2005**, *6*, 231.

[17] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, E. Teller, *The Journal of Chemical Physics* **1953**, *21*, 1087–1092.

[18] G. M. Torrie, J. P. Valleau, *Journal of Computational Physics* **1977**, *23*, 187–199.

[19] G. Pérez-Hernández, F. Paul, T. Giorgino, G. De Fabritiis, F. Noé, *Journal of Chemical Physics* **2013**, *139*, DOI 10.1063/1.4811489.

[20] C. R. Schwantes, V. S. Pande, *Journal of Chemical Theory and Computation* **2013**, *9*, 2000–2009.

[21] P. Baldi, K. Hornik, *Neural Networks* **1989**, *2*, 53–58.

[22] M. A. Kramer, *AIChE Journal* **1991**, *37*, 233–243.

[23] R. Lister, J. Stone in Proceedings of ICNN'95 - International Conference on Neural Networks, *Vol. 1*, **1995**, 237–241 vol.1.

[24] W. Chen, H. Sidky, A. L. Ferguson, *Journal of Chemical Physics* **2019**, *151*, DOI 10.1063/1.5112048.

[25] T. Langer, G. Wolber, *Drug Discovery Today: Technologies* **2004**, *1*, 203–207.

[26] G. Wolber, A. A. Dornhofer, T. Langer, *Journal of Computer-Aided Molecular Design* **2006**, *20*, 773–788.

[27] P. Ehrlich, *Berichte der deutschen chemischen Gesellschaft* **1909**, *42*, 17–47.

[28] C. G. Wermuth, C. R. Ganellin, P. Lindberg, L. a. Mitscher, *Pure Appl. Chem.* **1998**, *70*, 1129–1143.

[29] G. Schneider, W. Neidhart, T. Giller, G. Schmid, *Angewandte Chemie - International Edition* **1999**, *38*, 2894–2896.

[30] Y. Hu, D. Stumpfe, J. Bajorath, *Journal of Medicinal Chemistry* **2017**, *60*, 1238–1246.

[31] S. M. Kohlbacher, T. Langer, T. Seidel, *Journal of Cheminformatics* **2021**, *13*, 1–14.

[32] F. A. M. Opo, M. M. Rahman, F. Ahammad, I. Ahmed, M. A. Bhuiyan, A. M. Asiri, *Scientific Reports* **2021**, *11*, 1–18.

[33] G. Wolber, T. Langer, *Journal of Chemical Information and Modeling* **2005**, *45*, 160–169.

[34] M. Hoffmann, M. Scherer, T. Hempel, A. Mardt, B. de Silva, B. E. Husic, S. Klus, H. Wu, N. Kutz, S. L. Brunton, F. Noé, *Machine Learning: Science and Technology* **2022**, *3*, 015009.

[35] E. Ries, *The Lean Startup: How Constant Innovation Creates Radically Successful Businesses*, Portfolio Penguin, **2011**.

[36] pytest.org, *pytest-cov*, version 3.0.0, **2021**.

[37] N. Michaud-Agrawal, E. J. Denning, T. B. Woolf, O. Beckstein, *Journal of computational chemistry* **2011**, *32*, 2319–2327.

[38] R. Gowers, M. Linke, J. Barnoud, T. Reddy, M. Melo, S. Seyler, J. Domański, D. Dotson, S. Buchoux, I. Kenney, O. Beckstein, *Proceedings of the 15th Python in Science Conference* **2016**, 98–105.

[39] MDAnalysis, *MDAnalysis*, version 2.0.0, **2021**.

[40] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, *Journal of Machine Learning Research* **2011**, *12*, 2825–2830.

[41] scikit-learn.org, *scikit-learn*, version 1.0.2, **2021**.

[42] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T. E. Oliphant, *Nature* **2020**, *585*, 357–362.

[43] NumPy, *numpy*, version 1.22.0, **2021**.

[44] Wes McKinney in Proceedings of the 9th Python in Science Conference, (Eds.: Stéfan van der Walt, Jarrod Millman), **2010**, pp. 56 –61.

[45] pandas development team, *pandas-dev/pandas: Pandas*, version latest, **2020**.

[46] hmmlearn, *hmmlearn*, version latest, **2021**.

[47] M. Malumbres, M. Barbacid, *Nature Reviews Cancer* **2001**, *1*, 222–231.

[48] M. Malumbres, *Physiological Reviews* **2011**, *91*, 973–1007.

[49] S. Lapenna, A. Giordano, *Nature Reviews Drug Discovery* **2009**, *8*, 547–566.

[50] A. Echalier, A. J. Hole, G. Lolli, J. A. Endicott, M. E. Noble, *ACS Chemical Biology* **2014**, *9*, 1251–1256.

[51] J. Chen, L. Pang, W. Wang, L. Wang, J. Z. Zhang, T. Zhu, *Journal of Biomolecular Structure and Dynamics* **2020**, *38*, 985–996.

[52] C. C. Group, *Molecular Operating Environment (MOE)*, version 2020.09, **2022**.

[53] Schrödinger, *Maestro*, version 2022-1, **2022**.

[54] N. D. E. Shaw Research, New York, *Desmond Molecular Dynamics System*, version 2021-4, **2022**.

[55] SC '06: Proceedings of the 2006 ACM/IEEE Conference on Supercomputing, Association for Computing Machinery, Tampa, Florida, **2006**.

[56] R. T. McGibbon, K. A. Beauchamp, M. P. Harrigan, C. Klein, J. M. Swails, C. X. Hernández, C. R. Schwantes, L.-P. Wang, T. J. Lane, V. S. Pande, *Biophysical Journal* **2015**, *109*, 1528 –1532.

[57] L. Rong, H. Dahari, R. M. Ribeiro, A. S. Perelson, *Science Translational Medicine* **2010**, *2*, DOI 10.1126/scitranslmed.3000544.

[58] C. Sarrazin, S. Zeuzem, *Gastroenterology* **2010**, *138*, 447–462.

[59] C. D. Jones, D. M. Andrews, A. J. Barker, K. Blades, P. Daunt, S. East, C. Geh, M. A. Graham, K. M. Johnson, S. A. Loddick, H. M. McFarland, A. McGregor, L. Moss, D. A. Rudge, P. B. Simpson, M. L. Swain, K. Y. Tam, J. A. Tucker, M. Walker, *Bioorganic Medicinal Chemistry Letters* **2008**, *18*, 6369–6373.

[60] L. M. Stevenson, M. S. Deal, J. C. Hagopian, J. Lew, *Biochemistry* **2002**, *41*, 8528–8534.

[61] J. Wang, S. Olsson, C. Wehmeyer, A. Pérez, N. E. Charron, G. De Fabritiis, F. Noé, C. Clementi, *ACS Central Science* **2019**, *5*, 755–767.

[62] O. T. Unke, S. Chmiela, H. E. Sauceda, M. Gastegger, I. Poltavsky, K. T. Schütt, A. Tkatchenko, K. R. Müller, *Chemical Reviews* **2021**, *121*, 10142–10186.

[63] J. Wu, Y. Zhang, L. Zhang, S. Liu, *Physical Review B* **2021**, *103*, 1–26.

[64] J. Wu, L. Bai, J. Huang, L. Ma, J. Liu, S. Liu, *Physical Review B* **2021**, *104*, 1–31.

[65] J. S. Smith, B. T. Nebgen, R. Zubatyuk, N. Lubbers, C. Devereux, K. Barros, S. Tretiak, O. Isayev, A. E. Roitberg, *Nature Communications* **2019**, *10*, 1–8.

[66] B. E. Husic, N. E. Charron, D. Lemm, J. Wang, A. Pérez, M. Majewski, A. Krämer, Y. Chen, S. Olsson, G. De Fabritiis, F. Noé, C. Clementi, *Journal of Chemical Physics* **2020**, *153*, DOI 10.1063/5.0026133.