Chapter 3

Molecular Similarity

Molecular similarity is a widely used concept in “rational” drug design. Its use is based on the principle that structurally more similar molecules are more likely to exhibit similar properties than structurally less similar molecules [21]. Hence, searching for functionally similar molecules, which is a common aim in drug design, can be accomplished by searching for structurally similar molecules. Molecular similarity, however, is not a property of the molecules themselves but depends on an external criterion which defines the similarity [21]. In the context of drug design, molecular similarity is defined by the ability of the molecules to bind to the active site of some protein. On the one hand, this ability is driven by the possibility of the molecules to adopt conformers that are complementary to the active site of the protein. On the other hand, the binding is determined by the molecules' physico-chemical properties. Consequently, a molecular similarity measure capable of distinguishing between active and inactive molecules needs to take into account both the molecules' shapes and their physico-chemical properties.

In terms of binding of a drug molecule to the active site of a target protein, the shape of a molecule is best represented by some kind of molecular surface. Yet, most of the existing similarity measures are based on the molecular *scaffold* rather than on the molecule's outward facing surface. The quiet assumption underlying the use of scaffold based similarity measures is that if two molecules have a similar scaffold, they most probably also have a similar surface and, hence, similar binding properties. However, the reverse is not true. Two molecules having a similar surface do not necessarily have a similar scaffold. Hence, scaffold based approaches do not allow to identify molecules with similar binding properties yet dissimilar scaffolds, whereas surface based approaches do.

Many similarity measures have been proposed, which can be classified according to different categories, e.g. the dimensionality of the molecular structure representation used for determining the similarity. According to this category, we subdivided the similarity methods into methods based on the 2-dimensional structure, namely the molecular graph, and methods based on the 3-dimensional structure. Similarity methods belonging to the former class will be described in Section 3.1. Among these methods are exact and approximate graph isomorphism algorithms, but also feature trees and 2-dimensional fingerprints. The class of 3-dimensional similarity methods is much larger, and more diverse approaches are used. Hence, we decided to subdivide the rest of this chapter into basic concepts for 3-dimensional structure similarity, described in Section 3.2, and applications based on these concepts, described in Section 3.3. Of particular interest to us are methods based on surface similarity. Therefore, previous work in this field will be presented in greater detail than other methods.

3.1 Molecular Graph Based Similarity

In this section we will look at methods to compute the similarity of two molecular structures based on their molecular graph (cf. Definition 2.1.2). The first class of methods (Sections 3.1.1 and 3.1.2) directly compares the molecular graphs with each other and identifies common (or similar) subgraphs. The methods of this class relate parts of one molecule to parts of the other molecule. They define a matching which can be used to overlay the molecules. The second class (Section 3.1.3) generates so called fingerprints from the molecular graphs. Fingerprints constitute condensed representations of molecular graphs, which can be more easily compared.

3.1.1 Maximum Common Subgraph (MCS) Algorithms

An excellent review on this topic can be found in [145]. For an introductory text to graph theory, see, e.g., [48].

To the earliest methods of comparing molecular structures belongs the *substructure search*, which for a given molecular structure S checks whether its molecular graph $G(S)$ (cf. Definition 2.1.2) is a subgraph of the molecular graph $G(M)$ of some molecule M.

Definition 3.1.1 (Graph Isomorphism [145])**.** Two graphs G' and G'' are said to be isomorphic if there exists a one-to-one correspondence between their vertices and an edge exists between two vertices in G' if and only if an edge exists between their corresponding vertices in G''.

With the above definition we can more precisely formulate the substructure search as search for a subgraph $G(S') \subseteq G(M)$ which is isomorphic to $G(S)$. Note that the subgraph isomorphism problem is known to be NP-complete. However, for trees and almost trees of bounded degree it has been shown to be solvable in polynomial time [5]. Molecular graphs belong to this class of graphs.

Definition 3.1.2 (Induced Subgraph [48])**.** If $G' \subseteq G$ and $\forall v, w \in V(G') : (v, w) \in E(G) \Rightarrow (v, w) \in E(G')$, then G' is an *induced subgraph* of G (cf. Figure 3.1(b)).

Definition 3.1.3 (Maximum Common Induced Subgraph [145])**.** A graph G is a *common induced subgraph* (CIS) of two graphs G' and G'', if G is isomorphic to induced subgraphs of G' and G''. G is a *maximum common induced subgraph* (MCIS), if G is the CIS with the largest number of vertices.

Definition 3.1.4 (Maximum Common Edge Subgraph [145])**.** A graph G is a *common edge subgraph* (CES) of two graphs G' and G'', if G is isomorphic to subgraphs of G' and

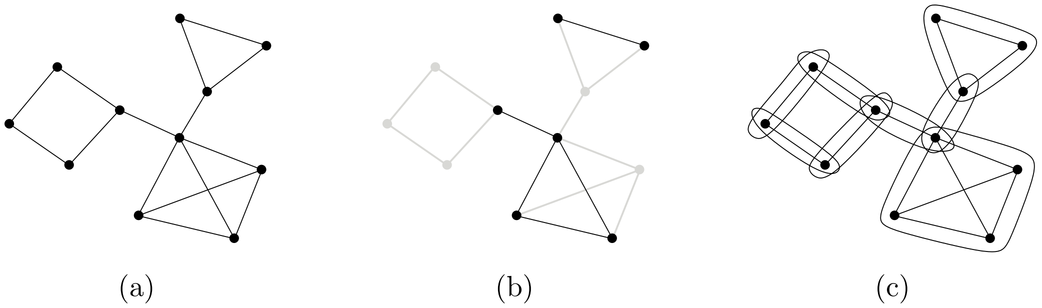

Figure 3.1: (a) Some arbitrary graph G. (b) The highlighted subgraph represents the subgraph of G induced by the black vertices. (c) All cliques of G. There exists one clique of size 4, one clique of size 3, and 6 cliques of size 2.

G''. G is a *maximum common edge subgraph* (MCES), if G is the CES with the largest number of edges.

In the literature, algorithms for both the MCIS and the MCES problem appear. Although the MCES is more intuitive in terms of molecular similarity, the number of algorithms for the MCIS problem is much larger, since it is easier to solve. However, Whitney [171] proved already in 1932 that edge isomorphism of two graphs G' and G'' is induced by a vertex isomorphism of the so called line graphs of G' and G'' provided that no ΔY exchange occurs when transforming G' and G'' to their line graphs [145]. This means, that algorithms solving the MCIS problem can be used to solve the MCES problem, since ΔY exchanges can be easily accounted for.

The algorithms for solving the MCS (MCIS or MCES) problem can be divided into algorithms computing the MCS exactly and algorithms approximating the MCS. That is to say, for algorithms of the latter class it is not guaranteed, that they will find the MCIS or MCES, but generally a CIS or CES that is close to it in size.

Exact Algorithms

Probably the most widely used class of algorithms for determining the MCIS of two molecular graphs utilizes clique detection on the *modular product graph*, which is an NP-complete problem, too.

Definition 3.1.5 (Modular Product Graph)**.** The *modular product* of two graphs G' and G'' is defined on the vertex set $V(G') \times V(G'')$ with two vertices (u', u'') and (v', v'') being adjacent if $(u', v') \in E(G') \Leftrightarrow (u'', v'') \in E(G'')$. The modular product will be denoted by $G' \diamond G''$.

Remark 3.1.6 (Compatibility Graph)**.** In the case of labeled graphs, such as molecular graphs, the modular product of two graphs can be further restricted by requiring that the vertex and edge labels satisfy some compatibility criterion. Hence, in the case of molecular graphs, the modular product graph is often called *compatibility graph*.

Definition 3.1.7 (Clique)**.** Let $G = (V, E)$ be a graph. We call a subgraph $G' = (V', E') \subseteq G$ *complete*, if $\forall u, v \in V' : (u, v) \in E'$. A vertex subset $S \subseteq V$ is called a *clique*, if the induced subgraph $G'' = (S, E'') \subseteq G$ is complete and there does not exist a complete subgraph $G''' \supset G''$ of G which contains G''. Figure 3.1(c) shows all cliques of the graph displayed in Figure 3.1(a).

Each clique of the compatibility graph $G' \diamond G''$ corresponds to a CIS of the graphs G' and G'', whose size is equal to that of the corresponding clique. Consequently, searching for the largest clique in $G' \diamond G''$ is equivalent to searching for the MCS of G' and G''. The most widely used algorithm for the MCS problem of molecular graphs is that proposed by Bron and Kerbosch [30] which enumerates all cliques of the graph. It applies a branch-and-bound technique to prune the search tree, which makes it very efficient.

Other exact algorithms include the *backtracking* algorithms by McGregor [122] and Wong [174], and the algorithm by Akutsu [5], which is based on *dynamic programming.* For further details see [145].

Approximate Algorithms

Another class of algorithms tries to reduce the cost of determining the MCS by approximate schemes. Wagener and Gasteiger [168], e.g., apply a *genetic algorithm*, where the population represents intermediate solutions to the MCS problem. Selection techniques such as cross-over and mutations constantly modify the population. According to the notion of Darwinian survival, the fitter an instance the more likely it is to survive. Consequently, at the end of the genetic algorithm run, most of the solutions that remain belong to the best solutions for the MCS problem. Other algorithms apply *simulated annealing* [98, 11], or *neural network optimization* [154]. For a more complete review, see [145].

3.1.2 Feature Trees

Feature trees, which are a more abstract way of representing a molecule by means of a graph, were introduced by Rarey and Dixon [142]. As the name suggests, feature trees are trees, i.e. connected acyclic graphs. Feature trees apply the concept of *super-atoms* introduced by Yuan et al. [175]. Here, each super-atom represents a group of atoms with some specific property, the feature. The nodes of the tree represent the super-atoms, where each atom belongs to at least one super-atom. Super-atoms that have atoms in common or contain atoms that are connected in the molecular graph will be connected by an edge in the feature tree. The difficulty lies in the choice of the atoms to be grouped into super-atoms. If the atoms to be grouped are not carefully chosen, the new graph will still contain cycles, which contradicts the notion of a tree.

Feature trees can represent the molecule at various levels of detail, hence a hierarchy can be build using feature trees. This is a central property of the feature tree concept which is exploited in the matching step. Feature trees are generated using a method based on biconnected components. No a priori knowledge of what constitutes an important feature is needed. Rarey and Dixon [142] propose two algorithms for comparing feature trees, namely *split search* and *match search*, both of which are based on subtree matching, but

Table 3.1: Some molecular graph based similarity measures collected by Raymond and Willett [144]. Here, $|G_1|$ and $|G_2|$ denote the size of the graphs G_1 and G_2, respectively, and $|G_{12}|$ denotes the size of the MCES of G_1 and G_2.

ID	Measure	Formula	Range
MCS1	Wallis et al.	$\frac{\vert G_{12}\vert}{\vert G_1\vert+\vert G_2\vert-\vert G_{12}\vert}$	[0.0,1.0]
MCS2	Bunk and Shearer	$\frac{\vert G_{12}\vert}{\max(\vert G_1\vert,\vert G_2\vert)}$	[0.0,1.0]
MCS3	Asymmetric	$\frac{\vert G_{12}\vert}{\min(\vert G_1\vert,\vert G_2\vert)}$	[0.0,1.0]
MCS4	Normalized Johnson	$\frac{2\cdot\vert G_{12}\vert}{\vert G_1\vert+\vert G_2\vert}$	[0.0,1.0]
MCS5	Johnson	$\frac{\vert G_{12}\vert^2}{\vert G_1\vert\cdot\vert G_2\vert}$	[0.0,1.0]
MCS6	Sokal-Sneath	$\frac{\vert G_{12}\vert}{2\vert G_1\vert+2\vert G_2\vert-3\vert G_{12}\vert}$	[0.0,1.0]
MCS7	Kulczynski	$\frac{\vert G_{12}\vert\cdot(\vert G_1\vert+\vert G_2\vert)}{2\cdot\vert G_1\vert\cdot\vert G_2\vert}$	[0.0,1.0]
MCS8	McConnaughey	$\frac{\vert G_1\vert\cdot\vert G_{12}\vert+\vert G_2\vert\cdot\vert G_{12}\vert-\vert G_1\vert\cdot\vert G_2\vert}{\vert G_1\vert\cdot\vert G_2\vert}$	[-1.0,1.0]

differ in the way matchings are computed and in the matchings themselves. For example, the split search algorithm might generate null-matches, in which some nodes, particularly nodes in the "middle" of the feature tree, are not matched. In order to circumvent these null-matches, the match search algorithm was developed.

One advantage of the feature tree approach is that similarities between molecules can be detected even if the molecular graphs are less similar. That is to say, feature trees are more robust in terms of substructure matching than MCS algorithms. Moreover, the feature tree approach is very fast, and therefore, it is applicable to data base screening.

3.1.3 2D Fingerprints

Fingerprints constitute 1-dimensional representations of molecular structures, since the molecular structures are encoded as strings, usually bit-strings, which can be efficiently compared. Fingerprints have been designed to efficiently search large data bases for molecules with certain structural features.

The earlier fingerprints, i.e. 2D fingerprints, are all based on the molecular graph. In recent years, fingerprints describing the 3-dimensional structure of the molecule have also been developed. Some of these will be described in subsequent sections. In this section, however, we will only deal with 2D fingerprints. A good introduction to 2D fingerprints can be found in [38].

The precursor of 2D fingerprints are *structural keys*, which encode the molecular structure into bit-strings using predefined substructures. Each bit of a structural key represents a certain substructure. If the substructure is present in the molecular graph, the bit is set, otherwise it is not set. There are certain drawbacks of this approach. One drawback is, that for each substructure to be represented, a substructure search needs to be carried out for each molecule in the data base. This is expensive in terms of computational cost. A second drawback is, that the success of a similarity search strongly depends on the choice of the substructures. Finally, in order to be representative, structural keys need to be very long, since the number of possible substructures is very large and including more substructures increases the specificity of the structural key. However, the longer the structural key is, the more bits will remain unset; hence, the density of a structural key will be rather low.

2D fingerprints address the above mentioned drawbacks. Fingerprints do not use predefined patterns. Instead, the patterns included in a fingerprint of some molecule are generated directly from the molecule's molecular graph. This is done by generating all groups of atoms that are connected by 0 (single atoms) up to N bonds, where N typically is in the range between 5 and 7. For each group of atoms, i.e. pattern, a bit-string is generated according to the atoms present in the group and the bonds connecting these atoms. The number of bits used for representing a single pattern is typically 4 or 5. In addition to the pattern bit-string, a position within the fingerprint bit-string is generated, i.e. the pattern is "hashed". The pattern bit-string is then added to the fingerprint bit-string using the logical OR operation. This means, that bits are shared between patterns in the fingerprint. Consequently, fingerprints cannot be used to state exactly whether a certain pattern is present in the molecular graph. If the bits for some pattern are set in the fingerprint, this pattern is only present in the molecular graph with some probability. This means, that if two fingerprints show a high similarity, the underlying molecular structures are not necessarily very similar. Similarity of the molecular structures is only present with some probability, which generally is rather high. But the opposite is true, i.e., if two fingerprints are very dissimilar, then their underlying molecular structures are very dissimilar, too. Hence, fingerprints can be efficiently used to filter out dissimilar molecular structures.

3.1.4 Similarity Measures

A large number of similarity measures has been proposed both for MCS based similarity (cf. Table 3.1) and for fingerprint similarity (cf. Table 3.2). Raymond and Willett [144] compared some of these similarity measures using the MCES algorithm RASCAL [143] as representative for MCS algorithms and three fingerprint representations, namely BCI [18], Daylight [37], and Unity [165]. Furthermore, they compared the effectiveness of these algorithms to identify bioactive compounds in a database of 2D structures. They found, that RASCAL is comparable in effectiveness to the fingerprint based methods. An interesting observation is that the set of active compounds identified by RASCAL and the fingerprint methods were, at least partially, non-identical. This suggests the use of RASCAL as complement to the widely used fingerprint methods.

Table 3.2: Some fingerprint based similarity measures collected by Raymond and Willett [144] and Holliday et al. [82]. Here, a and b denote the number of unique bits set in the fingerprints representing molecules A and B, respectively. The variable c denotes the number of bits set in both fingerprints, and d is the number of bits not set in both fingerprints.

ID	Measure	Formula	Range
F1	Cosine	$\frac{c}{\sqrt{(a+c)(b+c)}}$	[0.0,1.0]
F2	Dice	$\frac{2 \cdot c}{(a+c)+(b+c)}$	[0.0,1.0]
F3	Euclid	$\sqrt{\frac{c+d}{a+b+c+d}}$	[0.0,1.0]
F4	Forbes	$\frac{c \cdot (a+b+c+d)}{(a+c)(b+c)}$	$[0.0, \infty)$
F5	Hamman	$\frac{(c+d)-(a+b)}{a+b+c+d}$	[-1.0,1.0]
F6	Jaccard/Tanimoto	$\frac{c}{a+b+c}$	[0.0,1.0]
F7	Kulczynski	$0.5(\frac{c}{a+c} + \frac{c}{b+c})$	[0.0,1.0]
F8	Manhattan	$\frac{a+b}{a+b+c+d}$	[0.0,1.0]
F9	Matching	$\frac{c+d}{a+b+c+d}$	[0.0,1.0]
F10	Pearson	$\frac{c \cdot d - a \cdot b}{\sqrt{(a+c)(b+c)(a+d)(b+d)}}$	[-1.0,1.0]
F11	Rogers-Tanimoto	$\frac{c+d}{(a+b)+(a+b+c+d)}$	[0.0,1.0]
F12	Russell-Rao	$\frac{c}{a+b+c+d}$	[0.0,1.0]
F13	Simpson	$\frac{c}{\min((a+c),(b+c))}$	[0.0,1.0]
F14	Yule	$\frac{c \cdot d - a \cdot b}{c \cdot d + a \cdot b}$	[-1.0,1.0]
F15	Sokal-Sneath	$\frac{c}{2(a+c)+2(b+c)-3c}$	[0.0,1.0]
F16	McConnaughey	$\frac{(a+c) \cdot c+(b+c) \cdot c-(a+c) \cdot (b+c)}{(a+c) \cdot (b+c)}$	[-1.0,1.0]

3.2 Basic Concepts for 3-Dimensional Structure Similarity

In the previous section we reviewed molecular similarity methods based on the two-dimensional structure of molecules, namely the molecular graph representation. In this and the succeeding section, we want to look at similarity methods based on the 3-dimensional structure of molecules. Since this thesis only deals with drug-sized molecules, we will concentrate on methods that are mainly applied to this class of molecules.

It is outside the scope of this thesis to review all work in this field. We therefore refer the interested reader to a couple of very good review articles that appeared over the last two decades. Brint and Willett [28], and Martin et al. [116] review the developments in the field of 3-dimensional structure alignment predating the year 1990. A good collection of articles and reviews on this subject can also be found in [102]. A further excellent review on structural alignment roughly covering the years between 1994 and 2000 has been written by Lemmen and Lengauer [112]. Bender [21] gives a good introduction to similarity measures, in particular transformation-invariant similarity measures developed recently.

In this section we introduce basic concepts used for 3-dimensional similarity searching. In Section 3.3 we will then give examples of algorithms utilizing these concepts.

3.2.1 Clique Detection

In Section 3.1.1 we saw how clique detection on the compatibility graph (cf. Remark 3.1.6) of two molecular graphs G' and G'' can be utilized to find common subgraphs of G' and G''. In this section we will see how clique detection can be used to identify subsets of points of two point sets P and Q with similar inter-point distances. Here, P and Q can be sets of atomic nuclei, but they can also be other point sets used for representing two molecules, such as, e.g., surface points. In order to utilize clique detection for the identification of similar subsets of points, a *distance compatibility graph* needs to be constructed.

Definition 3.2.1 (Distance Compatibility Graph)**.** Let $d > 0$, $d \in \mathbb{R}$, be a constant and let $P = \{p_1, \dots, p_m\}$ and $Q = \{q_1, \dots, q_n\}$ be two point sets with coordinates $\mathbf{X}(P) = \{\mathbf{x}(p_i)\}_{i=1}^{m}$ and $\mathbf{X}(Q) = \{\mathbf{x}(q_i)\}_{i=1}^{n}$. Then, the *distance compatibility graph* of P and Q w.r.t. d is defined on the vertex set $P \times Q$ and has the following property: two vertices (p', q') and (p'', q'') are connected by an edge, if and only if the inequality

$$\big| \|\mathbf{x}(p') - \mathbf{x}(p'')\| - \|\mathbf{x}(q') - \mathbf{x}(q'')\| \big| \leq d$$

is satisfied. We denote the distance compatibility graph of P and Q w.r.t. d by $G(P,Q)_d = (P \times Q, E_d)$.

Each clique in the distance compatibility graph $G(P,Q)_d$ gives us two subsets $P' \subseteq P$ and $Q' \subseteq Q$ with similar distance matrices together with a one-to-one correspondence between the points of P' and Q'. Moreover, since a clique represents a maximal complete subgraph (cf. Definition 3.1.7), this one-to-one correspondence is also maximal, i.e., it cannot be extended by a further point pair such that the distance constraint imposed by d remains satisfied. Even though clique detection on the distance compatibility graph has proved to be very versatile, there exist two problems with it. First, since space reflections are not represented by distance matrices, clique detection also identifies space-reflected subsets of points. However, space reflections can be easily identified and eliminated. The second, more serious, problem is the computational cost. Clique detection on arbitrary graphs is known to be NP-complete. Thus, it is only applicable to small or sparse graphs. There are two possibilities to thin-out the distance compatibility graph. First, we can decrease the distance threshold d. Second, we can add further constraints, such as the requirement that corresponding points have common properties. These properties can vary according to the point sets to which clique detection is applied. For example, if the points are atoms, we could require the corresponding atoms to have equal type or charge. Due to the fact that clique detection is only applicable to small or sparse graphs, it is only of limited use for detecting structural similarities in large molecules such as proteins.

The most commonly used clique detection algorithm in the field of structural alignment is that proposed by Bron and Kerbosch [30]. As mentioned earlier, this algorithm enumerates all cliques of a graph. If one is interested in the largest clique only, there exist other approximate algorithms (see, e.g., [134, 10]).

3.2.2 Geometric Hashing

Geometric hashing is a technique originating from the field of computer vision. Originally, it was developed for matching geometric features against a data base of such features. A good introduction to geometric hashing can be found in [173]. As a method based on the indexing approach, geometric hashing was introduced by the work of Schwartz and Sharir [90, 156]. Nussinov and Wolfson [131] then were the first to apply the concept of geometric hashing to structural molecular data.

Geometric hashing is a two-step approach. In the first step, a highly redundant representation of the object data base is generated offline and stored in a hash table. Even though this representation is invariant under affine transformations, in the context of molecular data, we only need invariance under rotation and translation. In the second step, the recognition step, the hash table is queried with structural features from a further object, e.g. a molecule. Each hit in the table identifies a transformation of the query object with an object stored in the hash table. A large number of hits with the same object under the same transformation reflects a high similarity of the objects under consideration. In the context of molecular data, transformations that receive many hits are very likely to overlay essential structural features of the molecules in question.

3.2.3 Genetic Algorithms

Genetic algorithms are not specific to the field of structural analysis but represent a global optimization technique that has been applied in many fields [61]. Genetic algorithms simulate the evolution of populations of points in fitness space. The populations of points are coded via “chromosomes”, which are subject to manipulation operators mimicking genetic operators and sexual reproduction. The typical operators are *mutation* and *crossover*. While the mutation operator randomly modifies chromosomes of a single individual, the cross-over operator represents a recombination of parts of the chromosomes from two distinct individuals. The information encoded in the chromosomes varies from application to application. During the evolution of a population, new individuals are generated by applying the genetic operators. At the end of each such evolutionary step, all individuals are evaluated according to a fitness function, which gives the probability for each individual to survive. According to this probability, individuals do or do not carry on to the next generation.

In the context of structural molecular alignment, the chromosomes encode intermolecular matches and orientational degrees of freedom [112]. If molecular flexibility is considered, the chromosomes also encode torsional degrees of freedom [75].

Genetic algorithms have been shown to provide promising results in the entire area of computational structural biology [47]. While they are very flexible and enable the evolution of complex objects, genetic algorithms remain quite slow, which represents the main limitation to the use of this class of algorithms.

3.2.4 Point Matching

Given two sets of points in 3-dimensional space, the problem of determining a matching between these point sets can be described by means of a *bipartite graph.* A bipartite graph is a graph whose vertex set can be divided into two subsets, such that there do not exist edges between vertices of the same subset. The problem is now to find a *matching* of the bipartite graph maximizing a given scoring function, where a matching is a set of edges, such that no two edges are incident to the same vertex. In the discrete case, the optimal matching is the one with the largest number of edges satisfying some distance constraint for the matched points. In the continuous case, the distances of the points could be used as edge weights and the optimal matching is the one maximizing a scoring function w.r.t. the edge weights. The scoring function should be such that its values are large, if the edge weights are small.

Given an initial alignment of two point sets, point matching can be used to identify the optimal matching of the two point sets w.r.t. the current alignment. The point matching scheme can be embedded into an iterative optimization algorithm, where in each step the optimal matching is computed which yields a new initial alignment for the next iteration. Iteration is carried out until no further improvement is gained. Akutsu [6] uses a dynamic programming technique for determining the optimal matching for a given initial alignment. This algorithm runs in $\mathcal{O}(mn)$ time, where m and n are the number of atoms in the two point sets. Kirchner [96] gives an exact algorithm based on *augmented paths* which runs in time $\mathcal{O}(n^3)$, where n is the number of atoms in the larger point set.

3.2.5 Contact Maps

A *contact map* [76] of a folded macromolecule with n residues is an $n{\times}n$-matrix C, where each entry is either 0 or 1. A value of 1 of the matrix element C_{ij} denotes a contact between residues i and j. Typically, two residues i and j are said to be in contact, if the distance between two heavy atoms, one from residue i and one from residue j, is smaller then a predefined distance threshold, e.g. 5 Å. Contact maps constitute an intuitive and simple framework which is complex enough to capture most important properties of the folding phenomenon [31]. Because of these properties, contact maps are also well-suited to compare molecular structures. Since contact maps capture the most important properties of folding, similarly folded macromolecules should have similar contact maps. Hence, a comparison of the contact maps of two molecules can be used to determine the similarity of the two molecular structures.

The contact map of a molecule can be regarded as the adjacency matrix of a graph, where each node represents a single residue. In order to compare two molecular structures, the aim is to compute the maximal overlap of their contact graphs such that the order of the residues is preserved. This is known as the *contact map overlap (CMO) optimization problem*, which has been proved to be NP-hard by Goldman et al. [63].

The CMO optimization problem can be transformed into a *maximum independent set (MIS)* problem on a suitable graph [106]. However, such a graph is typically very large and, hence, the MIS problem is hard to solve. For dense graphs it is restricted to a few hundred nodes. In 2001, Lancia et al. [106] developed a new algorithm for the CMO optimization problem based on *Integer Linear Programming (ILP)*. Their approach is much more efficient than previous approaches, and it can solve graph instances with 10,000 and more nodes by exploiting the particular characteristics of the constructed graph. In the ILP approach, the problem is formulated as a problem of maximizing linear functions of some integer variables. Subsequently, this problem is solved via *branch-and-bound*. This approach was a major step in solving the CMO optimization problem. However, it was still only capable of aligning macromolecules of up to 80 residues [31]. This limitation was overcome by the algorithm proposed by Caprara and Lancia [31] in 2002. Instead of solving the ILP using branch-and-bound, they employed a *Lagrangian relaxation (LR)* based approach. The relaxation is solved by computing a sequence of simple alignments, which can be done in quadratic time for each alignment, and computing the Lagrangian multipliers using sub-gradient optimization [31]. With this new approach it became possible to align macromolecules with 1000 residues and 2000 contacts.

Recently, Bauer et al. [13, 14] applied Lagrangian relaxation to a combined sequence-structure alignment of RNA molecules. This is possible without knowing the 3-dimensional structure of the RNA molecules, since for RNA molecules there exist reliable methods for secondary structure prediction and base-pair annotation.

3.3 3-Dimensional Structure Similarity Algorithms

The algorithmic approaches for determining the similarity of the 3-dimensional structures of molecules can be divided into two broad categories, namely transformation-invariant approaches and approaches that need an alignment of the molecular structures to determine their similarity. While the latter are computationally more demanding, the former sacrifice quality for efficiency.

In this thesis we only deal with drug-sized molecules. Hence, in this section we will only consider algorithms that can be applied to this class of molecules. We have divided the algorithms according to the molecular representation used for determining the molecular similarity. Each such class of algorithms has been further subdivided according to whether the algorithms compute an alignment of the molecular structure or a descriptor from the molecular representation. Where possible, the alignment algorithms were even further subdivided into rigid and flexible algorithms. While rigid algorithms keep the molecular structure fixed, flexible algorithms modify the molecular structure during alignment.

3.3.1 Atom Based Similarity

In this section we will look at molecular similarity methods based on the positions of the atomic nuclei. This class of methods seems to be the most widely used class of methods for 3-dimensional structural alignment.

Alignment Based Similarity

Rigid Alignment. Among the rigid alignment approaches, there exist many algorithms utilizing clique detection [28, 20, 127, 10, 101, 15]. Since clique detection is limited to small or sparse graphs, it is often only possible to identify small common substructures with this technique. Therefore, Miller et al. [127] and Baum [15] use clique detection to generate feasible initial alignments only, which are used as starting points for local optimization. Krämer et al. [101] use clique detection only for the alignment of fragments which then need to be reassembled to align the whole molecule. Because of the small size of the fragments, clique detection can be efficiently applied. Bahadur et al. [10] developed a maximum clique detection algorithm for matching C_α atoms of proteins. However, their approach also uses information about the sequence of the C_α atoms in the amino acid chain.

Several alignment algorithms have been proposed that use a point matching approach to match the atoms of two molecules [6, 57, 96, 15, 163]. Finn et al. [57] compute alignments using triples of atoms which are randomly sampled in one of the two molecules. Then, similar triples are searched for in the other molecule. According to these triples, the molecules are aligned using rms-fitting [89], and close atoms are identified as matches. The probabilistic scheme ensures that the optimal solution is only missed with some low probability. Akutsu [6], Kirchner [96], and Baum [15] use an iterative point matching scheme, which first generates various initial alignments, all of which are locally optimized. While Akutsu and Kirchner use triples of atoms to generate these initial alignments, Baum employs clique detection to identify small common substructures. Akutsu proposed several strategies to reduce the number of initial alignments, e.g. a probabilistic scheme similar to that of Finn et al. [57]. Since Akutsu applies his method to the alignment of proteins, he only considers C_α atoms. The common triples or substructures thus identified are rigidly aligned using rms-fitting [89], resulting in initial alignments. Akutsu applies dynamic programming to optimize each of these alignments. In contrast, Kirchner and Baum use a greedy optimization strategy, which yields very good results. In order to efficiently determine the optimal matching, Kirchner applies a branch-and-bound approach.

In [163], Thimm proposed a new algorithm which allows to quickly determine the optimal matching of two molecules, each given by a set of conformers (cf. Definition 2.4.1). However, his approach identifies only substructures representing connected subgraphs of the molecular graphs of the two molecules. Consequently, it is restricted to sets of molecules with similar scaffolds, i.e. similar molecular graphs. In virtual drug design, however, one is often particularly interested in molecules showing distinct scaffolds. The algorithm proposed by Thimm will fail to identify a high similarity for such molecules.

Rigid alignment approaches do not consider the flexibility of the molecules at hand. However, most of these methods allow to consider multiple conformers. Since these methods do not consider the flexibility of the molecules directly, they are called *semi-flexible* methods. One advantage of *semi-flexible* approaches is the decoupling of the alignment method from the conformation analysis. Conformation analysis can be carried out as a preprocessing step and hence, it needs to be carried out only once per molecule. Consequently, conformation analysis can be done in more detail than if it were performed for each alignment.

Flexible Alignment. In order to account for the flexibility of the molecules during the alignment procedure, different approaches have been proposed. While some algorithms sample the configurational space during the alignment phase [124, 75], other algorithms only consider the molecule's flexibility implicitly by representing the molecule into such a form that allows to easily recombine rigid fragments in the alignment phase [149]. Recently, a new method was proposed which utilizes precomputed conformers that are interpolated to allow for an optimal superposition of atoms of similar chemical type [71].

McMartin and Bohacek [124] developed the program TFIT, which uses a molecular superposition force field to flexibly align two molecules. The program tries to maximize the overlap of atoms which are of similar chemical type. During the alignment phase, a Monte Carlo torsion perturbation scheme is applied, followed by torsional energy minimization to identify similar conformations co-minimizing the energy of the molecules. Handschuh et al. [75] developed a genetic algorithm for the flexible pairwise alignment of molecules. Here, "chromosomes" encode transformational as well as torsional degrees of freedom, together with an atom matching. Each individual of the population is uniquely defined by its chromosomes. Two genetic operators, mutation and cross-over, simulate the evolution of the population representing the alignments. In her dissertation, Handschuh [74] extended this approach to multiple alignment. In addition to the genetic operators, she developed two non-genetic operators which she calls *creep* and *crunch*. These operators are specific to the problem of multiple structure alignment.

Rigoutsos et al. [149] use a hybrid method between geometric hashing and pose clustering [160]. Geometric hashing is applied to quickly generate hypotheses about the positions and orientations that would allow to match substructures of the molecules stored in the hash table and the query molecule. Using pose clustering, these hypotheses are than clustered to identify larger common substructures. The crux of this approach is that only a single conformer of each molecule needs to be stored in the hash table, thus reducing the size of the hash table and making the approach very efficient.

Recently, Gürler [71] described a new flexible alignment approach, termed FADO (**F**lexible **a**lignment and **do**cking), which uses an ensemble of precomputed conformers. Each molecular structure generated during the alignment optimization represents a linear combination of the torsional angles of these conformers. The structure of one of the two molecules to be aligned is fixed, while the structure of the other one is modified w.r.t. the linear coefficients obtained from the optimization procedure. A simplified Lennard-Jones potential is applied as molecular superposition force field. The energy of this potential is minimized using the local optimization approach Rprop (**R**esilient Back**prop**agation) [27], which has the advantageous property to smooth the highly jagged potential energy surface. Thus, even though the optimization procedure is a local one, minor local minima are overleaped. As the name Fado suggests, the approach can also be applied to flexible docking, where the ligand is flexibly aligned to a *pharma site* previously generated in the active site of a protein [71]. This new approach has shown to give very promising results.

Descriptor Based Similarity

In the *atom-mapping* method proposed by Pepperrell et al. [139], the inter-atomic distances between non-hydrogen atoms are computed and stored in a matrix, termed the *atom match matrix*, which represents the molecular descriptor. Similarity of two atom match matrices is determined by pairwise comparisons of two rows, one from each matrix. Using the Tanimoto coefficient [86], the degree of similarity is computed from these pairwise row comparisons.

Sheridan et al. [157] define the *atom pair descriptor*, which is an extension of the two-dimensional *atom pairs* proposed by Carhart et al. [32]. In the atom pair descriptor, all pairs of non-hydrogen atoms are considered, hence its name. Each atom pair is described by the types of the atoms plus their Euclidean distance. Here, the atom type can be any description of the atom's chemical properties. In order to be able to count the number of similar atom pairs, the distances are partitioned into bins. Then, for each pair of atom types we maintain a list of bins and count the number of atom pairs with a certain distance by computing its bin and adding a single count to this bin. To compute the similarity of two molecules, for each pair of atom types one compares the counts of their bins and computes a similarity value according to the similarity measure proposed in [32].

3.3.2 Pharmacophore Based Similarity

Alignment Based Similarity

Probably the most widely used program using pharmacophore points to identify similar molecular structures is DISCO (**DIS**tance **CO**mparison), developed by Martin et al. [115]. In a first step, the locations of pharmacophore points are computed with the program ALADDIN [50]. This program not only identifies atoms or pseudo atoms (replacing rings, e.g.) that might play an important role in the binding process, but also projections of the molecule to potential hydrogen bond donors or acceptors, and charged groups in the binding site. As the next step, clique detection is applied to identify subsets of the pharmacophore points in both molecules having similar inter-point distances. DISCO has proved to yield very good results. One limitation, however, is that it does not consider the whole shape of the molecule but only regions where pharmacophore points are present. Ignoring the molecular shape might in some cases miss the correct alignment of the molecules.

Iwase and Hirono [85] apply their program SUPERPOSE to the structural alignment of proteins. The functional groups of the two proteins to be compared are replaced by pseudo-atoms, thus reducing the number of points to be considered. The alignment is carried out by systematically translating and rotating the smaller molecule w.r.t. the fixed position of the larger one. For each transformation, a score is computed. The transformation yielding the largest score gives the best alignment.

Descriptor Based Similarity

The program 3DSEARCH, developed by Sheridan et al. [158], can be used to find pharmacophore patterns in a data base of potential drugs. For each non-hydrogen atom, its chemical type is generated, consisting of five fields: element type, number of neighbored non-hydrogen atoms, number of π-electrons, expected number of attached hydrogens, and formal charge. Also, rings are replaced by dummy atoms. Then, for each pair of atoms and dummy atoms, the distance is computed and the distance bin is determined. The atom pairs together with their distance bins form the keys. These keys are used to quickly search the data base and eliminate the largest part of it. For the remaining candidates, a more detailed geometric search can be carried out.

3.3.3 Volume Based Similarity

Volume based similarity approaches measure the similarity of the molecules' volumetric representations. More precisely, if there exists an orientation in which the volumes of two molecules overlap well, the molecules are considered to be similar. Thus, the aim of volume based alignment is to find an orientation such that the volume overlap is large. One limitation of volume based alignment becomes obvious if one considers the alignment of two molecules, where one of them is considerably larger than the other one. Then, an alignment of the two molecules would be optimal, if the smaller molecule was completely enclosed by the larger one. However, if the larger molecule is sufficiently large, there might exist very many such orientations.

In order to measure the similarity of the volume overlap, there exist several measures, such as the *Carbó index* [172] or the *Hodgkin similarity index* [80], both of which are used in several applications.

Alignment Based Similarity

Rigid Alignment. With their program SEAL (**S**teric and **E**lectrostatic **AL**ignment), Kearsley and Smith [95] were the first to replace grid-based volumetric alignment techniques by the analytical evaluation of Gaussian functions. With the introduction of Gaussian functions to the volume overlap problem, an enormous speed up was gained. In SEAL, simplex optimization is applied to the volume overlap problem. Similarly, Good et al. [65] use a method based on simplex optimization to overlap molecular electrostatic fields of two molecules. To measure the volume overlap, they use the Hodgkin similarity index [80]. McMahon et al. [123] optimize the electrostatic field overlap represented by Gaussian functions utilizing gradient optimization.

Beside the electrostatic field, other fields can also get represented by Gaussian functions, such as the van der Waals volume [67] or the electron density [130]. Nissink et al. [130] apply Fourier space methods similar to the molecular replacement technique used in X-ray crystallography [112]. These have the advantageous properties that rotational and translational parameters can be optimized separately. Again, simplex optimization is used to locally optimize the volume overlap.

Some approaches use multiple molecular fields to represent a molecule [135, 109]. For example, Parretti et al. [135] use fields for steric and electrostatic properties and align these using Monte Carlo optimization. Apart from using multiple sets of Gaussian fields, the algorithm RigFit [109] extends the above mentioned approaches by sampling both the translational and the rotational degree of freedom more appropriately.

Flexible Alignment. Based on the RigFit algorithm [109], Lemmen and Lengauer developed the program FlexS [110, 111], which flexibly aligns molecular structures based on Gaussian representations of the molecules. One assumption of FlexS is that one of the molecules, the reference, is given in its active conformation. The second molecule, the query, is then flexibly aligned to the reference molecule. Here, the basic idea is to decompose the query molecule into fragments, each of which is represented by a set of Gaussian functions. The algorithm starts by positioning a particular fragment, called *base fragment*, onto the reference molecule using the RigFit algorithm. Step by step the remaining fragments are added to the base fragment such that the Gaussian functions of the fragments overlap well with the Gaussian functions representing the reference molecule.

Labute and Williams [104] presented a method to flexibly align two molecules based on random sampling of the configurational space. Here, a new conformer is generated by randomly perturbing the torsional angles and then applying a minimization algorithm that minimizes the energy w.r.t. the volume overlap. The reported times are in the order of 10 minutes for small molecules. Thus, the run times are much longer than that reported for FlexS.

Descriptor Based Similarity

In [136], Pastor et al. presented a 3-dimensional descriptor based on molecular fields which they call GRIND (**GR**id-**IN**dependent **D**escriptor). GRINDs are generated in three steps. First, *molecular interaction fields* (MIF) [66] are computed. Second, the fields are simplified by identifying those parts of the fields not carrying any information. Third, the remaining parts of the fields are encoded into a transformation-invariant description using autocorrelation transformations. These descriptions can be analyzed using principal component analysis (PCA) or partial least-squares (PLS) (cf. [107]). This enables one to identify the most common properties of a set of molecules.

3.3.4 Surface Based Similarity

In order to identify molecular similarities that explain the molecule's functionality, it is necessary to look at those parts of the molecule playing a major role in the binding process. These parts can be most effectively described using the concept of molecular surfaces. Due to this need, in the last one and a half decades many approaches to molecular surface similarity were proposed. Yet, compared to the overall number of molecular similarity approaches, these approaches still represent only a small fraction.

Alignment Based Similarity

Spherical Harmonics. Since many molecules have a more or less globular shape, some researchers described molecular surfaces by spherical harmonic functions [120, 52, 108, 150], which constitute an approximate parametric representation of a molecular surface. Max and Getzloff [120] showed that spherical harmonics of low order can already capture the main features of molecular surfaces. Duncan and Olson [52] and Leicester et al. [108] used spherical harmonics of high order to represent fine surface details. While previously the representation of molecular surfaces by spherical harmonics had mainly been used for visualization and classification of molecular shape, Ritchie and Kemp [150] were the first to use spherical harmonics for surface alignment. Since rotations of a surface represented by spherical harmonics can be simulated by rotating the harmonic expansion coefficients [150], the use of such a representation for molecular surfaces alignment seems a natural choice. Ritchie and Kemp accomplish the optimization of the harmonic expansion coefficients by applying a quasi Newton algorithm, which works very fast. However, using spherical harmonics for molecular surface alignment has two major drawbacks. First, as already mentioned, approximation of molecular surfaces by spherical harmonics is only sensible for globular molecules. By representing a non-globular molecule by spherical harmonics, important details may get lost. Second, the approach is only capable of identifying global matches, hence, it is not applicable to applications where partial matches have to be identified.

Correlation Plots. Katchalski-Katzir et al. [93] and Barequet and Sharir [12] proposed alignment strategies which are similar in spirit. Both strategies use discrete representations of the molecular surface. While Katchalski-Katzir et al. use a grid representation, where each grid node is labeled as being "inside" or "outside" the molecular surface, Barequet and Sharir use points sampled on the molecular surface. Now, assume two molecules A and B, centered at the origin. Then the space of possible rotations around the origin is sampled using Euler angles with some predefined angle step size. For each thus generated rotation, they compute the correlation of the surfaces for different translational vectors. Katchalski-Katzir et al. [93] use Fourier transformations to analyze the correlation functions while Barequet and Sharir [12] compute the translational vectors for each pair of points representing A and B. Since this method is quadratic in the number of points, Barequet and Sharir compute local foot prints for each point and consider only pairs of points with similar foot prints. For all rotations with a high peak in the correlation plot, the neighborhood of this rotation is sampled in more detail. This is done recursively until the optimal correlation is found. For this final rotation, the best translational vector can be easily computed yielding an alignment of the molecular surfaces. One obvious disadvantage of this approach is the large number of rotations that need be considered to “guarantee” not to overlook important transformations.

Surface Patches. Another class of algorithms identifies local shape features of the molecular surface [62, 35, 81, 55, 56]. The local shape at some point on the surface can be described by the principal curvatures which are given by the two eigenvalues of the Hessian matrix [100]. However, in the context of molecular surfaces a more global description of molecular curvature is desirable. Such a description is given by the parameters of a paraboloid fitted to the local neighborhood of some point [176, 62]. Similar to the local curvature defined by the two canonical curvatures, a paraboloid also specifies two main curvatures and their directions. These curvatures can then be used to compute shape descriptors, such as the *surface topography index* introduced by Heiden and Brickmann [78], or the *shape index* proposed by Duncan and Olson [52, 53]. According to these shape descriptors, the neighborhood of each point can be classified as belonging to one of five classes: convex, concave, saddle, cylinder, and flat.

Using shape indices, Goldman and Wipke pursue the following approach [62]. First, critical points are computed on the surface by placing a single point into the center of each topological surface patch of the solvent excluded surface. For each critical point the shape index is computed. Then, for each pair of critical points with similar shape indices on the surfaces of molecules A and B, the whole molecular surface of B is aligned according to the local coordinate system centered at the critical points and defined by the main curvature directions and the surface normals at the critical points. For each alignment thus gained, the number of critical points on the two molecular surfaces lying close to each other is counted, yielding an alignment score.

Cosgrove et al. [35] developed the program SPAt (**S**urface **P**atch **A**lignmen**t**). In SPAt, the shape indices for all points on the triangular surface are computed. These points are clustered into circular patches of approximately constant curvature. Each patch is then represented by a single point located at the center of the patch. Clique detection is then applied to identify similar arrangements of points defining an alignment. Only points with similar shape properties can be matched. All alignments are scored according to the rms distance and the number of matched points. A similar approach is followed by Hofbauer [81] in his program SurfComp (**Surf**ace **Comp**arison). Here, in addition to shape properties, physico-chemical properties are considered in the matching step. For determining the similarity of surface patches in terms of physico-chemical properties, fuzzy filters [55] are applied. To measure the shape similarity of surface patches, harmonic shape image filters [178] are used.

Exner et al. [55] partition the molecular surface into overlapping patches of well-defined size. The patches are generated such that within each patch the local shape defined by the shape index as well as the molecular properties, such as electrostatic potential, lipophilicity/hydrophilicity and hydrogen bond density, vary only slightly. To be able to compare regions according to several properties at once, they apply fuzzy logic using linguistic variables. The overlapping patches together with their local shape properties are then used for surface matching. Exner et al. apply the approach to complementarity searching [56], i.e. rigid docking, but it could also be applied to similarity searching. For the matching step, each patch is represented by two points, its center on the surface and its center of mass. Point matching of the points representing the surface patches and properties is then done using geometric hashing [105, 173].

Morphological similarity. Jain [87] proposed the concept of *morphological similarity* based on a set of observers looking at the molecular surface and molecular properties thereon. In particular, Jain considers the charge characteristics of each atom. Each surface is embedded into a 3-dimensional grid and each grid point outside the surface is considered as an observer. For each observer, the minimum distance to the molecular surface, the distance to the nearest hydrogen bond donor or positively charged atom, and the distance to the nearest hydrogen bond acceptor or negatively charged atom is computed and stored. In addition to that, directional information is used for observations of polar moieties. The similarity of two molecules is then defined as a normalized sum of weighted Gaussian-like functions of differences in distances from observation points to the molecules. Since this similarity function is continuous and piecewise differentiable, gradient-based optimization can be applied [87].

Others. Masek et al. [117, 118, 119] use a representation based on molecular surfaces which they call *molecular skin*. A molecular skin is the volume difference between two molecular surfaces, where one of them is computed with an offset added to the van der Waals radii of the molecule's atoms. Thus, the 2-dimensional surface representation is transformed to a 3-dimensional volume representation. For this volume representation, methods commonly used for volume overlap can be applied (cf. Section 3.3.3). Perkins et al. [140] described a similar approach based on a grid representation. Here, interior grid points, exterior grid points and grid points lying close to the molecular surface are marked. Then, simulated annealing is applied to optimize the position of one of the grids with respect to the other one. Grant and Pickup [68] modified their Gaussian volume overlay method to take into account the surface area represented by the Gaussian representation. *Gnomonic projection*, i.e. the projection of molecular properties and distances onto a sphere, is used by Dean et al. [41, 40]. Using gnomonic projection, two molecules are aligned to each other by rotating one sphere against the other one centered at the same point, such that the similarity of the points on the spheres is maximized. Poirrette et al. [141] also use points on a regular grid to represent each molecular surface. In order to match these points, they apply a genetic algorithm.

Descriptor Based Similarity

The early approaches to molecular surface similarity were all based on alignment, i.e. on the relative orientation of one molecular surface to the other one. More recently, transformation-invariant approaches were developed [159, 22, 177]. They use molecular surfaces to define 3-dimensional descriptors.

Stiefl and Baumann describe a molecular surface descriptor called MaP [159]. This descriptor is generated in three steps. First, a surface approximation of the molecular surface by equally distributed points, sampled on a regular 3D grid, is computed. Second, molecular properties are projected onto the surface and mapped to the surface points. Finally, the distribution of surface properties given at the surface points is encoded into a transformation-invariant descriptor. Each surface point is assigned to one of four property classes: hydrophobicity, hydrophilicity, hydrogen bond donor, and hydrogen bond acceptor. The theoretical basis for the MaP descriptor are radial distribution functions, i.e. distance dependent count statistics. These distribution functions are computed as follows. For each pair of surface points the distance is measured and the occurrence of this distance is added to the descriptor vector. To store the distances between pairs of surface points, a discretization into c bins is used. The property class of the surface points is also considered, i.e., for each pair of property classes there exist distinct bins. Thus, the dimension of the descriptor vector is $(p(p+1)/2)c$, where p is the number of property classes. In order to make the distribution of the distances into bins more soft, fuzzy counts are used [157]. Comparison of molecules using MaP descriptors is done as follows. First, all vector elements having zero variance are excluded. Second, a regression model is computed using principal component regression (PCR) or partial least-squares regression (PLS) [114]. The correlation coefficient and the mean-squared error of calibration give information about the similarity of the molecules. Stiefl and Baumann also mention that the model is easily interpretable, i.e., the similarity of the descriptors can be back-projected into molecular space and thus allows for an easy understanding of the results.

Bender et al. [23] propose a transformation-invariant surface descriptor, which they call Molprint 3D. In contrast to MaP, Bender et al. use points on the solvent excluded surface for van der Waals spheres whose radii have been scaled by a factor of 2.0. The points are not approximated by a grid, but points given by the surface triangulation are used directly. These are, however, not equally distributed. Unlike MaP, continuous variables representing molecular properties are used. Also, the binning is replaced by neighbor/non-neighbor relationships between points, where only small (local) neighborhoods are investigated. For each point on the surface, an individual descriptor is computed. The union of these individual descriptors represents the Molprint 3D descriptor. For the pairwise comparison of descriptors, the Tanimoto coefficient [86] is used. In order to compare multiple structures, a Bayesian classifier can be applied. Properties responsible for the similarity can also be back-projected into molecular space, thus allowing an interpretation of the results.

Yet another descriptor, termed *shape signature*, was introduced by Zauhar et al. [177]. A shape signature of a molecular surface is generated by means of ray-tracing [60]. Starting from a random point on the surface, a ray is initiated with a random direction to the inside of the molecular surface. This ray is propagated inside the molecular surface by rules of optical reflection. Following this ray, a large number of reflections is computed. From the lengths of the reflection lines, a histogram is computed which describes the shape of the molecule given by its molecular surface. Once the histogram is computed for each molecule in a data base, the molecules can be efficiently compared using this 1-dimensional descriptor. In order to take into account molecular properties, 2-dimensional and higher dimensional descriptors can also be generated [177].

3.4 Summary

In this chapter we gave an overview of existing approaches to measure the similarity of molecular structures. We mainly concentrated on approaches that deal with drug-sized molecules. Although the number of articles that were considered is rather large, it is by no means complete. Nevertheless, we believe that the most interesting ideas in the field of drug-sized molecular similarity were covered.

The approaches presented in this chapter were categorized according to the molecular representation used for determining the similarity. However, a user might be more interested in the scenarios each approach can be successfully applied to, and in the expected outcome when applying a certain method. Three such scenarios are described below.

Scaffold vs. Surface Similarity. As already mentioned in the introduction to this chapter, it can generally be said that if we are searching for molecules with similar function yet different scaffold, pharmacophore based and surface based approaches are the most promising ones. These approaches are of particular interest, if new lead structures need to be identified. If, however, we are searching for very similar molecules, or if we are interested in clustering molecules with similar molecular graphs, than skeleton based approaches are more successful.

Data Base Search. If we want to search large data bases, the most restricting factor for the choice of method is time. In the presence of data bases with hundreds of thousands to millions of potent drugs, we need fast methods. In this case, descriptor based approaches are the method of choice. These include 2D or 3D fingerprints, but also molecular surface descriptors like MaP, Molprint 3D, or shape signatures. Molecular surface descriptors are particularly useful, if scaffold-hopping is desired, i.e., if we are looking for molecules with distinct molecular scaffolds. Feature trees represent an alternative to descriptor based approaches. Even though the feature tree approach computes an alignment (in 2D) of molecular structures, it is fast enough to be applicable to data base searching.

Pharmacophore Elucidation. If pharmacophore elucidation is the desired task, an alignment of molecular structures is needed. Alignment approaches are more expensive than descriptor based approaches, since they need to sample the space of rigid body transformations. But they also allow for more insight into the system, i.e. the set of molecular structures. Alignment methods are a prerequisite for quantitative structure activity relationship (QSAR) analyses, but they are also a necessity for the visual inspection of molecular structures.