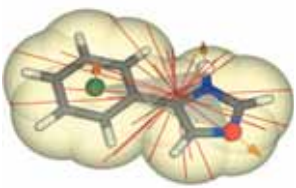
Structure-Based Virtual Screening Using Index Technology



Dissertation
zur Erlangung des akademischen Grades
Dr. rer. nat.
an der Fakultät für Mathematik, Informatik und Naturwissenschaften
der Universität Hamburg

eingereicht beim Department Informatik von

Jochen Schlosser
aus Eschweiler

Januar 2010

1. Reviewer: Prof. Dr. Matthias Rarey
2. Reviewer: Prof. Dr. Norbert Ritter
3. Reviewer: Prof. Dr. Dr. Thomas Lengauer

Date of thesis defense: November 10, 2010

Berichte aus der Medizinischen Informatik und Bioinformatik

Jochen Schlosser

Structure-Based Virtual Screening Using Index Technology

Shaker Verlag
Aachen 2011

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

Zugl.: Hamburg, Univ., Diss., 2010

Printed in Germany.

ISBN 978-3-8322-9874-6
ISSN 1432-4385

Shaker Verlag GmbH • P.O. BOX 101818 • D-52018 Aachen
Phone: 0049/2407/9596-0 • Telefax: 0049/2407/9596-9
Internet: www.shaker.de • e-mail: info@shaker.de

Zusammenfassung

Traditionelle Ansätze des strukturbasierten virtuellen Screenings arbeiten sequentiell: Jedes Molekül einer Bibliothek wird einzeln für das Zielprotein untersucht und die zu erwartende Bindungsgaffinität berechnet.

In dieser Arbeit wird ein Verfahren namens TrixX BMI vorgestellt, welches diesen sequentiellen Prozess umgeht. Der Ansatz basiert auf einem innovativen Deskriptor, welcher physikochemische Eigenschaften von Proteinen und Wirkstoffen kodiert. Der Deskriptor wird in einem Vorverarbeitungsschritt für alle Liganden unter Berücksichtigung ihrer Flexibilität berechnet und indiziert gespeichert. Mit Hilfe komplementärer, auf dem aktiven Zentrum beruhenden Anfragedeskriptoren, ist es möglich Ligandplatzierungen innerhalb des Proteins zu identifizieren. Sowohl chemische, als auch räumliche Komplementarität wird bereits auf der abstrakten Deskriptorebene behandelt. Moleküle für die kein passender Anfragedeskriptor vorliegt werden von weiteren Berechnungen ausgeschlossen. Dieser nicht-sequentielle Zugriff auf die Molekülbibliothek beschleunigt das virtuelle Screening im Vergleich zu anderen Methoden um eine Größenordnung, ohne dass ein signifikanter Qualitätsverlust der Resultate auftritt. Weiterhin ist es möglich pharmakophore Eigenschaften, die ein bevorzugtes Interaktionsmuster des Proteins darstellen, als Bestandteil der Deskriptoranfrage zu verwenden. In diesem Fall wird das Verfahren um eine weitere Größenordnung beschleunigt.

TrixX BMI kann in parallelen Rechnerumgebungen eingesetzt werden und das System skaliert mit der Anzahl an verfügbaren Rechenkernen. Somit können virtuelle Hochdurchsatzexperimente mit TrixX BMI durchgeführt werden.

Abstract

The standard approach to structure-based high-throughput virtual screening is a sequential procedure: Each molecule of a given library is evaluated against the target protein, eventually generating a ranked list of molecules.

A new approach, TrixX BMI, avoiding the sequential screening pipeline is presented in this thesis. It is based on a novel descriptor that encodes physicochemical properties of small molecules and proteins. The descriptor is calculated as part of a preprocessing step for all molecules in a compound library. Compound flexibility is accounted for using pre-enumerated conformational ensembles and the descriptors are stored in an indexed database. Complementary site descriptors of the protein are used to identify matching compounds and possible placements within the active site. Chemical and shape complementarity is evaluated solely on the descriptor level. During this process, molecules that are not part of any descriptor match are discarded. This non-sequential access to the compound library results in a speed-up of one order of magnitude compared to competing approaches while yielding results of comparable quality. Furthermore, TrixX BMI can incorporate requests for a certain pharmacophore interaction pattern to the protein. In this scenario, the speed-up increases to two orders of magnitude.

TrixX BMI can be deployed in a parallel computing environment and scales with the number of available cores. Thus, it is suited for virtual high-throughput experiments.

Acknowledgements

First of all, I would like to thank my research advisor Professor Dr. Matthias Rarey, not only for his excellent guidance during the course of my study, but for bringing my attention to the field of cheminformatics in the first place. Thank you for entrusting me with this challenging project.

Without Ingo Schellhammer the entire project would not have been possible: The results of his work are the basis for my own research project. Furthermore, I want to thank John Wu and Kurt Stockinger for supplying the FastBit indexing system and our cooperation during the last years.

The research work was partially funded by c.a.r.u.s. and based on the software library of BioSolveIT. Being a member of a large team and seeing how the different software products evolve was a great experience. I am confident that you will push TrixX BMI to the market and that our project will become a valuable tool in the process of drug discovery.

I had a great time at the Center for Bioinformatics (ZBH) in Hamburg, and all colleagues contributed to this. I would especially like to thank Axel Griewel for helpful discussions, being the best office co-worker I could wish for, and working together on various projects. During these years many people have come and gone: A whole generation of PhD-students — Patrick, Ingo, Jörg, Juri, Paul, and Gordon — left the ZBH, and a new generation picked up where they left off — Angela, Nadine, Katrin, Lennart, Matthias, Robert, Karen, Christin, Birte, Tobias, Adrian, Christian, Sascha, and Andrea. Thanks to all of you and also the past and present guest scientists and postdocs Andrea and Björn. I learnt a lot concerning our interdisciplinary field of research, and I really appreciate your extensive proofreading of my manuscript. Another thank you goes to Jörn for his 24/7 IT support and

to Melanie for keeping up with my constantly flawed forms and requests for leave.

Even though my family never showed particular interest in the subject of my research, I am glad that they have always shown this interest in me. Thank you for your support and help during all these years. Michael, Achim, and Christian, thank you for final proofreading.

Finally, I would like to thank my girlfriend Nora, not only for enduring numerous discussions about my research but especially for her expertise as professional illustrator. Thank you for designing the complex graphics in this thesis and for teaching me the very basics of your profession. Thank you for being there whenever I need(ed) you.

Contents

Acronyms

Å	Ångström
1D	One dimension, one-dimensional
2D	Two dimensions, two-dimensional
3D	Three dimensions, three-dimensional
ADMET	Absorption, distribution, metabolism, excretion, and toxicology
BLOB	Binary large object
CIAC	Compound interaction center
CPU	Central processing unit
CSD	Cambridge Structural Database
DPF	Depth-probe frequency
e.g.	exemple gratia (for example (Latin))
EF	Enrichment factor
ESS	Explored search space
et al.	et alii (and others (Latin))
GA	Genetic algorithm
GB	Gigabyte
GPU	Graphics processing unit
H-bond	Hydrogen bond
HTS	High-throughput screening
i.e.	id est (that is (Latin))
IUPAC	International Union of Pure and Applied Chemistry
MC	Monte Carlo
MNC	Minimum number of conformations
n.a.	Not applicable
NHR	Nuclear hormone receptor
PDB	Protein data bank
RMSD	Root mean square deviation

SGE	Sun Grid Engine
SIAC	Site interaction center
SQL	Structured query language
TB	Terabyte
TCG	TrixX Conformer Generator
VS	Virtual screening
WAH	Word-Aligned Hybrid code

Symbols

A	Descriptor attribute
G	Gibbs free energy
H	Enthalpy
S	Entropy
T	Temperature
Δ	Indicates a change
α	Angle in degree
β	Angle in degree
b_j	Distance value (steric bulk)
c_i	Compound interaction center
d_c	Compound descriptor
d_s	Site descriptor
dir_i	Coordinates (interaction direction)
i	Integer number
id	Integer identifier
j	Integer number
l_i	Distance value (side length)
r_j	Direction vector (steric bulk)
t	Interaction type as integer

1 Introduction

1.1. Background

Over the last decades virtual screening (VS) has become an integral part of modern drug discovery. The increase in computational power, detailed theoretical models, and the large number of available molecular structures, provides researchers with valuable information during early stages of the drug discovery pipeline.

In contrast to VS, automated experimental technologies like high-throughput screening (HTS) require the availability of library compounds as well as target proteins. In practice, identifying initial hits or lead structures in silico is both faster and more economical compared to solely experimental approaches. VS can be seen as a complementary approach, helping to identify drug candidates for the protein of interest. In combination, experimental and virtual techniques play an essential role in the process of drug discovery as recent success stories demonstrate [1, 2].

Among the most commonly used tools for virtual screening are molecular docking methods. Numerous approaches using different models and methodologies have been implemented so far. Nevertheless, the increasing requirements from academia and industry still leave many open questions and room for optimization. Some aspects of protein-ligand recognition have yet not been modeled and implemented well enough to deploy them automatically in a large-scale virtual screening campaign. To name a few: Protein flexibility, entropic effects, the role of water and changes in protonation states. This and the fact that standard docking methodology is an iterative process which docks compounds individually into the target protein leads to an increase in runtime requirements. Therefore, fundamentally new docking concepts and efforts to parallelize existing approaches are necessary.

1.2. Research Goal

In contrast to the iterative screening paradigm of most docking algorithms, the aim of this thesis is to develop and validate a hierarchical screening pipeline able to access compounds by usage of modern descriptor and indexing technology. The purpose of this nonsequential workflow is the rapid identification of drug candidates. These can then be subject to more detailed and time consuming postoptimization routines. In the following, a more detailed requirement analysis is presented:

- The fast prediction of binding molecules in the compound library is vital for the overall process. The average runtime for each compound in the library should decrease significantly in comparison to other tools, even for nonselective target proteins.
- The descriptor to be developed should enable highly selective queries. This ensures that the initial filtering step produces only few hits and that the above mentioned runtime requirements hold. Furthermore, each attribute of the descriptor should be accessible using index structures.
- The final result needs to be a ranked hitlist including pose predictions in three-dimensional space. This means that not only an activity value should be predicted but also the corresponding binding mode of the compound.
- The overall quality of the resulting predictions should be comparable with those of competing approaches. A gain in runtime is worthless if the predictions are error prone. Therefore, a comparable prediction quality in terms of root mean square deviation (RMSD) to cocrystallized structures, as well as capability to enrich known active compounds, must be achieved.
- A-priori knowledge like pharmacophore information and molecular properties of known binders should be incorporated into the docking engine.
- The models to be generated should have the potential to integrate concepts of protein flexibility.
- The overall approach should be deployable in a parallel computing environment as most pharmaceutical research institutes have access to large compute clusters. A

nonparallel methodology, which does not scale with the available infrastructure, would otherwise miss the aim of runtime reduction in a general setting.

The development to be presented is based on two components. First, the FlexX [3] library that supplies basic models, data structures, and algorithms for molecular docking. Second, the TrixX [4] approach which demonstrates that a descriptor-based index of physicochemical properties, like pharmacophore information and shape, suffices to identify drug candidates within large compound libraries.

1.3. Structure of the Thesis

In the following chapters a new molecular docking tool named TrixX BMI [5] is presented. It differs from the original TrixX approach in fundamental aspects: Its basic docking strategy, the prediction of molecular coordinates compared to just filtering compounds, and a high-dimensional description of shape which necessitates the usage of a specialized indexing system. Currently, there is no other structure-based molecular docking tool, that generates pose predictions of library compounds based on queries to indexed descriptor data.

Chapter 2 summarizes the history of drug design and especially VS. The focus of this chapter is on a general introduction to different methods and applications of VS screening and emphasizes its importance to the field of drug discovery.

Chapter 3 provides an in-depth overview of the field of structure-based virtual screening. Underlying concepts, basic models and algorithms, as well as the state of the art of structure-based tools, are presented. General problems and challenges are outlined and motivate the development of TrixX BMI.

Chapter 4 focuses on data organization, especially index structures that can be used to support multi-dimensional descriptor queries.

Chapter 5 spotlights the direct prerequisites of TrixX BMI. A detailed introduction into FlexX and TrixX is given. In addition, the underlying indexing technology to search in a database of molecular descriptors is presented.

Chapter 6 provides information about the TrixX BMI approach to molecular docking and highlights its differences to the previously introduced methods. The concept of the TrixX BMI descriptor and details of the overall workflow are shown. This

includes compound fragmentation, the handling of compound flexibility via conformational sampling using the TrixX Conformer Generator (TCG), descriptor indexing, index-based identification of pose predictions, and postprocessing routines.

Chapter 7 supplies the validation of TrixX BMI. First, a general introduction to different evaluation methodologies like redocking accuracy and enrichment experiments is given, followed by the actual results and their discussion.

As a conclusion, Chapter 8 summarizes the thesis and provides an outlook concerning future challenges and extensions of TrixX BMI.

2
Drug Discovery

2.1. Historic Background

Based on Fischer's principle of lock and key [6], which describes complementary physicochemical and steric fit of an enzyme and its substrate, the concept of drugs binding selectively to a receptor was developed by Ehrlich at the end of the nineteenth century. It was extended in 1905 by Langley [7] who postulated the idea of a receptor as molecular switch that can be activated and deactivated. During the evolution of drug discovery, other disciplines like chemistry, pharmacology, microbiology, and biochemistry have helped to raise it to a level where drugs are no longer discovered by chance or as a result of a chemists imagination but from cooperations of interdisciplinary research teams.

The discovery of x-ray crystallography in 1913 by W.L. Bragg [8] and nuclear magnetic resonance spectroscopy by Purcell and Bloch [9, 10] supplied scientists with the foundations to elucidate molecular structures on atomic detail level. In the beginning, this technique could only be applied to rather simple and symmetric structures. However, over the last decades improvements in experimental and also computational methods made it feasible to deduce atomic positions at high resolution for complex molecular structures.

Due to the growing number of discovered target proteins and their structural elucidation, pharmaceutical research shifted its focus to rational or semi-rational approaches to identify bioactive compounds. Starting in the 1980s, experimental high-throughput screening and combinatorial chemistry were developed. Improvements in robotics, control software, and data processing led to large scale automated screening platforms which were expected to increase the number of novel lead candidates and thus the

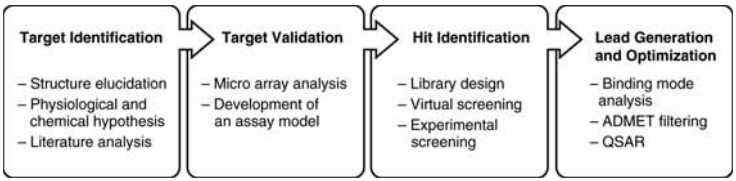


Figure 2.1.: The drug discovery pipeline

number of available drugs. These expectation could not be fulfilled. In many cases initial HTS hits could not be validated or optimized into actual leads. High error rates [11, 12] and low ligand efficiency [13, 14] were observed.

In order to cope with the significant expenses and hit rates below expectations, alternative techniques needed to be developed. This led to the implementation of new computer-aided approaches that are nowadays an essential part of the drug discovery pipeline.

2.2. The Drug Discovery Pipeline

The modern drug discovery pipeline (see Figure 2.1) is a complex multi-stage process involving in silico, in vitro, and in vivo technologies.

- **Target identification**
  The first stage involves the identification of a potential therapeutic target, its function, and an understanding of its role in the disease. If the target is an enzyme, catalytic activity and downstream substrates also need to be analyzed. In addition, structural elucidation using x-ray crystallography, NMR spectroscopy or protein homology modeling is applied.

- **Target validation**
  After the identification of a drug target, it must be verified that modulating the targets activity actually causes the desired therapeutic effect. This can be achieved by substituting the natural ligand, a small molecule responsible for the proteins mode of action, with a different ligand in the corresponding binding site. This ligand can either increase the level of activation (an agonist) or decrease it (an antagonist). The validation is usually done using in vitro and in vivo studies.

At the end of this stage, sufficient models to develop assays for experimental HTS need to be available.

- **Hit identification**
  In this stage, VS methods are frequently applied. They can be used to generate and filter large compound libraries or to predict preferred binding modes in order to identify essential interactions and features (pharmacophores) that trigger biological activity. Since current in silico approaches lack accuracy especially concerning bound water, affinity prediction, and protein flexibility, subsequent experimental screening experiments are performed. This results in a number of ligands likely to bind: The initial hits. Eventually, these hits are validated via further screening experiments to identify false positives and to reduce the risks of off-target effects.

- **Lead generation and optimization**
  The optimization of hit candidates regarding potency, selectivity, and ADMET (absorption, metabolism, excretion, and toxicology) properties is of central importance in the process of drug discovery [15]. Lack of efficacy (29 %) and low bioavailability and toxicity (39 %) account for the majority of failing drug candidates [15, 16]. This stage is often supported using docking techniques, for instance binding mode analysis and in silico ADMET prediction.

2.3. Virtual Screening

According to the International Union of Pure and Applied Chemistry (IUPAC), VS is defined as the "selection of compounds by evaluating their desirability in a computational model" [17]. This rather general definition is reflected by a wide range of methods. The most important applications of VS are library design and compound screening, whereas the methodology can be split into four different categories each requiring different information as input [18].

- **Substructure search** can be performed if substructures common to some actives are available. It is based on the hypothesis that such a "privileged scaffold" is associated with activity and a compound sharing this scaffold is also likely to be active [19].

- **QSAR** (quantitative structure-activity relationship) methods try to link chemical structure with biological activity and express this relation quantitatively [20]. This kind of analysis can only be performed if experimental data for active ligands is available. It is based on the following assumption: Activity can be expressed as a function of chemical and structural properties. This means that similar structures have similar activities.

The above mentioned methodologies are beyond the scope of this thesis. The remaining ones are now introduced in more detail.

2.3.1. Ligand-based virtual screening

In the absence of structural information of the target protein, ligand-based techniques are the method of choice. Their focus is on molecular similarity searching with active molecules as search templates. It is based on the assumption that compounds that are globally similar to known binders are likely to also show biological activity. Subsequently, different ligand-based methodologies are presented.

Descriptor-based similarity

In order to encode the properties of a compound a special index value or binary feature vectors (fingerprints) are used as descriptors. The actual search is performed using criteria like Tanimoto similarity for bit strings [21]. On a finer grain, these descriptors can be grouped into one-dimensional (1D), two-dimensional (2D), and three-dimensional (3D) approaches:

- 1D methods are based on properties that can either be derived from the structural formula of a compound like molecular weight, number of rotatable bonds, and number of hydrogen bond acceptors/donors or approximated by summation of atomic contributions, for instance the logP value to estimate the membrane permeability of a compound. Combinations of these numeric properties can then be applied as a filter which selects only compounds within certain ranges. Examples for filtering based on 1D descriptors are Lipinski's rule of five [16] and Oprea's lead-like criteria [22].

- 2D methods use descriptors derived from the covalent bonding pattern of a molecule. Since this topology is conformation independent, 3D atomic coordinates of a flexible compound do not influence the resulting descriptor. Prominent examples of 2D methods are the Wiener index [23] and Daylight fingerprints [24]. In general, index values encode similar properties as fingerprints and are easier to generate but harder to interpret.

- 3D methods rely on atomic coordinates and are therefore conformation dependent. Due to the flexible nature of organic molecules they can adopt different states in 3D space. These so called conformers share the same topology but differ in their geometric arrangement. The problem of molecular flexibility can either be addressed by pre-enumeration or by an alignment procedure that adapts the coordinates as needed (for details see 3.5). Examples for 3D descriptors are numerical values like van der Waals volume, molecular-, and polar surface area. More sophisticated 3D similarity methodologies include shape matching [25], shape-based fingerprints [26], and molecular field descriptors [27].

Small molecule alignment

These methods use molecular superpositioning algorithms to calculate similarity based on a 3D alignment between a template ligand and the ligand to be evaluated. The actual alignment can be calculated using different methods, for instance RMSD-fitting, geometric hashing, genetic algorithms, and Gaussian overlap optimization. Numerous methods differing widely in their underlying algorithmic concept, employed similarity criteria, superposition, and optimization have been developed [28]. Since these calculations are based on 3D coordinates, molecular flexibility needs to be addressed. Some approaches regard the reference ligand as rigid: The superpositioning is performed using either a flexible alignment [29] or by solid-body optimization based on precomputed conformer databases [25]. Details concerning the generation of these databases is discussed in Section 3.5. Other approaches perform fully flexible alignments, again significantly differing in the underlying methodology [30, 31, 32].

Since ligand-based methods are not in the main focus of this thesis, please refer to [33, 34] for a comprehensive review of the field.

2.3.2. Structure-based virtual screening

For structure-based approaches, the prediction of protein-ligand complex geometries and binding affinities are at the center of attention. Therefore, structural information on atomic detail level or models based on protein homology must be available.

The most prevalent structure-based methodology is molecular docking: In a first step, small molecules are placed into the active site of the protein. This is a complex task since it involves a continuous search space including many degrees of freedom like translation, rotation, and molecular flexibility of the ligand as well as the target protein. In a second step, the experimental binding affinity is estimated using so called *scoring functions* for each of the resulting pose predictions of a ligand. Since TrixX BMI is a structure-based approach, models for search space discretization, as well as different approaches to molecular docking and scoring are presented in more detail later (see Chapter 3).

2.3.3. Pharmacophores

Pharmacophores can be used in ligand-based as well as structure-based scenarios. The concept was introduced already in 1909 by Ehrlich as "a molecular framework that carries the essential features responsible for a drug's biological activity" [35]. It is independent of computational methods and has been successfully applied already before computers were used in chemistry [36, 37].

Since then the definition has not changed much, according to IUPAC, "a pharmacophore is supposed to represent electronic and steric features necessary to trigger or block a compounds biological response" [17]. This fuzzy definition demands an abstract perception of pharmacophoric features: A pharmacophore encodes chemical functionality rather than specific functional groups, for instance: Hydrogen bond interactions, aromatic interactions, and lipophilic area. It can be derived based on superpositioning of active ligands as part of a ligand-based search or by analyzing complementarities between a ligand and its position in the protein as part of a structure-based analysis. The extracted pharmacophore information can then be encoded using adequate descriptors and used as input to a similarity search. Again, 3D descriptor methods are conformation dependent and rely on handling of molecular flexibility. A more detailed overview

of pharmacophore type constraints that are used in structure-based VS is presented in Section 3.4.

2.3.4. Applications of virtual screening

The modern drug discovery process uses VS methods during two of its four different development stages: *Hit identification* as well as *lead generation and optimization* both use computer-aided methodologies to support and complement wetlab experiments.

Library design

At the start of a virtual screening campaign, a library of compounds needs to be generated. This process, called library design, can be split in two phases. Phase one — *library generation* — can be performed using two opposing objectives [38]:

- Diversity oriented libraries aim at covering the chemical space in a representative way. The ideal diversity oriented library would be with no voids, no redundancy, and an even distribution with regard to a given chemical space. These libraries are searched using multiple targets in order to discover novel, unexpected hits.

- Target libraries are generated with the focus on a specific protein target or target family. Thus, chemical properties believed to be of importance for biologically active compounds of the current target are overrepresented. Here, all compounds are designed to be similar to already known hit or lead structures.

On the algorithmic side, the design process is often supported using clustering, classification approaches as well as (dis)similarity analysis [39].

During phase two — *library filtering* — compounds with unfavorable ADMET properties are identified using for instance molecular property filters [40]. The applied filtering rules and models are based on property ranges derived from known drugs. Prominent example are Lipinski's rule of five and Oprea's more detailed lead-like criteria. Both filters rely on basic molecular properties like molecular weight, number of hydrogen bond acceptors/donors, lipophilicity (logP), and in case of Oprea's criteria number of rings and their flexibility. Compounds exceeding Lipinki's filter values tend to have solubility and permeability problems which would lead to poor oral bioavailability.

Compound screening and optimization

The screening of library compounds is the second major application of VS technology. Based on previously generated libraries, compound screening methods try to subselect compounds which are likely to show biological activity with the protein of interest. Thus, a subset of the original library is identified as suitable for downstream experimental screening experiments. Furthermore, if molecular docking approaches are employed, knowledge about the preferred binding modes or scaffolds can be extracted and deployed as part of further screening experiments aiming at optimizing initial hits.

3

Structure-Based Virtual Screening

For structure-based VS approaches, the prediction of protein-ligand complex geometries and their respective binding affinities are at the center of attention. Thus, two separate problems need to be solved:

First, the ligand's binding mode within the active site of the protein needs to be predicted. A so-called pose must imply a reasonable steric fit and must form favorable protein-ligand interactions. This reflects the previously mentioned complementarity of "lock and key". Therefore, the first part of this chapter focuses on basic principles and methodologies for binding mode prediction. This is a computationally expensive task, with an infinite number of valid placements; starting from any valid initial placement an infinite number of transformations can be applied. Since there are high-energy barriers between different local minima of the search space efficient gradient search methods to "dock" a compound into the protein active site cannot be applied. Thus, the search space needs to be discretized via means of simplification and placement constraints.

Second, for each resulting pose, the experimental binding affinity needs to be estimated. The binding affinity reflects the energy difference between the unbound apo structure of protein and ligand, compared to the energy of the protein-ligand complex. Since an exact calculation of this difference is computationally expensive and thus not suited for high-throughput docking approaches, different approximative methods, so called scoring functions, are introduced.

The next part focuses on the state of the art of existing docking technology. Since pharmacophores can also be used within a structure-based VS campaign, the next section summarizes this field. Since some of the previous approaches depend on the generation of conformational ensembles to handle molecular flexibility, this research

area is outlined next. Eventually, the last section summarizes the current challenges of structure-based VS.

3.1. Models for Molecular Docking

3.1.1. Search space

The intuitive representation of molecules is in form of a graph, using atoms as nodes and bonds as edges between them. Based on this graph representation, chemical and steric properties of molecules can be described. Unfortunately, the number of possible placements of protein and ligand is infinite. Thus, further discretization and limitation of the search space is inevitable: The structural information needs to be reduced and the law of parsimony should be applied.

Simplified molecular representations

Instead of the complete molecular graph of atoms and bonds, a molecule can be represented using surface descriptors, physicochemical descriptors, or a 3D grid representation of its interaction potential. In all cases, individual atom types can be neglected and represented in a reduced form [41]. Thus, a coarser model based on the molecules functional groups like hydrogen bond donors/acceptors, hydrophobic groups, charged groups, and metal coordination sites can be used instead [42]. This model can be applied to both protein and ligand, if the definition of compatible interaction groups is adjusted accordingly.

The calculation of discretized molecular properties in order to generate a reduced molecular representation can be done in a multitude of ways:

- Sphere-based reduction approaches place atom sized spheres into the binding site, thus trying to describe the protein surface and to identify possible anchor points for the placement of compound functional groups [43].

- As part of an empiric analysis of known protein-ligand complexes, each functional group of the protein can be assigned with preferred interaction geometries. Based on these geometries, discretized interaction patches can be generated and used as physicochemical descriptors [44].

- Probe-based methods place small molecular fragments into the binding site, thus capturing not only its geometric but also chemical properties. Subsequently, a mapping of each derived property to a corresponding grid point can be applied [45, 46].

All of the above approaches can be employed using different levels of granularity, depending on the desired level of accuracy and the employed docking algorithm.

Placement constraints

One of the most efficient ways to limit the search space is the introduction of placement constraints. A-priori knowledge about the binding site, e.g. the binding mode, can be used to limit the search to certain regions of the protein and to neglect poses which do not meet the desired criteria.

The active site covers only a rather small part of the entire protein, thus the search radius should be restricted. Usually, this is done by selecting only amino acids within 5 to 10 Ångström (Å) around the atoms of a cocrystallized ligand. If no ligand is available, active site specification must be performed by visual inspection of the protein or by detection algorithms based on geometric and energetic type information [47, 48]. This restriction on the active site of the protein prevents unfavorable placements outside the region of interest. For instance, large convex areas of the protein, offer numerous possible placements which do not contribute to the proteins mode of action and can often be neglected.

An even more restrictive constraint is the request for a certain interaction pattern each pose has to obey in order to be a valid placement. These so-called pharmacophore type constraints require knowledge about the preferred binding mode and describe chemical and steric constraints like hydrogen bonds, hydrophobic contacts, and spatial arrangements. If multiple binding modes are known, they can be combined into a pharmacophore by boolean combination. Since one objective of VS is the generation of novel lead candidates, pharmacophores should be applied with great care; their usage can increase the performance of VS significantly but also bears the risk of generating a nondiverse set of solutions.

3.1.2. Search methodologies

Due to the discretization of the search space, superpositioning algorithms can be used to identify physicochemically complementary placements of the compound within the active site of the target protein. Assuming protein and ligand to be inflexible, rigid body transformations can be used to search for reasonable placements [49]. However, protein-ligand complexation is much more dynamic in nature. Neither lock (the protein) nor key (the ligand) can be considered as rigid. Koshland postulated already in 1958 [50] his *induced fit* theory which states that protein and ligand adapt their conformation upon binding. Thus, molecular flexibility is not only a result of inter- but also intramolecular interactions.

Ligand flexibility

The classic approach to docking is to keep the protein target rigid and only consider the flexibility of the ligand to be docked. According to Brooijmans [51] the treatment of ligand flexibility can be divided into three basic categories.

- Systematic search methods use a set of values for each formal degree of freedom and explore these in a combinatorial fashion. As the computational complexity increases exponentially with the number of degrees of freedom, cut-off criteria need to be introduced. Examples for systematic search methods are *incremental construction*, *placement and linking*, and *multi-conformer* algorithms.

- Random or stochastic methods apply random changes, usually changing one degree of freedom at a time. A concern is the uncertainty of convergence that can be improved by performing multiple, independent runs of the same experiment. Examples of stochastic searching are Monte Carlo (MC) methods and approaches using a genetic algorithm (GA).

- Deterministic methods always end up in exactly the same result state if multiple runs on the same system are performed. Each state determines the move to the next state that is less or equal in energy. A common problem of these methods is the tendency to get trapped in local minima due to the inability to cross energy barriers. An example for deterministic search is force-field based energy minimization.

Some of the above methods can only be applied as online procedure during the search, others can also incorporate a preprocessing step. Since all of them are implemented in widely used docking programs, a detailed presentation is given in Section 3.3 where the state of the art of docking tools is presented.

Protein flexibility

The treatment of protein flexibility is less advanced than that of ligand flexibility. Various models have been developed to represent either just side chains movements or alternatively, a fully flexible protein including backbone movements.

The different methods can be categorized according to the type of binding event they are modeling:

- Kohslands's *induced fit* effect that is based on the assumption that the binding site adapts its conformation upon ligand binding.
- The hypothesis of *conformational selection* [52] that is based on the co-existence of multiple protein conformers and thus the believe that conformational change happens prior to protein-ligand complexation.

Algorithmically, different implementations again use systematic, random, and deterministic approaches.

Induced fit approaches One way to treat flexibility is according to the theory of *induced fit*: Structural changes of protein and ligand are considered as consequence of the binding event, thus are handled online during docking. These changes can be computed either consecutively or simultaneously.

Examples for consecutive methods are soft-docking approaches: These allow significant steric overlaps for initial, rather coarse placements [53]. In order to refine these initial poses, postoptimization routines for conformational adaptation need to be applied [54].

Quasi-simultaneous conformational search is often restricted to approximative calculations since computationally expensive simulation methods need to be performed. Often, these approaches are restricted to side-chain flexibility using rotamer libraries based on experimentally observed orientations. Analog to the problem of covering

ligand flexibility, these algorithms need to avoid combinatorial explosion. First implementations of this approach are the A*-based search of Leach [55] and Sternberg's self-consistent mean field [56]. More sophisticated methods which also cover backbone flexibility are often based on principal component analysis or normal mode detection [57, 58].

Conformational selection Another way of treating protein flexibility is to use ensembles of protein conformations as target for docking. These ensembles are able to represent not only side-chain but also backbone flexibility. They can either be used in a sequential manner by standard docking tools or by novel approaches which handle conformational ensembles in a single docking run. For this purpose heuristics can be used to select the correct protein conformation [59, 60]. Alternatively, a united protein structure to identify common rigid protein regions [61, 62], or 3D grids to map precalculated protein contributions can be employed to speed up the search and energy evaluation [63, 64].

Of course, combinations of the above mentioned methodologies for *conformational selection* and *induced fit* can also be followed [65]. One of the main challenges in the context of flexible receptor docking is the estimation of binding affinity using multiple protein structures. Since a detailed view on protein flexibility is beyond the scope of this thesis, please refer to [66, 67] for more details.

3.2. Scoring Functions

The evaluation and ranking of ligand pose predictions is a crucial aspect of structure-based VS. Even if binding conformations are correctly predicted, VS ultimately cannot succeed if it is not able to correctly differentiate "true" binders from non-binders.

3.2.1. Introduction

The basic physical principles in molecular recognition are governed by thermodynamics. The binding affinity of ligand and protein can be estimated based on the fundamental thermodynamic equation that relates changes in inner enthalpy (ΔH) and

entropy (ΔS) in combination with temperature (T) to a change of the Gibbs free energy (ΔG)

$$\Delta G = \Delta H - T\Delta S \tag{3.1}$$

A reaction occurs spontaneously if the resulting change in free energy is negative and the system reaches a level of lower energy. Thus, the quantity of interest is the difference in free energy between the complexed state of protein and ligand (G_{PL}) and their unbound states (G_P and G_L).

$$\Delta G_{PL} = G_{PL} - (G_P + G_L) \tag{3.2}$$

The exact calculation of the free energy of binding is currently not feasible. Most approaches estimating the binding free energy make various assumptions, simplifications, and do not fully account for all physical phenomena of molecular recognition. Therefore, many so-called scoring functions factor the problem into independent components.

- Non-covalent interactions contribute favorably to enthalpy, e.g. hydrogen bonds, metal coordination, and hydrophic interactions, which account for the stabilizing effects of aggregated hydrocarbons in solvent [68, 69, 70]. Hydrogen bonds are of special importance for protein-ligand complexation and molecular docking. Typically a hydrogen bond (H-bond) describes an interaction between two electronegative atoms, e.g. oxygen and nitrogen, one of which is bound to a hydrogen atom. In this constellation, the hydrogen can no longer be clearly allocated to one of the hetero atoms. Non-classic H-bonds also involve CH groups, sulfur, fluorine, and pi-electrons. Since H-bonds contribute significantly to the overall enthalpy and have specific geometric preferences concerning atom-atom distance and angle of interaction, they are of high importance in molecular docking. Their energy contribution ranges from 4–15 kJ mol^{-1} for classic H-bonds up to 60–170 kJ mol^{-1} for strong/charged H-bonds [71]. Ligand atoms coordinating a metal ion of the protein have a similar stabilizing effect.

- Further enthalpic contributions are van der Waals forces. The energy is favorable if two molecules fit closely and no significant atom-atom overlaps occur.[70, 72]

- Desolvation effects represent the influence of the solvent on the overall energy. On the one hand, desolvation can lead to an unfavorable change in enthalpy due to the breaking of H-bonds between hydrophilic molecules and solvent. Here, the enthalpic change usually outweighs the increase in entropy of the solvent. On the other hand, in case of hydrophobic molecules, the H-bond network of the surrounding solvent is destabilized, which leads to an increase in entropy. In this scenario, desolvation results in a favorable energy contribution, since no H-bonds between hydrophobic molecules and solvent are broken upon complexation.[73, 74]

- Complex formation often entails additional constraints on rotatable single bonds and thus reduced transformational and rotational freedom. This reflects a loss in entropy and thus an unfavorable energy contribution.[75]

Essentially, three basic types or classes of scoring functions are currently applied: Force-field based, empirical and knowledge-based scoring functions. Additionally, a combination of these approaches can be employed. Details are discussed in the following.

3.2.2. Force-field based scoring

Force fields usually quantify the sum of two energies, the protein-ligand interaction energy and internal ligand energy based on classic potentials. Since binding affinity is influenced by entropy and force fields do not accommodate for it, they are often applied in combination with molecular dynamics or MC simulations. Here, solvent can either be explicitly simulated or is part of an implicit calculation based on the Poisson-Boltzmann model [76]. Since these simulations are computationally expensive, force-field terms can alternatively be combined with empirical terms to account for solvation and ligand entropy [77]. Most force-field scoring functions only consider a single protein conformation, which allows to omit the calculation of internal protein energy, and thus simplify scoring.

3.2.3. Empirical scoring

Empirical scoring functions are fit to reproduce experimental data. Their design is based on the idea that binding energies can be approximated by a sum of individual uncorrelated energy terms. The coefficients ΔG of the individual terms are obtained

from statistical analysis of experimentally determined binding energies and structural information. Binding energies are calculated as a sum of these terms, each of these terms represents a different physicochemical property. An example of empirical socring is the FlexX scoring function, an adaptation of Böhm's function from 1994 [78]:

$$\begin{aligned}\Delta G_{binding} = \quad & \Delta G_0 \\ & + \Delta G_{rot} N_{rot} \\ & + \Delta G_{hb} \sum_{hb} f(\Delta R, \Delta\alpha) \\ & + \Delta G_{io} \sum_{io} f(\Delta R, \Delta\alpha) \\ & + \Delta G_{aro} \sum_{aro} f(\Delta R, \Delta\alpha) \\ & + \Delta G_{lipo} f^*(\Delta R)\end{aligned} \tag{3.3}$$

Here, the Gibbs free energy is estimated by the sum of contributions of H-bonds ΔG_{hb}, ionic interactions ΔG_{io}, pi-interactions ΔG_{aro}, hydrophobic contacts ΔG_{lipo} between protein and ligand, and a term ΔG_{rot} for rotatable bonds. Each individual contribution is scaled using the distance R and, in case of directed interactions, the angle α between the corresponding functional groups. The remaining constant term ΔG_0 describes the loss of entropy due to reduced transformational freedom after complex formation. A problem of empirical scoring functions is that the ΔG terms are derived from experimental data and thus stable complexes. This means that destabilizing contributions are inadequately represented.

3.2.4. Knowledge-based scoring

In knowledge-based functions, protein-ligand complexes are evaluated using relatively simple interaction-pair potentials. The corresponding energy contributions are based on the frequency of actual pairwise atom contacts in crystallized protein-ligand complexes. The fundamental hypothesis is that the experimentally determined crystal structure corresponds to the energetically optimal structure. Thus, the usage of a Boltzmann statistic on distance distributions yields pairwise atom potentials, which are used as a scaling factor during summation of all relevant atom pairs as in the PMF score [79]. Another prominent example using an additional term describing desolvation effects is

Gohlke's DrugScore function [80]. Since the pair potentials are derived using a Boltzmann distribution a common problem of these methods is the choice of the reference system: If it does not represent the entire space of stable protein-ligand complexes, the resulting potentials are inherently affected and do not comprise information about the missing states.

3.2.5. Consensus scoring

Given the imperfections of current general purpose scoring functions, consensus scoring schemes are often employed. Consensus scoring combines information from different scores to balance errors in single scores and reduces the probability of outliers. Nevertheless, a consensus score also includes the weaknesses of the individual members, thus cannot compete with a scoring function ideally suited for the target of interest. However, if no or no sufficient knowledge about the protein is available, and thus no suited scoring approach can be identified [81, 82, 83], consensus scoring often yields the best results.

3.3. Molecular Docking Implementations

This section covers existing docking implementations and is grouped into systematic, random, and simulation methods. Furthermore, combinations of these approaches are presented. Since the focus of this thesis is on high-throughput molecular docking, computationally expensive methods to solve problems like protein flexibility and energy minimization during postoptimization are only discussed briefly. For more details on these methods please refer to [84, 66, 67].

Table 3.1 gives an overview of the presented methodologies and their different implementations. Since a full summary of the entire field cannot be given as part of this work, please refer to the following reviews for more details on methodology, implementations, and comparative studies [77, 85, 86].

3.3.1. Systematic methods

Incremental construction

Docking programs based on the *incremental construction* approach build up the ligand on the fly as first proposed by Leach and Kuntz [87]. Initially, the ligand is split into

Systematic methods	
Incremental construction	Dock 4.0, HammerHead, FlexX
Placement and linking	eHITS, Surflex
Multi-Conformer approaches	Slide, Flog, FTDock, Fred, ShapeSignatures
Cluster-Based approaches	PHDock, TrixX
Random methods	
Monte Carlo	ICM, QXP
Genetic algorithms	Gold, AutoDock, MolDock
Others	Pro_Leads, Plants
Simulation methods	not presented
Multistep methods	HierVls, Glide

Table 3.1.: Different docking methodologies and implementations.

fragments, before an anchor fragment, a small part of the compound, is placed into the active site using matching algorithms. Adjacent parts are added incrementally, often relying on libraries of preferred torsional angles. This strategy, also called "anchor and grow", is efficient for small compounds and relies on the hypothesis that a protein-ligand complex can be reconstructed via placement of a characteristic substructure. Since the search space is still potentially exponential, larger and extremely flexible compounds lead to a significant increase of computational costs.

DOCK 4.0 The original version of Dock [43] was one of the first developments for molecular docking. Introduced by Kuntz in 1982, it uses spheres as protein descriptors and performs 4-point superpositioning between the sphere centers and ligand atoms. This first approach regards both, protein and ligand, as rigid. In 1998 Dock introduced ligand flexibility [88]. Ligand conformations are precalculated, such that each conformer shares a common rigid fragment with all other conformations. These conformers are then docked as a group into the receptor binding site. The latest major release [89] features an *incremental construction* routine: The compound is divided into rigid overlapping fragments. By default the largest of these fragments is chosen as anchor. The initial orientation of the anchor fragment is then computed using a geometric matching protocol. Subsequently, nonoverlapping fragments that are connected

via rotatable bonds are generated. These fragments are iteratively added using bond type specific torsional angles. After each construction step, score and diversity filters are applied. Only a reasonable number of partial solutions remains for further calculations. The scoring function, which is not only used for evaluating final pose predictions, but also to guide intermediate search stages, is based on precomputed AMBER [90] force-field terms combined with intra-molecular energies like Lennard-Jones potentials for clash detection.

HAMMERHEAD developed by Welsh in 1996 [91] uses a probe-based method to describe the favorable interaction spots of the protein active site. The compound is split into fragments, these are conformationally sampled and aligned to the protein by matching fragment atoms with probe atoms. The highest scoring fragment placements are retained as anchors (head placements). The remaining fragments are aligned iteratively into the probe neighborhood next to the end of the current head. Each successful fragment placement is then merged with the head using gradient-descent methods to optimize the alignment and to reduce steric clashes. This fragment chaining procedure is repeated until the full ligand is rebuilt. The search process and the final scoring is performed using an empirically derived scoring function [92].

FlexX has been developed by Rarey et al. since 1996 [3]. The work to be presented in this thesis is based on the models and algorithms of FlexX: Please refer to Section (5.1) which introduces the underlying molecular models and the employed *incremental construction* algorithm in detail.

Placement and linking

Another variation is docking of all fragments followed by fragment reconnection instead of incrementally constructing the ligand starting with anchor placements. A problem with *placement and linking* approaches is the risk of not finding reasonable fragment poses. If only one fragment cannot be docked correctly, the regeneration of the bioactive pose will eventually fail.

eHITS employs a classic *placement and linking* scheme: In a first step the ligand is split into fragments. Then, each fragment is docked into the active site using clique detection based on a so-called "Geometric Shape and Chemical Feature Graph". This graph description is used for both, protein cavity and ligand, and represents the essential functional features of a molecule. Based on these fragment poses all possible pose sets are enumerated. A pose set is defined as a set of fragment poses (one for each fragment) that are "distance compatible". This means, capable of reforming the input ligand. Based on an empirical scoring function the most promising pose sets are retained for further processing and optimization. Thus, also fragments with poor individual scores can be part of a high-scoring pose set and thus part of a final pose prediction.[93]

SURFLEX uses the same binding site definition (so-called protomols) as HammerHead. The compound to be docked is fragmented and each fragment is aligned with the protomol based on molecular similarity. In contrast to HammerHead, corresponding transformations are applied to the complete molecule, not only the current fragment. The resulting putative poses are scored using the Hammerhead scoring function, which also serves as an objective function for local optimization. Flexible docking proceeds by linking pieces of initial pose predictions, such that the resulting solutions possibly consist of fragments from different initial fragment predictions.[94, 95]

Multi-conformer docking

These algorithms use pregenerated libraries (see 3.5) of likely conformations of library compounds to model ligand flexibility. The actual docking is performed using rigid-body methods based on either superpositioning and scoring or shape complementarity. The major strength of these approaches is speed, while their predictive power is constrained to the explicit conformations in the library. This makes it difficult for the assignment of an accurate score. However, the usage of postprocessing routines can be used to address this issue.

SLIDE uses a multi-level hash index based on triplets of functional groups. The index captures basic properties like the interaction types of the triangle as well as conformation dependent properties, e.g. perimeter, longest and shortest side of the triangle. The different conformations of the compounds are taken from crystal structures of the

Cambridge Structural Database (CSD) [96] or from rule-derived models of the NCI database [97]. In addition, the database can be enriched by a series of computationally generated low-energy conformers. During the search, a grid-based representation of the receptor is employed to generate complementary triplets of functional groups, which are used as input to a geometric hashing algorithm. This algorithm identifies initial compound placements within the binding site based on the hash index. Single bonds connected to these anchor groups are considered as rotatable and can be adapted to improve the initial fit of the ligand. Furthermore, SLIDE also models protein side chains as flexible, thus the active site can adapt towards the docked ligand according to the induced fit paradigm. The resulting predictions can be postprocessed using mean-field optimization and an empirical scoring function. Another innovative feature of SLIDE is the incorporation of water particles into the binding site. These can either be displaced, if they clash with the ligand placement, or be evaluated considering enthalpic and entropic effects as part of the score.[98]

FLOG searches over so-called Flexibases using an approach similar to the original version of DOCK. A Flexibase stores a small, yet diverse set of conformations for each ligand in the library. On average it stores 8 and at most 25 conformations per compound. Steric and geometric filters are applied to increase the diversity of the conformations. During the search, clique detection algorithms are used to generate trial orientations which are then scored based on a grid representation of molecular properties. Eventually, the resulting placements are postprocessed using a simplex-based optimizer.[99]

FTDOCK is a shape-based approach. The original field of application is protein-protein docking, but it has also been applied to protein-ligand complexes. The shape is represented by a simple numerical grid for protein and solvent, and ligand and solvent. Each grid point is assigned with a value determining whether it is part of the molecular surface, of the core, or it is space that is occupied by solvent. Complementary values are used for protein and ligand. Additionally, each grid node holds a value representing the local electric field. It performs a global search of translational and rotational space followed by refinement of the best pose predictions. Potential complexes

are scored on the basis of shape complementarity and favorable electrostatic interactions. FTDock, in its original version does not reflect flexibility and does not produce 3D coordinates for the resulting complex, it only produces the correlation score for a similarity analysis.[100]

FRED uses Gaussian functions to represent molecular shape of the protein active site as well as the ligand. First, a precalculated set of conformers is randomly positioned into the binding site. These initial poses are then scored and iteratively optimized using a fast Gaussian overlap function and systematic solid body optimization. Since the employed docking function is strictly shaped-based, multiple empirical scoring functions can be applied either individually or as a consensus score to produce the final rank order. Optionally, force-field optimization can be performed as postprocessing step.[101]

ShapeSignatures describes the shape of molecules using distance histograms based on a ray tracing approach: Starting from a random position, a ray is casted. For each reflection point the length of the previous ray and the electrostatic property of the reflection point on the Connolly surface [102] are annotated and stored in a histogram. In case of ligands, the rays are casted and reflected inside of the molecule. In case of proteins, the ray initiation and reflections are calculated using the outside of the molecule starting within the active site of the protein. Thus, complementary histograms for ligand and protein are generated that can be compared using simple metrics. Just like FTDock, this approach does not yield a valid superposition, it only returns a similarity value.[103]

Cluster-based methods

Cluster-based screening approaches exploit redundancies in compound libraries by grouping compounds, fragments, or pharmacophores into clusters.

PHDOCK employs *multi-conformer* docking similar to DOCK 3.5 and is implemented using Dock 4.0. The authors introduce a reduced active site representation. Instead of representing the site using Dock's sphere description, chemically labeled site points are generated using randomly distributed molecular probes. These are then energy minimized and clustered. Instead of docking individual compounds, pharmacophores

are docked into the active site. Pharmacophores represent basic functional groups and are derived for each conformation of the library compounds. The resulting set of pharmacophores is clustered, thus identifying redundancies within the set. After docking the pharmacophores, each successfully placed pharmacophore is replaced with its associated compound in the respective conformation and scored. The authors report a speed-up of a factor $5 - 8$ while the resulting enrichment factors and poses are of comparable quality to the regular DOCK results.[104, 105]

TrixX was published by Schellhammer and Rarey in 2007 and is the direct predecessor of the work presented in this thesis. Based on queries to a molecular index, which identifies redundancies in molecular fragments, initial fragment placements are generated and recombined according to the *placement and linking* paradigm. The TRIXX concept of searching a structure-based molecular index, some of its descriptor properties, and its active site representation are reused in the TRIXX BMI approach. Details are given in Section (5.2).[4]

3.3.2. Random methods

These algorithms (also called stochastic methods) operate by making random changes to either a single compound or a population of compounds. A newly obtained molecule is evaluated on the basis of a fitness function which guides the search. This function needs to be able to cross energy barriers in order to explore the full search space of the protein's binding site. Furthermore it needs to be sensitive to minor changes in order to optimize an already useful pose. Different models like MC methods, GA, tabu search, and ant colony optimization meet these criteria. An advantage of random searching is the independence of a characteristically binding anchor fragment. Thus, large and rather hydrophobic molecules can often be docked successfully using random approaches. However, they can get stuck in local minima and do not reward a-priori knowledge about specific geometric preferences of known binders. Also, limited reproducibility and convergence are connected to their stochastic foundation [51].

Monte Carlo methods

In MC implementations, the molecule is considered as a whole and random changes are made to adapt the translation and rotation of the compound, as well as its torsion

angles. After each move, the structure is minimized, and the energy of the new structure is determined.

ICM implements so-called biased probability MC. Translational and rotational degrees of freedom are sampled by pseudo-Brownian motion, while torsion angles are sampled using biased probability moves. These probabilities can be derived from known structures. Thus, the resulting moves likely sample regions of high probability. Before the fitness function is evaluated, local energy minimization is performed. The search is guided using the following selection criteria: A surface-based solvation energy, an entropy calculation, and the energy from the minimization. The sum of the three energy terms determines whether the new structure is accepted or rejected.[106]

QXP initially uses MC procedures to explore the torsional conformation space compound. This is followed by rigid body rotations and translations to align the compound onto "guide atoms" within the active site. These guides are atoms in van der Waals contact with atoms of the protein active site. The resulting poses are optimized using MC calculations based only on rigid body rotations and translations. Eventually, QXP uses a combination of conjugate-gradient minimization and MC for optimization of torsion angles. [107]

Genetic algorithms

GAs try to mimic the process of evolution during their search for the global energy minimum. Abstract representations (chromosomes) of possible solutions (phenotypes) need to be generated and are the foundation for the search process. Evolutionary algorithms demand that only the fittest phenotypes are carried on to the next generation. Each generation is created based on random or biased mutations, which are applied to increase genetic diversity, and prevent early convergence. Crossover, a process that swaps large regions of the "parents", is permitted in GAs. In general, minimization of the phenotypes is not applied until convergence is reached.

GOLD was introduced in 1995 by Jones et al. [108] and is still one of the most widely used docking tools. It uses multiple subpopulations of the molecule, rather than a single large population, and manipulates these simultaneously. Migration of members

from one subpopulation to another can occur, which increases the efficiency of the GA. The poses are represented as chromosomes. In each step, a classic GA operation like mutation and crossover is applied. Each chromosome encodes torsional angles, protein side-chain bonds, and integer values describing hydrogen bonding. The operation and the parent chromosome are both chosen using a roulette-wheel mechanism. The resulting new chromosome then replaces the least fit member of the current subpopulation. In 1997, support for metal coordination, partial ring-system flexibility, and better handling of small, rather hydrophobic molecules was introduced [109]. Since 2003, an adaptation of the empirical scoring function ChemScore [110] is used to guide the search: In addition to the weighted sum of H-bond energies, van der Waals contact energy and intra-molecular energy of the original scoring function, terms for H-bond geometries and entropy loss due to limited conformational freedom of the compound are employed [111].

AUTODOCK employs a Lamarckian genetic algorithm that incorporates local minimization of a certain fraction of the population. The program switches between a "genotypic" phase for global conformational and translational search and a "phenotypic" phase with an adaptive local energy minimization based on a force-field energy function. Phenotypic changes due to the energy minimization are mapped back onto the genes via updating the explicit coordinate representation in the chromosome.[112]

MOLDOCK applies a differential evolution [113] approach towards molecular docking. First, all individuals are initialized and evaluated according to the fitness function, which is an extension of the PLP [114] scoring function including new hydrogen bonding and electrostatic terms. Afterwards, the following process is iteratively applied until the termination condition is fulfilled. The novel idea in differential evolution approaches is to create an individual of the next generation using a weighted difference of all possible parent solutions in combination with a specific parent. This parent is replaced by the new individual only if the newly generated one is fitter according to the scoring function. The termination condition is enforced if a maximum number of evaluations has been performed or the variance within the solution set drops below a certain threshold.[115]

Other approaches

PRO_LEADS implements a "Tabu search", similar to MC. It also uses random moves to explore conformational and translational space. Furthermore, it maintains a list of tabu conformations that represents previously accepted pose predictions. A new solution, lower in energy, is only kept if it is dissimilar to anything in the "tabu" list. This procedure stimulates sampling of space that has not been sampled before. A modified version of ChemScore is applied as scoring function . Eventually, a local minimization is applied to the lowest energy conformations as postoptimization step.[116]

PLANTS is based on stochastic optimization algorithms called ant colony optimization [117]. These algorithms are inspired by the behavior of real ants, which try to find a shortest path between a food source and their nest. Pheromone trails mark the best paths and are used as means of communication. In the context of VS, the artificial ant colony is employed to find the lowest energy conformation of a compound within the binding site. The solution space consists of different "pheromone trails", each describing different rotational and transformational settings. The trails are iteratively modified to increase the probability of generating low energy conformations. Intermediate solutions are improved using a local simplex search algorithm. Then, diversity filters are applied. After a certain number of iterations, depending on the size of the ant colony, the flexibility of the molecule, and the number of heavy atoms, the search concludes. The remaining solutions are ranked using an empirical scoring function based on combination of GOLD's ChemScore and PLP scoring.[118, 119]

3.3.3. Simulation methods

The most popular simulation method is currently molecular dynamics (MD). This approach tries to approximate known physics by modeling individual atom movements. However, MD simulations are often unable to cross high-energy barriers within feasible simulation time. Therefore, one might only retrieve molecules within local minima of the energy surface. Two strategies can be applied to solve this issue: On the one hand, searching can be started at different temperatures. On the other hand, multiple molecular dynamics calculations starting from different positions can be executed.

In contrast to molecular dynamics simulation, energy minimization approaches are rarely used as stand-alone search, since they only reach local energy minima. Instead, energy minimization often complements other search methods aiming at refining initial, rather coarse pose predictions during postoptimization steps.

Since simulation methods are computationally expensive and thus not suited for high-throughput VS, details and implementations are beyond the scope of this thesis.

3.3.4. Multistep methods

These methods use hierarchical combinations of different methodologies. Starting from rather coarse, but fast approaches they continue to increase the granularity and thus the computational complexity in a stepwise manner. Thus, these approaches are often suited for high-throughput VS: Initially, fast filtering methods are applied yielding a low false negative rate but still filter out significant parts of the viable search space. The question of false positives is then addressed during subsequent processing steps. Generally speaking, most available VS tools can be applied in a meaningful hierarchical order. However, a successful screening pipeline needs to be parametrized and executed in an adjusted way, such that known weaknesses of the single methods are addressed and thus corrected by the downstream ones.

HIERVLS starts with a coarse grain conformational search. The up to 300 resulting conformers are docked into the active site using Dock 4.0 with 25 % reduced van der Waals radii. After this first phase, close protein-ligand contacts are still accepted. The 50 best scoring placements according to the grid-based Dock 4.0 score are handed over to the next phase. Here, the buried surface area is evaluated and poses with a value below 30 % are filtered. Within the next stage, the remaining top five poses are subject to energy minimization. This also includes solvent- and protein side-chain movements. Eventually, the lowest energy solution is rescored explicitly accounting for solvation/desolvation effects.[120]

GLIDE is currently one of the most successful and widely used docking tools. It combines coarse *multi-conformer* docking with subsequent gradient-descent methods and MC optimization. Initially, the conformers are clustered and the resulting representatives are docked into the binding site which is represented as an equally spaced grid.

A cluster representative is defined by a rigid core region and flexible side chains. The actual docking uses histograms that encode the protein and ligand shape as either the distances of grid points to the receptor surface or distances from the ligand center to the ligand surface. If the histograms match sufficiently the corresponding conformer is placed using a number of predefined orientations. GLIDE uses 20 % reduced van der Waals radii to account for minor readjustments during complexation. The resulting poses are scored using a grid adaptation of the ChemScore function. Then, the top 5000 poses are subjected to refinement and are subsequently rescored. In the next phase, force-field minimization and scoring is applied using the 400 previously best scored poses. Here, the scoring function is evaluated on a grid-based discretization of van der Waals and electrostatic energy. During the last phase, MC optimization is applied to the six best scoring poses in order to adapt nearby torsional minima and orientations of peripheral groups of the ligand.[121]

3.4. Pharmacophore Search

Apart from molecular docking, similarity search methods from ligand-based VS can be adapted, such that structure-based searching can be performed. In this scenario, pharmacophore models (see 2.3.3) derived from protein structure by determining complementarities between a ligand and the corresponding binding site are employed [122, 123]. Since structure-based VS aims at the prediction of protein-ligand complex geometries the following methods are based on pharmacophore alignment. In this field two different approaches can be distinguished. While most approaches use point-based methods relying on chemical features, property-based approaches representing molecular field descriptors are also feasible [123]. As in molecular docking, the question of molecular flexibility must be addressed. This is done by either pre-enumeration or via online sampling algorithms. Different algorithmic approaches again rely on systematic, random, or simulation methods. In the following, a brief introduction to successful implementations and their underlying methodology is presented.

GASP [124, 30] and GALAHAD [125, 126] are genetic algorithms, which both handle molecular flexibility online during the search process. A chromosome represents all bond angles and all feature mappings between pharmacophore and the reference compound. A new offspring is created by applying random movements in torsion space. Both

tools differ concerning their employed fitness function and the fact that GALAHAD also allows the usage of pregenerated conformations as base for torsion updates. This bears a significant speed advantage.

Most other tools rely on conformational ensembles and employ rigid body docking combined with maximum common substructure searching for the minimum pharmacophoric requirements. Most applications, e.g. DISCOVERY STUDIO [127], LIGAND SCOUT [128], PHASE [129], and UNITY [130], explicitly model steric constraints using inclusion and exclusion volumes. This representation of shape, however, reduces the computational efficiency, since it necessitates frequent checks for ligand clashes as part of a postprocessing routine. A variant of these tools is SHAPE4 [131] which generates a negative image of the active site and employs a modified search based on the OE SHAPE TOOLKIT [132]. This approach incorporates shape in form of Gaussian functions into the pharmacophore description. During the search only compounds with matching substructure and a reasonable shape fit are selected for postprocessing.

For more details concerning this field of research please refer to [133, 34].

3.5. Conformational Sampling

Already during the 1970s, conformational ensembles were used to represent flexibility of small molecules. Back then, ensemble generation was based on experimentally determined crystal structures [134]. During the 1980s, first knowledge-based approaches were published that addressed the computational generation of a single low energy conformer for a given input ligand, e.g. CORINA [135] and CONCORD [136].

Since molecular docking of conformational ensembles depends on a representative sampling of the entire space of bioactive conformations [137, 138, 139], tools which generate multiple low energy compound conformations needed to be developed. This led to the development of stand-alone tools like OMEGA [140], CATALYST [141], ROTATE [142], and the TRIXX CONFORMER GENERATOR [143].

OMEGA utilizes fragment template libraries and histograms to represent torsion-angle energies. First, exhaustive sampling of molecular fragments with up to five rotatable bonds is performed. Then, the molecule is reassembled by successively choosing fragments with lowest energy. The search terminates if no more conformations can be built,

a predefined number of conformers is reached, or if a global energy cut-off is reached. Eventually, diversity filters are applied. [144].

Catalyst starts with a coarse but fast conformational sampling of the molecule [145, 146]. Subsequently, conformational diversity is introduced to the ensemble via energy minimization using a regular molecular force field with one additional term. This term penalizes similar conformers and forces them to diverge [147]. In order to speed up calculations a fragment library can be applied.

Rotate uses a depth-first search based on a tree representation of molecular fragments using the central fragment as root. For large and flexible molecules only the most central bonds can be processed. This reflects the strong influence of the corresponding bond angles on the overall shape of a molecule. During the search, each bond is rotated in 30 degree increments. In each step, only the six best-scoring angles according to an empirical energy potential are retained. Geometry optimizations, energy cut-offs, and diversity filters are applied to the final result set.

The TCG is used as conformer generator within the TrixX BMI workflow. It uses a tree representation of the molecule and traverses it using a best-first search guided by force-field energies. Details of the conformational sampling algorithm are presented in Section 6.4.1.

3.6. General Problems

Today, VS based on 3D structural information of the protein target provides an opportunity for identification of novel lead candidates. Yet, certain challenges remain: 85 % of the about 60.000 proteins in the protein databank (PDB) [148], the worldwide repository for 3D macromolecules, contain one to three flexible side chains [149]. This seems to be a small number, but in context of docking a small adjustment can have a tremendous impact. Thus, protein flexibility is of central importance, particularly when it comes to scoring pose predictions in different protein conformations. Also, the presence or absence of water molecules within the active site needs to be addressed in more detail. Water particles have significant energetic contributions and are of high importance for the reliable prediction of ligand binding. Furthermore, entropy and solvation effects play a key role in the further development of accurate docking and scoring methods. Another issue is connected to the coverage of conformational space. The

right balance between accuracy and speed, between coarse conformational ensembles and a full molecular dynamics approach, needs to be determined since computational resources are limited.

4
Data Indexing

Searching in a database of molecular descriptors can be a computationally expensive task. In order to avoid a sequential scan over the whole data set, index structures must be applied. Different kinds of queries and data environments call for specialized indexing systems appropriate for the data to be searched. In general, index structures can be classified into hash-based and interval-based structures. The former use a hash function to map descriptors to certain hash bins. Thus, a search corresponds to a look-up in the respective hash bin. Subsequently, the resulting candidates need to be checked, since hash functions are not bijective. Interval-based index structures, for example the B-Tree [150] and kd-Tree [151], organize the data hierarchically.

The data volume to be analyzed in high-throughput VS approaches is not suitable for main memory structures. Therefore, the next section focuses on index structures that consider the effects of secondary storage persistence. These indices aim at minimizing I/O operations and thus the access to disk pages.

4.1. Index Structures

Different index structures and their associated algorithms support different types of data and queries. Point-access methods usually perform well if the application primarily searches for points in multi-dimensional space. In contrast, space-access methods are optimized to support efficient spatial selection, for instance range queries on spatial objects. In order to enable efficient data access, it is essential to analyze the data to be indexed regarding its properties. This means that the performance of an indexing system depends on the dimensionality of the data, its distribution, and its frequency of change as well as the query type that is predominantly used when searching the data.

In the following, some widely used indexing approaches and their strengths and weaknesses are presented [152, 153].

B-Trees are balanced search trees optimized for storing and searching 1D data. A node of a B-Tree is usually mapped to a disk page. If a page overflows, the data needs to be split based on a total ordering of the keys. Insertions, deletions, and updates of data can be performed in $O(\log_k(n))$, with n being the number of search keys and k the maximum number of keys in one page. In order to support different types of queries efficiently, several B-Tree like structures can be employed. B^+-Trees solely store data on leaf level. This increases the fan-out of the inner nodes and thus reduces the height of the tree. Range queries are efficiently supported by linking each leaf with its predecessor and successor. Multi-key data can be accessed and stored efficiently using kdb-Trees [154] that combine the balanced structure of a B-Tree with the multidimensional features of the kd-Tree. However, this index structure is only suited for point data [155].

R-Trees [155] are a multidimensional generalization of the B^+-Tree and thus also map data to disk pages. In contrast to B^+-Trees that use 1D intervals to structure the data, the R-tree uses multidimensional rectangles, which allow efficient handling of spatial objects. All nodes store a minimum bounding rectangle of its children, which is used to guide the search. Similar to the B^+-Tree, data is only stored in leaf nodes. In case of a page overflow, a split needs to be made aiming at the minimization of rectangle overlap and future page overflows. A simple heuristic for this is to minimize the area of the two resulting rectangles. Since the original R-Tree contains overlapping rectangles, it cannot guarantee a logarithmic worst-case performance. Applications like the R*-Tree [156] overcome this problem by usage of sophisticated insertion and split strategies. However, in higher dimensional spaces the overlaps tend to affect the majority of the data. One possible way to solve this problem is to use so-called supernodes as employed by the X-Tree [157]. These nodes can exceed the usual page size, and the index reverts to a linear data scan. Supernodes exceed the usual page size and are generated in cases where only overlapping splits are possible. The data in these nodes is linearized, and the index accesses it sequentially.

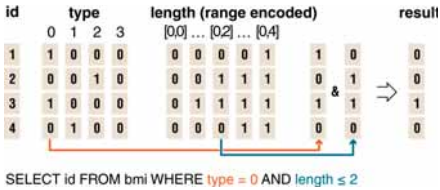


Figure 4.1.: Example query on a table holding 2D data that is indexed by Bitmap Indices. The type information is encoded using equality encoding, whereas the index on the length dimension utilizes range encoding.

Grid Files [158] use a multi-dimensional hashing scheme. The underlying data is partitioned with a k-dimensional grid using nonuniform scales for each dimension. On top of the grid, a k-dimensional dynamic array — the grid directory — is used to map each grid cell to a specific disk page. Different grid cells can be combined to regions, which are mapped onto the same page if the covered region is convex in k-dimensional space. Over- and underflowing pages require split and merge strategies. These strategies must preserve the convexity of all regions in the grid directory.

A Grid File can efficiently answer fully qualified point queries with just two disk accesses. Furthermore, the convexity of the directory regions allow efficient k-dimensional range queries. A disadvantage of Grid Files is the superlinear growth of the grid directory. Adaptations, like the Two-Level Grid File [159], introduce a second layer Grid File to handle the grid directory itself.

Bitmap Indices [160] are often used to answer complex queries in read-mostly environments. A Bitmap Index builds one index for each dimension A. Each index consists of $|A|$ bitmaps. Bit value 1 in row i of bitmap j of the index then represents the value of $\text{row}_i[A] = j$. Multi-dimensional queries can be answered efficiently by boolean combinations of the corresponding bitmap vectors. Furthermore, it is possible to adjust the bit encoding to support different types of queries [161], for example range encoding. In this case a set bit in row i in bitmap j represents $\text{row}_i[A] \leq j$. The choice for a specific encoding scheme decreases the number of bitmaps to be evaluated significantly. If a dimension is queried predominantly using range conditions, its corresponding bitmaps should be encoded accordingly. In case of range encoding, a range query can be answered using a single bitmap. For equality encoded bitmaps, a range query needs to

combine all bitmaps within this range using a logical OR operation. This resuls in $\frac{|A|}{2}$ bitmap combinations in the worst case. An example query using two dimensions (type and length), as well as different encoding schemes (equality and range), is illustrated in Figure 4.1.

Bitmap Indices can efficiently answer multi-dimensional queries and are especially useful for ad-hoc queries where the involved data dimensions are previously unknown. Depending on the cardinality of an indexed dimension, bitmap indexing can be space consuming. Binning and compression techniques can be used to approach this problem. Furthermore, insert and update operations are more expensive than they are for tree-based structures.

5
Background Overview

In this chapter prerequisites and predecessors of TrixX BMI are presented. First, the molecular docking program FlexX is introduced which supplies basic routines for molecule I/O, molecule initialization, scoring, and an *incremental construction* procedure. Furthermore, TrixX BMI employs some concepts of TrixX, a tool for prefiltering large compound collections. This includes its model of the active site and parts of the TrixX molecular descriptor. To conclude this chapter, the indexing system used to search in high-dimensional descriptor space and the database to organize descriptor matches is introduced.

5.1. FlexX

FlexX employs a systematic approach to protein-ligand docking. According to Chapter 3, such a docking algorithm is based on three components: A model for the search space, a search algorithm, and a scoring function to evaluate the resulting predictions.

- The search space is represented using an empirical interaction model as described in 3.1.1.
- The search for protein-ligand poses is performed systematically by an incremental construction algorithm (see 3.1.2).
- Intermediate and final poses generated by FlexX are evaluated using an empirical scoring function as presented in 3.2.3.

In the following, each of these components is presented in more detail.

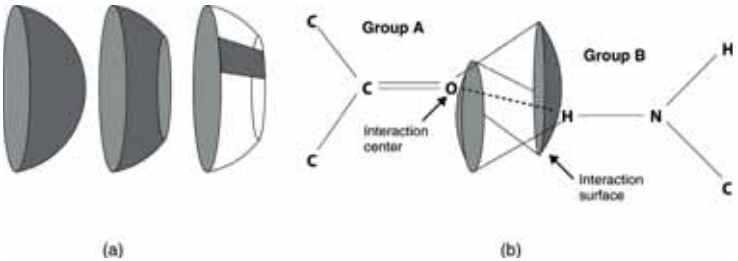


Figure 5.1.: (a) Different interaction geometries of FlexX. (b) Favorable superposition of two opposing chemical groups

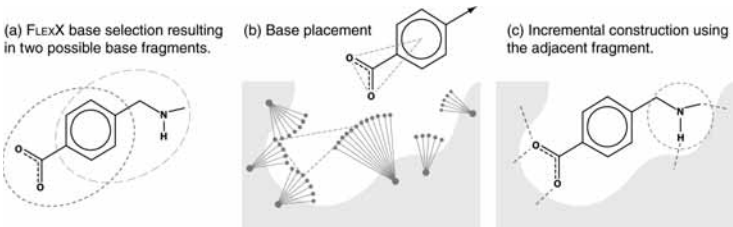


Figure 5.2.: Three steps of the FlexX *incremental construction* algorithm.

Interactions FlexX represents the protein active site using a modified version of the empirical interaction model of Böhm [44], later adapted by Klebe [162]. It consists of a set of rules describing preferred interaction geometries for H-bonds and hydrophobic contacts between functional groups of protein and ligand. These rules result from a statistical analysis of nonbonded protein-ligand contacts in complexes of the CSD. Each interacting group of a molecule is assigned with a specific geometry. This geometry is defined by an interaction center, a radius, and an interaction surface as shown in Figure 5.1 (a). Depending on the interaction type of the functional group, different surface radii and surface shapes are used [3]. An interaction is formed if the involved interaction types are complementary and the interaction center of both groups is located approximately on the interaction surface of the counter group (see Figure 5.1 (b)). For computational reasons, the interaction surfaces of protein functional groups are discretized resulting in a finite set of so-called interaction dots.

Docking The docking algorithm of FlexX consists of three different phases, which are illustrated in Figure 5.2.

- Initially, a *base selection* routine identifies one or more possible anchor fragments. This is done using selection criteria like rigidity and hydrophilicity and results in up to ten base fragments. For each of these anchors, the remaining compound is further fragmented at each acyclic rotatable bond yielding a set of fragments to be used during incremental construction. These fragments consist of either ring systems or rigid components. Eventually, base fragments and flexible ring systems are conformationally sampled.[163]

- During *base placement*, FlexX tries to identify high-affinity placements for the anchor fragments. First, all triplets of interaction groups within the individual base fragments are identified. Second, triangle geometries for the different fragmentations are generated. Then, an efficient pose clustering [164] algorithm generates matches between these compound triangles and triangles derived from protein interaction centers. Each of the resulting matches corresponds to a transformation of a base fragment into the active site and thus yields an initial anchor placement. Subsequently, all valid placements concerning interaction directions and clashes are clustered using RMSD of atom coordinates as diversity measure [165]. After an optimization step to merge members within each cluster, clashing placements are discarded. The remaining ones are scored using a heuristic term to represent the maximal possible energy contribution of the nonplaced parts of the compound.

- The anchor placements serve as starting point for the *incremental construction* routine that uses a k-greedy heuristic. Starting with the k best base placements, only the k best scored placements resulting from each iterative construction step are retained for further processing. In each construction step, the algorithm places all conformers of the next component using a set of predefined torsion angles of the connecting bond. The resulting set of partial solutions is optimized using a weighted superpositioning scheme, and similar placements are clustered. Then, the resulting placements are subject to a clash test. After scoring, the k best ranked solutions are further processed. This process continues until all fragments are added.

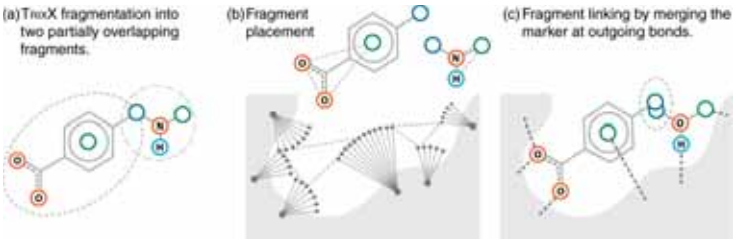


Figure 5.3.: Three steps of the TrixX *placement and linking* algorithm.

Scoring The empirical scoring function of FlexX, as presented in Section 3.3, is tightly coupled to the interaction model: The optimal alignment of two compatible groups is associated to a value that estimates its contribution to the binding free energy according to the empirical analysis it is based on. Deviations from this optimal alignment are penalized using distance and angle deviations. The remaining entropic contributions to the binding free energy is estimated based on pairwise atom-atom distances to account for the hydrophobic effect. Since all parameters and geometry specifications are user configurable, different empirical models like ChemScore and PLP score can also be employed during a FlexX docking run.

5.2. TrixX

In 2007 Schellhammer and Rarey published TrixX, a new method for molecular docking based on the fundamental models of FlexX and the *placement and linking* docking paradigm (see Figure 5.3). Its unique contribution towards molecular docking is its search strategy for initial fragment placements: TrixX uses relational database techniques to access molecular descriptors and exploits redundancies within the data set.

As illustrated in Figure 5.4, the TrixX workflow can be split into five phases.

- Phase one (*compound cataloging*) is independent of the protein target and needs to be performed only once. Based on functional groups of the molecule, so-called compound interaction center (CIAC), are identified (see Figure 5.5). It then decomposes the compound into partially overlapping fragments and stores these in a

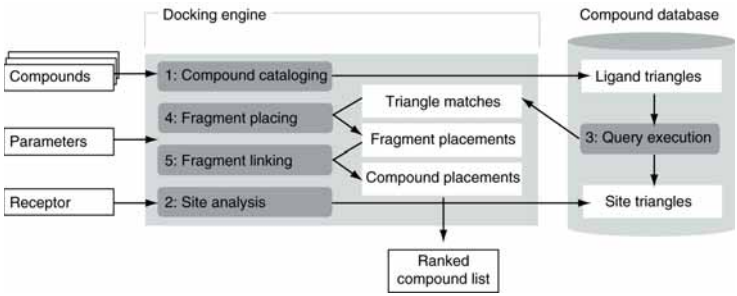


Figure 5.4.: Workflow of the TrixX docking algorithm.

Figure 5.5.: A compound (left) and the corresponding CIAC assignments (right).

relational database system. Since TrixX fragments are rather small, consisting of only up to five rotatable bonds, a duplicate check is performed. New fragments are conformationally sampled and registered. Duplicates are annotated at the corresponding fragment. Thus, fragment redundancies in the compound library are exploited. Then, TrixX molecular descriptors are generated. Each descriptor is assigned to a specific fragment and represents a conformation dependent triangle between functional groups of a compound. A TrixX descriptor encodes the following molecular properties: The types of the involved interaction centers, their interaction direction as Euler angles, their pairwise distances, and a representation of nearby steric properties of the fragment. Eventually, the descriptors are also stored in the compound database.

- During phase two (*site analysis*), TrixX generates favorable interaction spots, so-called site interaction center (SIAC), based on type dependent clustering of FlexX interaction dots of the protein. The resulting SIACs represent favorable positions for CIAC placements. Then, triplets of SIACs are used to generate TrixX descriptors for the protein active site. These site descriptors are complementary to the compound descriptors and are also stored in a table of the database.

- In phase three (*query execution*), the previously stored TrixX site descriptors are used to query the table of compound descriptors in the compound database. Each descriptor match describes a valid transformation of the annotated fragment into the active site of the protein. Two descriptors match if the involved interaction types are compatible (see Table 5.1), the pairwise interaction distances and interaction directions are equal within certain thresholds, and the TrixX description of bulk yields a reasonable steric fit.

- The next phase (*fragment placing*) performs the actual transformation of each fragment into the active site using the conformation and transformation identified via the corresponding descriptor match. After clash testing the resulting fragment placements, TrixX employs a coarse scoring function that evaluates the superposition of CIACs and SIACs. The highest scoring placements are stored in fragment-specific priority queues.

- Within the final phase (*fragment linking*), TrixX tries to link fragment placements to compound placements. Therefore, each fragment placement is expanded with its associated compound fragments as stored in the compound database. Fragment placements of the same compound are merged if they are distance and overlap compatible. The resulting compound placements are again checked for clashes. Then, TrixX searches for additional interactions that are not part of initial descriptor matches. Finally, the resulting placements are sorted into compound specific priority queues. These are used to generate the final hitlist of the current target protein.

TrixX uses distance and overlap criteria to merge fragment placements to pose predictions. These criteria are rather coarse, and the resulting superpositions often

Interaction type	CIAC type	SIAC type
Hydrogen bond	H-Acceptor	H-Donor
	H-Donor	H-Acceptor
Metal coordination	H-Acceptor	Metal
Hydrophobic contact	Phenyl ring, methyl, ethyl, halogen	Hydrophobic spot

Table 5.1.: TrixX main interaction types between functional groups of a compound (CIACs) and of a receptor (SIACs).

result in chemically nonvalid bond lengths and angles. Therefore, TrixX should only be applied as a prefilter to more accurate docking tools.

5.3. Descriptor Indexing

As already mentioned, TrixX uses a relational database system to identify initial fragment placements and exploits redundancies within the library. Its descriptor consists of only few individual attributes since the representation of steric bulk is stored in a single bitmask. The evaluation of steric fit is performed within a postprocessing step as part of the database query is not supported by any index structures.

TrixX BMI uses an entirely new concept for the description of molecular shape. It employs an 80-dimensional shape representation that is supported using indices. The index structure to be used must be able to cope with high-dimensional queries. Furthermore, the descriptor data is part of a read-mostly environment where data is changed only infrequently. Since all these criteria are met by Bitmap Indices (see 4.1), TrixX BMI employs the FastBit [166] system.

5.3.1. FastBit

FastBit is a stand-alone Bitmap Index system. Bitmap Indices are well suited in read-mostly scenarios of high-dimensional data but have certain drawbacks concerning space overhead for attributes of high cardinality. FastBit addresses the issue of high cardinality attributes by applying a binning scheme, such that the index is generated based on a coarser representation of the data. The bin boundaries can either be user-defined or derived using a statistical analysis of the data. FastBit generates candidates using the Bitmap Indices. These candidates are then checked individually: The system reverts

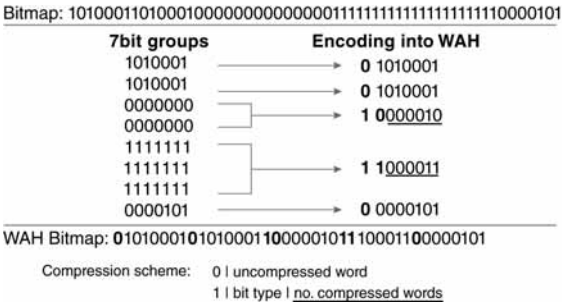


Figure 5.6.: Example of the WAH compression using a CPU word size of 8 bit.

to the raw data to decide whether a candidate fulfills the query condition or not. This scheme can also be applied in a two-level hierarchy to increase the grain of the index and to reduce the number of candidates.

Another aspect of Bitmap Indices is the possibility to compress each bitmap. During query processing, bitmaps are logically combined using boolean operations as for instance AND, OR, and NOT. Regular central processing unit (CPU) operations are not applicable to compressed bitmaps. Thus, standard compression algorithms lead to an increase in runtime since each involved bitmap needs to be decompressed in order to answer a query. However, FastBit uses a special compression method called Word-Aligned Hybrid (WAH) code [167]. This adaptation of run-length encoding enables the usage of standard operations without the need to decompress the bitmaps. The main idea of WAH is to compress bits in groups of CPU-wordsize-1, for example 31 bits on a 32 bit CPU. In combination with an additional bit flag, each group fits exactly into one word of the CPU: A compressed group consists of a leading 1 indicating compression, followed by the bit value that is compressed, and the number of compressed words. Uncompressed groups are stored literally, prepended with a 0 bit. Due to this alignment, standard bitwise operations can be executed efficiently [168, 169]. An example of the WAH compression is given in Figure 5.6.

Another feature of FastBit is the availability of different bitmap encoding schemes (see Section 4.1 and Figure 4.1 for details), for instance equality- and range encoding which can be employed to speed up corresponding query types. Furthermore, FastBit

enables the user to supply fixed memory boundaries: The maximum memory usage can be restricted by assigning a global cache limit.

5.3.2. SQLite

As a result of FastBit queries, descriptor matches need to be stored and organized. This task is handled by SQLite [170] that offers a transactional database engine accessible via the structured query language (SQL).

SQLite is an embedded SQL database engine that does not have a separate server process. Instead, it directly accesses regular disk files. A single file includes a database with multiple tables, indices, triggers, and views. The database file format is implemented for different platforms. It should not be used as a replacement for a grown database management system but can be used to structure and search data efficiently. Similar to FastBit, SQLite allows the definition of fixed memory boundaries: Cache sizes and thus memory consumption can be adapted and constrained during runtime.

6 Methods

In this chapter the ideas behind TrixX BMI are presented. It restates the aims of this project and summarizes the drawbacks of previously developed tools. Subsequently, the concepts and methods of TrixX BMI are described. This includes the introduction of the novel TrixX BMI descriptor, the hierarchical docking pipeline, details about individual phases of the overall workflow, and the TrixX Conformer Generator that handles large parts of compound flexibility. Eventually, the TrixX BMI approach to parallelization is presented.

6.1. Motivation and Goals

Nowadays, VS is a sequential procedure. Only few molecular docking tools deviate from the concept of iteratively placing each compound into the binding site. Cluster-based approaches exist, but these do not result in significant improvements of runtime and often do not produce results of comparable prediction quality. For instance, the original TrixX version is not suited for molecular docking and should only be used to prefilter large compound libraries. VS tools from the field of structure-based pharmacophore search also follow the iterative screening paradigm. Furthermore, these tools do not generate 3D placements but only rank the library compounds.

Most importantly, none of these methods employs a 3D shape matching routine that uses a molecular descriptor which is accessible using modern indexing technology. Therefore, the goal of the TrixX BMI development is to achieve the following:

- Fast prediction of binding molecules based on a selective molecular descriptor that incorporates physicochemical information, especially a canonical representation

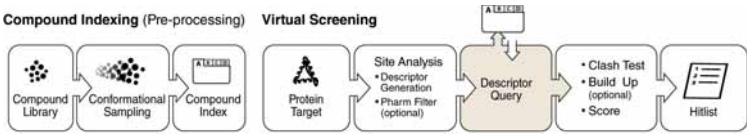


Figure 6.1.: TrixX BMI workflow: The preprocessing phase on the left and the virtual screening part on the right.

of molecular shape. Each descriptor attribute should be accessible using index structures.

- Accurate 3D pose prediction and significant enrichment of active compounds. The results should be of comparable prediction quality to state-of-the-art approaches.
- Incorporation of a-priori knowledge into the docking engine, this means, pharmacophore information and molecular properties of known binders.
- The models to be generated must have the potential to integrate concepts of protein flexibility.
- Scalability in a parallel computing environment.

To achieve these goals, TrixX BMI is modeled as a hierarchical screening pipeline that employs and extends techniques of previously developed tools: It uses the scoring scheme and incremental construction algorithm of FlexX, it adapts and extends the molecular description of TrixX, and incorporates a sophisticated indexing system.

6.2. Workflow

The TrixX BMI workflow, presented in Figure 6.1, can be split into two disjoint phases.

- **Preprocessing**
  The information computed in the first phase is independent of the actual target protein and thus needs to be calculated only once. In this preprocessing step, all library compounds are analyzed to identify their physicochemical features and to generate TrixX BMI *compound descriptors*. These are stored in the so-called

compound database which uses an indexing system optimized for retrieval of high-dimensional data. The problem of conformational flexibility is addressed by using TCG which creates conformational ensembles of the library compounds.

- **Virtual screening**
  In the second phase, structure-based VS is performed. TrixX BMI *compound descriptors*, generated in the preprocessing step, and target dependent *site descriptors* are utilized to rapidly identify candidate compounds and corresponding 3D pose predictions. These initial poses are then analyzed in more detail, and flexible, nonleadlike compounds are subject to an incremental construction algorithm.

The central concept connecting these two parts of the workflow is the TrixX BMI descriptor. This novel methodology of modeling molecular properties separates TrixX BMI from other docking approaches: Molecular compounds can be directly accessed and 3D properties can be compared solely on the abstract descriptor level. SQL queries, based on site descriptors, are utilized in combination with modern indexing technology to discard incompatible compounds and identify likely binding molecules. Since only compounds identified by descriptor matches are subject to docking calculations, TrixX BMI breaks the iterative screening paradigm.

6.3. TrixX BMI Descriptors

The most prominent concept used in TrixX BMI is the descriptor that is shown as *compound* and *site descriptor* in Figure 6.2. The motivation for its development is the idea, that an abstract representation suffices to identify reasonable pose predictions and to discard unlikely binders. Thus, the descriptor is modeled to resemble a three-point pharmacophore of interactions between functional groups. It captures steric and electronic features of a ligand or a protein active site, respectively.

A descriptor is based on an ordered triplet of interaction centers: For a compound, this triplet consists of CIACs (c_1, c_2, c_3) of compound functional groups. For a protein active site, the descriptor is based on SIACs (s_1, s_2, s_3), interaction spots which discretize the interaction geometries of a protein. For details about the CIAC/SIAC model of functional groups, please refer to Section 5.2. The corresponding interaction types

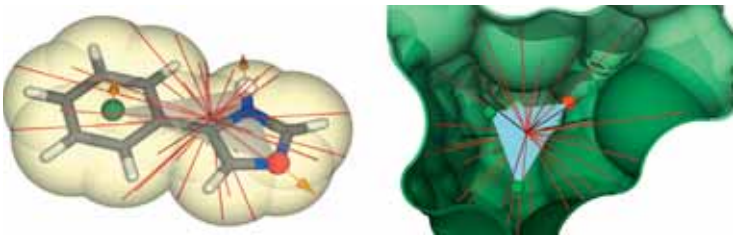


Figure 6.2.: TrixX BMI descriptor of a compound (left) and a protein active site (right). Three interaction centers (red, green, and white spheres), their interaction direction (orange arrows), and 40 of the 80 rays representing shape relative to the triangle (red rays).

of the ordered triplet are encoded into a type attribute $t \in \{1, ..., 10\}$. For example, the type $t = 0$ corresponds to a descriptor based on three H-bond donor SIACs.

Pairwise Euclidean distances between SIAC coordinates are stored as (l_1, l_2, l_3), $l_i \in \mathbb{R}$. Shape is modeled using a novel 80-dimensional distance vector $(b_1, ..., b_{80})$, $b_j \in \mathbb{R}$ and the individual interaction directions are stored as (dir_1, dir_2, dir_3), $dir_i \in \mathbb{R}^3$. Furthermore, each descriptor can be identified using a unique identifier $id \in \mathbb{N}$.

TrixX BMI descriptors for compounds and protein active site both employ this basic descriptor but differ regarding the calculation of individual descriptor attributes. Individual properties are encoded complementary, most notably regarding the description of shape: TrixX BMI uses a distinct alignment based on local descriptor properties in order to describe the global shape of a compound or an active site, respectively. Furthermore, *compound descriptors* are augmented with additional attributes which are necessary to transfer a descriptor match into a 3D pose prediction.

6.3.1. Modeling of 3D properties

In the following, the models for the description of 3D properties of TrixX BMI descriptors are presented. In contrast to competing approaches, TrixX BMI exploits these properties already on the descriptor level by formulation of highly selective, descriptor-based queries. This is realized using translationally and rotationally invariant models to represent molecular shape and directional interaction constraints.

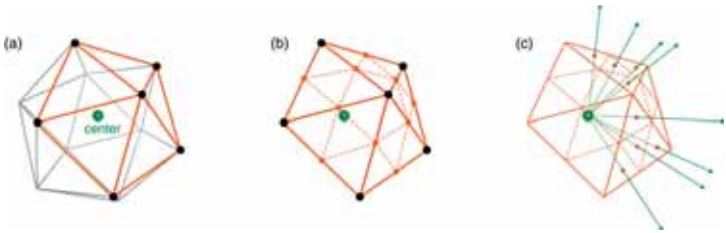


Figure 6.3.: Example of the icosahedron refinement and the generation of the 80 rays which are used for shape description. (a) Basic icosahedron consisting of 20 triangle faces. Its center is depicted as green sphere. (b) Subdivision of each triangle into four subtriangles. (c) Eight rays of the 80-dimensional shape description of TrixX BMI.

Steric bulk

The average depth of a drug-binding cavity is in between 6.8–11.4 Å, and the maximum depth is in the range of 13.0–22.9 Å [48]. In order to identify valid poses, TrixX BMI needs to consider the global shape of the protein active site and the ligand. At the same time, constraints on descriptor size have to be regarded. Most importanly, the description of shape should enable a descriptor based detection of protein-ligand overlap.

TrixX BMI approaches this problem with an 80-dimensional distance vector which is locally aligned to the descriptor geometry. The description includes all atoms within a radius of 7.5 Å around the geometric center of the descriptor and thus covers 15 Å in diameter. As shown later, the choice for this boundary allows a compact representation using just one byte for each dimension and still allows detailed overlap predictions.

The description of steric bulk is based on a refined icosahedron, which consists of 80 triangle faces. Each of the 20 regular icosahedron triangles is subdivided into four subtriangles using the center points of the original triangle edges as additional corner points. Based on these three and the original triangle corners, four nonoverlapping subtriangles are generated (see Figure 6.3). The resulting triangles are then used to generate 80 different direction vectors, so-called rays r_j, for later shape calculations: All r_j originate from the center of the icosahedron. Their individual direction is determined by the direction from this origin to the geometric center of the different triangle faces.

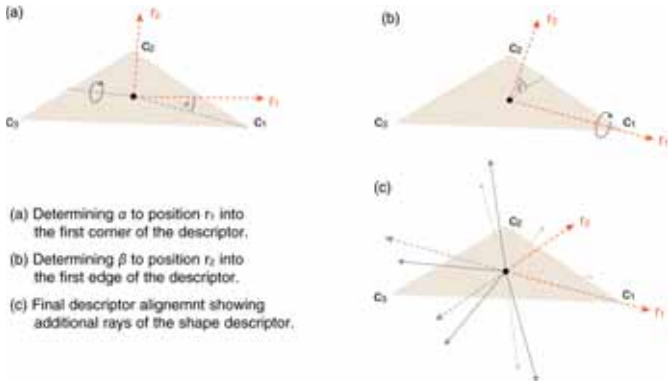


Figure 6.4.: Example alignment of the bulk descriptor rays into the triangle geometry of a TrixX BMI descriptor.

These rays are used to generate a representation of global shape relative to the current descriptor. For a shape description that is invariant to translation and rotation, a unique descriptor alignment based on local descriptor properties is necessary: First, the origin of the rays is translated into the geometric center of the descriptor. Then, a rotation α is determined that positions the first ray r_1 into the descriptor plane, such that it coincides with the first corner of the descriptor triangle as given by the canonical order, e.g. c_1 in case of a *compound descriptor* and its ordered CIACs (c_1, c_2, c_3). In more detail: α is the angle between vector r_1 and the vector from the descriptor origin to corner c_1 using the cross product of the two involved vectors as rotation axis (see Figure 6.4 (a)). After applying α to each r_j, the alignment is fixed by determining a second rotation β around r_1 as rotation axis that causes r_2 to be in the plane of the descriptor triangle and to be directed towards the edge between (c_1, c_2) (see Figure 6.4 (b)). Eventually, β is applied to each r_j. The final result of this alignment is shown in Figure 6.4 (c). SIACs are used instead of CIACs if the rays r_j are aligned to a *site descriptor*.

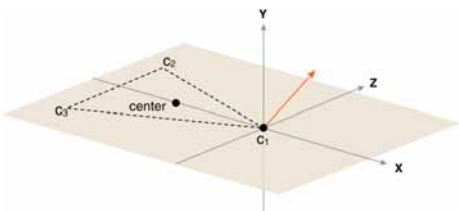


Figure 6.5.: Alignment of a descriptor into the local reference system according to CIAC c_1.

Interaction directions

In order to generate a translational and rotational invariant representation of interaction directions, the descriptor is aligned to a local reference system for each of its three interactions individually. In case of a *compound descriptor* and one of its CIACs c_i, this is done as follows: The CIAC c_i is translated into the coordinate origin, such that the descriptor center lies on the negative x-axis. Then, the descriptor is rotated around this axis until the next corner according to the canonical order is positioned in the x-z plane with $z > 0$. Finally, the main direction of the current interaction c_i is stored using its coordinate values. The result of this alignment process exemplified for CIAC c_1 is depicted in Figure 6.5. Again, SIACs are used instead of CIACs if a *site descriptor* is aligned.

6.3.2. Compound descriptors

For each compound in the library, TrixX BMI *compound descriptors* are calculated, each based on a triplet of CIACs. TrixX BMI uses solely topological information to subselect these triplets: Two CIACs need to be at least two bonds apart, no more than two CIACs reside in the same ring system, and at least one of them is hydrophilic. This reduces the number of descriptors that are derived from a single ringsystem and removes entirely hydrophobic, thus unspecific, descriptors. If a compound consists of only one ring system and all CIACs are within this system, the corresponding constraint is not enforced. No additional heuristics are utilized. Then, a canonical order scheme is employed that reorders the involved interactions depending on their type and pairwise

Euclidean distances in 3D space. This routine results in the descriptor base order (c_1, c_2, c_3) that is used to calculate the following descriptor properties.

- **Interaction type**
  The types of the involved CIACs are encoded into a single number. Since a descriptor is based on three interactions and there are four different types (H-donor, H-acceptor, hydrophobic, and metal), this results in $\binom{6}{3} = 20$ possible arrangements. Metal CIACs are in general not part of leadlike compounds, thus there are actually just ten different interaction types for *compound descriptors* $(t \in \{1, ..., 10\})$.

- **Side length**
  The side length attribute depends on the conformation of a compound but not on its position in 3D space. Depending on the compound's current conformation and the descriptor's base order (c_1, c_2, c_3), pairwise Euclidean distances $d(c_1, c_2)$, $d(c_2, c_3)$, and $d(c_3, c_1)$ are calculated and stored.

- **Steric bulk**
  The descriptor values $(b_1, ..., b_{80})$ are computed using the previously described alignment of the 80 rays r_j and the current *compound descriptor* (see 6.3.1): Each b_j encodes a so-called *exit distance* that represents the extension of the molecule with respect to the descriptor. Starting from the center of the descriptor, the distance in direction of ray r_j is measured until this ray exits and leaves the interior of the compound. A 2D example using three arbitrary rays of the icosahedron is illustrated in Figure 6.6.

- **Interaction direction**
  For each CIAC in the ordered triplet (c_1, c_2, c_3) of the descriptor, the alignment scheme for interaction directions (see 6.3.1) is applied. The resulting interaction direction of CIAC c_i is stored using 3D coordinates in dir_i.

- **Reconstruction data**
  Since each *compound descriptor* is linked to a specific molecule, fragmentation, and conformation, the basic descriptor is augmented with a compound identifier $m_{id} \in \mathbb{N}$, a compound fragmentation $f_{id} \in \mathbb{N}$, and a specific conformation $c_{id} \in \mathbb{N}$.

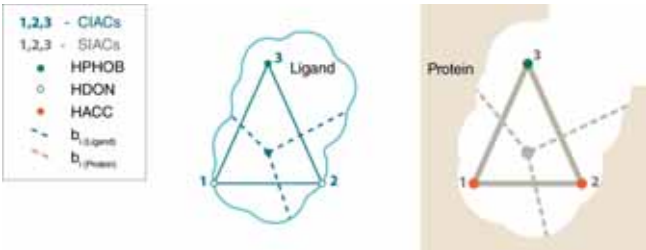


Figure 6.6.: 2D example of descriptor bulk generation using three arbitrary directions for ligand (left) and site descriptor (right).

In combination with a unique compound specific descriptor identifier, this data is later used for reconstructing 3D placements, so-called *descriptor poses*, from matching *site*- and *compound descriptors*.

Flexible CIACs

Molecular flexibility involves terminal rotatable bonds, as for instance in hydroxyl groups. These groups are not considered during conformational sampling. Since RMSD clustering usually accounts only for heavy atom distances, a rotation of such a bond does not increase the diversity and is therefore discarded. However, *descriptor poses* rely on correct orientations of hydrogen atoms since donor CIACs are registered at these atoms.

TrixX BMI approaches this problem by applying an orientation independent representation for flexible hydrogen donors. For *compound descriptors* the model is adapted as follows: Donor CIACs are not assigned with an interaction direction if they belong to a terminal flexible group. Instead, they are marked with a special value indicating flexibility. Furthermore, the calculation of triangle side lengths is adjusted and is now based on the mean of all possible CIAC positions. Further adjustments are made during molecular docking as presented in 6.5.1.

6.3.3. Site descriptors

Site descriptors are generated analogously to *compound descriptors* and are also based on the basic TrixX BMI descriptor. Individual attributes are calculated complemen-

tary to enable the comparison of 3D properties solely based on the descriptor level.

Instead of CIACs (c_1, c_2, c_3), *site descriptors* are based on SIACs (s_1, s_2, s_3) which are ordered according to the same scheme. The basic descriptor attributes are calculated as follows:

- **Interaction type**
  The interaction type t of a *site descriptor* is encoded using its complementary interaction type as given in Table 5.1. A Metal SIAC is encoded as an H-Acceptor, and again only ten different values — as for the interaction type of a *compound descriptor* — can occur.

- **Side length**
  The generation of the side length attributes (l_1, l_2, l_3) remains unchanged. Pairwise Euclidean distances $d(s_1, s_2)$, $d(s_2, s_3)$, and $d(s_3, s_1)$ are calculated and stored.

- **Steric bulk**
  Site descriptors also employ the alignment procedure as described in 6.3.1. The calculation of the shape values $(b_1, ..., b_{80})$ is adapted in order to describe shape complementarity to a *compound descriptor*. A *site descriptor* represents the cavity relative to the descriptor. Each b_j encodes a so-called *clash distance* that represents the empty space between the center of the descriptor and the protein surface. The value of b_j is calculated by identifying the closest active site atom in direction of r_j and encoding the corresponding distance. Figure 6.6 illustrates the complementary approach of *exit* and *clash distances* to model molecular shape for *compound*) and *site descriptors*.

- **Interaction direction**
  For each SIAC in the ordered triplet (s_1, s_2, s_3) of the descriptor, the alignment scheme for interaction directions (see 6.3.1) is applied. In order to enable a direct comparison of *compound* and *site descriptors*, a *site descriptor* stores the direction from a SIAC to its corresponding heavy atom. This inversion of the interaction direction and the usage of the same reference system yields comparable interaction coordinates for *compound* and *site descriptors*. In case of perfectly aligned functional groups the coordinates in the reference system, thus the vectors

dir_i, are identical. Protein flexible groups are not modeled in TrixX BMI, and there is no adaptation towards flexible SIACs associated to terminal rotatable groups of the active site.

TrixX BMI *site descriptors* reside in main memory and are associated to the current protein target in its given conformation. Therefore, *site descriptors* are not augmented with additional attributes and the standard TrixX BMI descriptor suffices.

The selection algorithm choosing the triplets of SIACs for descriptor generation reuses the original constraints of TrixX: A minimum and maximum SIAC–SIAC distance between 1 Å and 9.5 Å and a constraint describing the pocket environment of a descriptor triangle. A descriptor must not clash with the protein. Thus, *site descriptors* cover only concave parts of the protein.

6.3.4. Descriptor size and binning

Since a large number of descriptors are necessary to represent a single compound, descriptor size is an important factor concerning disk space in general and I/O load during VS. Especially, the storage requirements of a Bitmap Index depend on the cardinality of the attribute to be indexed. Thus, continuous descriptor attributes should be subjected to binning (see Table 6.1). However, the granularity of the scheme should not be to coarse.

- The type attribute, which is already discrete and has only ten distinct values, is not binned and is stored using 1 byte.
- Interaction directions are binned using 100 equidistant 0.1 Å buckets to represent their coordinate values in the interval $[-5.0, 5.0]$. In total, 18 bytes are necessary, 6 bytes for each interaction direction, respectively 2 bytes for each coordinate.
- The side length attribute is also binned into equidistant 0.1 Å buckets. Here, 85 buckets are used to cover the minimum/maximum range of site descriptors $l_i \in [1.0, 9.5]$. Thus, 1 bytes for each side and in total 3 bytes are necessary.
- The steric bulk description is 80-dimensional and represents the majority of the descriptor data. A lower granularity of 15 equidistant buckets of size 0.5 Å is selected to map the individual $b_j \in [0, 7.5]$. Thus, 1 byte suffices for each of the 80 dimensions.

Furthermore, there is the reconstruction data consisting of molecule identifier (4 bytes), fragmentation number (1 byte), conformation number (2 bytes), and descriptor identifier (2 bytes) adding up to 9 bytes. This leads to a total of 111 bytes storage requirements for each descriptor.

A compound library of 1 million ligands, having 10 conformations on average and about 100 descriptors each, needs about 100 gigabyte (GB) to store the raw descriptor data. The Bitmap Index description based on the binned data requires about twice the space of the original data, such that the on disk requirements of such a compound database are about 300 GB.

Adapting the resolution of the binning scheme would alter space requirements, especially in case of the shape description. In the current settings, the descriptor range of 7.5 Å covers large parts of the protein active site and only one byte suffices to store a single attribute. Most importantly, the 0.5 Å bin sizes still allow for detailed query formulation.

6.3.5. Index encoding

For Bitmap Indices different encoding schemes can be selected to efficiently support the predominant query type (e.g. equality or range) to be used on the attribute (for details, please refer to Section 4.1). It is essential to select the appropriate bitmap encoding in order to efficiently access descriptor data on the basis of Bitmap Indices. In case of TrixX BMI, no ad-hoc queries are executed. Thus, each Bitmap Index can be encoded as it is necessary for descriptor matching (see 6.3.6).

All attributes of a TrixX BMI descriptors are predominantly queried using one specific condition. Steric bulk, interaction directions, and side length are primarily used in range or interval conditions. Therefore, these are stored using range encoding. This encoding suffices to answer a range query on a specific attribute by reading a single bitmap. Since an interval query ($min \leq x \leq max$) for discretized descriptor data can be rewritten as $((\neg(x < min)) \wedge (x \leq max))$, two bitmaps suffice to answer this kind of query. The remaining type attribute is equality encoded. A summary of encoding types is given in Table 6.1.

Descriptor attribute	Encoding	Bucket size	#Buckets	Tolerance
Interaction type	equality	n.a.	10	n.a.
Side length	range	0.1 Å	85	±1.2 Å
Interaction direction	range	0.1 Å	100	±1.5 Å
Steric bulk	range	0.5 Å	15	+0.5 or +1.0 Å

Table 6.1.: Bitmap encoding, bucket size, number of buckets, and query tolerances of the descriptor attributes.

6.3.6. Descriptor matching

Based on the TrixX BMI descriptor model, compounds can be directly accessed by their chemical- and shape complementarity to a given protein active site. In contrast to descriptor models applied by other approaches (see 3.3.1, 3.4), the TrixX BMI description of shape is aligned via a reference system based on local descriptor properties. This distinct alignment yields an automatically derived structure-based pharmacophore that includes a global description of shape and can be used to identify complementary *compound descriptors*.

Since all descriptor properties are calculated invariant to translation and rotation, they can be assessed as part of the database query. For each *site descriptor* a query can be formulated, such that a descriptor match implies a reasonable 3D pose prediction. Noncompatible, clashing compounds are discarded early during the search. Within a query, the interaction type is tested for equality. The remaining attributes are not checked for exact matches but use certain thresholds to model imperfect placements and to account for the discrete model of molecular flexibility: Side lengths and interaction directions of a *compound* and a *site descriptor* have to be within a certain tolerance interval (see Table 6.1), whereas the compound's *exit distances* must be smaller than the corresponding active site's *clash distances*. The tolerance values for steric bulk depend on the length of the current *clash distance*: Small distances (≤ 4 Å) employ a tolerance of 0.5 Å, whereas larger distances use a tolerance of 1.0 Å.

The query tolerances of each attribute are mapped on the corresponding binning scheme (see Table 6.1). Thus, the bin boundaries for each attribute are exactly met during the query phase: The index alone suffices to answer each query. No candidate check needs to be performed since all descriptor matches are already described in sufficient detail by the index itself.

A match m_{d_s,d_c} of a *site descriptor* d_s and a *compound descriptor* d_c must therefore meet the following condition:

$$\begin{aligned}
\forall_{i,j,k} \quad & t[d_s] = t[d_c] \\
& \wedge \ (dir_i[d_s][k] - \Delta_{dir} \leq dir_i[d_c][k] \leq dir_i[d_s][k] + \Delta_{dir} \\
& \quad \vee \ dir_i[d_c][k] = \text{FLEX}) \\
& \wedge \ l_i[d_s] - \Delta_l \leq l_i[d_c] \leq l_i[d_s] + \Delta_l \\
& \wedge \ b_j[d_c] \leq b_j[d_s] + \Delta_b, \\
& i, k \in \{1, 2, 3\}, \ j \in \{1, ..., 80\}
\end{aligned}$$

The term $dir_i[d_c][k]$ denotes the coordinate value of the kth dimension of the interaction direction i of a descriptor d_c. Thus, the condition for a single interaction direction dir_i can be depicted as a box query around its 3D coordinate.

The case of flexible donors, for instance the donor CIAC on a hydroxyl group, is handled separately. The interaction directions of these groups are rather unspecific and eventually determined during protein-ligand complexation. Such an interaction is marked as flexible ($dir_i[d_c][k] = \text{FLEX}$) and is not suited for filtering.

Molecular flexibility and imperfect matching entail another drawback concerning the representation of the active site: A strict canonical order of a *site descriptor* potentially misses reasonable descriptor matches. If a *site descriptor* does not have pairwise dissimilar interaction types, a strict canonical order of a *site descriptor* does not suffice. There can be more than one reasonable superposition with a corresponding *compound descriptor*. Due to molecular flexibility and thus thresholds as part of the query condition, a strict canonical order misses matching descriptors, e.g. for an almost isosceles triangle of identical interaction types. In such ambiguous cases, where the canonical order is almost arbitrary and a small shift in the compounds conformation results in a different base order, all possible orders are used for descriptor calculation. For example, a triangle consisting of three identical interaction types yields six possible superpositions.

To illustrate the overall process of matching a 2D example is given in Figure 6.7. Part (a) depicts a compound and a protein active site as well as corresponding CIACs and SIACs. The SIACs $\{s_1, s_2, s_3\}$ yield two valid site descriptors differing only in their base ordering. These two *site descriptors* and their superposition with the compound

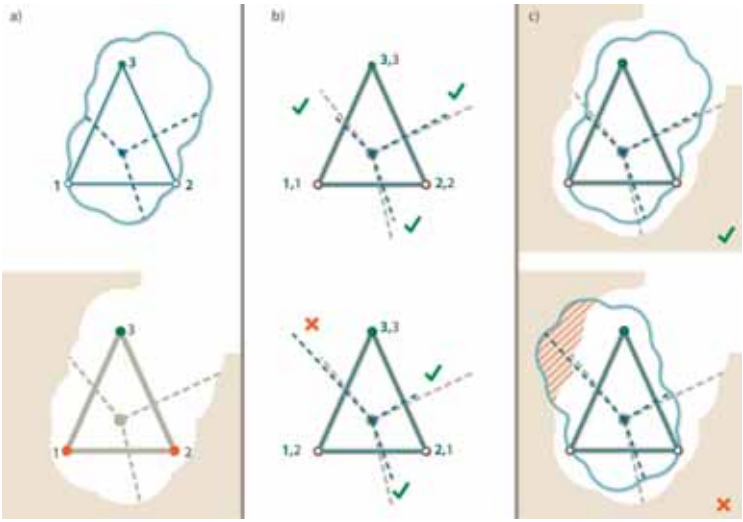


Figure 6.7.: Multiple descriptor alignments and steric clash detection in 2D using the TrixX BMI descriptor. (a) Triangle descriptor and three corresponding bulk directions for ligand and site descriptor. (b) The two possible alignments and the TrixX BMI steric bulk comparison on the abstract descriptor level. (c) The corresponding poses demonstrate the correctness of the clash prediction.

descriptor (c_1, c_2, c_3) are shown in Part (b). Additionally, the corresponding clash predictions based on the TrixX BMI description of shape is illustrated. The resulting protein-ligand complexes in Part (c) reveal that the descriptor accepts the nonclashing and discards the clashing pose. The filter decision is already made on the descriptor level by simply comparing corresponding *clash* and *exit distances*.

6.3.7. Data handling

TrixX BMI minimizes recurring operations by grouping and sorting descriptor matches. For example, each compound is loaded and initialized only once. To achieve this, all descriptor matches are inserted into a temporary database during the query phase and later, during the actual docking, extracted in an appropriate order. Each descriptor

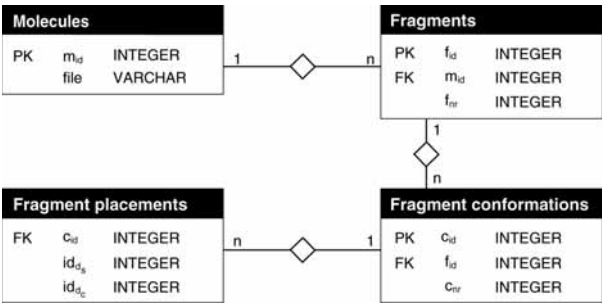


Figure 6.8.: Entity relationship diagram of the storage model for descriptor matches which are returned by the indexing system.

match entails a *molecule identifier* m_{id}, a *fragmentation identifier* f_{id}, a *conformation identifier* c_{id}, and an identifier for the *compound descriptor* itself id_{d_c}. In addition, there is the identifier of the matched site descriptor id_{d_s}. This data is stored persistently using a simple relational schema. A table *fragment placements* storing the attributes id_{d_c} and id_{d_s} is linked using a foreign key constraint to a specific conformation c_{id} in a table *fragment conformations*. Each entry in this table is associated to an f_{id} in the table *fragments* that is connected to a molecular compound via m_{id} in the table *molecules*. The corresponding entity relationship diagram is depicted in Figure 6.8. Eventually, all primary and foreign key constraints in these tables are indexed using regular B-Trees to decrease the complexity of joining and sorting the tables during later docking calculations.

6.4. Preprocessing

As already mentioned, the preprocessing phase is target independent and performed only once for a given compound library. The molecules within that library are represented by the TrixX BMI *compound database* in the *compound index*. This index holds the information necessary to perform molecular docking. An essential aspect of this descriptor representation is that it is based on conformational ensembles. Thus, compound flexibility is handled largely within the preprocessing phase.

6.4.1. Compound flexibility

Before the calculation of compound descriptors commences, TrixX BMI analyses a compound regarding its flexibility. In cases violating a certain flexibility threshold, a fragmentation routine splits the compound into large fragments. Fragments of non-leadlike compounds and leadlike compounds are then sampled individually.

Compound analysis

Before a compound is passed to the conformational sampling phase, its flexibility expressed as number of rotatable bonds (RTB) is estimated using the flexibility definition of Oprea [22]:

$$\mathrm{RTB} = N_{nt} + \sum_i (n_i - 4 - \mathrm{RGB}_i + \mathrm{ShB}_i) \tag{6.1}$$

In this formula N_{nt} is the number of nonterminal freely rotatable bonds, excluding single bonds in groups like sulfonamides or esters. Within the sum, n_i is the number of single bonds in a nonaromatic ring i of six or more bonds. RGB_i and ShB_i represent the number of rigid, respectively shared bonds in ring i. This measure of flexibility incorporates ring bonds, e.g. macrocycles, into the flexibility estimation and neglects terminal rotatable bonds that do not influence the overall shape of a compound.

Compound fragmentation

The default value that triggers compound fragmentation is an RTB of ten. This reflects the maximal size of leadlike structures according to Oprea's analysis [22]. The fragmentation routine then chooses base fragments of seven rotatable bonds. This is done by enumerating all connected subtrees of the so-called *component tree* that is generated by splitting a compound at each acyclic, nonterminal rotatable bond according to the FlexX model. Each of these subtrees is evaluated using a scoring function to assess its adequacy to serve as base fragment. This score is based on two criteria.

- First, the algorithm chooses those fragments that cover the most components of the compound which are currently not present in any of the previously chosen base fragments.

- Second, these candidates are evaluated concerning their interaction potential by scoring their covered interactions using a weighting scheme rewarding hydrophilicity, analogous to the base selection routine of FlexX.

This process is continued until the compound's components are all covered by base fragments, or else a maximum number (the default value is four) of base fragments is selected. Leadlike compounds, respectively corresponding base fragments of nonleadlike compounds, are then handed over to the next phase of preprocessing.

There is no process for identification and registration of fragment redundancies. TrixX BMI fragments consist of up to seven rotatable bonds, which means that identical fragments occurring in multiple compounds are unlikely. Furthermore, the TrixX concept of dummy CIACs to identify reconnection points between fragments is discarded. TrixX BMI uses *incremental construction* in case of compound fragmentation and thus, individual fragment placements do not need to be merged.

Conformational sampling

After compound analysis, the actual conformational sampling is performed using TCG[1]. TCG also treats molecules according to the FlexX model for molecular flexibility and is based on the same data structure, the *component tree*, which is used during compound fragmentation. Each node of this tree consists of either rigidly connected atoms or all atoms of a ring system. The bonds connecting molecular components are used as edges.

Search space TCG combines the *component tree* with the Mimumba model [171] for torsion angles: Each bond of the *component tree* is assigned with the preferred torsions depending on the bond's specific molecular environment. The resulting structure —

[1]TCG is a joint project of Axel Griewel and the author of this thesis, Jochen Schlosser. It is based on the diploma thesis of Ole Kayser which was supervised by the author. The original version featured a best-first search augmented by static depth probes. During further development, flexibility dependent thresholds were incorporated that limit the explored search space size and require a minimum number of conformations. In addition, different quality settings were introduced. The scientific focus of Jochen Schlosser was on adequate sampling of leadlike compounds as necessary for the TrixX BMI screening pipeline: The resulting ensembles should comprise only few conformations, and the individual conformers should resemble bioactive structures. The result of this work, the high-throughput sampling mode of TCG (quality level one), is used as default setting for conformational sampling in TrixX BMI.

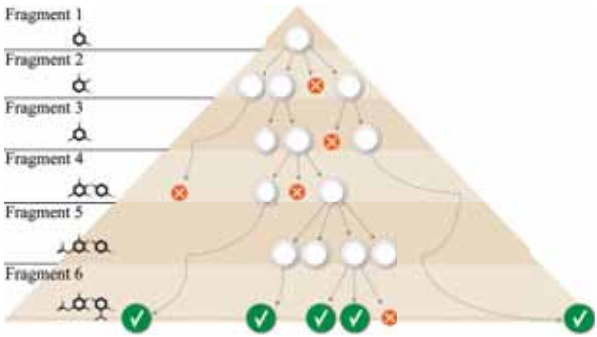


Figure 6.9.: The *conformation tree* and its traversal for an example molecule during the course of the TCG build-up algorithm.

the TCG *conformation tree* — describes the conformational degrees of freedom of a molecular compound: If component C_i is connected via bond b with n torsion angles to component C_j, this results in n edges from node C_i to n nodes $C_{j,k}$ ($1 \leq k \leq n$). In case of flexible ring systems, ring conformations are generated and are also considered while expanding the search. Thus, each $C_{j,k}$ represents a different conformation of a (partial) molecule in 3D space. Leaf nodes represent conformations of a complete molecule and thereby valid solutions, inner nodes correspond to partially built-up molecules. The actual conformation of a (partial) molecule represented by a node of the *conformation tree* can be determined by traversing the path from that node to the root node and collecting the specific torsion angles of the edges on this path.

Search strategy The actual sampling algorithm of TCG traverses this tree, guided by a force-field energy function [172], in a best first manner. In order to account for nonplaced parts of a compound, the force field is augmented with an additional heuristic term: Rigid components account with their internal energy, for ring components the energy of the lowest energy conformation is selected. Starting from the molecule's central component as root, one component after another is added using the previously assigned torsion angles and the pregenerated conformations for ring components. Similar to the strategy of the A^*-based sampling algorithm by Leach [173], TCG selects the currently best ranked solution for further expansion. The traversal continues un-

Quality level (q)	1	2	3	4	5
b_q	2	3	4	6	8

Table 6.2.: Base exponent of the quality levels b_q used in $f_{ESS}(k, q)$ to determine the amount of search space to be explored (see Equation 6.2).

til thresholds depending on the compound's flexibility and a user-defined quality level are reached. These thresholds include limits on the amount of explored search space (ESS) in the *conformation tree* and the minimum number of conformations (MNC) to be produced.

While expanding the *conformation tree*, TCG employs so-called *depth-probes* to generate fully built-up molecules from various starting points during the search. With a depth-probe frequency (DPF), also depending on molecular flexibility, the best-first-search is converted to a depth-first-search: The currently best scored partial solution is expanded in a depth-first-manner until either a clash occurs or a complete molecule is generated. Subsequently, TCG reverts to the best-first strategy. An example for the traversal of the conformation tree is given in Figure 6.9.

Search boundaries The idea behind the ESS constraint is based on the exponential growth of the total search space. Depending on the number of rotatable bonds k of the compound and a user-defined quality level q, it is calculated according to Equation 6.2. During the search, at most $f_{ESS}(k, q)$ nodes of the *conformation tree* are expanded. The base b_q of the exponential function is dependent on the desired quality level q: Larger quality levels are associated with larger values of b_q (see Table 6.2) and thus control the granularity of the sampling. In addition, a minimum ESS of 2^{k+5} is guaranteed as lower bound.

$$f_{ESS}(k, q) = min\{2^{q+14}, max\{b_q^k, 2^{k+5}\}\} \tag{6.2}$$

$$f_{MNC}(k, q) = min\{2^{k+2}, 2^{q+4}\} \tag{6.3}$$

The MNC limit, which also depends on quality level q and the compound's flexibility k, guarantees a minimum number of conformations to be generated before the ESS constraint is enforced. It is calculated according to Equation 6.3. The resulting boundaries for ESS and MNC are depicted in Figure 6.10.

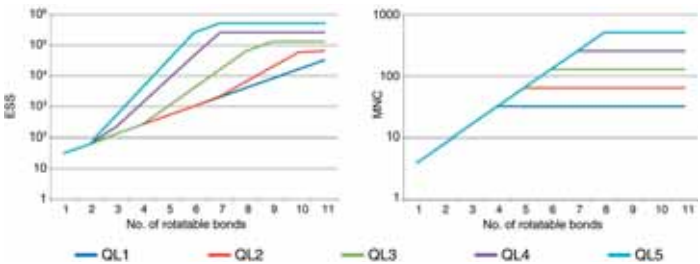


Figure 6.10.: Explored search space (ESS) as a function of a molecule's flexibility and quality level (QL) (left). Quality level one (QL1) is used as a lower bound. Minimum number of conformations (MNC) as a function of a molecule's flexibility and quality level before clustering (right).

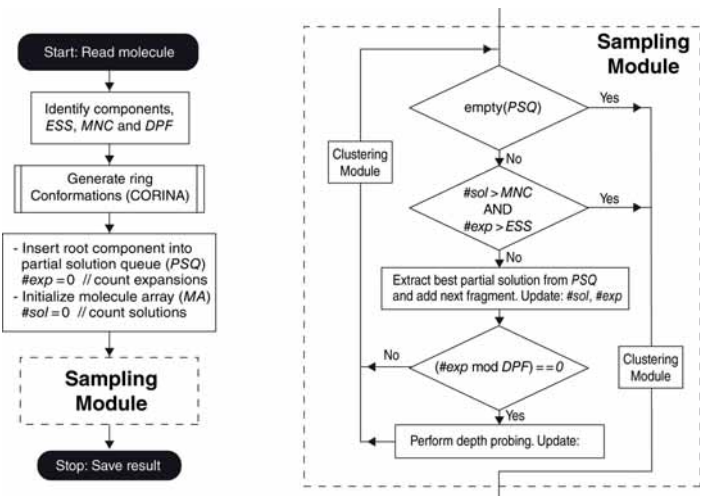


Figure 6.11.: Flowchart of the TCG conformational sampling algorithm.

In cases where the conformation tree is fully traversed and the ESS and MNC limits cannot be reached, TCG returns the conformations that have been found so far.

Clustering During the entire search process, an RMSD clustering routine is employed within a sliding window of fixed size. Besides a reduction of memory and runtime requirements, this intermediate online clustering procedure ensures a reasonably sized and diverse conformer set that is clustered as a whole once the algorithm concludes.

The overall sampling strategy of TCG including search boundaries, depth probing, and clustering module is illustrated as a flowchart in Figure 6.11.

6.4.2. Descriptor indexing

During the last phase of preprocessing, TrixX BMI descriptors are generated for each conformation in the library. TrixX BMI uses two layers of partitions — *descriptor-* and *type partitions* — to limit main memory requirements and to speed up descriptor matching. The corresponding partitioning scheme is illustrated in Figure 6.12.

Descriptor partitioning As part of descriptor matching, Bitmap Indices are loaded into memory, descriptor-based queries are executed, and the resulting matches are stored in a database. TrixX BMI partitions the descriptor data horizontally in order to reduce memory requirements of these calculations, and to generate pose predictions early during VS. The default number of descriptors that causes descriptor partitioning is 2 million. It corresponds to roughly 2000 leadlike molecules with an average of 100 compound descriptors and 10 conformations. Descriptor partitioning yields smaller

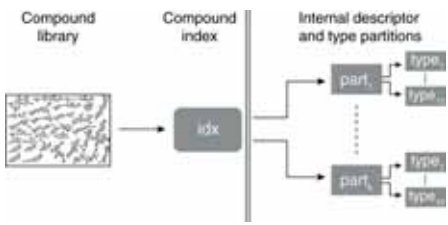


Figure 6.12.: TrixX BMI *internal partitioning* scheme using *descriptor-* and *type partitions*.

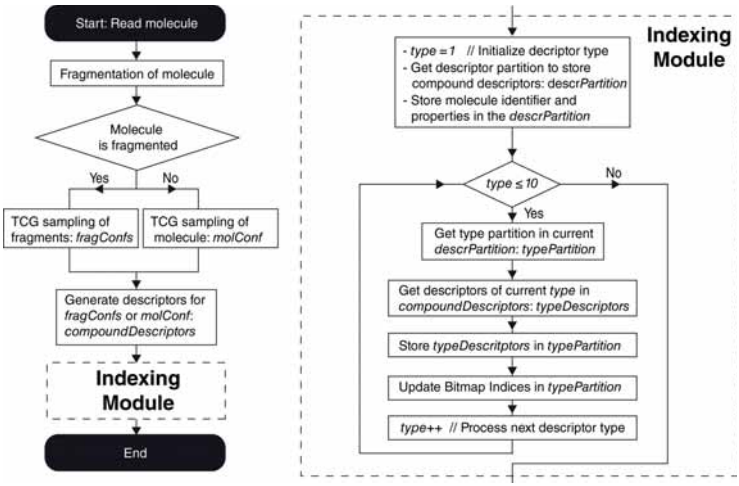


Figure 6.13.: Flowchart for the preprocessing phase including conformational sampling and compound indexing.

individual indices and thus reduced memory requirements. Furthermore, only a subset of the total number of descriptor matches needs to be handled within each partition. This reduces the requirements of organizing these matches.

Type partitioning The descriptor type attribute has only ten distinct values. It is not stored explicitly but is used to split the data again horizontally. Instead of creating one Bitmap Index for each of the 93 descriptor attributes, TrixX BMI partitions the data by descriptor type: Each attribute index is split into 10 smaller subindices, one for each type. Thus, a descriptor match no longer needs to explicitly check the equality of the type attribute. The query is directly posed to the corresponding subindex of the current attribute. This subindex stores only descriptors of matching type. I/O load and the number of CPU operations to logically combine the individual Bitmap Indices are reduced.

6.4.3. Synopsis

The flowchart in Figure 6.13 summarizes the preprocessing phase of TrixX BMI. First, the compound is loaded and, if necessary, fragmented. Subsequently, TCG is used to generate molecular ensembles of the compound or its corresponding fragments. Based on these ensembles, TrixX BMI generates compound descriptors, which are then passed to the indexing module. Then, a *descriptor partition* of the *compound index* is selected for appending the data. Molecule identifier and molecular properties of the compound are stored. For each descriptor type, the corresponding descriptors are selected and stored in the *type partition*. Subsequently, the WAH compressed Bitmap Indices are updated.

6.5. TrixX BMI Virtual Screening

The second phase of the TrixX BMI workflow is target dependent. It relies on the pregenerated *compound index* and the associated compound conformations of phase one.

- At the start of each experiment the target protein is analyzed. Then, the active site is selected and prepared manually. Favorable interaction spots are identified automatically using the models of FlexX and TrixX (see Section (5.1) and 5.2).
- *Site descriptors* are generated and translated into SQL queries to the database holding TrixX BMI *compound descriptors*. Matching *site* and *compound descriptors* are identified as described in Section 6.3.6 — individual compounds are not loaded within this phase. The resulting candidates are stored in a candidate database.
- The matches within the candidate database are handed over to the TrixX BMI docking engine. Only compounds that are identified by an m_{id} of a corresponding descriptor match are subject to actual docking calculations.

6.5.1. TrixX BMI docking engine

The docking phase itself also consists of different subphases, which first generate *descriptor poses*, then TrixX BMI poses, and (optionally) optimized poses.

The docking engine uses the set of all matches M_{d_s,d_c} of *site* and *compound descriptors* to generate *descriptor poses*. Each match $m_{d_s,d_c} = (id_{d_c}, m_{id}, f_{id}, c_{id}, id_{d_s})$ in the set M_{d_s,d_c} is extracted from the database as described in 6.3.7. Each molecule, each fragmentation, and conformation that is part of any descriptor match m_{d_s,d_c} is loaded and initialized only once. This is achieved via sorting and grouping the data using SQL, first by molecule, then fragmentation, and eventually fragment conformation.

Descriptor poses

Starting with a coarse grid-based scoring and clashing scheme, the initial descriptor matches are used to generate *descriptor poses*. Each descriptor match implies a triangle superposition using the associated SIACs (s_1, s_2, s_3) given by id_{d_s} and CIACs (c_1, c_2, c_3) identified by id_{d_c} of the compound m_{id}. In addition, a compound fragmentation and a fragment conformation are given via f_{id} and c_{id}. Thus, 3D coordinates for the involved molecules are available and a triangle superposition can be performed. The current compound is initialized using all conformations from pregenerated conformer files that are part of at least one match. Then, each match m_{d_s,d_c} of this compound is used to transform the conformation c_{id} into the active site according to the RMSD optimal superposition of the SIAC, CIAC coordinate pairs (s_1, c_1), (s_2, c_2), and (s_3, c_3). Thus, the results returned by the query engine already suffice to perform rigid body docking of leadlike compounds or to place anchor fragments in case of large, fragmented compounds.

As already mentioned, TrixX BMI models flexible donor CIACs as unspecific. During superpositioning, these CIACs are not located at the corresponding hydrogen as regular donors, but at the connected heavy atom. Therefore, the superpositioning routine is adapted: Instead of superposing donor CIAC and acceptor SIAC coordinates, the routine translates the acceptor SIAC by 1 Å along its associated main direction. The distance of 1 Å approximates the length of a bond between a heavy atom and its connected hydrogen(s) as for instance in terminal OH, NH_2, and SH groups. The subsequent superposition of the CIAC and the shifted SIAC position results in a reasonable placement, independent of the position of the actual hydrogen donor in the pregenerated conformation (see Figure 6.14).

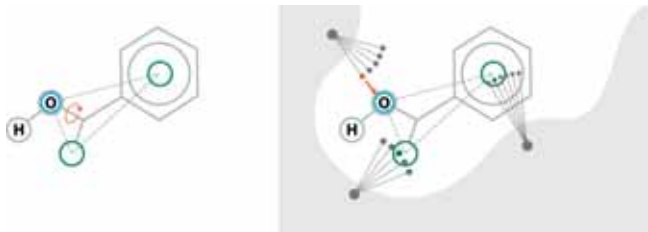


Figure 6.14.: Compound (left) with a hydroxyl group and the TrixX BMI superposition independent of the orientation of this group (right).

The resulting poses are then tested for clashes and scored on a grid representation of the active site. Finally, the best scored 200 pose predictions per compound are handed over to the next stage of the docking engine.

TrixX BMI poses

In this stage, all *descriptor poses* are evaluated more thoroughly resulting in so-called TrixX BMI pose predictions. The grid-based scoring of the previous phase is replaced with a more accurate empirical scoring function, as for instance the FlexX score or any other available scoring function (see Section 5.1). Furthermore, the clash calculations are refined by computing pairwise atom overlaps. So far, *descriptor poses* are rigidly placed into the binding site and the protein environment is not considered. Thus, interactions of *descriptor poses* are not optimally aligned with respect to protein interaction geometries. These flexible interactions are locally optimized, and all poses are rescored. If a compound is fragmented and thus a descriptor match only identifies anchor placements, the compounds remaining fragments are added using the FlexX incremental construction routine.

If no postoptimization of TrixX BMI poses is requested, which reflects the default setting, the resulting predictions and their scores are written to a solution database. The database scheme that is used to store these poses is shown in Figure 6.15. Here, the pose coordinates are not stored in the same table as the remaining pose attributes, i.e. the different energy contributions of the final score. The energy values are inserted using standard prepared statements and a manual commit strategy. The insertion of

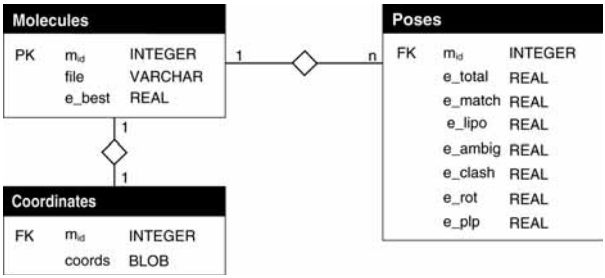


Figure 6.15.: Entity relationship diagram for pose predictions and their associated scores.

coordinates as binary large object (BLOB) data — a variable number of atom coordinates for each compound and pose needs to be stored — utilizes different internal storage mappings. This yields a performance decrease in the employed database system if the mapping procedure is called for each resulting pose. Thus, the storage of coordinates is realized as a bulk load for all pose predictions of a compound at once in a single BLOB. Within that BLOB, the coordinate sets are sorted by score, thus linking the ith best pose to the ith coordinate set in the compound's coordinate BLOB. Again, all primary and foreign key attributes of the different tables are indexed using B-Trees to speed up the generation of result lists or the visualization of pose predictions.

Optimized poses

An optional step of the TrixX BMI docking engine can be used to enhance the quality of protein-ligand complex predictions. Applications like binding mode analysis for lead optimization rely on highly accurate pose predictions that reflect the bioactive binding mode. TrixX BMI poses can therefore be used as input to a multi-objective optimization routine provided by FlexX. It performs a simplex-based optimization of the FlexX score to refine interaction geometries, a Lennard-Jones potential to reduce intra- and intermolecular clashes, and a term reflecting the torsional strain energy of the compound.

6.5.2. Pharmacophore type constraints

One of the goals for the development of TrixX BMI is the incorporation of pharmacophore information. In the following, the issue of how to handle this information and how to filter the resulting poses is addressed.

Based on the FlexX-Pharm [174] interface, pharmacophore type constraints can be formulated. Specific interactions, as well as spatial constraints using inclusion and exclusion volumes can be required. In addition, spatial constraints can be associated to certain molecular groups by using the SMARTS language [24] that supplies a notation for describing molecular patterns and properties. Furthermore, boolean combinations to combine alternative constraints into one pharmacophore description can be specified.

Since TrixX BMI is integrated into the FlexX library and reuses some of its internal data structures, it is possible to utilize the aforementioned pharmacophore module to specify pharmacophore type constraints and to filter pose predictions. Thus, regular pharmacophore type constraints are supported in TrixX BMI.

Apart from these filters, TrixX BMI allows the application of pharmacophores also during an earlier part of its docking pipeline. *Site descriptors* are usually generated for each triplet of SIACs that does not violate certain requirements. Pharmacophore type constraints establish a reasonable possibility to select descriptors based on their potential to fulfill these requirements. A pharmacophore P consisting of a set of directional constraints p_d and spatial constraints p_s can be used to deselect *site descriptors* that are not likely to contribute to a pose fulfilling the constraints in P. Therefore, each directed interaction in p_d is mapped to its corresponding SIACs resulting in a set P_d of SIACs. Furthermore, the inclusion volumes in p_s are used to identify all protein SIACs that are covered by these constraints resulting in a set P_{s+}. The calculations can be performed accordingly for exclusion volumes which yields a set P_{s-}. Based on these sets, the following condition is used as filter for each *site descriptor* d_s and the SIACs $\{s_1, s_2, s_3\}$ it is based on:

$$|\{s_1, s_2, s_3\} \cap (P_d \cup P_{s+})| \geq 2 \wedge |\{s_1, s_2, s_3\} \cap P_{s-}| = 0$$

A *site descriptor* is valid only if this condition holds, otherwise it is discarded. Thus, a *site descriptor* has to cover at least two SIACs associated to pharmacophore type constraints and must not cover any SIACs within any of the exclusion volumes. In

Molecular property	Filter condition
Molecular weight	≤ 450
Number of H-bond acceptors	≤ 8
Number of H-bond donors	≤ 5
Number of rings	≤ 4
Number of rotatable bonds	≤ 10
logP	$[-3.5, 4.5]$

Table 6.3.: Default values used for the TrixX BMI property filter.

cases where only one constraint $|\{P_d \cup P_{s+}\}| = 1$ is given, the first part of the condition must be relaxed, such that one covered interaction suffices. TrixX BMI also reverts to this relaxed filter criterion in cases where no or only few descriptors (≤ 50) remain after standard filtering.

6.5.3. Molecular property filters

Often, a compound library and thus the resulting index is designed with respect to certain molecular properties of known binders. However, during a VS campaign new knowledge might become available. Thus, the hypothesis used during the initial setup of the library is not static and can change over time. Regular, sequential docking approaches target this issue by filtering the library according to the desired properties. Only molecules that fulfill these properties are selected for the next sequential VS run. A straightforward approach to solve this problem is to delete compounds that do not fulfill the criteria. Depending on the number of affected compounds, this strategy can be computationally expensive since TrixX BMI uses a precalculated index of molecular descriptors as basis for VS. Rather than updating the compound database in order to meet the current filter properties, TrixX BMI follows an integrated approach. In addition to the *descriptor table D* of *compound descriptors*, a second table holding molecular properties is generated. Each library compound and its properties are inserted into this *molecule table M* using a primary key m_{id} that is employed as foreign key attribute m_{id} in the *descriptor table*. In a relational database system, these tables are naturally joined and can then be queried with the additional filter conditions. This approach is not feasible using the FastBit system since it does not efficiently support joins on large data sets. Instead, TrixX BMI employs a filtering approach using an

additional filter bitmap F to identify descriptors that are derived from compounds within the desired property ranges.

This filtering step works as follows: TrixX BMI inserts all primary keys m_{id} of all molecules in M matching the desired criteria into a set M_{id}. The keys in M_{id} are then used to generate a set D_{id} of bitmap identifiers. This set can later be used to construct the aforementioned filter bitmap F. For a given *compound descriptor* d_c the function $id(d_c)$ uniquely identifies its position in a bitmap. D_{id} is defined as:

$$D_{id} = \{id(d_c) \mid m_{id}[d_c] \in M_{id}\}$$

In the above set definition, $m_{id}[d_c]$ denotes the access to the molecule identifier m_{id} of a descriptor d_c. Thus, the set D_{id} identifies the bitmap positions of all descriptors that are derived from a molecule in the set M_{id}. Eventually, the filter bitmap F is constructed: It has a set bit at position j for each $j \in D_{id}$.

During querying, each regular TrixX BMI query is appended with F using a boolean AND operation. Thus, the final set of descriptor matches comprises only *compound descriptors* that match the current *site descriptor* and are also derived from compounds that fulfill the desired properties. The default values used for property filtering, which reflect the leadlikeness criteria of Oprea, are given in Table 6.3.

6.5.4. Synopsis

The flowchart in Figure 6.16 summarizes the virtual screening phase of TrixX BMI. First, the protein active site is loaded and SIACs are generated. Optionally, a pharmacophore type constraint can be read. Then, TrixX BMI site descriptors are generated. During the actual screening phase, each *descriptor partition* is handled individually. TrixX BMI switches between descriptor matching and docking calculations. First, the descriptor matching module generates matches between *site-* and *compound descriptors* of the current *descriptor partition*. Then, the resulting matches are handed over to the docking engine. This process continues until there are no further partitions.

During descriptor matching, *type partitions* and the optional molecular property filter are handled. If filter criteria are supplied, the identifier of each molecule that fulfills the criteria is added to a set of *filterIds*. The actual property filter is generated for each *type partition* individually. It is implemented as a bitmap (*bitMapFilter*) that is used as additional query dimension. The *bitmapFilter* identifies all descriptors in the current

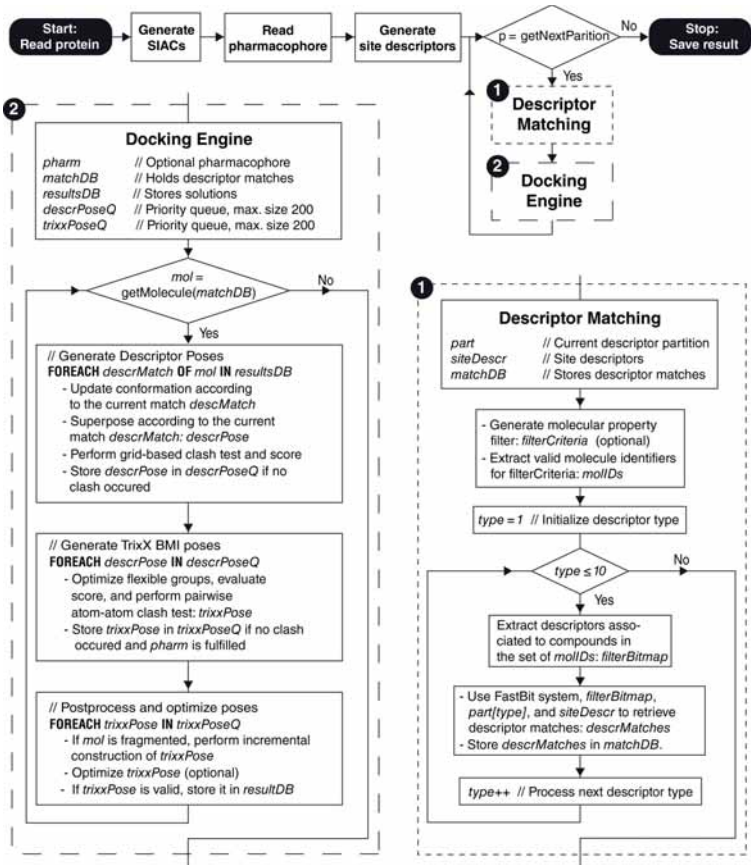


Figure 6.16.: Flowchart of the TrixX BMI virtual screening phase.

partition that are associated to the molecules in the set *filterIds*. Subsequently, FastBit is used to identify actual descriptor matches. The resulting matches are stored in a database. Before the descriptor matching of the current *descriptor partition* concludes, the FastBit cache is freed.

The docking phase of TrixX BMI consists of three different stages. For each

molecule in the result database, descriptor matches are extracted. Each match describes a fragmentation and conformation that is used to update the current molecule. Then, descriptors are superposed. The resulting descriptor pose is checked for clashes and scored on a grid. During the next stage, TrixX BMI poses are generated. It involves reorientation of the compound's flexible groups, the calculation of interaction scores based on the FlexX interaction model, and a refined clash test on pairwise atom-atom overlap. If pharmacophore information is available, it is used for filtering.

If a molecule is fragmented, incremental construction is performed as part of a post-processing stage. Optionally, TrixX BMI poses can be optimized. Eventually, the resulting poses are written to the result table.

6.6. Parallelization

TrixX BMI always searches the complete database of compound descriptors. Complex operations like joins on multiple tables and attributes are not performed. Furthermore, the read-mostly data of the compound database is updated infrequently: Updates can be executed as a batch load. This suggests to parallelize TrixX BMI using a shared-nothing architecture. It is defined as a system that does not share main memory or peripheral storage among processors [175].

Figure 6.17 illustrates the partitioning schema employed by Trixx BMI. The schema consists of four different levels:

- The original compound library consists of all compounds that are supposed to be screened in a TrixX BMI VS experiment. Typically, a few thousand up to several million compounds and their properties are stored in a regular database system.
- The compute grid consists of n different nodes, each with distinct peripheral storage and main memory.
- Each node consists of m cores. On current platforms, this number is in the range of 1–24 and is likely to increase in the next years.
- Each of these cores is assigned with a unique *compound index* that uses the internal partition scheme of TrixX BMI (see 6.4.2).

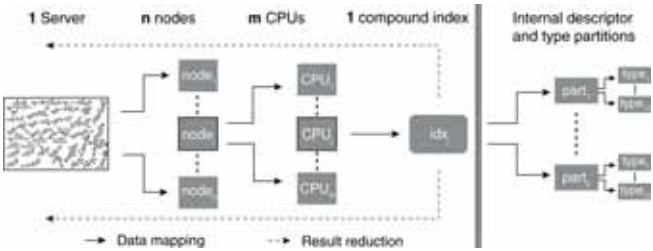


Figure 6.17.: TrixX BMI approach to parallelization using *compound partitions*.

Preprocessing In a first step, TrixX BMI splits the compound library into $n \cdot m$ *compound partitions*. This process is currently performed in a random manner. The resulting *compound partitions* are assigned to individual nodes and their cores using the Sun Grid Engine (SGE) [176]. Each core executes the standard preprocessing phase of TrixX BMI (*conformational sampling* and *compound indexing*). In total, $2 \cdot n \cdot m$ messages suffice to initiate the preprocessing phase and to collect the resulting log files. Eventually, the resulting *compound index* is synchronized to other nodes in order to increase its availability in the system. In the default setup, TrixX BMI distributes each *compound index* to three different nodes.

Virtual screening The actual VS is executed analogously. The SGE is used to submit $n \cdot m$ screening runs: For each *compound partition*, a job request is generated that explicitly maps the task to the associated nodes of the partition. If all nodes associated to a *compound partition* are occupied, an additional node is selected and the data is synchronized. Each VS experiment runs independently. Only the mapping of the individual hitlists of the *compound partitions* into a global hitlist of the entire VS experiment needs to be coordinated. Again, $2 \cdot n \cdot m$ messages suffice to initiate the experiment and to collect the results.

7
Results and Discussion

TrixX BMI is a multiphase process and thus the evaluation is also split into multiple parts. The first section discusses the experimental methodology used for this evaluation. Then, results of conformational sampling using the TCG[1] during the preprocessing phase are presented. Subsequently, the redocking and virtual screening performance of TrixX BMI is presented. The chapter concludes with results concerning runtime and space requirements of TrixX BMI and its performance in a parallel computing environment.

7.1. Experimental Methodology

There are two main applications of docking programs as part of the drug discovery process: Binding mode prediction and VS experiments. For each of these, a different measure is used to evaluate the performance of a docking algorithm.

7.1.1. Binding mode prediction

The ability of a particular program to predict the correct binding mode of a protein-ligand complex is usually assessed based on the RMSD of the predicted ligand placement to the structure of the cocrystallized ligand. A widely used standard in the field of structure-based VS is a 2 Å cut-off for correctly docked poses, whereas poses between

[1] The evaluation of the TCG results was carried out in collaboration with Axel Griewel. The author of this thesis focused on the results for quality level one that yields compact ensembles with good accuracy. Axel Griewel performed the analyses concerning higher quality levels and the trade-off between ensemble size and accuracy. He also generated the results on the CSD data and performed the detailed case studies.

2 Å and 3 Å are considered as partially docked. Predictions beyond 3 Å are considered a docking failure.[177]

Due to an increasing number of crystallized complexes, large test sets are available [178, 179] to evaluate a novel docking algorithm and compare it to state-of-the-art approaches.

7.1.2. Virtual Screening

During the course of screening, a docking algorithm has to identify a small number of active compounds in a database of mostly inactive compounds (decoys). The metric used commonly to evaluate the success in virtual screening is the so-called enrichment factor (EF). It is defined as:

$$EF = \frac{a}{A} * \frac{N}{n} \tag{7.1}$$

Here, a is the number of active compounds among the n best ranked compounds of the database, and A is the total number of active compounds in the whole database of N compounds. The EF thus compares the ratio of actives found to the ratio of the database that is considered relevant for more detailed postprocessing routines or wet lab experiments. The problem of the EF is that it becomes smaller if fewer decoy structures are present in the data set. Enrichment does not only depend on the algorithm being analyzed but also on the experimental data that is used: A certain EF cannot be associated with a general quality of the docking algorithm, it only reflects the quality concerning a specific experiment. Therefore, a comparison based on EF as evaluation metric should be performed using a publicly available data set which provides a well balanced ratio of actives to decoys.

The limited robustness of EF can be addressed by slightly adapting the above formula. Instead of the fraction of all compounds, so-called ROC enrichment employs the fraction of decoys, the so called false positive rate.

$$ROC_{EF} = \frac{a}{A} * \frac{N - A}{n - a} \tag{7.2}$$

This adaptation makes enrichment more robust and independent of extensive quantities for actives and decoys. [180]

7.2. TrixX Conformer Generator

This chapter presents the evaluation results for the TCG. The results presented here, focus on leadlike structures as they are employed in the TrixX BMI screening pipeline.

7.2.1. Sampling data

TCG is evaluated by calculating the accuracy in terms of RMSD to a biologically active structure. Success and failure of the conformer sampling are discriminated by this accuracy: Cases in which the accuracy of the ensemble is larger than 2 Å are generally considered a failure. For molecules with an RMSD between 1.5 Å and 2.0 Å, the overall structure of the conformer is usually close to the bioactive conformation while structural details may differ significantly. An RMSD below 1.5 Å indicates an acceptable reproduction of the conformer, while an RMSD below 1.0 Å is considered a good fit between generated and biologically active conformer. During all experiments, input structures for TCG sampling are generated by Corina. This provides an unbiased starting point since the sampling algorithm is not influenced by the conformation of the crystal structure.

First, TCG's capabilities to reproduce conformations found in the CSD [181] are analyzed. For this purpose, a previously published subset [182] of the CSD, consisting of approximately 71,200 high quality structures, is utilized. These molecules contain only H, C, N, O, S, and halogen atoms. Since TrixX BMI is based on conformational ensembles of leadlike molecules and leadlike fragments of larger molecules, the original test set is filtered using the leadlike criteria of Oprea [22]. The remaining set consists of 43,047 structures.

In a second experiment, TCG is compared to two widely used tools: Omega 2.0 and Catalyst 4.11. A publicly available test set consisting of 778 druglike molecules bound to their receptors from the PDB is used as benchmark [183]. This set is filtered retaining only molecules with ≤ 11 rotatable bonds. This cut-off exceeds the default TrixX BMI setting for fragmentation and is selected since it corresponds to the data in the original publication. The corresponding distribution of molecular flexibility within this test set of size 644 separated by the number of rotatable bonds is shown in Figure 7.1.

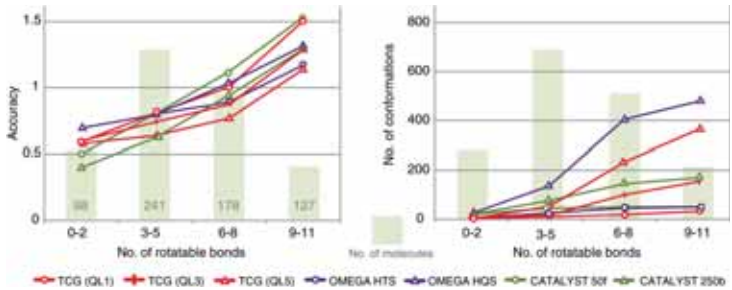


Figure 7.1.: Comparative results of conformational sampling using TCG in differen quality levels (QL).

7.2.2. Conformational sampling

In order to show TCG's ability to reproduce conformations found in the CSD within small error boundaries, the default settings as used in the TrixX BMI pipeline are employed: The quality level is set to one and the clustering threshold to 1.2 Å. The resulting conformational ensembles have an average RMSD of less than 1.0 Å. This demonstrates that TrixX BMI settings of the TCG produce high-quality conformations which can subsequently be used for molecular docking.

The comparative results to Omega and Catalyst are based on TCG quality levels one, three, and five. All TCG settings employ a clustering threshold of 1.2 Å. In this experiment, not only the accuracy in terms of RMSD but also the number of conformers in the generated ensembles is analyzed. Both properties are separated by the number of rotatable bonds. The results are given in Figure 7.1. As expected, the average number of conformers as well as RMSD rise with the molecule's flexibility. For molecules with up to eight rotatable bonds, high-quality ensembles with an average accuracy below 1.0 Å are generated for all three quality levels. In the high-throughput setting of TCG (quality level one) an average of 15 conformations suffices to achieve an average accuracy of 0.98 Å. For larger compounds with 9–11 rotatable bonds the average accuracy is between 1.2 Å and 1.5 Å and thus reflects the overall structure of the corresponding molecule.

The comparison to Omega and Catalyst shows that TCG produces ensembles with similar accuracy. All tools are run using a high quality and a high-throughput

setting. In case of TCG, different objectives concerning accuracy versus the size of the generated ensembles are followed by employing different quality settings. For each of the presented tools, numerous more settings are available to better adapt to the user's objective. However, the experiments show that TCG performs well with respect to the trade-off between the number of conformers per ensemble and resulting accuracy: Already few TCG conformations suffice to generate accurate ensembles. This property is essential for downstream virtual screening experiments such as the TrixX BMI screening pipeline.

Case studies The quality of the generated conformers is studied in detail using two ligands that occur in multiple PDB structures. The corresponding crystal structures are selected as case studies. To ensure high structural quality, all complexes with a resolution above 2 Å are discarded. Furthermore, a 0.5 Å RMSD filter is employed to focus on conformationally different structures. Again, TCG experiments are performed for quality settings one, two, and three. A clustering threshold of 0.8 Å is utilized.

The first ligand is 4-hydroxy-tamoxifen (OHT), which has eight rotatable bonds. It is present in nine PDB structures that are filtered down to three using the previously mentioned criteria. The corresponding PDB identifiers, the intrinsic RMSDs, and the results of TCG sampling are presented in Table 7.1. For all examples in the test set, a conformation with accuracy below 0.8 Å was found, making the ensemble a good approximation of the experimentally determined conformers. In Figure 7.2 (a) a superposition of the crystal structure from PDB entry 2gpu and the best conformation from quality level one and five is depicted. The overall conformations align well already in quality level one. Only the alignment of the aliphatic part of the molecule is less precise. The alignment improves largely if higher quality levels are used: The RMSD improves to 0.47 Å.

In a second case the binding modes of indomethacin (IMN) are investigated. This molecule comprises four rotatable bonds and is contained in 11 PDB structures, respectively four after diversity and quality filtering. Two of these are reproduced in high quality below 0.5 Å RMSD in quality level one. The remaining structures are reproduced reasonably with an RMSD of 1.2 Å respectively 1.5 Å. The resulting RMSDs improve if higher quality levels are employed. Quality level three and five both yield further improvements, such that finally all crystal structures can be reproduced at high

OHT	2bj4	2gpu	3ert	**IMN**	1s2a	2dm6	2zb8	3fo7
2bj4	0	1.17	1.02	1s2a	0	2.28	0.90	1.12
2gpu		0	1.33	2dm6		0	2.23	2.35
3ert			0	2zb8			0	1.13
				3fo7				0
TCG QL1	0.76	0.61	0.75		0.46	1.51	0.89	1.21
TCG QL3	0.38	0.47	0.46		0.46	1.14	0.89	1.06
TCG QL5	0.38	0.47	0.55		0.46	0.52	0.69	0.65

Table 7.1.: RMSD between conformers in the case study test set and the RMSD between these structures and TCG ensembles generated in quality level one, three, and five.

quality with an RMSD below 1 Å. In Figure 7.2 (b) the alignment of the best conformer from level one and five with the crystal structure from 2zb8 is shown. The overall alignment of the level one solution is already acceptable. However, the high-quality level (green), yields a better alignment of structural details for the methoxy- and carboxylate group of IMN.

Runtime The runtime of the TCG is assessed on single 2.4 GHz Xeon CPUs with 4 GB of main memory and is presented for the entire test set consisting of 778 druglike molecules. It ranges from 5.2 *s* in the lowest quality setting to 201.6 *s* in the highest quality setting. The increase in computing time is due to the enlarged search space. The performance of Catalyst and Omega on the test set has been assessed on Intel Pentium IV 2.8 GHz workstations with 1 GB RAM [183]. The average run times of Omega ranged from 6.0 s to 12.9 s, while those of Catalyst were reported to be in the range of 1.5 s to 155.0 s. Taking similar setups into account, the runtime for TCG is in the same range as that of Omega and Catalyst.

7.3. Redocking Experiments

7.3.1. Redocking data

First, the overall redocking performance using the Astex Diverse Set [179] is measured in terms of RMSD to the cocrystallized ligand in order to compare TrixX BMI, FlexX,

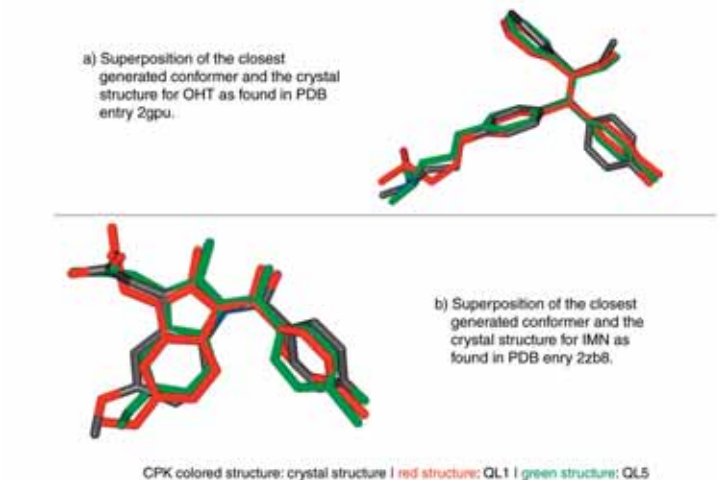


Figure 7.2.: TCG case studies for a tamoxifen derivate (OHT) and indomethacin indomethacin (IMN).

and Gold. This test set consists of 85 high resolution protein-ligand crystal structures that have been retrieved from the PDB. The individual protein targets are manually prepared by including essential metals, altering protonation states, and adapting torsional angles in order to capture the binding mode of the cocrystallized ligand. The set of ligands is energy minimized using Corina to create an unbiased conformation to start the search.

7.3.2. Redocking performance

The redocking studies performed use different sets of ligand conformations to assess the performance of TrixX BMI. In a first experiment, the bioactive conformation, as found in the protein-ligand crystal structure, is redocked. This scenario reflects the performance of the docking algorithm independent of the quality of the conformational search. Subsequent experiments use ligand conformations produced by the TCG as input. Three different settings of the TCG using an RMSD clustering threshold of 1.2 Å and different quality levels are employed. The results for different RMSD cut-

		RMSD[Å] ≤		
Best(x)	quality	1.0	2.0	3.0
1	crystal	49	63	70
1	1	15	39	51
1	3	19	41	54
1	5	14	46	50
20	crystal	58	74	78
20	1	25	61	71
20	3	26	68	72
20	5	28	65	74
200	crystal	58	75	78
200	1	27	68	78
200	3	28	73	79
200	5	29	72	78

Table 7.2.: Redocking results of TrixX BMI on the Astex Diverse Set when looking at the x best scored pose predictions using TCG ensembles of different quality q.

offs, which correspond to accurate predictions (≤ 1.0 Å), correct predictions (≤ 2.0 Å), and partially correct predictions (≤ 3.0 Å), are shown in Table 7.2.

The usage of cocrystallized 3D structures clearly yields the best results when it comes to overall prediction quality considering the top 200 ranks: 75 of 85 can be reproduced correctly. Additional three complexes are partially redocked. Thus, TrixX BMI poses of a known active conformation can be found for about 90 % of the complexes in the Astex Diverse Set if the effects of scoring and conformational flexibility are neglected; The best ranked 200 poses of the redocked crystal structure are considered.

Since the eventually bioactive conformation of a ligand is not known at the start of a VS campaign, this does not reflect a real world scenario. Further experiments using different quality and cluster settings of the TCG demonstrate, that the overall redocking rate of TrixX BMI based on conformational ensembles is at the same level of about 90 % of the data set. This obverservation only holds for (partially) correct predictions. Accurate placements below 1 Å cannot be reproduced on the same level. Based on the crystal structures, 58 complexes are reproduced accurately, compared to 27–29 if TCG ensembles are employed.

In the following, the effect of rank ordering is also accounted for. If only the best rank is considered, the usage of a bioactive conformation as input to TrixX BMI clearly outperforms the experiments based on TCG conformations. In experiments redocking the crystal structure, 49 complexes can be accurately docked and for 70 complexes the best ranked pose is (partially) correct. The ensemble-based experiments are able to place at most 19 ligands accurately and at most 54 ligands (partially) correct.

In a realistic redocking scenario, an application scientist checks the resulting predictions by visual inspection. Therefore, not only the top ranked pose is used to gather knowledge about the protein of interest. In the above experiments the analysis of the top 20 poses does not significantly increase the performance considering accurate pose predictions based on TCG performance: Accurate placements are found for 25–28 compared to 58 if the bioactive conformation is used. However, the performance concerning (partially) redocked poses significantly increases and 71–74 poses with an RMSD below 3.0 Å can be found. Only four of the 78 partially correct pose predictions of cocrystallized ligands are not found.

In order to compare TrixX BMI to FlexX and Gold, the following data is selected in order to reflect publicly available results: The 20 best scored poses and only correct placements with an RMSD of 2.0 Å or less are considered. The results (see Table 7.3) show that for accurate placements below 1.0 Å FlexX outperforms TrixX BMI. Concerning predictions within 2.0 Å RMSD, TrixX BMI is on a par with FlexX and misses only two poses correctly predicted by Gold. At least partially correct pose predictions below 3.0 Å are found for 71 ligands compared to 65 found by FlexX. As further comparison the results for all 200 poses predicted by TrixX BMI are given. These results demonstrate that ranking accounts for an 3 % to 8 % loss of precision depending on the accuracy of the prediction.

All redocking experiments are performed using the standard FlexX score. Different scoring functions lead to different predictions, although the overall picture stays the same. The ScreenScore is able to rank further two poses below 2.0 Å in the top 20 %. In general though, the available scoring functions are on the same level and only minor differences in ranking can be observed.

Tool	Poses	RMSD[Å] ≤ 1.0	1.5	2.0	3.0
GOLD	avg(20)	n.a.	n.a.	64[75]	n.a.
FLEXX	top(20)	38[45]	53[62]	61[72]	65[76]
TRIXX BMI	top(20)	25[29]	47[55]	61[72]	71[84]
TRIXX BMI	top(200)	27[32]	53[62]	68[80]	78[92]

Table 7.3.: Number ([%]) of poses found (within the best n ranks) out of 85 protein-ligand complexes using TCG conformational ensembles based on CORINA structures (TRIXX BMI) respectively minimized CORINA structures (GOLD, FLEXX) as input.

Best(x)	q	RMSD[Å] ≤ ≤ 1.0 Å	≤ 2.0 Å	≤ 3.0 Å
20	crystal	55(−3)	75(+1)	78(±0)
20	1	24(−1)	63(+2)	70(−1)
20	3	23(−3)	64(−4)	73(+1)
20	5	26(−2)	68(+3)	74(±0)

Table 7.4.: Redocking results of TRIXX BMI on the Astex Diverse Set using the internal optimization. The number in parenthesis indicates the change in overall accuracy of optimized poses compared to the corresponding nonoptimized poses.

7.3.3. Optimization

As part of redocking experiments, a user can require highly accurate predictions for a detailed analysis of the binding mode. Therefore, an optional step within the TRIXX BMI docking pipeline is the optimization of TRIXX BMI poses. In the following, experiments using this integrated optimization option are introduced. Furthermore, experimental results of external optimizations based on YASARA [184] and energy minimization with the Amber [185] force field are presented. All poses that are input to optimization are based on TRIXX BMI pose predictions of TCG ensembles sampled in quality level 1 with a clustering threshold of 1.2. Table 7.4 shows the results of the internal optimization. Here, a clear trend cannot be observed. None of the experiments shows a significant improvement after optimization. However, the resulting poses are subject to a more restrictive clash test and minor clashes of TRIXX BMI poses are removed. The integration of the internal optimizer is not sufficiently tested.

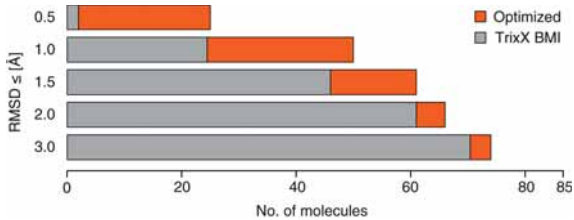


Figure 7.3.: Comparison of TrixX BMI poses and the corresponding optimizations using Yasara and the Amber force field.

A parametrization focused on the removal of the aforementioned clashes and a sophisticated force field is currently not available.

The optimization results of TrixX BMI poses using Yasara and Amber is visualized in Figure 7.3. After optimization the number of accurate predictions increases from 25 to 50 and from 61 to 66 for correctly docked poses. Highly accurate predictions are generated that are suited for detailed binding mode analysis. This demonstrates the potential of TrixX BMI poses for redocking. Further experiments suggest, that similar results can be achieved for crossdocking a ligand into a different protein conformation.

7.4. Enrichment Experiments

7.4.1. Enrichment data

A common problem when setting up enrichment studies is how to choose a library of inactive compounds to avoid an artificial enrichment of known actives. Therefore, the DUD data set that supplies tailored sets for 40 different target proteins is used. Each target has its own set of actives and decoys which are chosen with respect to similar molecular properties but differing chemical structure. Thus, the DUD poses a challenging test set. Furthermore, this data set is publicly available [186] and results for widely used docking tools (Dock, FlexX, Glide, ICM, PhDock, and Surflex) have been published by Cross et al. [187].

The active site is defined as all atoms within a radius of 6.5 Å around any atom of the cocrystallized ligand. The protein targets are subjected to visual inspection, and

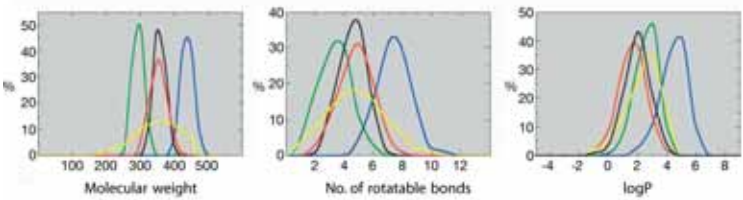


Figure 7.4.: Property distributions of the used ligand sets (actives+decoys). Left: molecular weight, center: number of rotatable bonds, right: logP. (CDK2: black, DHFR: red, ER agonist: green, ER antagonist: blue, random set Z_2 : yellow).

protonation states for histidine and conformations of the amino acids asparagine and glutamine are adjusted. Apart from these, all heavy atoms remain fixed in their x-ray positions and no energy minimization is performed. The actives and decoys are used as provided by the DUD. No hand-crafted or computational adaptations are made. Molecular ensembles for each molecule are built with TCG defaults for quality level (one) and RMSD clustering threshold (1.2 Å) based on CORINA generated structures.

For further analysis, four pharmaceutically relevant protein targets — Cyclin Dependent Kinase 2 (PDB entry 1ckp), Dihydrofolate Reductase (3dfr), Estrogen Receptor (ER) Agonist (1l2i), and ER Antagonist (3ert) — are chosen. These targets represent different classes of interest in pharmaceutical research. In addition, pharmacophores for these targets are available from the literature (see Table 7.6) and the corresponding actives and decoys represent a diverse set of compounds with regards to molecular properties. Figure 7.4 shows property distributions of these sets compared to a random set of 2000 leadlike molecules (Z_2) from the ZINC database [188] of commercially available compounds.

7.4.2. External data analysis

A problem that is common to most publications in the field of virtual high throughput screening is the availability of detailed experimental data. This is also true for the experiments of Cross et al. which are used to evaluate TRIXX BMI's enrichment capabilities. The individual target proteins and their corresponding actives and decoys

sets are publicly available as part of the DUD. However, the resulting hitlists and thus the raw data of the experiments is not. Results are only presented as summary data. Individual enrichment plots are not scaled logarithmically and the important first percent of corresponding enrichment experiments can hardly be analyzed by visual inspection.

Unfortunately, there seems to be a flaw in the evaluation of Cross' enrichment data. In cases where a compound cannot be docked into the active site, this is considered a *docking failure*. However, the statistical analysis of the predictions must be based on the entire data set of positives (P) and negatives (N), not on the number of actually docked compounds. ROC enrichment relates the true positive (TP) rate with the false positive (FP) rate, the former is defined as TP/P, the latter as FP/N. Thus, a TP rate of 1.0 can only be reached, if all positives (actives) in the data set are correctly predicted. The same holds true for the FP rate and the prediction of negatives (decoys). In VS experiments this is often not the case, some of the actives/decoys cannot be docked into the active site of interest. Cross et al. mention that they treat this problem by appending the nonpredicted compounds by distributing actives and decoys evenly at the bottom of the hitlist. Thus, each tool that cannot successfully dock all compounds must have a corresponding quadrant in its plot that depicts this selection. Such a distribution results in a diagonal across a rectangular area on the upper right part of the ROC plot. This area corresponds to the fraction of actives (y-axis) and decoys (x-axis) that cannot be docked. Cross' analysis does not present plots that depict this kind of distribution even though, docking tools like ICM and Glide produce numerous docking failures. Therefore, it seems as though only docked compounds are used for statistical evaluation. For instance the *purine nucleoside phosphorylase* (PNP), where only 15.8 % of all compounds are successfully docked with Glide, or the *HIV protease* (HIVPR), where only 2.2 % can be docked with ICM, should be associated with a ROC plot that is in large parts evenly distributed.

Individual plots offer further evidence for errors in the statistical evaluation: Glide is only able to dock 4 of 15 actives of the mineralocorticoid receptor (MR), in case of PNP only 7 of 25 actives can be docked. In both plots, the number of docked actives corresponds to the number of increments along the y-axis until the TP rate reaches 1.0. The same holds true for ICM and targets like the *peroxisome proliferator activated receptor* (PPAR) with 6 of 81 actives and the *S-adenosyl-homocysteine hydrolase*

(SAHH) with 3 of 33 actives. Since visual extraction of ROC enrichments at 5 and 10 % FP rate corresponds to the available summary data in the publication, it is likely that all results are based on the same data foundation. Unfortunately, the data is not supplied in a fine enough grain to rescale the plots and generate the associated ROC enrichments.

Surflex, which is able to predict poses for 100 % of all compounds, is not affected by this. For two of the six tools, the error should be small. Dock and FlexX are able to successfully dock more than 95 % of all compounds. The worst percentage of docked compounds for a single target is 88 % for Dock, 86 % for FlexX, and the ratio between nondocked actives and decoys is often close to one. Therefore, the averages based on rescaled ROC plots are likely to yield similar enrichments for these tools.

The same does not hold for Glide, ICM, and PhDock. In case of the nuclear hormone receptor family Glide is able to dock only 56 % of the actives and 43 % of the decoys. For three of the DUD targets (*glucocorticoid receptor*, MR, PNP) the success rate drops to less than 30% of the actives. For ICM, the serine proteases are problematic with only 21 % of actives and 20 % of decoys that are successfully docked. In total, more than six targets yield docking success for less than 30 % (PPAR, *factor Xa* (FXA), thrombin, trypsin, HIVPR, SAHH) of the actives, three of these even less than 10 %. PhDock is also not able to successfully dock into serine proteases. Only 63 % actives and 61 % decoys can be placed into the corresponding binding sites. FXA with 35 % and the *P38 mitogen activated protein* with 36 % are most problematic.

The authors of the publication have yet not answered a request to supply detailed data. Therefore, the published results are the only available source of information. Since most other comparative studies are based on nonpublic data, the software of other vendors is not publicly available, and the usage of DUD is widely spread in the field of structure-based VS, evaluating the enrichment capabilities of TrixX BMI on the DUD data set is nevertheless the best current practice.

7.4.3. Enrichment performance

Similar to the presentation of data in the study of Cross et al., the enrichment performance of TrixX BMI is analyzed using multiple criteria.

- The overall docking success is evaluated by checking the percentages of successfully docked actives and decoys. If a compound could not be place into the binding site, it is considered a docking failure and is either a true negative prediction in case of a decoy structure or a false positive one if an active cannot be docked.
- ROC enrichments are presented for each protein family in the DUD data set to evaluate the enrichment capability of TrixX BMI. Enrichment data for numerous other tools is supplied in the publication of Cross et al. and is used for comparison with TrixX BMI.
- Enrichment plots are presented to assess the performance on individual targets. For these, also the performance of TrixX BMI in combination with pharmacophore constraints is presented. Compounds that failed to dock are not placed randomly at the bottom of the hitlist. In such cases, the depicted enrichment does not reach the 100 % plateau for actives or decoys.

The overall docking success of TrixX BMI is close to 100 %. Averaged over all targets, 99.8 % of the active compounds and 99.1 % of the decoys are successfully docked. About 33 % of the failing compounds cannot be conformationally sampled using TCG. The remaining ones cannot be placed during the course of TrixX BMI docking.

The enrichment capabilities of TrixX BMI are summarized in Figure 7.5. It shows the mean ROC enrichments at 0.5 %, 1.0 %, 2.0 %, and 5 % false positive rate for the different protein families (nuclear hormone receptor (NHR), kinases, serine proteases, metallo-, and folate enzymes) in the DUD and the overall performance on the entire DUD, which includes 14 additional proteins that are not categorized. The data is supplied for Dock, FlexX, Glide, ICM, PhDock, Surflex, and TrixX BMI. In some cases a second scoring option of TrixX BMI apart from the ScreenScore is presented. The results of the other docking tools are presented using their default settings. As suggested in the original paper [187], the enrichment data of Surflex with activated ring flexibility is considered as default setting.

In the following the enrichment results are presented in more detail. Table 7.5 provides ROC enrichments at 0.5 %, 1.0 %, 2.0 %, and 5.0 % of TrixX BMI for different scoring functions (FlexX score, ChemScore, PLP score, and ScreenScore). The results are presented and discussed by protein family. This includes the performance of

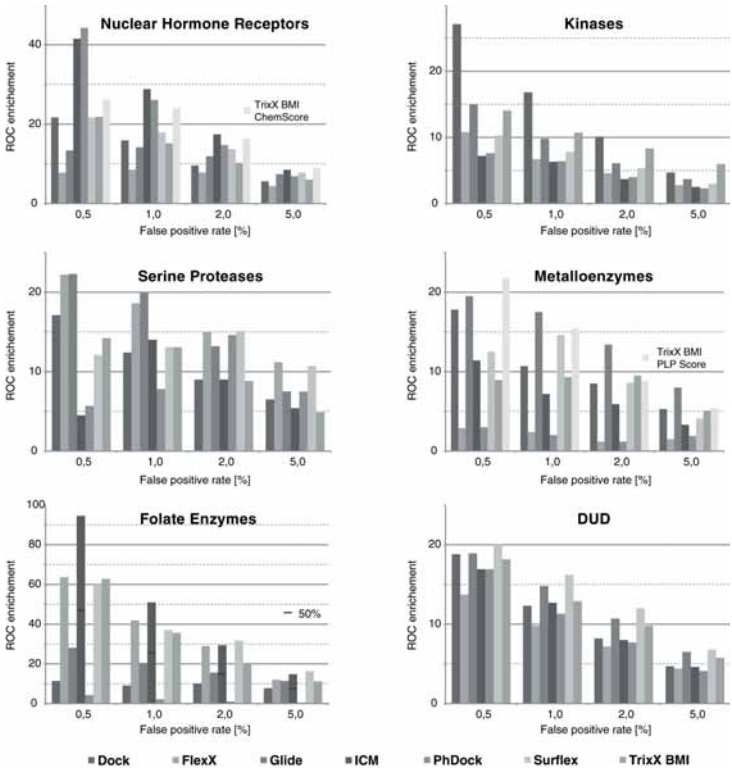


Figure 7.5.: ROC enrichments at 0.5 %, 1.0 %, 2.0 %, and 5.0 % false positive rate categorized by individual protein families and the results on the entire DUD data set.

different TrixX BMI scoring functions, especially in cases where the default option ScreenScore does not yield the best results. In contrast to previous redocking experiments, the choice of scoring function has a significant impact on the overall prediction quality. Also, the comparison to other docking tools is presented.

Nuclear hormone receptor In case of the NHR protein family, the ChemScore function clearly shows the best performance. Its ROC enrichment is higher than that of all other scoring functions for each observed false positive rate. The only other docking tool that produces consistently higher enrichments for the NHR family is ICM whereas PhDock outperforms TrixX BMI and ChemScore regarding the measurements at 0.5 % and 1.0 % but not at 2.0 % and 5.0 %. The performance of the other TrixX BMI scoring options cannot compete on the same level. The ScreenScore is on par with Dock and Surflex. TrixX BMI enrichment based on FlexX scoring is on the performance level of Glide, and still better than the results of PLP scoring which still outperforms FlexX.

Kinases For the kinase family ScreenScore and FlexX score yield almost identical enrichments and are superior to PLP and ChemScore. Regarding the other tools, only the program Dock outperforms TrixX BMI concerning early enrichment. The remaining tools, with the exception of Glide that exhibits a similar performance at 0.5 %, yield consistently lower ROC enrichments. At 5 % false positive rate TrixX BMI outperforms all competing approaches including Dock.

Serine proteases The TrixX BMI performance on serine proteases is again best for ScreenScore and FlexX score. Concerning the competing approaches, FlexX and Glide yield higher enrichments at all points of measurement. At 0.5 % and 1.0 %, TrixX BMI, Dock, and Surflex perform comparably and follow the previously mentioned tools. The situation shifts from there on, and PhDock and Surflex exhibit similar results as the leading tools FlexX and Glide. The remaining tools, TrixX BMI, Dock, and ICM yield lower but overall significant ROC enrichments of about 9 at 2 % false positive rate.

Metalloenzymes In combination with metalloenzymes, TrixX BMI and the PLP scoring option performs best for early enrichment, followed by similar results of FlexX score and ScreenScore. At an FP rate of 2.0 %, FlexX score and ScreenScore yield the best performance closely followed by PLP scoring which again performs best at 5 % FP rate. In comparison to other tools, the PLP score has the highest overall enrichment at 0.5 %. At 1.0 %, 2.0 %, and 5.0 % only Glide has better enrichments.

The other scoring options also perform well: Again, ScreenScore and FlexX score outperform ChemScore. Averaged over all points of measurement, TrixX BMI in combination with ScreenScore or FlexX score yield results that are comparable to Dock and Surflex. The enrichments are significantly higher than those of the remaining approaches FlexX, ICM, and PhDock.

		False positive rate [%]			
Family	Score	0.5	1.0	2.0	5.0
NHR	FX	17.7	12.8	10.4	6.1
	CS	26.2	24.0	16.4	9.1
	PLP	17.1	11.0	8.9	5.4
	SCREEN	21.8	15.2	10.3	6.0
Kinases	FX	14.0	10.6	8.2	6.0
	CS	7.3	6.7	6.0	4.2
	PLP	10.9	7.7	6.0	4.0
	SCREEN	14.1	10.7	8.3	6.0
Serine Proteases	FX	15.6	13.1	8.7	4.8
	CS	9.1	4.8	4.8	2.9
	PLP	0.5	2.1	1.7	1.9
	SCREEN	14.2	13.1	8.8	4.9
Metalloenzymes	FX	9.1	9.0	10.4	5.7
	CS	10.9	6.2	6.1	3.5
	PLP	21.8	15.5	8.8	5.4
	SCREEN	9.0	9.3	9.5	5.1
Folate Enzymes	FX	62.6	35.4	20.4	10.9
	CS	22.2	15.5	10.1	6.1
	PLP	47.0	29.5	22.2	11.4
	SCREEN	62.8	35.6	20.4	11.1
All	FX	17.5	12.4	9.9	5.0
	CS	14.0	12.0	9.0	5.3
	PLP	15.8	10.2	7.9	5.2
	SCREEN	18.2	12.9	9.8	5.0

Table 7.5.: Mean ROC enrichment at 0.5 %, 1.0 %, 2.0 %, and 5.0 % false positive rate for protein families and the entire DUD data set. Furthermore, different scoring functions are employed: FlexX score (FX), ChemScore (CS), PLP score (PLP), and ScreenScore (Screen).

Folate enzymes TrixX BMI performs well if used with folate enzymes. Again, ScreenScore and FlexX scoring perform best, followed by PLP scoring with a slightly diminished performance and eventually ChemScore. The comparison to other tools shows that only ICM yields significantly higher early enrichments at 0.5 % and 1.0 %. One has to keep in mind, that ICM is not able to dock any actives of the GART protein and thus the reported enrichment is for DHFR only. Thus, the bars of ICM performance on folates in Figure 7.5 are marked at 50 % to depict its performance on both proteins within this family. In a DHFR-only scenario, TrixX BMI outperforms ICM. The corresponding ROC enrichments are 115.6, 66.1, 37.1, and 16.3 at 0.5 %, 1.0 %, 2.0 %, 5.0 % FP rate.

TrixX BMI, FlexX, and Surflex perform best and are on a comparable level on the entire folate family including GART. Although, Surflex moves ahead for later points of measurement. Glide and Dock yield considerably lower enrichments, and PhDock basically fails to enrich active compounds in this setup.

Entire DUD The overall best results are produced using ScreenScore, followed by FlexX score. The remaining two options, PLP scoring and ChemScore, show a diminished overall performance, even though they are suited best for individual protein families, metalloenzymes and nuclear hormone receptors, respectively. The comparison to the state of the art of molecular docking tools reveals that only Surflex and Glide outperform TrixX BMI on the DUD data set. Thus, TrixX BMI produces the third best results followed by Dock which has a slightly higher mean enrichment at 0.5 % but falls behind from 1.0 % FP rate on. Compared to the remaining tools, TrixX BMI yields higher mean ROC enrichments for all points of measurement.

Furthermore, TrixX BMI yields a balanced performance over the different protein families: Enrichment never drops below a factor of 8 at 2 %. All other tools, with the exception of Dock, have significantly lower enrichments (less than factor 6.1) at 2.0 % for at least one of the protein families.

It must be noted that the results of all tools but Surflex and TrixX BMI — especially Glide and ICM — are associated with some uncertainty concerning the current statistical evaluation. Manual rescaling of the plots and visual extraction of ROC enrichments seems to suggest that the performance of these tools diminishes. Especially, early enrichment is sensitive to small changes of true positive and false

Target	Type	Detail[a]	e/o[b]	P_{min}/P_{max}[c]	Ref[d]
	h_{don}	N LEU 83	e	1,2	[189]
CDK2	h_{acc}	O LEU 83	o		
	h_{acc}	O GLU 81	o		
	h_{acc}	_OD2 ASP 26	e		[174]
DHFR	h_{acc}	O LEU 4	e		
	$phen_{center}$	_CG PHE 30	e		
	h_{don}	_NH2 ARG 394	e		[189]
ER agonist	h_{acc}	_OE1 GLU 353	e		
	spatial	$1.4, -1.4, -3.4$ (2.5 Å)	e		
	h_{don}	_NH2 ARG 394	e		[189]
ER agonist	h_{acc}	_OE1 GLU 353	e		
	spatial	$34.1,\ 0.5, 27.9$ (2.5 Å)	e		

[a]For interaction constraints, the name of the receptor atom (PDB nomenclature: atom name, amino acid code, amino acid number) is given. For a spatial constraint, the coordinates and the sphere radius are given. [b]Denotes an essential constraint, o an optional constraint. [c]P_{min} is the minimum number of optional constraints allowed. P_{max} is the maximum number of optional constraints allowed. [d]Here, the literature reference is provided.

Table 7.6.: Pharmacophore type constraints of four chosen target from the DUD.

positive rate. Thus, a concluding examination, which re-examines the currently not available original data, can hopefully be executed in the near future.

7.4.4. Pharmacophores

Four targets in the DUD — CDK2, DHFR, ER agonist, and ER antagonist — are selected as case studies for a more detailed analysis. Pharmacophore type constraints that describe the preferred binding mode in the active site and trigger biological activity are extracted from the literature (see Table 7.6) for each target. TrixX BMI pose predictions are provided in Figure 7.6: All targets are shown with the corresponding pharmacophore, the cocrystallized ligand structure in orange, and the pose prediction of the top ranked active in the final hitlist. The enrichment results are depicted in Figure 7.7.

CDK2 In case of CDK2 interactions to the flexible hinge region of the protein need to be established. The central donor interaction is essential and at least one of the two optional acceptors must be saturated. Changes in the hinge region can open or close the cleft between two subdomains of the kinase for ATP binding. The kinase is either activated or deactivated and thus regulates aspects of cell growth.

The enrichment plot shows that all four experiments select actives significantly over decoys. The usage of pharmacophore constraints in TrixX BMI removes pose predictions of active compounds. The current implementation does not differentiate between optional and essential constraints during triangle subselection. Some of the CDK2 actives are placed using only SIACs of the essential interaction and none of the optional ones. The corresponding descriptors are filtered and the placements are not found. Therefore, early enrichment is actually better using TrixX BMI without constraints. From 1 % false positive rate on, the pharmacophore experiment yields the highest enrichments. Furthermore, significantly more decoys than actives are removed using the filter: About 80 % of the actives but less than 50 % of the decoys are placed. Compared to FlexX and FlexX Pharm, TrixX BMI performs best concerning early enrichment and TrixX BMI in pharmacophore mode best starting at 1.0 % of the database.

DHFR The pharmacophore type constraint for DHFR is derived from the PDB entry 1rh3, which contains the protein cocrystallized with methotrexate. Two H-bonds formed between protonated nitrogens of the ligand's diaminopteridin ring system and two hydrogen acceptors of the protein are essential. These acceptors are nested deep within the active site. In addition, a hydrophobic interaction to a phenyl ring is required. The rather rigid scaffold of methotrexate or close analogs of it are found in many DHFR inhibitors. DHFR reduces dihydrofolic acid, which is essential for rapidly dividing cells to build the nucleobase thymine.

The enrichment behavior of all analyzed tools are almost perfect regarding the first percent of the database. At approximately 2 %, the constraint runs of TrixX BMI and FlexX outperform the nonconstrained experiments. Again, the pharmacophore filter works as expected: TrixX BMI Pharm finds predictions for almost all actives but only 17 % of the decoys.

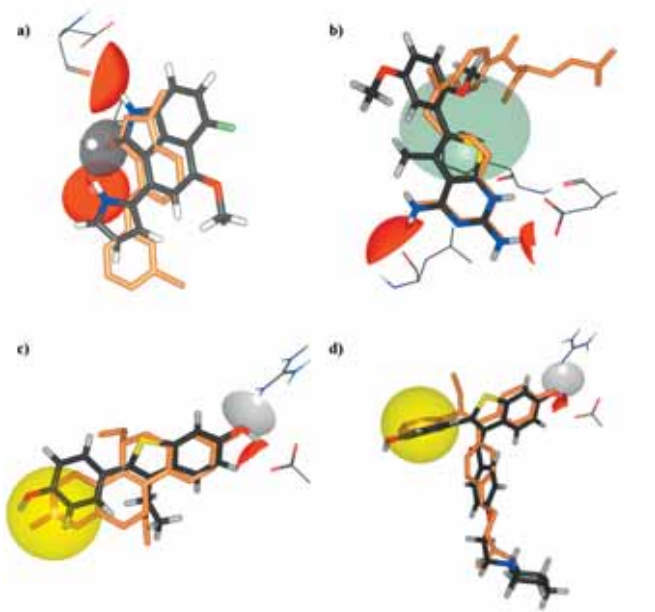


Figure 7.6.: TrixX BMI poses of the top ranked active of the target specific DUD actives and decoys that obeys the pharmacophore (rank with | without constraint): (a) CDK2 (1|9), (b) DHFR (1|2), (c) ER agonist (4|41), and (d) ER antagonist (5|13). Donors are depicted in grey, acceptors in red and spatial constraints as yellow sphere.

ER agonist The estrogen receptor used in the enrichment experiments is a nuclear hormone receptor and regulates gene expression. These receptors have a rather hydrophobic pocket. Thus, the pharmacophore for ER agonists includes a spatial constraint that is introduced due to this nonspecific and for this protein also flexible part of the pocket. The spatial constraint forces the ligands to be in a rather strained conformation thus filling the entire active site. In addition, two hydrophilic anchor interactions are employed.

The enrichment plot reveals that TrixX BMI in its basic version outperforms FlexX significantly. In combination with pharmacophore constraints, both tools yield

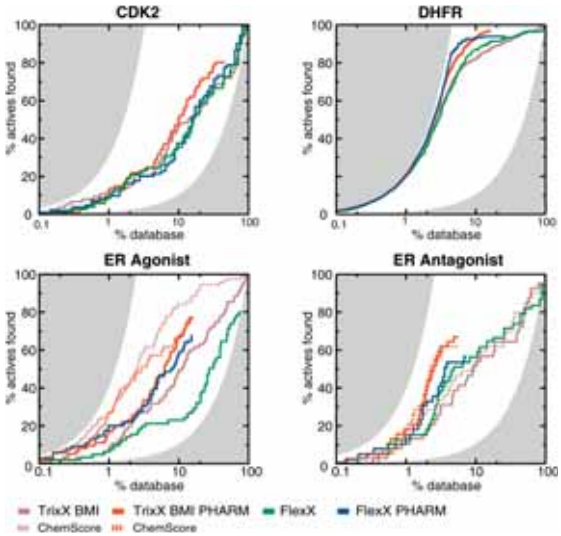


Figure 7.7.: Enrichment plots for CDK2, DHFR, and ER agonist/antagonist using TrixX BMI and FlexX, both with and without usage of pharmacophore type constraints. The plots of ER also includes the results based on ChemScore. The gray shaded areas represent maximal (top) and random enrichment (bottom). The x-axis is scaled logarithmically to focus on the important range of percentages.

a comparable performance. FlexX performs better concerning early enrichment below 0.5 %. TrixX BMI yields higher enrichment from about 5.0 % rate on. Furthermore, TrixX BMI Pharm finds 76 % of all actives but only 16 % of the decoys.

Since the ChemScore scoring option shows the best performance on the nuclear receptors from the DUD, the corresponding results are also presented. The usage of ChemScore yields significantly higher enrichments for TrixX BMI with and without pharmacophore constraints. This is especially true for early enrichment.

ER antagonist The enrichment of the ER antagonist is evaluated to demonstrate that TrixX BMI is also able to dock larger compounds that are not necessarily leadlike. The pharmacophore is identical to the one used for the agonist. Since here only antagonists

and similar decoys are searched, no further constraint is utilized. In a real world scenario where scientist are explicitly looking for antagonists a second spatial constraint is employed. This constraint is used to filter smaller compounds that cannot adopt the typical T-shape of estrogen antagonists.

FlexX has slightly better performance compared to TrixX BMI if no constraints are employed. Again, the ChemScore scoring function which exhibits the best results for nuclear hormone receptors yields enrichments that are comparable to FlexX. If pharmacophore type constraints are used, TrixX BMI clearly outperforms FlexX independent of the scoring function being used. The filtered run yields pose predictions for about 60 % of the actives and less than 6 % of the decoys.

All four protein targets have active compounds that cannot be placed using the TrixX BMI Pharm approach. In some cases this might be due to a different binding mode of the active ligands which means that the given pharmacophore cannot be obeyed by all of them. Another aspect is the quality of the conformational ensembles. If none of the conformations within an ensemble represents a conformation close to the bioactive one, this can also prevent a correct pose prediction due to clashes or a wrong geometry arrangement of the required pharmacophore interactions. However, FlexX Pharm is also not able to place all actives. This suggests that different binding modes of these ligands are more likely to be the reason for docking failures.

On average, TrixX BMI generates poses fulfilling the corresponding pharmacophore type constraints for 78 % of the actives and 35 % of the decoys. This and the previous analysis of the individual enrichment plots clearly demonstrate that pharmacophore filtering in combination with the TrixX BMI subselection of site descriptors significantly enriches actives over decoys.

7.5. Runtime and Space Requirements

This section analyzes the requirements of TrixX BMI concerning runtime and storage space. First, some statistics about TrixX BMI descriptors for protein active sites as well as molecular compounds from the DUD are shown. Then, runtime measurements based on experiments using these targets and compounds are discussed. Based on these numbers, the average selectivity of TrixX BMI descriptor queries, hard disk-, and

Target	actives +decoys[a]	no. of conf.[b]	no. of descr.[c]	no. of site descriptors[d] without	no. of site descriptors[d] with constraints
CDK2	72+2069	21	147	10568	253
DHFR	410+8353	22	181	18895	898
ER agonist	67+2557	14	98	3010	260
ER antagonist	39+1446	48	155	15251	160
Astex set	Z_2	8	60	16067	n.a.

[a]Number of actives and decoys of the target protein in the DUD. In case of the Astex Diverse Set the Z_2 set of leadlike compounds is used. [b]Average number of conformations. [c]Average number of descriptors. [d]Number of site descriptors with and without pharmacophore constraints. In case of the Astex Diverse Set average values are presented.

Table 7.7.: Descriptor statistics of four selected DUD proteins and their compound sets. In addition, average values for the Astex Diverse Set and 2000 leadlike compounds.

memory requirements are analyzed. Finally, a large scale VS experiment is presented to demonstrate the scalability of TrixX BMI.

7.5.1. Descriptor analysis

The overall runtime and space requirements are connected to the number of descriptors in the compound index as well as the number of site descriptors used for querying. Therefore, Table 7.7 gives some statistics on the chosen targets from the DUD and their corresponding active and decoy set. Furthermore, average values for the random set of 2000 leadlike compounds from the ZINC database (Z_2) are presented. As expected, the average number of conformations depends on the flexibility of the compounds within the target specific set of actives and decoys (see Figure 7.4). Additionally, the average number of descriptors in the different sets of compounds is presented. If pharmacophore information is available, the TrixX BMI subselection of site descriptors yields a significant reduction: On average, only 4.2 % of the site descriptors are retained as query templates, the maximum being 8.6 % for DHFR, the mimimum 1.0 % for the ER antagonist.

	Runtime [s] on							
	DUD index				Z_2 index			
Target	TrixX BMI		FlexX		TrixX BMI		FlexX	
CDK2	0.48	(0.05)	6.4	(5.5)	0.25	(0.02)	5.4	(3.5)
DHFR	3.16	(0.13)	9.2	(14.4)	0.37	(0.04)	9.1	(3.2)
ER agonist	0.20	(0.06)	6.2	(6.2)	0.13	(0.04)	3.1	(3.3)
ER antagonist	1.84	(0.10)	16.6	(21.7)	0.32	(0.03)	6.8	(6.3)
Astex Diverse Set	n.a.	(n.a.)	n.a.	(n.a.)	0.24	(n.a.)	7.8	(n.a.)

Table 7.8.: Average runtimes of TrixX BMI and FlexX without (and with) pharmacophore constraints for each compound in the index.

7.5.2. Runtime requirements

Runtime experiments compare TrixX BMI to FlexX and FlexX Pharm. Average runtimes for a set of 2000 leadlike compounds are shown. Furthermore, the runtime of TrixX BMI is analyzed with respect to the number of queries and the I/O load of the underlying system is evaluated.

In Table 7.8 the resulting runtime measurements are presented. It is obvious that TrixX BMI offers a substantial speed-up over FlexX. The average runtime of all 85 targets in Astex Diverse Set is 0.24 s per compound compared to 7.8 s for FlexX. This average speed-up of factor 32.5 and the individual runtime of the four DUD targets on the Z_2 compound index demonstrate that the descriptor-based look-up technology yields an improvement in runtime of more than one order of magnitude to an iterative screening approach like FlexX. The targets from the DUD show that TrixX BMI has good runtime behavior on the given data sets even though these sets are not strictly leadlike.

Pharmacophore mode

As already mentioned, the introduction of pharmacophore type constraints reduces the number of site descriptors and thus queries significantly. In addition, pharmacophore constraints tend to be nested deep in active site. This often results in higher descriptor selectivity. Table 7.9 shows, that the selectivity of TrixX BMI pharmacophore queries is considerably higher for all proteins with exception of the ER agonist. The estrogen

	Descriptor selectivity [‰] on Z_2		
Target	without	with	Factor
CDK2	0.13	0.014	8.7
DHFR	0.02	0.001	22.3
ER agonist	0.01	0.011	0.9
ER antagonist	0.02	0.009	1.8
Astex Diverse Set	0.17	n.a.	n.a.

Table 7.9.: TrixX BMI descriptor selectivity with and without pharmacophore type constraints and the corresponding factor quantifying the increase in selectivity.

receptor with a bound agonist exhibits a closed and small pocket. There are no descriptors that entail solvent exposed compound placements and as such yield numerous possible compound placements. In such a scenario, the selectivity of TrixX BMI does not increase if pharmacophore type constraints are employed.

Due to nonavailable pharmacophores only unconstrained runtimes against the Z_2 are supplied for the Astex Diverse set. For the targets from the DUD, the reduction of queries and the increase of selectivity results in decreasing runtime. TrixX BMI needs 82 ms for each compound in the database averaged over all targets and their target specific compound catalogs. In case of the leadlike Z_2 compound index, 30 ms suffice. This corresponds to a speed-up of another order of magnitude compared to the nonconstrained experiments and results in a runtime that is two orders of magnitude faster than FlexX and FlexX Pharm.

On average, TrixX BMI runtimes include an I/O overhead of 4 % with a standard deviation of 1.8 %. The different phases of TrixX BMI virtual screening, the initial identification of descriptor matches and the actual docking calculations, each account for about 50 % of the total runtime.

Contribution of FastBit

The FastBit indexing system speeds up descriptor matching by a factor of five. Thus, reverting to a raw data scan yields an increase in total runtime by a factor of three since the query phase accounts for about 50 % of the overall TrixX BMI runtime. Without FastBit, the average runtime on a library of leadlike compounds drops to 0.72 s. This

	TrixX BMI[a]		FlexX[b]	
Target	without	with constraints	without	with constraints
CDK2	2:04	0:09	37:54	24:20
DHFR	9:43	0:05	58:20	17:06
ER (agonist)	0:53	0:27	43:45	28:36
ER (antagonist)	6:01	0:25	48:37	33:03

[a]Maximum runtime [h:min]. [b]Runtime [h:min] estimated based on representative results.

Table 7.10.: Total VS runtimes using the DUD targets and 1.7 million compounds on 48 cluster nodes.

demonstrates that FastBit is an important part of the TrixX BMI screening pipeline and efficiently exploits the high selectivity of the TrixX BMI molecular descriptor.

7.5.3. Space requirements

Another important aspect is the time needed to build the compound indices. It is obviously correlated to the average number of conformations and the molecular properties of the library compounds. For the Z_2 data set the compound indexing phase takes about 0.5 s per compound. On average, this time is split in half between conformational sampling and subsequent descriptor generation. Thus, the setup of a large compound collection consisting of millions of individual compounds can be accomplished overnight on a reasonable sized compute cluster. The space requirements correspond to the approximation presented in Section 6.3.4: A test on 1.7 million randomly chosen leadlike compounds from the ZINC database yields a compound database of about 500 GB.

The main memory requirements are basically fixed by the cache sizes of the employed indexing system and storage system. In the previous experiments each TrixX BMI process is supplied with 2 GB of main memory. Thus, on each node at most 4 GB and in total 192 GB are used.

7.5.4. Scalability

The last part demonstrates the scalability of TrixX BMI on a medium-sized compute cluster. Again, the 1.7 million random leadlike ZINC compounds are used. The corresponding compound indices are distributed on 48 cluster nodes of 2.4 GHz Dual

Xenon CPUs with 4 GB of main memory on each node. According to the shared-nothing paradigm of TrixX BMI parallelization, the compound library is split into 96 packages, one for each core. These packages are distributed to the individual nodes. In order to reduce network load each package is assigned to two additional cores. Thus, each node manages six different packages. This leads to about 30 GB on each nodes local hard drive and adds up to about 1.5 terabyte (TB) on the whole compute cluster. The resulting system is used to perform virtual high-throughput screening of the four DUD targets that are also used in the enrichment studies.

Table 7.10 shows the maximum runtime for each target on 48 cluster nodes, respectively the 96 cores. TrixX BMI is able to perform virtual screening of 1.7 million ligands using four different target proteins in between 5 and 27 minutes with pharmacophore constraints. Without constraints it takes between 53 minutes and just below 10 hours. In comparison, a sequential docking tool like FlexX in the same parallel setup needs at least 17 hours in the constraint search and up to more than 2.5 days in the nonconstraint search.

8 Conclusion & Outlook

This thesis presents TrixX BMI, a new approach to structure-based VS. In the following, the main concepts and validation results are summarized. The first section introduces key concepts of TrixX BMI. Then, the results of the validation experiments are analyzed with respect to the original research goals and related to the results of other docking tools. The next section focuses on limitations of the current TrixX BMI version. Subsequently, the overall applicability of the approach is analyzed. The last section shows future perspectives and describes possible extensions. This involves new application scenarios as well as adaptations of the methodology.

8.1. Overview

Based on an innovative description of molecular properties, TrixX BMI implements an index-driven approach to VS. The validation shows that the new approach reproduces protein-ligand complexes and successfully enriches known binders. Most notably, TrixX BMI offers a speed-up of more than one order of magnitude compared to state-of-the-art approaches. Pharmacophore information can be incorporated resulting in an increase of enrichment and a speed-up that excels two orders of magnitude. Since the system scales in a parallel computing environment, it is applicable to high-throughput VS experiments.

Key components Trixx BMI is implemented as a hierarchical screening pipeline based on a preprocessed database of descriptors. In the following, its key components are highlighted:

- The most prominent concept in TrixX BMI is the descriptor: It encodes physicochemical properties and can be generated for molecular compounds and protein active sites. Individual descriptor properties are encoded complementarily including a unique, high-dimensional model of molecular shape. A descriptor-based alignment scheme enables TrixX BMI to identify reasonable placements solely on the abstract descriptor level. All descriptor attributes are independent of the target protein and are persistently stored on peripheral storage, and each descriptor attribute can be accessed via fast Bitmap Indices. This unique feature of TrixX BMI combined with high descriptor selectivity contributes significantly to the overall performance.

- The TrixX Conformer Generator is used to handle large parts of molecular flexibility by pre-enumeration of compound conformations. The resulting conformations are of high quality, comprise only few conformations, and are basis for the calculation of compound descriptors. The algorithm is based on a force-field guided, best-first search that uses flexibility-dependent thresholds to constrain the search space and to guarantee a reasonable coverage of conformational space.

- TrixX BMI employs a hierarchical docking pipeline to generate 3D pose predictions. Initially, matching site and compound descriptors are generated. These are transformed into descriptor poses using a grid-based clash- and scoring routine. During the next stage, the best ranked descriptor poses are refined by reorienting flexible groups of the compound, performing pairwise atom-atom clash tests, and employing an empirical scoring function. This results in so-called TrixX BMI poses that can optionally be handed over to the final stage of optimization.

- A multi-level partitioning scheme enables TrixX BMI to be used in a parallel, shared-nothing computing environment. Three horizontal partitions — compound partitions, descriptor partitions, and type partitions — are used to distribute library compounds to different cluster nodes, lower the main memory requirements, and reduce CPU costs.

Accomplishments The validation of TrixX BMI shows that most of the original research goals are reached.

- The average runtime of TrixX BMI compared to other tools is improved by more than one order of magnitude. Compared to FlexX, the runtime drops by an average factor of 32.5. If pharmacophore information is employed, which typically is the case in large-scale VS experiments, this factor increases to 141 averaged over the four proteins of the case studies.

- The quality of TrixX BMI's results is comparable to that of competing approaches. The redocking accuracy, benchmarked on 85 high-quality complexes of the Astex Diverse Set, is on par with Gold and FlexX. The same holds true for the enrichment performance. It is evaluated on the publicly available DUD data set that allows comparison to six other tools. Averaged over the entire set of 40 proteins, TrixX BMI ranks third overall, closely following the two top ranked approaches.

- Scalability is validated on a compute cluster comprising 96 cores and a library of 1.7 million leadlike compounds. VS of this library using TrixX BMI takes in between 1 hours and 6 hours depending on the chosen target protein. If pharmacophore information is added, less than 30 minutes suffice for all case studies. As comparison, FlexX runtimes are presented. These are in between 17 hours and 58 hours using the same target proteins and pharmacophores.

8.2. Limitations

Regarding the original research goals, TrixX BMI has only one limitation: The current version does not incorporate protein flexibility. A minor weakness of the approach concerns its ability to produce highly accurate poses for binding mode analysis. In the following, these two aspects are discussed in more detail.

Protein flexibility The need to account for the dynamic behavior of proteins is a deficiency of modern computational drug design. Most applications, including TrixX BMI, assume the protein to be rigid.

A naive approach towards models for protein flexibility, sequential docking of multiple protein conformations, is possible. However, a general solution needs to consider more details: Even for a rigid protein, the problem of scoring and optimizing compound

placements is not yet satisfactorily solved. For flexible-protein docking, the energetic differences of each protein conformation need to be considered. A fast and reliable scoring scheme is currently not available and remains a challenge for future research.

Binding mode prediction A minor weakness of TrixX BMI concerns its ability to generate highly accurate predictions below 1 Å RMSD to the cocrystallized ligand. TrixX BMI docks pre-enumerated conformations rigidly into the binding site and reorients their flexible groups. The resulting placements correctly reflect the overall binding mode as validation experiments show. However, structural details of the protein-ligand complex are not considered. Such details can obviously not be part of precalculated conformational ensembles. Therefore, sophisticated postoptimization routines need to be applied, for instance force-field optimization. The internal optimization option of TrixX BMI is not able to generate highly accurate predictions. Clashes are reliably resolved but the overall prediction accuracy does not improve. Therefore, external optimization tools need to be employed.

8.3. Applicability

The validation experiments demonstrate that TrixX BMI is suited for high throughput VS experiments. The quality of the resulting placements and the enrichment factors of active compounds are on par with industry-leading tools. The speed-up of two orders of magnitude enables researchers to screen millions of compounds on a reasonably-sized compute cluster within minutes, rather than days.

TrixX BMI does not include models for protein flexiblity. Since this aspect is not part of any other approach suited for high-througput VS experiments, it does not impact the overall applicability of TrixX BMI. As part of redocking and crossdocking experiments, fewer compounds are subjected to docking calculations and more sophisticated approaches are available. In this scenario, TrixX BMI can be pipelined with an external optimization tool to account for the requirements of detailed binding mode anlysis and also to incorporate protein flexibility.

8.4. Outlook

The validation studies suggest that TrixX BMI is an efficient approach towards structure-based VS. However, certain challenges remain. In the following, some ideas on how to augment the descriptor and the overall docking methodology are presented. Furthermore, the applicability of TrixX BMI descriptors in related fields is discussed.

Protein flexibility and postoptimization The overall approach of TrixX BMI is highly suited for adaptations towards protein flexibility. Its tremendous speed offers a path towards fully flexible VS in reasonable time.

In a first step, the descriptor could be augmented towards multiple protein conformations. Rigid parts of the protein would be captured using a single descriptor instance while flexible sidechains could be handled separately. Each descriptor would thus be associated to multiple conformations of the protein. Subsequently, the hierarchical docking engine of TrixX BMI could be augmented with additional stages. Especially, the clash, scoring, and postoptimization routines should be adapted. Clash calculations would consider multiple protein conformation simultaneously. The same could be done for scoring and postoptimization. Protein flexibility could be included on each level of the screening pipeline. This would involve grid-based operations as well as calculations on atomic coordinates.

Descriptor extensions The TrixX BMI descriptor as it is does not carry any electrostatic information. Electrostatic potential could be mapped onto molecular surface patches. The description of shape could then be augmented as follows: Each of the 80 rays describing the molecular shape could be attributed with an electrostatics value describing the environment where it intersects with the surface. These new descriptor attributes could either be used to prefilter compounds during descriptor matching or to score descriptor matches using electrostatic compatibility. Thus, not only clash predictions but also scoring could be performed already on the abstract descriptor level.

Application to other problems Generally speaking, any problem that involves physicochemical properties in 3D space can benefit from using TrixX BMI descriptors.

Since the TrixX BMI descriptor is applicable to compounds and to the active site of proteins, it can be used to handle problems closely related to protein-ligand docking and VS:

- A tool for ligand-based VS has already been implemented as part of a diploma thesis. A reference ligand is used to generate compound descriptors which then serve as query templates. The corresponding query conditions need to be adapted in order to accommodate for partial shape matches. The objective is no longer the detection of clashes but the maximization of shape overlap.

- The comparison of protein active sites could be performed analogously to ligand-based VS. Site descriptors would have to be used instead of compound descriptors.

- Another application is inverse docking. Here, multiple protein targets are screened using a single compound in order to evaluate its selectivity and thus the risk of side effects. To perform inverse docking, the overall TrixX BMI process could be inverted: Site descriptors would be subject to indexing and compound descriptors would serve as query templates. The remainder of the worfklow would stay unchanged.

- TrixX BMI descriptors are also suited to support pharmacophore searching. Instead of generating query descriptors based on compounds or active sites, pharmacophore models would be employed. Corresponding chemical feature points of the pharmacophore could be used as basis for the descriptor geometry, and shape would be handled as in the ligand-based VS scenario.

Tools for de-novo or focused library design often rely on the selection of appropriate fragments from a large library of candidate fragments. This selection mechanism could be efficiently supported using TrixX BMI descriptors. As part of library design, the descriptors could also be used to evaluate diversity. Here, the set of all library descriptors for a given library would be used to decide whether a new compound should be added. Only if a compound's descriptors introduces diversity to the library, meaning that they differ significantly from the ones already in the set, the compound should be added. Based on this criterion, optimization algorithms could be designed that focus on the selection of an optimal subset of compounds with respect to chemical diversity.

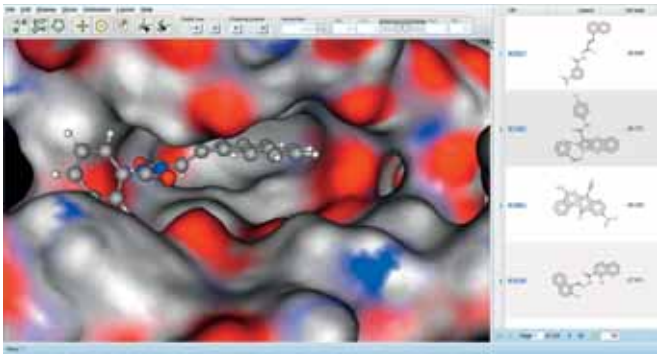


Figure 8.1.: ViSor 3D viewer (left) and result browser (right).

Technology TrixX BMI can be efficiently employed in a parallel computing environment. However, the current implementation is based on scripts and needs supervision. Multi-user capabilities and automatic load-balancing routines which distribute and share the data efficiently among the available compute nodes are not available. These tasks are realized as part of an industry cooperation under the ViSoR project [190]. ViSoR is a browser-based platform (see Figure 8.1) that offers an integrated approach to knowledge management in the context of drug discovery. It includes experimental data, as well as computer-aided methods, for instance TrixX BMI as VS application.

The concept of general-purpose computing on a graphics processing unit (GPU) can be utilized to further reduce the runtime of TrixX BMI. Grid-based algorithms for clash detection and scoring can be transformed to correspond to the single-instruction-multiple-data paradigm of data level parallelism. Multiple poses need to be checked for clashes and need to scored using one instance of a protein. The current version of TrixX BMI performs these task iteratively, one pose at a time. Kernel functions for GPU execution would be straightforward to develop and could provide a large speed-up. This is especially true for unselective active sites which generate thousands of descriptor poses for a single compound. In such a case, the identification of the most promising descriptor poses for downstream processing could be accelerated enormously by usage of GPU technology.

References

[1] Kubinyi, H. *Wiley Series in Drug Discovery and Development*, chapter Docking and scoring for structure-based drug design, 377–424. Wiley-VCH, Weinheim (2006). 1

[2] Shoichet, B. Virtual screening of chemical libraries. *Nature* **432**(7019), 862–865 (2004). 1

[3] Rarey, M., Kramer, B., Lengauer, T., and Klebe, G. A fast flexible docking method using an incremental construction algorithm. *Journal of Molecular Biology* **261**(3), 470–489 (1996). 3, 24, 42

[4] Schellhammer, I. and Rarey, M. Trixx: structure-based molecule indexing for large-scale virtual screening in sublinear time. *Journal of Computer-Aided Molecular Design* **21**(5), 223–238 (2007). 3, 28

[5] Schlosser, J. and Rarey, M. Beyond the virtual screening paradigm: Structure-based searching for new lead compounds. *Journal of Chemical Information and Modeling* **49**(4), 800–809 (2009). 3

[6] Fischer, E. Einfluss der configuration auf die wirkung der enzyme. *Berichte der deutschen chemischen Gesellschaft* **27**(3), 2985–2993 (1894). 5

[7] Langley, J. N. On the reaction of cells and of nerve-endings to certain poisons, chiefly as regards the reaction of striated muscle to nicotine and to curari. *Journal of Physiology* **33**(7019), 373–413 (1905). 5

[8] Bragg, W. L. The specular reflexion of x-rays. *Nature* **90**, 410 (1912). 5

[9] Purcell, E. M., Torrey, H. C., and Pound, R. V. Resonance absorption by nuclearmagnetic moments in a solid. *Physical Review* **69**, 37–38 (1946). 5

[10] Bloch, F., Hansen, W. W., and Packard, M. Nuclear induction. *Physical Review* **69**, 127 (1946). 5

[11] Gribbon, P., Lyons, R., Laflin, P., Bradley, J., Chambers, C., Williams, B., Keighley, W., and Sewing, A. Evaluating real-life high-throughput screening data. *Journal of Biomolecular Screening : the Official Journal of the Society for Biomolecular Screening* **10**(2), 99–107 (2005). 6

[12] Leach, A. and Hann, M. The in silico world of virtual libraries. *Drug Discovery Today* **5**(8), 326–336 (2000). 6

[13] Reynolds, C., Tounge, B., and Bembenek, S. Ligand binding efficiency: trends, physical basis, and implications. *Journal of Medicinal Chemistry* **51**(8), 2432–2438 (2008). 6

[14] Hann, M., Leach, A., and Harper, G. Molecular complexity and its impact on the probability of finding leads for drug discovery. *Journal of Chemical Information and Computer Sciences* **41**(3), 856–864 (2001). 6

[15] Venkatesh, S. and Lipper, R. Role of the development scientist in compound lead selection and optimization. *Journal of Pharmaceutical Sciences* **89**(2), 145–154 (2000). 7

[16] Lipinski, C., Lombardo, F., Dominy, B., and Feeney, P. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Advanced Drug Delivery Reviews* **46**(1-3), 3–26 (2001). 7, 8

[17] Wermuth, C. G., Ganellin, C. R., Lindberg, P., and Mitscher, L. A. *Glossary of terms used in medicinal chemistry (IUPAC recommendations)*, volume 70, 1129–1143. International Union of Pure and Applied Chemistry (1998). 7, 10

[18] Sheridan, R. and Kearsley, S. Why do we need so many chemical similarity search methods? *Drug Discovery Today* **7**(17), 903–911 (2002). 7

[19] Klekota, J. and Roth, F. Chemical substructures that enrich for biological activity. *Bioinformatics (Oxford, England)* **24**(21), 2518–2525 (2008). 7

[20] Kubinyi, H., Folkers, G., and Martin, Y. C., editors. *3D QSAR in drug design*, volume 1-3. Kluwer/ESCOM, Dordrecht, Boston, London, (1998). 8

[21] Bajorath, J. Integration of virtual and high-throughput screening. *Nature Reviews Drug Discovery* **1**(11), 882–894 (2002). 8

[22] Oprea, T., Davis, A., Teague, S., and Leeson, P. Is there a difference between leads and drugs? a historical perspective. *Journal of Chemical Information and Computer Sciences* **41**(5), 1308–1315 (2001). 8, 67, 87

[23] Wiener, H. Structural determination of paraffin boiling points. *Journal of the American Chemical Society* **69**(1), 17–20 (1947). 9

[24] James, C. and Weininger, D. *Daylight theory manual*, chapter 6. Daylight Chemical Information Systems (2006). 9, 78

[25] Rush, T., Grant, J., Mosyak, L., and Nicholls, A. A shape-based 3-d scaffold hopping method and its application to a bacterial protein-protein interaction. *Journal of Medicinal Chemistry* **48**(5), 1489–1495 (2005). 9

[26] Haigh, J., Pickup, B., Grant, J., and Nicholls, A. Small molecule shape-fingerprints. *Journal of Chemical Information and Modeling* **45**(3), 673–684 (2005). 9

[27] Cheeseright, T., Mackey, M., Rose, S., and Vinter, A. Molecular field extrema as descriptors of biological activity: definition and validation. *Journal of Chemical Information and Modeling* **46**(2), 665–676 (2006). 9

[28] Lemmen, C. and Lengauer, T. Computational methods for the structural alignment of molecules. *Journal of Computer Aided Molecular Design* **14**(3), 215–232, March (2000). 9

[29] Lemmen, C., Lengauer, T., and Klebe, G. Flexs: a method for fast flexible ligand superposition. *Journal of Medicinal Chemistry* **41**(23), 4502–4520 (1998). 9

[30] Jones, G., Willett, P., and Glen, R. A genetic algorithm for flexible molecular overlay and pharmacophore elucidation. *Journal of Computer-Aided Molecular Design* **9**(6), 532–549 (1995). 9, 33

[31] Labute, P., Williams, C., Feher, M., Sourial, E., and Schmidt, J. Flexible alignment of small molecules. *Journal of Medicinal Chemistry* **44**(10), 1483–1490 (2001). 9

[32] Cho, S. and Sun, Y. Flame: a program to flexibly align molecules. *Journal of Chemical Information and Modeling* **46**(1), 298–306 (2006). 9

[33] Eckert, H. and Bajorath, J. Molecular similarity analysis in virtual screening: foundations, limitations and novel approaches. *Drug Discovery Today* **12**, 225–233 (2007). 9

[34] Klebe, G. Virtual ligand screening: strategies, perspectives and limitations. *Drug Discovery Today* **11**(13-14), 580–594 (2006). 9, 34

[35] Ehrlich, P. Present status of chemotherapy. *Berichte der Deutschen Chemischen Gesellschaft* **42**, 17–47 (1909). 10

[36] Dodds, E. and Lawson, W. Molecular structure in relation to oestrogenic activity. compounds without a phenanthrene nucleus. In *Proceedings of the Royal Society of London. Series B, Biological Sciences*, volume 125, 222–232, (1938). No. 839. 10

[37] Schueler, F. Sex hormonal action and chemical constitution. *Science (New York, N.Y.)* **103**(2669), 221–223 (1946). 10

[38] Beno, B. and Mason, J. The design of combinatorial libraries using properties and 3d pharmacophore fingerprints. *Drug Discovery Today* **6**(5), 251–258 (2001). 11

[39] Van Drie, J. H. and Lajiness, M. S. Approaches to virtual library design. *Drug Discovery Today* **3**(10), 274–283 (1998). 11

[40] Mannhold, R., editor. *Molecular Drug Properties*. Wiley-VCH, Weinheim, Germany, (2008). 11

[41] Gillet, V., Downs, G., Holliday, J., Lynch, M., and Dethlefsen, W. Computer storage and retrieval of generic chemical structures in patents. 13. reduced graph generation. *Journal of Chemical Information and Modeling* **31**(2), 260–270 (1991). 14

[42] Rarey, M. and Dixon, J. Feature trees: a new molecular similarity measure based on tree matching. *Journal of Computer-Aided Molecular Design* **12**(5), 471–490 (1998). 14

[43] Kuntz, I. D., Blaney, J. M., Oatley, S. J., Langridge, R., and Ferrin, T. E. A geometric approach to macromolecule-ligand interactions. *Journal Molecular Biology* **161**(2), 269–288, October (1982). 14, 23

[44] Bohm, H. The computer program ludi: a new method for the de novo design of enzyme inhibitors. *Journal of Computer-Aided Molecular Design* **6**(1), 61–78 (1992). 14, 42

[45] Goodford, P. A computational procedure for determining energetically favorable binding sites on biologically important macromolecules. *Journal of Medicinal Chemistry* **28**(7), 849–857 (1985). 15

[46] Verdonk, M., Cole, J., Watson, P., Gillet, V., and Willett, P. Superstar: improved knowledge-based interaction fields for protein binding sites. *Journal of Molecular Biology* **307**(3), 841–859 (2001). 15

[47] Weisel, M., Proschak, E., and Schneider, G. Pocketpicker: analysis of ligand binding-sites with shape descriptors. *Chemistry Central Journal* **1**, 7 (2007). 15

[48] Nayal, M. and Honig, B. On the nature of cavities on protein surfaces: application to the identification of drug-binding sites. *Proteins* **63**(4), 892–906 (2006). 15, 55

[49] Kabsch, W. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A* **32**(5), 922–923, Sep (1976). 16

[50] Koshland, D. Application of a theory of enzyme specificity to protein synthesis. *Proceedings of the National Academy of Sciences of the United States of America* **44**(2), 98–104 (1958). 16

[51] Brooijmans, N. and Kuntz, I. Molecular recognition and docking algorithms. *Annual Review of Biophysics and Biomolecular Structure* **32**(1), 335–373 (2003). 16, 28

[52] Kumar, S., Ma, B., Tsai, C., Sinha, N., and Nussinov, R. Folding and binding cascades: dynamic landscapes and population shifts. *Protein Science : a Publication of the Protein Society* **9**(1), 10–19 (2000). 17

[53] Jiang, F. and Kim, S. "soft docking": matching of molecular surface cubes. *Journal of Molecular Biology* **219**(1), 79–102 (1991). 17

[54] Schaffer, L. and Verkhivker, G. Predicting structural effects in hiv-1 protease mutant complexes with flexible ligand docking and protein side-chain optimization. *Proteins* **33**(2), 295–310 (1998). 17

[55] Leach, A. Ligand docking to proteins with discrete side-chain flexibility. *Journal of Molecular Biology* **235**(1), 345–356 (1994). 18

[56] Jackson, R., Gabb, H., and Sternberg, M. Rapid refinement of protein interfaces incorporating solvation: application to the docking problem. *Journal of Molecular Biology* **276**(1), 265–285 (1998). 18

[57] Zacharias, M. Rapid protein-ligand docking using soft modes from molecular dynamics simulations to account for protein deformability: binding of fk506 to fkbp. *Proteins* **54**(4), 759–767 (2004). 18

[58] Sandak, B., Wolfson, H., and Nussinov, R. Flexible docking allowing induced fit in proteins: insights from an open to closed conformational isomers. *Proteins* **32**(2), 159–174 (1998). 18

[59] Huang, S.-Y. and Zou, X. Ensemble docking of multiple protein structures: considering protein structural variations in molecular docking. *Proteins* **66**(2), 399–421 (2007). 18

[60] Bottegoni, G., Kufareva, I., Totrov, M., and Abagyan, R. Four-dimensional docking: a fast and accurate account of discrete receptor flexibility in ligand docking. *Journal of Medicinal Chemistry* **52**(2), 397–406 (2009). 18

[61] Claussen, H., Buning, C., Rarey, M., and Lengauer, T. Flexe: efficient molecular docking considering protein structure variations. *Journal of Molecular Biology* **308**(2), 377–395 (2001). 18

[62] Wei, B., Weaver, L., Ferrari, A., Matthews, B., and Shoichet, B. Testing a flexible-receptor docking algorithm in a model binding site. *Journal of Molecular Biology* **337**(5), 1161–1182 (2004). 18

[63] Knegtel, R., Kuntz, I., and Oshiro, C. Molecular docking to ensembles of protein structures. *Journal of Molecular Biology* **266**(2), 424–440 (1997). 18

[64] Sotriffer, C. and Dramburg, I. "in situ cross-docking" to simultaneously address multiple targets. *Journal of Medicinal Chemistry* **48**(9), 3122–3125 (2005). 18

[65] Nabuurs, S., Wagener, M., and De Vlieg, J. A flexible approach to induced fit docking. *Journal Of Medicinal Chemistry* **50**(26), 6507–6518 (2007). 18

[66] Teodoro, M. and Kavraki, L. Conformational flexibility models for the receptor in structure-based drug design. *Current Pharmaceutical Design* **9**(20), 1635–1648 (2003). 18, 22

[67] Totrov, M. and Abagyan, R. Flexible ligand docking to multiple receptor conformations: a practical alternative. *Current Opinion in Structural Biology* **18**(2), 178–184 (2008). 18, 22

[68] Pauling, L. *The nature of the Chemical Bond*. Cornell University Press, (1960). 19

[69] Harding, M. The geometry of metal-ligand interactions relevant to proteins. *Acta Crystallographica. Section D, Biological Crystallography* **55**(Pt 8), 1432–1443 (1999). 19

[70] Dunitz, J. and Gavezzotti, A. How molecules stick together in organic crystals: weak intermolecular interactions. *Chemical Society Reviews* **38**(9), 2622–2633 (2009). 19

[71] Parthasarathi, R. and V., S. *Hydrogen Bonding–New Insights*, chapter Characterization of Hydrogen Bonding: From van der Waals Interactions to Covalency, 1–50. Springer Netherlands (2006). 19

[72] Dunitz, J. and Gavezzotti, A. Attractions and repulsions in molecular crystals: What can be learned from the crystal structures of condensed ring aromatic hydrocarbons? *Accounts of Chemical Research* **32**(8), 677–684 (1999). 19

[73] Eisenberg, D. and McLachlan, A. D. Solvation energy in protein folding and binding. *Nature* **319**(6050), 199–203 (1986). 20

[74] Whitesides, G. and Krishnamurthy, V. Designing ligands to bind proteins. *Quarterly Reviews of Biophysics* **38**(4), 385–395 (2005). 20

[75] Chang, C.-e. A., Chen, W., and Gilson, M. Ligand configurational entropy and protein binding. *Proceedings of the National Academy of Sciences of the United States of America* **104**(5), 1534–1539 (2007). 20

[76] Fogolari, F., Brigo, A., and Molinari, H. The poisson-boltzmann equation for biomolecular electrostatics: a tool for structural biology. *Journal of Molecular Recognition : JMR* **15**(6), 377–392 (2002). 20

[77] Moitessier, N., Englebienne, P., Lee, D., Lawandi, J., and Corbeil, C. Towards the development of universal, fast and highly accurate docking/scoring methods: a long way to go. *British Journal of Pharmacology* **153 Suppl 1**, S7–26 (2008). 20, 22

[78] Böhm, H. The development of a simple empirical scoring function to estimate the binding constant for a protein-ligand complex of known three-dimensional structure. *Journal of Computer-Aided Molecular Design* **8**(3), 243–256 (1994). 21

[79] Muegge, I. and Martin, Y. A general and fast scoring function for protein-ligand interactions: a simplified potential approach. *Journal of Medicinal Chemistry* **42**(5), 791–804 (1999). 21

[80] Gohlke, H., Hendlich, M., and Klebe, G. Knowledge-based scoring function to predict protein-ligand interactions. *Journal of Molecular Biology* **295**(2), 337–356 (2000). 22

[81] Bissantz, C., Folkers, G., and Rognan, D. Protein-based virtual screening of chemical databases. 1. evaluation of different docking/scoring combinations. *Journal of Medicinal Chemistry* **43**(25), 4759–4767 (2000). 22

[82] Ferrara, P., Gohlke, H., Price, D., Klebe, G., and Brooks, C. Assessing scoring functions for protein-ligand interactions. *Journal of Medicinal Chemistry* **47**(12), 3032–3047 (2004). 22

[83] Wang, R., Lu, Y., and Wang, S. Comparative evaluation of 11 scoring functions for molecular docking. *Journal of Medicinal Chemistry* **46**(12), 2287–2303 (2003). 22

[84] Alonso, H., Bliznyuk, A., and Gready, J. Combining docking and molecular dynamic simulations in drug design. *Medicinal Research Reviews* **26**(5), 531–568 (2006). 22

[85] Warren, G., Andrews, C., Capelli, A.-M., Clarke, B., LaLonde, J., Lambert, M., Lindvall, M., Nevins, N., Semus, S., Senger, S., Tedesco, G., Wall, I., Woolven, J., Peishoff, C., and Head, M. A critical assessment of docking programs and scoring functions. *Journal of Medicinal Chemistry* **49**(20), 5912–5931 (2006). 22

[86] Rester, U. From virtuality to reality – virtual screening in lead discovery and lead optimization: a medicinal chemistry perspective. *Current Opinion in Drug Discovery & Development* **11**(4), 559–568 (2008). 22

[87] Leach, A. R. and Kuntz, I. D. Conformational analysis of flexible ligands in macremolecular receptor sites. *Journal of Computational Chemistry* **13**(6), 730–748 (1992). 22

[88] Lorber, D. and Shoichet, B. Flexible ligand docking using conformational ensembles. *Protein Science : a Publication of the Protein Society* **7**(4), 938–950 (1998). 23

[89] Ewing, T., Makino, S., Skillman, A., and Kuntz, I. Dock 4.0: search strategies for automated molecular docking of flexible molecule databases. *Journal of Computer-Aided Molecular Design* **15**(5), 411–428 (2001). 23

[90] Cornell, W., Cieplak, P., Bayly, C., Gould, I., Merz, K., Ferguson, D., Spellmeyer, D., Fox, T., Caldwell, J., and Kollman, P. A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. *Journal of the American Chemical Society* **117**(19), 5179–5197, May (1995). 24

[91] Welch, W., Ruppert, J., and Jain, A. Hammerhead: fast, fully automated docking of flexible ligands to protein binding sites. *Chemistry & Biology* **3**(6), 449–462 (1996). 24

[92] Jain, A. Scoring noncovalent protein-ligand interactions: a continuous differentiable function tuned to compute binding affinities. *Journal of Computer-Aided Molecular Design* **10**(5), 427–440 (1996). 24

[93] Zsoldos, Z., Reid, D., Simon, A., Sadjad, B., and Johnson, A. ehits: an innovative approach to the docking and scoring function problems. *Current Protein & Peptide Science* **7**(5), 421–435 (2006). 25

[94] Jain, A. Surflex: fully automatic flexible molecular docking using a molecular similarity-based search engine. *Journal of Medicinal Chemistry* **46**(4), 499–511 (2003). 25

[95] Ruppert, J., Welch, W., and Jain, A. Automatic identification and representation of protein binding sites for molecular docking. *Protein Science : a Publication of the Protein Society* **6**(3), 524–533 (1997). 25

[96] The cambridge structural database. `http://www.ccdc.cam.ac.uk/products/csd`. [Online; accessed 09-Januar-2010]. 26

[97] Developmental therapeutics program (nci/nih). `http://dtp.nci.nih.gov`. [Online; accessed 09-Januar-2010]. 26

[98] Schnecke, V. and Kuhn, L. Virtual screening with solvation and ligand-induced complementarity. *Perspectives in Drug Discovery and Design* **20**(1), 171–190 (2000). 26

[99] Miller, M., Kearsley, S., Underwood, D., and Sheridan, R. Flog: a system to select quasi-flexible ligands complementary to a receptor of known three-dimensional structure. *Journal of Computer-Aided Molecular Design* **8**(2), 153–174 (1994). 26

[100] Gabb, H., Jackson, R., and Sternberg, M. Modelling protein docking using shape complementarity, electrostatics and biochemical information. *Journal of Molecular Biology* **272**(1), 106–120 (1997). 27

[101] McGann, M., Almond, H., Nicholls, A., Grant, J., and Brown, F. Gaussian docking functions. *Biopolymers* **68**(1), 76–90 (2003). 27

[102] Connolly, M. Solvent-accessible surfaces of proteins and nucleic acids. *Science (New York, N.Y.)* **221**(4612), 709–713 (1983). 27

[103] Zauhar, R., Moyna, G., Tian, L., Li, Z., and Welsh, W. Shape signatures: a new approach to computer-aided ligand- and receptor-based drug design. *Journal of Medicinal Chemistry* **46**(26), 5674–5690 (2003). 27

[104] Joseph-McCarthy, D., Thomas, B., Belmarsh, M., Moustakas, D., and Alvarez, J. Pharmacophore-based molecular docking to account for ligand flexibility. *Proteins* **51**(2), 172–188 (2003). 28

[105] Joseph-McCarthy, D. and Alvarez, J. Automated generation of of mcss-derived pharmacophoric dock site points for searching multiconformation databases. *Proteins* **51**(2), 189–202 (2003). 28

[106] Abagyan, R., Totrov, M., and Kuznetsov, D. Icm – a new method for protein modeling and design – applications to docking and structure prediction from the distorted native conformation. *Journal Of Computational Chemistry* **15**(5), 488–506 (1994). 29

[107] McMartin, C. and Bohacek, R. Qxp: powerful, rapid computer algorithms for structure-based drug design. *Journal of Computer-Aided Molecular Design* **11**(4), 333–344 (1997). 29

[108] Jones, G., Willett, P., and Glen, R. Molecular recognition of receptor sites using a genetic algorithm with a description of desolvation. *Journal of Molecular Biology* **245**(1), 43–53 (1995). 29

[109] Jones, G., Willett, P., Glen, R., Leach, A., and Taylor, R. Development and validation of a genetic algorithm for flexible docking. *Journal of Molecular Biology* **267**(3), 727–748 (1997). 30

[110] Eldridge, M., Murray, C., Auton, T., Paolini, G., and Mee, R. Empirical scoring functions: I. the development of a fast empirical scoring function to estimate the binding affinity of ligands in receptor complexes. *Journal of Computer-Aided Molecular Design* **11**(5), 425–445 (1997). 30

[111] Verdonk, M., Cole, J., Hartshorn, M., Murray, C., and Taylor, R. Improved protein-ligand docking using gold. *Proteins* **52**(4), 609–623 (2003). 30

[112] Garrett M. Morris, D. Automated docking using a lamarckian genetic algorithm and an empirical binding free energy function. *Journal of Computational Chemistry* **19**(14), 1639–1662 (1998). 30

[113] Storn, R. and Price, K. Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical report, International Computer Science Institute: Berkley, (1995). 30

[114] Gehlhaar, D., Verkhivker, G., Rejto, P., Sherman, C., Fogel, D., Fogel, L., and Freer, S. Molecular recognition of the inhibitor ag-1343 by hiv-1 protease: conformationally flexible docking by evolutionary programming. *Chemistry & Biology* **2**(5), 317–324 (1995). 30

[115] Thomsen, R. and Christensen, M. Moldock: a new technique for high-accuracy molecular docking. *Journal of Medicinal Chemistry* **49**(11), 3315–3321 (2006). 30

[116] Baxter, C., Murray, C., Clark, D., Westhead, D., and Eldridge, M. Flexible docking using tabu search and an empirical estimate of binding affinity. *Proteins* **33**(3), 367–382 (1998). 31

[117] Dorigo, M. and Stuetzle, T. *Ant colony optimization.* MIT Press, Cambridge, MA, USA, (2004). 31

[118] Korb, O., Stutzle, T., and Exner, T. Plants: Application of ant colony optimization to structure-based drug design. *Swarm Intelligence* **1**(2), 115–134 (2007). 31

[119] Korb, O., Stutzle, T., and Exner, T. Empirical scoring functions for advanced protein-ligand docking with plants. *Journal of Chemical Information and Modeling* **49**(1), 84–96 (2009). 31

[120] Floriano, W., Vaidehi, N., Zamanakos, G., and Goddard, W. r. Hiervls hierarchical docking protocol for virtual ligand screening of large-molecule databases. *Journal of Medicinal Chemistry* **47**(1), 56–71 (2004). 32

[121] Friesner, R., Banks, J., Murphy, R., Halgren, T., Klicic, J., Mainz, D., Repasky, M., Knoll, E., Shelley, M., Perry, J., Shaw, D., Francis, P., and Shenkin, P. Glide: a new approach for rapid, accurate docking and scoring. 1. method and assessment of docking accuracy. *Journal of Medicinal Chemistry* **47**(7), 1739–1749 (2004). 33

[122] Guner, O. *Pharmacophore Perception, Development, and Use in Drug Design.* International University Line, La Jolla, CA, (2000). 33

[123] Langer, T., Hoffmann, R., and Mannhold, R. *Pharmacophores and Pharmacophore Searches.* Wiley-VCH, Weinheim, Germany, (2006). 33

[124] Guner, O. *Pharmacophore Perception, Development, and Use in Drug Design*, chapter Gasp: genetic algorithm superimposition program. International University Line, La Jolla, CA (2000). 33

[125] Cottrell, S., Gillet, V., Taylor, R., and Wilton, D. Generation of multiple pharmacophore hypotheses using multiobjective optimisation techniques. *Journal of Computer-Aided Molecular Design* **18**(11), 665–682 (2004). 33

[126] Richmond, N., Willett, P., and Clark, R. Alignment of three-dimensional molecules using an image recognition algorithm. *Journal of Molecular Graphics & Modelling* **23**(2), 199–209 (2004). 33

[127] Discovery Studio, 2008, Version 2.1, Accelrys: San Diego, CA, U.S.A. 34

[128] Wolber, G. and Langer, T. Ligandscout: 3-d pharmacophores derived from protein-bound ligands and their use as virtual screening filters. *Journal of Chemical Information and Modeling* **45**(1), 160–169 (2005). 34

[129] Dixon, S., Smondyrev, A., Knoll, E., Rao, S., Shaw, D., and Friesner, R. Phase: a new engine for pharmacophore perception, 3d qsar model development, and 3d database screening: 1. methodology and preliminary results. *Journal of Computer-Aided Molecular Design* **20**(10-11), 647–671 (2006). 34

[130] Unity, 2008, Version 8, Tripos Inc.: St. Louis, MO, U.S.A. 34

[131] Ebalunode, J., Ouyang, Z., Liang, J., and Zheng, W. Novel approach to structure-based pharmacophore search using computational geometry and shape matching techniques. *Journal of Chemical Information and Modeling* **48**(4), 889–901 (2008). 34

[132] OEShape Toolkit, 2006, Version 1.6, OpenEye Scientific Software: Santa Fe, NM, U.S.A. 34

[133] Wolber, G., Seidel, T., Bendix, F., and Langer, T. Molecule-pharmacophore superpositioning and pattern matching in computational drug design. *Drug Discovery Today* **13**(1-2), 23–29 (2008). 34

[134] Gund, P., Wipke, W., and Langridge, R. Computer searching of a molecular structure file for pharmacophoric patterns. *Proceedings of the International Conference on Computers in Chemical Research and Education* **3**, 33–38 (1974). 34

[135] Gasteiger, J., Rudolph, C., and Sadowski, J. Automatic generation of 3d-atomic coordinates for organic molecules. *Tetrahedron Computer Methodology* **3**(6, Part 3), 537–547 (1990). 34

[136] Perlman, R. Computer searching of a molecular structure file for pharmacophoric patterns. *Rapid generation of high quality approximate 3D molecular structures.* **2**, 5–7 (1987). 34

[137] Gunther, S., Senger, C., Michalsky, E., Goede, A., and Preissner, R. Representation of target-bound drugs by computed conformers: implications for conformational libraries. *BMC Bioinformatics* **7**, 293 (2006). 34

[138] Perola, E. and Charifson, P. Conformational analysis of drug-like molecules bound to proteins: an extensive study of ligand reorganization upon binding. *Journal of Medicinal Chemistry* **47**(10), 2499–2510 (2004). 34

[139] Wang, Q. and Pang, Y.-P. Preference of small molecules for local minimum conformations when binding to proteins. *PLoS ONE* **2**(9), e820 (2007). 34

[140] OMEGA, 2009, OpenEye Scientific Software, Inc.: Santa Fe, NM, U.S.A. 34

[141] CATALYST, 2009, Accelrys: San Diego, CA, U.S.A. 34

[142] ROTATE, 2009, Molecular Networks: Erlangen, Germany,. 34

[143] Griewel, A., Kayser, O., Schlosser, J., and Rarey, M. Conformational sampling for large-scale virtual screening: Accuracy versus ensemble size. *Journal of Chemical Information and Modeling* **49**, 2303 (2009). 34

[144] A.R., L. and A., S. A combined model-building and distance-geometry approach to automated conformational analysis and search. *Journal of Chemical Information and Computer Sciences* **32**(4), 379–385 (1992). 35

[145] Smellie, A., Kahn, S., and Teig, S. Analysis of conformational coverage. 2. applications of conformational models. *Journal of Chemical Information and Modeling* **35**(2), 295–304 (1995). 35

[146] Smellie, A., Kahn, S., and Teig, S. Analysis of conformational coverage. 1. validation and estimation of coverage. *Journal of Chemical Information and Modeling* **35**(2), 285–294 (1995). 35

[147] Smellie, A., Teig, S., and Towbin, P. Poling: Promoting conformational variation. *Journal of Computational Chemistry* **16**(2), 171–187 (1995). 35

[148] Bernstein, F., Koetzle, T., Williams, G., Meyer, E., Brice, M., Rodgers, J., Kennard, O., Shimanouchi, T., and Tasumi, M. The protein data bank: a computer-based archival file for macromolecular structures. *Journal of Molecular Biology* **112**(3), 535–542 (1977). 35

[149] Najmanovich, R., Kuttner, J., Sobolev, V., and Edelman, M. Side-chain flexibility in proteins upon ligand binding. *Proteins* **39**(3), 261–268 (2000). 35

[150] Bayer, R. and McCreight, E. "organization and maintenance of large ordered indexes". *Acta Informatica* **1**(3), 173–189 (1972). 37

[151] L., B. J. Mulidimensional binary search trees. *Communications of the ACM* **18**(9), 509–517 (1975). 37

[152] Ramakrishnan, R. *Database Management Systems.* WCB/McGraw-Hill, (1998). 38

[153] O'Neil, P. E. and Quass, D. Improved query performance with variant indexes. In *SIGMOD Conference*, 38–49, (1997). 38

[154] Robinson, J. T. The k-d-b-tree: a search structure for large multidimensional dynamic indexes. In *SIGMOD '81: Proceedings of the 1981 ACM SIGMOD international conference on Management of data*, 10–18. ACM, (1981). 38

[155] Guttman, A. R-trees: A dynamic index structure for spatial searching. In SIGMOD'84, Proceedings of Annual Meeting, Boston, Massachusetts, June 18-21, 1984, Yormark, B., editor, 47–57. ACM Press, (1984). 38

[156] Beckmann, N., Kriegel, H.-P., Schneider, R., and Seeger, B. The r*-tree: An efficient and robust access method for points and rectangles. In *SIGMOD Conference*, 322–331, (1990). 38

[157] Berchtold, S., Keim, D. A., and Kriegel, H.-P. The x-tree : An index structure for high-dimensional data. In VLDB'96, Proceedings of 22th International Conference on Very Large Data Bases, September 3-6, 1996, Mumbai (Bombay), India, Vijayaraman, T. M., Buchmann, A. P., Mohan, C., and Sarda, N. L., editors, 28–39. Morgan Kaufmann, (1996). 38

[158] Nievergelt, J., Hinterberger, H., and Sevcik, K. C. The grid file: An adaptable, symmetric multikey file structure. *ACM Transactions on Database Systems* **9**(1), 38–71 (1984). 39

[159] Hinrichs, K. Implementation of the grid file: Design concepts and experience. *BIT* **25**(4), 569–592 (1985). 39

[160] O'Neil, P. E. Model 204 architecture and performance. In *HPTS*, 40–59, (1987). 39

[161] Chan, C. Y. and Ioannidis, Y. E. Bitmap index design and evaluation. In *SIGMOD Conference*, 355–366, (1998). 39

[162] Klebe, G. The use of composite crystal-field environments in molecular recognition and the de novo design of protein ligands. *Journal of Molecular Biology* **237**(2), 212–235 (1994). 42

[163] Rarey, M., Kramer, B., and Lengauer, T. Multiple automatic base selection: protein-ligand docking based on incremental construction without manual intervention. *Journal of Computer-Aided Molecular Design* **11**(4), 369–384 (1997). 43

[164] Linnainmaa, S., Harwood, D., and Davis, L. Pose determination of a three-dimensional object using triangle pairs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **10**(5), 634–647 (1988). 43

[165] Rarey, M., Wefing, S., and Lengauer, T. Placement of medium-sized molecular fragments into active sites of proteins. *Journal of Computer-Aided Molecular Design* **10**(1), 41–54 (1996). 43

[166] Kesheng, W. Fastbit: an efficient indexing technology for accelerating data-intensive science. *Journal of Physics: Conference Series* **16**, 556 (2005). 47

[167] Wu, K., Otoo, E. J., and Shoshani, A. Compressing bitmap indexes for faster search operations. In *SSDBM'02, Proceedings of 14th International Conference on Scientific and Statistical Database Management*, 99, (2002). 48

[168] Wu, K., Otoo, E., and Shoshani, A. Nordberg, H. Notes on design and implementation of compressed bit vectors. Technical Report LBNL/PUB-3161, Lawrence Berkeley National Laboratory, (2001). 48

[169] Wu, K., Otoo, E., and Shoshani, A. An efficient compression scheme for bitmap indices. Technical Report LBNL-49626, Lawrence Berkeley National Laboratory, (2004). 48

[170] sqlite.org. Sqlite3, a self-contained, serverless, zero-configuration, transactional sql database engine. `http://www.sqlite.org`. [Online; accessed 09-Januar-2010]. 49

[171] Klebe, G. and Mietzner, T. A fast and efficient method to generate biologically relevant conformations. *Journal of Computer-Aided Molecular Design* **8**(5), 583–606 (1994). 68

[172] Clark, M., Cramer III, R. D., and van Opdenbosch, N. Validation of the general purpose tripos 5.2 force field. *Journal of Computational Chemistry* **10**(8), 982–1012 (1989). 69

[173] Leach, A. R. and Prout, K. Automated conformational analysis: Directed conformational search using the a* algorithm. *Journal of Computational Chemistry* **11**(10), 1193–1205 (1990). 69

[174] Hindle, S., Rarey, M., Buning, C., and Lengauer, T. Flexible docking under pharmacophore type constraints. *Journal of Computer-Aided Molecular Design* **16**(2), 129–149 (2002). 78, 104

[175] Stonebraker, M. The case for shared nothing. *IEEE Database Eng. Bull.* **9**(1), 4–9 (1986). 82

[176] Sun grid engine. http://gridengine.sunsource.net. [Online; accessed 09-Januar-2010]. 83

[177] Vieth, M., Hirst, J. D., Kolinski, A., and C.L., B. Assessing energy functions for flexible docking. *Journal of Computational Chemistry* **19**, 1612–1622 (1989). 86

[178] Kramer, B., Rarey, M., and Lengauer, T. Evaluation of the flexx incremental construction algorithm for protein-ligand docking. *Proteins* **37**(2), 228–241 (1999). 86

[179] Hartshorn, M., Verdonk, M., Chessari, G., Brewerton, S., Mooij, W., Mortenson, P., and Murray, C. Diverse, high-quality test set for the validation of protein-ligand docking performance. *Journal of Medicinal Chemistry* **50**(4), 726–741 (2007). 86, 90

[180] Nicholls, A. What do we know and when do we know it? *Journal of Computer-Aided Molecular Design* **22**(3-4), 239–255 (2008). 86

[181] Allen, F. H., Davies, J. E., Galloy, J. J., Johnson, O., Kennard, O., Macrae, C. F., Mitchell, E. M., Mitchell, G. F., Smith, J. M., and Watson, D. G. The development of versions 3 and 4 of the cambridge structural database system. *Journal of Chemical Information and Modeling* **31**(2), 187–204 (2002). 87

[182] Maass, P., Schulz-Gasch, T., Stahl, M., and Rarey, M. Recore: a fast and versatile method for scaffold hopping based on small molecule crystal structure conformations. *Journal of Chemical Information and Modeling* **47**(2), 390–399 (2007). 87

[183] Kirchmair, J., Wolber, G., Laggner, C., and Langer, T. Comparative performance assessment of the conformational model generators omega and catalyst: a large-scale survey on the retrieval of protein-bound ligand conformations. *Journal of Chemical Information and Modeling* **46**(4), 1848–1861 (2006). 87, 90

[184] YASARA Version 8, 2008, YASARA Biosciences: 8042 Graz, Austria 2008. 94

[185] Duan, Y., Wu, C., Chowdhury, S., Lee, M., Xiong, G., Zhang, W., Yang, R., Cieplak, P., Luo, R., Lee, T., Caldwell, J., Wang, J., and Kollman, P. A point-charge force field for molecular mechanics simulations of proteins based on condensed-phase quantum mechanical calculations. *Journal of Computational Chemistry* **24**(16), 1999–2012 (2003). 94

[186] ShoichetLaboratory. Dud a directory of useful decoys. http://dud.docking.org. [Online; accessed 09-Januar-2010]. 95

[187] Cross, J., Thompson, D., Rai, B., Baber, J., Fan, K., Hu, Y., and Humblet, C. Comparison of several molecular docking programs: Pose prediction and virtual screening accuracy. *Journal of Chemical Information and Modeling* **49**(6), 1455–1474 (2009). 95, 99

[188] Irwin, J. and Shoichet, B. Zinc–a free database of commercially available compounds for virtual screening. *Journal of Chemical Information and Modeling* **45**(1), 177–182 (2005). 96

[189] Stahl, M., Todorov, N., James, T., Mauser, H., Böhm, H., and Dean, P. A validation study on the practical use of automated de novo design. *J Comput -Aided Mol Des* **16**(7), 459–478 (2002). 104

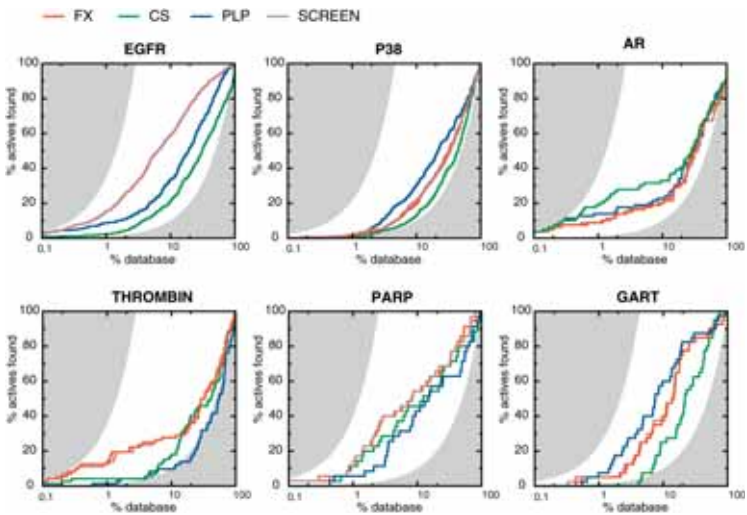
[190] c.a.r.u.s. Visor (virtual screening optimizing the reality. http://www.carus-it.com/de/portfolio/life-science/carus-visor.html. [Online; accessed 09-Januar-2010]. 121

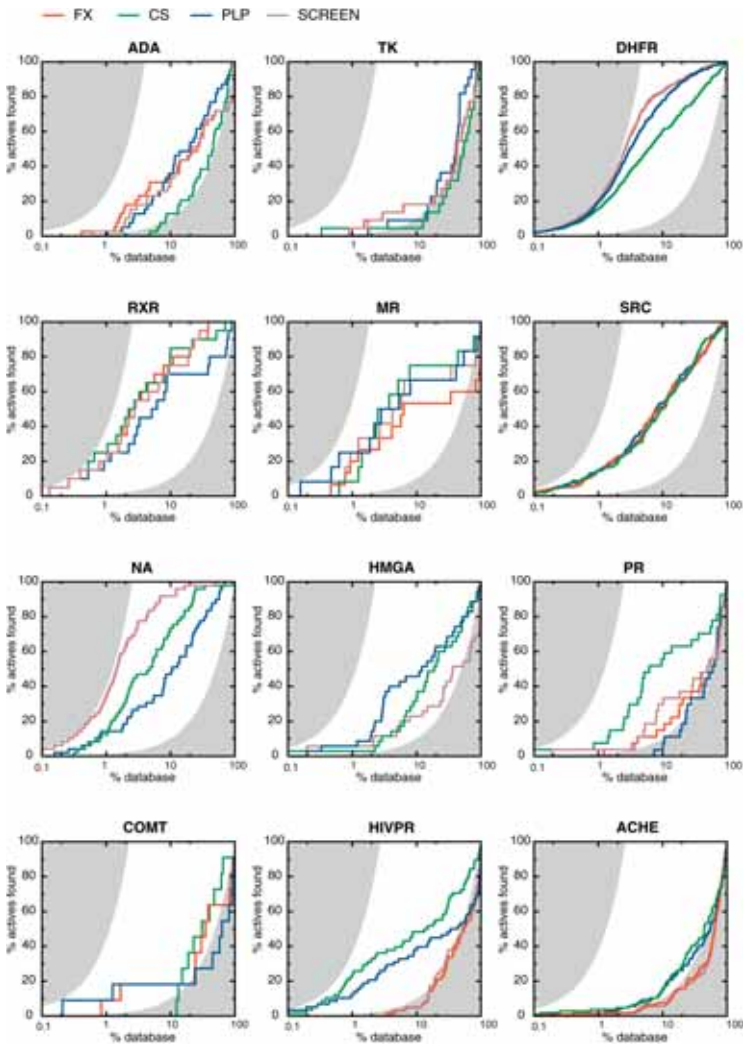
Appendix

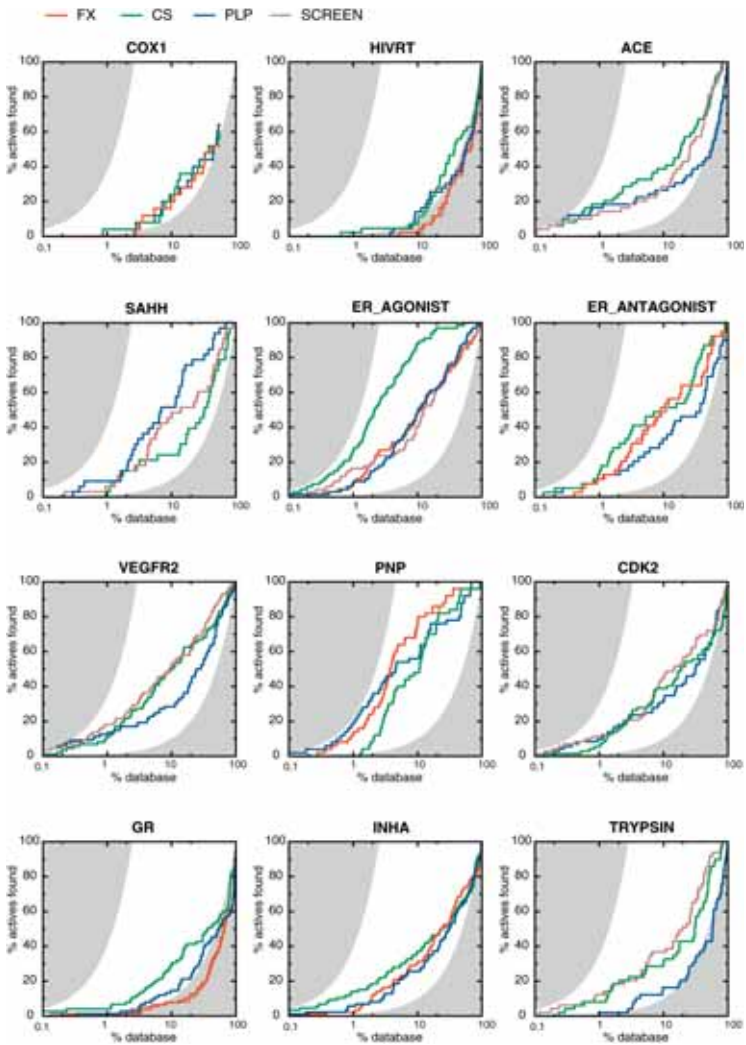
A
Results

A.1. Enrichment Experiments

On the next pages, 40 individual enrichment plots of the experiments on the DUD are illustrated. This includes results using four different scoring options: FlexX score (FX), ChemScore (CS), PLP Score (PLP), and ScreenScore (SCREEN).




FX
CS
PLP
SCREEN
ADA
TK
DHFR
RXR
MR
SRC
NA
HMGA
PR
COMT
HIVPR
ACHE
% actives found
% database


FX
CS
PLP
SCREEN
COX1
HIVRT
ACE
SAHH
ER_AGONIST
ER_ANTAGONIST
VEGFR2
PNP
CDK2
GR
INHA
TRYPSIN
% actives found
% database

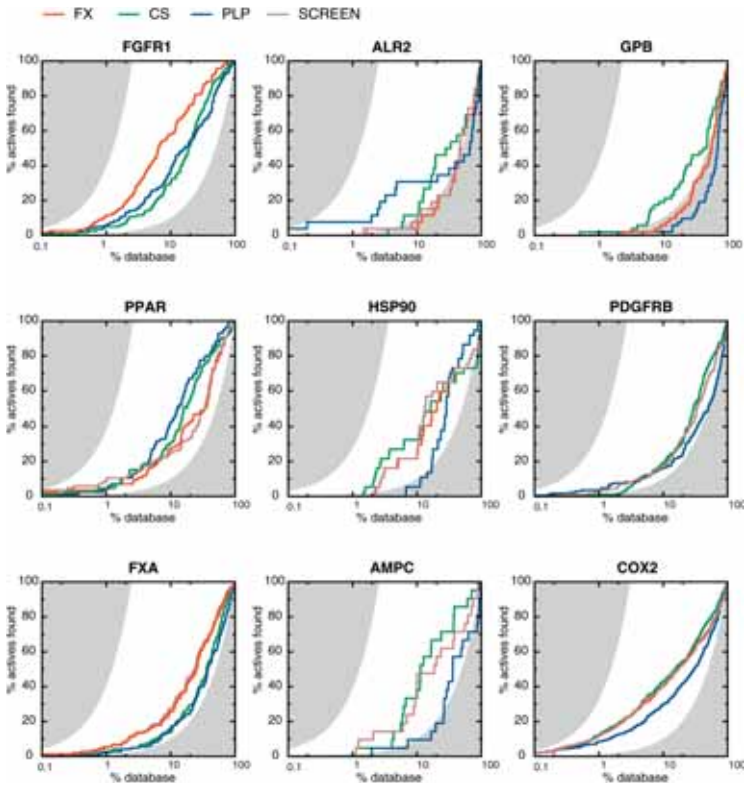


Figure A.1.: Invividual enrichment plots for all 40 targets of the DUD data set.

A.2. Redocking Experiments

Table A.1 presents detailed redocking results for the Astex Diverse Set based on TCG ensembles of quality level one and a clustering threshold of 1.2 Å. Additionally, Table A.2 supplies summary data for quality levels one, two, three, four, and five.

	RMSD[Å] ≤					
PDB ID	0.5	1.0	1.5	2.0	2.5	3.0
1g9v	5.793	5.793	4.893	3.659	3.112	3.112
1gkc	3.586	3.257	2.296	2.296	2.296	1.843
1gm8	3.042	3.042	2.901	2.237	2.237	1.321
1gpk	1.355	1.355	1.355	1.355	0.796	0.796
1hnn	5.491	1.392	1.392	1.392	1.392	1.392
1hp0	1.949	0.733	0.733	0.607	0.607	0.607
1hq2	0.616	0.616	0.541	0.429	0.429	0.429
1hvy	11.162	2.778	2.296	2.140	1.439	1.439
1hwi	1.504	1.334	1.282	1.282	1.282	1.282
1hww	0.429	0.429	0.429	0.429	0.429	0.429
1ia1	1.490	0.985	0.681	0.681	0.681	0.681
1ig3	0.942	0.931	0.717	0.717	0.717	0.717
1j3j	0.376	0.317	0.317	0.317	0.317	0.317
1jd0	2.254	1.904	1.904	1.724	1.618	1.618
1jje	7.910	1.403	1.070	1.070	1.070	1.070
1jla	6.006	1.941	1.514	1.514	1.514	1.514
1k3u	1.594	1.485	1.485	1.485	1.485	1.485
1ke5	2.429	2.200	2.200	2.200	2.200	2.200
1kzk	9.270	8.787	8.787	8.787	8.787	8.787
1l2s	2.316	1.447	1.447	1.056	1.056	1.056
1l7f	0.613	0.613	0.425	0.425	0.425	0.425
1lpz	7.720	2.280	2.280	2.280	2.280	2.280
1lrh	4.370	3.578	3.578	3.578	3.578	3.578
1m2z	0.818	0.691	0.691	0.691	0.691	0.691
1meh	5.983	4.071	3.823	3.823	3.045	2.051
1mmv	4.155	3.827	3.827	2.911	2.888	2.629
1mzc	4.973	4.842	4.787	4.787	2.351	2.351
1n1m	0.840	0.598	0.598	0.598	0.577	0.577
1n2j	3.519	2.161	1.932	1.644	0.650	0.650
1n2v	2.446	2.412	2.412	1.275	1.147	1.147
1n46	1.377	1.377	1.377	1.377	1.377	1.377
1nav	14.948	14.948	14.948	14.948	14.948	14.948
1of1	0.731	0.731	0.731	0.731	0.731	0.731
1of6	0.803	0.803	0.803	0.710	0.710	0.686

	RMSD[Å] ≤					
PDB ID	0.5	1.0	1.5	2.0	2.5	3.0
1opk	2.711	1.279	1.279	1.279	1.279	1.279
1oq5	5.043	4.570	3.419	3.419	2.624	2.624
1owe	2.189	1.719	1.391	1.135	1.135	1.135
1oyt	1.583	1.454	1.454	1.454	1.159	1.159
1p2y	5.001	4.606	4.216	4.137	3.985	3.444
1p62	1.474	0.664	0.486	0.486	0.486	0.486
1pmn	2.273	1.922	1.922	1.810	1.810	1.810
1q1g	5.673	1.094	1.094	1.094	1.094	1.094
1q41	0.600	0.600	0.600	0.600	0.600	0.600
1q4g	0.775	0.775	0.775	0.775	0.775	0.775
1r1h	1.047	0.962	0.962	0.962	0.962	0.962
1r55	2.564	2.558	2.558	2.169	1.986	1.774
1r58	2.187	1.635	1.341	1.341	1.189	1.189
1r9o	0.718	0.718	0.718	0.665	0.665	0.665
1s19	1.686	1.634	1.634	1.588	1.588	1.588
1s3v	5.135	0.982	0.982	0.982	0.982	0.982
1sg0	8.029	7.895	1.481	0.972	0.972	0.972
1sj0	4.682	4.682	4.666	3.841	2.879	2.643
1sq5	5.284	5.284	3.937	3.845	1.456	1.456
1sqn	0.850	0.850	0.850	0.850	0.850	0.850
1t40	2.893	2.773	2.773	2.773	2.773	2.252
1t46	6.081	6.081	6.081	6.081	6.081	6.081
1t9b	9.084	4.875	4.269	3.765	2.648	2.029
1tow	3.736	1.620	1.620	1.401	1.401	1.401
1tt1	1.306	1.306	1.306	1.306	1.306	1.187
1tz8	1.104	1.104	1.104	1.104	1.104	1.104
1u1c	1.956	1.041	1.041	0.998	0.998	0.998
1u4d	3.788	1.503	1.503	1.503	1.503	1.142
1uml	2.116	1.811	1.811	1.811	1.804	1.804
1unl	6.262	1.556	1.556	1.556	1.556	1.556
1uou	1.579	1.400	1.400	1.400	1.303	1.303
1v0p	1.138	1.138	1.138	1.138	1.138	1.138
1v48	1.764	1.185	1.185	1.185	1.185	1.185
1v4s	7.514	5.633	1.867	1.867	1.867	1.867
1vcj	1.878	1.653	1.653	1.589	1.589	1.589
1w1p	0.445	0.445	0.445	0.445	0.445	0.445
1w2g	1.388	1.388	1.264	0.973	0.973	0.973
1x8x	6.675	5.507	3.040	0.837	0.837	0.837
1xm6	3.607	1.601	1.601	1.601	1.526	1.225
1xoq	2.827	2.731	2.409	2.380	1.897	1.801

	RMSD[Å] ≤					
PDB ID	0.5	1.0	1.5	2.0	2.5	3.0
1xoz	7.473	5.950	5.248	5.248	5.248	5.248
1y6b	7.753	2.237	2.237	2.237	2.237	1.639
1ygc	1.318	1.303	1.303	1.303	1.303	1.303
1yqy	1.541	0.979	0.979	0.979	0.979	0.979
1yv3	1.033	1.033	1.033	1.033	1.033	1.033
1yvf	3.603	3.097	3.097	3.097	3.071	2.551
1ywr	1.554	1.554	1.554	1.554	1.554	1.554
1z95	1.252	1.252	1.252	1.252	1.252	1.252
2bm2	3.815	2.340	2.340	1.905	1.905	1.905
2br1	1.678	1.678	1.583	1.569	1.569	1.516
2bsm	0.790	0.790	0.786	0.786	0.786	0.786

Table A.1.: Detailed TrixX BMI redocking results of the 85 protein-ligand complexes in the Astex Diverse Set based on TCG ensemble using default setting of quality level one and clustering threshold 1.2 Å.

		RMSD[Å] ≤		
Best(x)	quality	1.0	2.0	3.0
1	crystal	49	63	70
1	1	15	39	51
1	2	16	40	50
1	3	19	41	54
1	4	14	44	53
1	5	14	46	50
5	crystal	55	70	74
5	1	21	55	65
5	2	19	53	65
5	3	22	51	65
5	4	21	57	62
5	5	21	57	61
10	crystal	58	71	76
10	1	21	58	69
10	2	19	59	70
10	3	22	60	71
10	4	22	62	68
10	5	23	60	67

		RMSD[Å] ≤		
Best(x)	quality	1.0	2.0	3.0
20	crystal	58	74	78
20	1	25	61	71
20	2	25	62	71
20	3	26	68	72
20	4	26	65	72
20	5	28	65	74
50	crystal	58	75	78
50	1	27	65	76
50	2	24	65	75
50	3	28	69	76
50	4	27	73	75
50	5	29	68	76
200	crystal	58	75	78
200	1	27	68	78
200	2	24	68	79
200	3	28	73	79
200	4	27	76	78
200	5	29	72	78

Table A.2.: Redocking results of TrixX BMI on the Astex Diverse Set when looking at the x best scored pose predictions using TCG ensembles of different five different qualities.

B

Developer & User Information

In the first section, the implementation of TrixX BMI is summarized. This involves individual modules and their tasks as well as a diagram that illustrates their usage. The second section focuses on user information and on how to use TrixX BMI for VS.

B.1. Developer Information

Module	File name	Description
trixxBmiMenu	menu_screenin	Menu command of TrixX BMI. All TrixX BMI commands except pharmacophore and receptor specific routines are called from here. This includes the TrixX BMI sampling routine which includes compound fragmentation.
dataHandling	flex_admin	Routines for data management and initialization.
descriptor	screening_data	Module holding generic descriptor functionality which is not specific for site- and compound descriptors.
	decode	Maps descriptor type attributes to triplets of interaction centers.
siteDescr	sitequery	Implements routines specific for site descriptors.
	pharmacophore	FlexX implementation of pharmacophore type constraints augmented with TrixX BMI specific routines for descriptor subselection.
compDescr	catalog	Implements routines specific for compound descriptors.

Module	File name	Description
	base_select	FlexX fragmentation module, extended with TrixX BMI functionality.
compoundIdx	trixx_idx	Implements a facade to the the TrixX BMI compound index. It hides the internal partitions. Cataloging and querying both access the compound index using this module.
	trixx_idx_part	Holds all algorithms that are executed inside of internal partitions, e.g. appending descriptors, deleting descriptors, and generating Bitmap Indices.
	trixx_properties	Encapsulates the property filter of TrixX BMI.
	interface_fastbit	Provides the interface to FastBit and transforms descriptors into queries. This mechanism is already abstracted and allows the incorporation of different query transformations, e.g. for ligand-based VS. Matches are stored using the *dataStorage* module.
dataStorage	trixx_sql	Interface to the SQLite storage system. Prepares and binds statements, performs transaction processing and regulates cache usage.
dockingEngine	trixx_placement	Implements the hierarchical docking pipeline. This includes access to FlexX internal routines for scoring and incremental construction using appropriate data structures (dock_entry, match_entry). This mechanism is also abstracted and allows the usage of different docking routines. Matches are extracted and results are stored using the *dataStorage* module.
	trixx_optimize	Interface to the internal optimization of FlexX.
visualization	trixx_display	Visualizes the resulting poses which are organized using the module *dataStorage*.
TcgMenu	menu_smacks	Menu command of TCG. Stand-alone sampling is called from here.
TcgSampling	smacks_milk	Setup routines for TCG sampling.
	smacks_crunch	Holds the TCG sampling algorithm.

Module	File name	Description
	smacks_cluster	TCG clustering module for online- and final clustering.

Table B.1.: TrixX BMI and TCG modules.

In Table B.1 the different modules and corresponding source files are explained in detail. First, the modules of TrixX BMI are presented, followed by the modules of the TCG. Access to the compound index is controlled by the *trixxIdx* module which is implemented as a facade to the indexing subsystem (see Figure B.1). Descriptor indexing and descriptor matching do not call any functions from within the subsystem. Thus, the implementation of the single components in the subsystem can easily be changed without affecting the entire system. The interdependencies of the individual modules is illustrated using arrows. The compound index (*compoundIdx*) is made up of descriptor- (*descrPart*) and type partitions (*typePart*). The type partitions have access to FastBit and can initiate the generation of Bitmap Indices and the matching of descriptor attributes. The resulting matches are stored and later forwarded to the *dockingEnginge* using the *dataStorage* modul that is based on SQLite.

B.2. User Guide

On the next pages, the user guide of TrixX BMI is presented. This includes the presentation of some FlexX commands necessary to intialize the protein target, read the pharmacophore, and adapt general settings of the program.

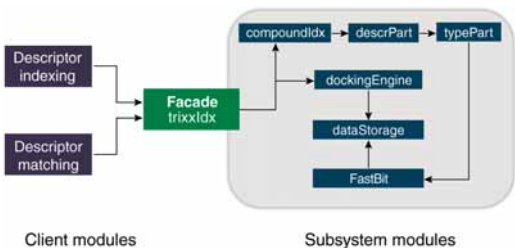


Figure B.1.: Encapsulation of the compound index of TrixX BMI.

TrixX BMI

Version 1.2.0

User Guide

Jochen Schlosser
Matthias Rarey

Center for Bioinformatics (ZBH)
University of Hamburg, 20146 Hamburg, Germany

BioSolveIT GmbH
An der Ziegelei 75, 53757 St. Augustin, Germany

Contents

1 Introduction

1.1 About TrixX BMI

TrixX BMI is a computer program tool for screening a virtual compound library against an active site of a protein. The active site of the protein has to be specified; the compound library also is provided by the user.
Sometimes additional information about a potential inhibitor or even the complex is known. You can integrate this knowledge in the screening process with TrixX BMI by creating a pharmacophore. This information will then be used when searching tlhe ligand space.
Before you start working with TrixX BMI, we would remind you that TrixX BMI is software under steady and current development. We do test the program with a continuously growing set of proteins and ligand spaces, but we are sure that TrixX BMI is not "error-free".

To understand and interpret the results produced with TrixX BMI, it is necessary to know something about the underlying models and algorithms. This topic is not covered in this User Guide.

TrixX BMI originates from research done at the group for Computational Molecular Design at the Center for Bioinformatics of the University of Hamburg, Germany. It is based on the *FlexX* docking program and makes use of its functionality and of the incremental construction algorithm in particular. In addition *FlexX-Pharm* functionality can be used. For details refer to the following literature [1, 2, 4, 6, 8, 11, 9, 10].

Further development of the TrixX BMI system is carried out at the Center for Bioinformatics of the University of Hamburg. The *FlexX* program is under steady development at BiosolveIT GmbH.

1.2 How to read this guide

The user interface of TrixX BMI is similar to those of the programs in the Flex* software suite. If you are familiar with the protein-ligand docking program *FlexX* [7] or with the feature-tree descriptor program *FTrees* [12] in particular, you should find it easy to learn to use TrixX BMI.

Most commands and file formats are self-explanatory, there is no need to go through the

whole User Guide in order to work with *TrixX BMI*. In this User Guide the focus is set on commands and menu options that are important when performing a virtual screening run with TrixX BMI, while other less important and basic *FlexX* commands are not explained in detail. However, a detailed description of these can be found in the *FlexX* manual [7].
We have used the following styles or fonts to highlight specific parts of the text. The most important style is the environment of examples, as follows:

```
Example
```

```
This is an example
```

The descriptions of commands and global parameters of TrixX BMI have a special list structure, which is self-explanatory. In the text, we use the following fonts: this is a `command`, this is a <parameter>, and this is a `filename`, a `path`, or a `program`.
A syntax description looks like this:

```
command <parameter> ...
```

Parameters which occur only in special cases or which are optional are set in parentheses: [<optional parameter>]. If the line ends with a \-character, the command line is continued in the next line. Note that in TrixX BMI itself it is not possible to escape a carriage return character by using a \-character.

1.3 Important program and documentation issues

As already mentioned earlier, TrixX BMI is based on the molecular docking program *FlexX*. Because of this, most of the basic installation, the configuration steps and requirements also hold for TrixX BMI. Therefore, the description of all the relevant *FlexX* program parameters, the configuration data, the installation issues and so on are not described but can be found in the *FlexX* User Guide [7].

1.4 Additional copyright notes

The following software/data can be used in/with *TrixX BMI*:

- **Base software**: Copyright ©2001 by Fraunhofer Gesellschaft (FhI-SCAI)
- **getline** library: Copyright ©1993 by Chris Thewalt
- **SMARTS™** may be a registered trademark of Daylight Chemical Information Systems.
- The torsion angle data (torsion_standard.dat) is derived from the Cambridge Structural Database. The copyright © of these file is shared by GMD – Forschungszentrum Informationstechnik GmbH, the Cambridge Crystallographic Data Center (CCDC), and BASF AG, Ludwigshafen.

2 Installation

2.1 Parts of *TrixX BMI*

After unpacking (`tar -xzvf <file>`) the *TrixX BMI* software package, you will have the following files:

Filename	Description
`bat/`	Batch script files
`bin/`	*FlexV* binary for architecture
`example/`	First examples of ligands, proteins, etc.
`Fastbit/`	Libraries for indexing
`pharm/`	Pharmacophore example
`static_data/`	Static data files of *TrixX BMI*
`test_idx/`	Standard *TrixX BMI* index directory
`trixx_data/`	Libraries and some shared data sources
`tmp/`	Standard *TrixX BMI* directory for temporary data

TrixX BMI is an executable. If it does not have the 'x' flag, set it with `chmod +x trixx`.

2.2 License scheme

Our software is license key protected. Please be aware that you cannot run *TrixX BMI* under any circumstances without a valid license. To obtain a valid license, please visit `http://www.biosolveit.de/license/`.
After receiving your license keys, you will need to edit the configuration file `config.dat`. Enter the path and name of the license files containing the license keys after the keyword `@license_files`.

2.3 Installing *TrixX BMI*

Various settings need to be set in the configuration file `config.dat`. The second entry in the configuration file is the root directory. All paths specified further on are relative to this path except those starting with `/` or `./`. You can define default paths to various types of data files in the `@directories` section. The `@static_data` section contains paths and filenames of the static data files of *TrixX BMI*, and the `@programs` section contains paths and filenames of executables.

For a first in the current directory you can write the current path into the definition of the root directory and just leave all the rest as it is. You can customize the configuration of *TrixX BMI* later on, also individually for each user.

2.4 A first simple test

For a first simple test, go to the directory where you installed *TrixX BMI* and type `./trixx`. After displaying a startup message, *TrixX BMI* will read the configuration file and the static data files.

Now you should see the *TrixX BMI* prompt (`TRIXX>`) that is waiting for your input. For explicitly testing the TrixX BMI software (as it is built into the *FlexX* framework, you have to go to a submenu to test TrixX BMI) type `screenin`, then you should see the prompt `TRIXX>SCREENIN>`. Type `quit` to terminate *TrixX BMI*.

2.5 Essential Libraries

2.5.1 Fastbit

Fastbit (used for the indexing system) is supplied in binary form. The libraries were compiled using GLIBCXX.3.4.9. If this version is not available on your system, please add them or alternatively recompile the Fastbit sources from `https://codeforge.lbl.gov/projects/fastbit` and copy the resulting libraries to `./trixx_data/library` as target directory.

2.6 External programs and data

Some features of *TrixX BMI* are based on external data and software. Although *TrixX BMI* can be used as a stand-alone program, we advise you to make the following facilities available to *TrixX BMI*.

2.6.1 Graphics

TrixX BMI has no internal graphics. For visualization, *TrixX BMI* must be coupled with an external program. Currently, interfaces to the following software are provided:
FlexV is an in-house visualization tool based on OpenGL. *FlexV* supports all graphic features of *TrixX BMI*.

2.6.2 Torsion angles

The static data file `torsion_standard.dat` contains energetically favorable torsion angles for specific molecular fragments.
The torsion data files in this software package have been derived by Gerhard Klebe [5] from the Cambridge Structural Database (CSD), licensed by the Cambridge Crystallographic Data

Centre (CCDC). The torsion data files are under copyright of GMD, BASF AG, and CCDC. An end-user license for the torsion data files is included in the *TrixX BMI* software license.

2.6.3 Conformer Generation

TrixX BMI uses preprocessed conformations to perform its docking calculations. In order to generate meaningfull conformers using the torsion driver TrixX Conformer Generator (TCG), it is vital to supply it with an external program which generates ring conformations and, if the molecules are supplied in SMILES format, also generates the initial 3d structures.

Flexible ring systems

The conformations of flexible ring systems can be computed by the 3D structure generator CORINA [3, 13]. Your CORINA version is suitable for use with *TrixX BMI*/TCG if the driver option 'flexx' is available (set the CORINA executable to your $path variable, then type 'corina -h d' to check). CORINA or CORINA-F can be obtained from Molecular Networks GmbH (see `http://www.mol-net.de` for detailed information). Alternatively, the program CONFORT can be used. (Please contact Tripos Inc. for more information on this.) If no ring conformation generator is available, the flag RING_MODE must be set to 0 in the used configuration file. Again, this is only important if you are generating your own conformations using TCG.

Conversion of SMILES strings

For the use of SMILES strings within *TrixX BMI*, an external program must be used that is capable of converting the string to a 3D representation of the corresponding ligand or fragment. This can be done by using, for example, CORINA, although a full version of the program is required to do this in contrast to the computation of ring conformations.

2.6.4 Fastbit

Fastbit is a bitmap indexing system suited for high-dimensional data retrieval. For more information concerning Fastbit see 2.5.1.

2.6.5 SQLite

SQLite is a software library that implements a self-contained, serverless, almost zero-configuration, transactional SQL database engine. If you use LD_LIBRARY_PATH in your system, please supply version 3.6.4 or above. The correct libraries are also supplied under `./trixx_data/library` which is part of the trixx runpath.

3

Getting started — a tutorial introduction

This tutorial section is meant to be an introduction to the possibilities of virtual screening with TrixX BMI. *TrixX BMI* is extremely flexible and configurable, and you will learn in the later sections where to tune to get your desired screening results. With the help of two example script files for *TrixX BMI* you first get to know the two-step process of the virtual screening program. Afterwards you will get into the more sophisticated section of applying the settings of the program for your own purpose.

3.1 Configuration

The most important thing to do before you start is to provide *TrixX BMI* with a config.dat file in the working directory. This file tells *TrixX BMI* the location of files that it needs at runtime and that are important for correct execution, e.g. template files for the receptor amino acids, helper programs such as a graphic visualizer (e.g. *FlexV*, our free 3D graphics program supplied with *TrixX BMI*), or a text editor of your choice.
Create a working directory where no valuable data can be destroyed and copy the example config.dat file (located in <trixx_installation_dir>/config.dat) into this directory. Open the config.dat file with a text editor of your choice. The file is separated into 9 subsections, each one marked with a '@'.Some oft hese have subitems, where you can for example specify the location of your ligand files:

```
@LICENSE_FILES
@ROOTDIR
@DIRECTORIES
-PHARM
-RECEPTOR
-PDB
-SURFACE
-SITE
-LIGAND
-INDICES
-SCRIPT
-HELP
-PREDICT
```

```
-TEMP
@STATIC_DATA
@PROGRAMS
@FLAGS
@ID_STRINGS
@ALIASES
```

These are explained in detail in section 4.1 of the *FlexX* User Guide [7]. In this tutorial, we only need to tinker with `@ROOTDIR` and the linking of a library; we assume you have already entered the information about the license file in `@LICENSE_FILES` (if you have any problems with this, please refer to the *FlexX* User Guide [7]).

In order for *TrixX BMI* to find the important files for error-free execution, the `@ROOTDIR` can point either to the directory where you installed the software or to your current working directory if the *TrixX BMI* binary is linked here. Let's assume that your installation directory is `/home/user`. In this case, `@ROOTDIR` can be `.` or `/home/user/trixx`. In both cases *TrixX BMI* needs to find the appropriate libraries, especially Fastbit which is most likely not part of your basic installation. Fastbit libraries can be found in your installation folder under `trixx_data/library`. In order to find these, *TrixX BMI* needs a soft link named `trixx_data` in the installation folder (an alternative way is to set an appropriate environment variable).

TrixX BMI should now find the location of `static_data`, `objects`, necessary libraries, etc.

In this tutorial we use *FlexV* as a molecular viewer, therefore you must make sure that under `@PROGRAMS` the path points to its executable. Alternatively, you can create a link in the `<trixx_installation_dir>/bin` to wherever you have installed *FlexV*, the path to *FlexV* would then be the same as to *TrixX BMI*.

You should now have a readily configured config.dat for this tutorial.

Please note: *TrixX BMI* does not come with a 3D generator, so the path in `@PROGRAMS` following `RCGENERATOR`, `3DGENERATOR` and `CONV_SMILES` is not configured. You should point these three variables to the location of your own 3D generator (e.g. CORINA, CONFORD, CONCORD etc.) and SMILES conversion tool, respectively. Without this information, *TrixX BMI* cannot convert SMILES strings to 3D-geometries and also cannot generate ring conformations for those molecules.

3.2 Running a virtual screening

When you type `trixx` in your working directory, *TrixX BMI* will say HI, and the command prompt will be waiting for your valuable input:

```
                                ##
                  ######                        ##  ##
                  # ## # ## ###  ###   ### ##  ####
                    ##    ### #   ##    ####    ##
                    ##    ##      ##     ##     ##
                    ##    ##      ##    ####   ####
                   ####  ####   ###### ## ### ##  ##

 Copyright        Screening with indexing technology

 BioSolveIT GmbH         Version:   2.3.0  (pre)   (30.10.08)
 An der Ziegelei 75      Modules:   [PHARM] [SCREEN]
 53757 St. Augustin
 Germany                 Original Author:    Matthias Rarey
 www.biosolveit.de       Contact:            flexx@biosolveit.de

 For information about additional contributors and copyright notes
 please consult the user guide or type 'help about'.

 >> Running on palermo (Linux 2.6.18.8-0.7-bigsmp) with 4 processors.
 >> TrixX configuration file 'config.dat' loaded.
 >> FlexX_base license check (BioSolveIT keys): succeeded.
 >> Licensed modules: TrixX [PHARM] [SCREEN]
 >> SETTINGS = 'static_data/flexx_settings.dat' loaded.
 >> CHEMPAR  = 'static_data/chempar.dat' loaded.
 >> CONTYPE  = 'static_data/contype.dat' loaded.
 >> GEOMETRY = 'static_data/geometry.dat' loaded.
 >> AMINO    = 'static_data/amino.dat' loaded.
 >> CHARGES  = 'static_data/amino_pcharges.dat' loaded.
 >> TRANSFORM = 'static_data/transform.dat' loaded.
 >> FCHARGES = 'static_data/fcharges.dat' loaded.
 >> DELOC    = 'static_data/delocalized.dat' loaded.
 >> CONTACT  = 'static_data/contact.dat' loaded.
 >> TORSION  = 'static_data/torsion_standard.dat' loaded.
 >> LOGP     = 'static_data/logp.dat' loaded.
 >> GRAPHIC  = 'static_data/graphic.dat' loaded.
    Process time used:  2.76 s.    Current process size: 67712 kB.
TRIXX>
```

In the following, the two step process of virtual screening is described by running through the basic routines with example script files. These are to be found in `bat`. First, a routine is called to catalog ligands.

3.2.1 How to catalog ligands?

The script for this section can be found in `bat/catalog.bat`. The first step in the virtual screening process is to catalog your ligands. Call the script with the following line from your shell:

```
trixx -b bat/catalog.bat -a '%{liglist}'=MY_LIG_LIST
```

This line means that you call the program TrixX BMI with the script `bat/catalog.bat`. The variables in the script have to be substituted with real file names. Here, this means you have to substitute MY_LIG_LIST with for example 1phd.list (provided for tutorial purposes) or your own list of ligands.
You can also start the script from within TrixX BMI by using this command

```
TRIXX> SCRIPT catalog %{liglist}= MY_LIG_LIST
```

from the main menu. Use a real name as ligand list as described above for MY_LIG_LIST. However, before you start the cataloging step, make sure that the entry "INDICES" in `config.dat` is set to an empty index directory!

Script file `bat/catalog.bat`:

set verbosity 1	Set the detail level of the output
SCREENIN	Enter screenin menu
FOR_EACH $(n) IN $(liglist)	Iterate over each ligand in the file
output $(n)	Print ligand name
catalog $(n)	Generate raw data
END_FOR	
CREATEDB	Generate meta data necessary to build indices
CREATIDX	Build indices

Now you have an index-based catalog of your ligands. The output command displays which ligands you have cataloged and how much time was spent on this procedure. The commands and options will be described in detail in section (4.2).

3.2.2 How to screen a target?

After having cataloged your ligands, you can start screening against your target. This section exemplifies how to perform this second step of the actual virtual screening procedure.

The screening procedure is also described by an example script to be found at `bat/screen.bat`. Call the script file from your shell with the following line:

```
trixx -b bat/screen.bat -a '%{target}'=MY_TARGET
```

Furthermore, you can start the script from within TrixX BMI by typing

```
TRIXX> SCRIPT screen %{target}=MY_TARGET
```

Again, you have to substitute MY_TARGET with a real filename. For this tutorial you may use 4dfr (valid pharmacophore provided for tutorial purpose). Before you start screening, be sure that the entry "INDICES" in `config.dat` is set to an existing compound index directory.

Script file `bat/screen.bat`:

RECEPTOR	Change to receptor menu
READ $(target)	Read the receptor via rec/$(target).rdf
DRAW	Draw receptor
SPOTS	
GENERATE	Generate interaction spots
PHARM	Change to pharmacophore menu
READ $(target)	Read in a pharmacophore (optional) from pharm/$(target).phm
DRAW	Draw pharmacophore
SCREENIN	Change to screenin menu
GETSTRI	Get site descriptors
FASTBIT	Start indexing
MATCH	Postprocess matches
STATS	Prints statistics about the screening run
LISTALL 20	Show the 20 or less highest scoring hits
DRAW 20 1	Draw top scored pose of the 20 highest scoring hits

Now you have performed a complete virtual screening experiment, and the best results have been reported to you. In the following sections, you get to know more about the commands and how to prepare you target for screening.

4

Working with *TrixX BMI*

4.1 Starting *TrixX BMI*

4.1.1 Interactive mode

To start *TrixX BMI* in interactive mode you must enter `trixx` from the operating system shell. You are then transferred to the *TrixX BMI* shell, i.e. you will see the *TrixX BMI* prompt on the screen:

```
TRIXX>
```

Historically, command line options and filenames are linked with a colon, for example `-l:<logfile>`. Because the filename extension mechanism does not work with this syntax, *TrixX BMI* also allows blanks as a separator, i.e. `-l <logfile>`.

4.1.2 Arguments for batch processing (-a)

If *TrixX BMI* is started in batch mode (see -b, below), you can define an argument string for the batch program. Variables in scripts are either an alphanumeric string or 0, 1, ..., 9.

4.1.3 Batch mode (-b)

For users experienced with scripts it may sometimes be desirable to start *TrixX BMI* in batch mode. One advantage of this mode is that you can redirect the screen output of *TrixX BMI* into a file. To start *TrixX BMI* in batch mode, type `trixx -b:<script filename>`. *TrixX BMI* will then execute the script <script filename>. If *TrixX BMI* is started with the -b option, it never waits for a keypress and terminates whenever an error occurs.

4.1.4 Specifying an alternative configuration file (-c)

When started, *TrixX BMI* normally tries to read `config.dat` from the current (startup) directory or from the directory specified by the *TrixX BMI* home directory. It is possible to tell *TrixX BMI* to use another configuration file. To do this, start *TrixX BMI* by typing `trixx -c:<filename>`, and *TrixX BMI* will then use <filename.dat> as its configuration file.

4.1.5 Specifying the execution directory (-d)

In order to execute *TrixX BMI* in an alternative directory, *TrixX BMI* can be called with option `-d:<execute dir>`.

4.1.6 Help for command line options (-h, ?)

Type `trixx -h` to get a short help text about the command line options.

4.1.7 Output the processor ID or system ID (-i)

Type `trixx -i` to output the processor or system ID of the machine it is running on.

4.1.8 Logging the *TrixX BMI* session (-l)

If *TrixX BMI* is started with the `-l:<logfile>` option, all commands executed are written with their parameters into a log file named `logfile` stored in the current directory.

4.1.9 Nice value (-n)

The *TrixX BMI* session can be started with a specific nice value given after the -n option.

4.1.10 Redirecting output (-o, -om)

By default *TrixX BMI* sends all text output to `stdout` and error messages to `stderr`. Starting *TrixX BMI* with `trixx -o:<outputfile>` causes text output to be redirected to `outputfile` and the error messages to be redirected to `outputfile.err`. The output of stdout and stderr can be merged using the parameter -om instead of -o.

4.1.11 Interface options (-p, -r, -s)

The options -p, -r, and -s are interface options to control *TrixX BMI* behavior in combination with calling programs and should therefore not be used as command line options.

4.1.12 Version information (-v)

Type `trixx -v` to get detailed information about the *TrixX BMI* version you are using.

4.2 The *TrixX BMI* shell

4.2.1 Menu navigation

When you see the *TrixX BMI* prompt on the screen, you can work with the *TrixX BMI* shell. The *TrixX BMI* shell is menu-driven, and the menus are hierarchically organized in a tree structure. In each menu you have specific valid commands (called *menu commands*). You can execute these commands by typing their names. Entering a name of a submenu brings you to the submenu, entering `END` to the parent menu. You can also directly go to a menu available in a parent menu by typing its name. The *TrixX BMI* prompt will always reflect the name and location of the current menu. There are some commands which are valid for all menus. These are called *global commands*.
You can get a list of all global commands, menu commands, submenu and parent menu names which are valid in a given menu by pressing the RETURN key after the prompt.

4.2.2 Global commands

Here, the commands are only described with a short information sentence. For further, detailed information, please refer to the command in the specified section of the *FlexX* User Guide [7]. Additional information about parameters for the commands can also be found there.

These commands are available in each menu.

Command	Explanation	Chapter in FlexX Manual
MAIN	Return to main Flexx menu from any submenu	6.2.2
END	Return to the parent menu	6.2.3
QUIT	End program, clear memory	6.2.1
!	Execute a unix command	6.2.15
EDITCFG	Open config.dat with the defined editor	6.2.7
RELOADCFG <filename>	Load filename as the new configuration file	6.2.8
LIST <parameter>	Lists environment variables and values of the given parameter, e.g. list all	6.2.9
SET <variable> <value>	Changes the environment variable to the new value	6.2.10
SELOUTP <destination>	Directs the output generated by the LIST, LISTALL, LISTSOL, LISTMAT, QUERY and INFO commands to <destination>.	6.2.11
EXEC	Execute a unix command and store output in $(UNIX_OUTP)	6.2.15
TOFLEXV <command>	Send a command string to the graphic mode FlexV	6.2.12
DISPLAY	Displays objects previously produced with the DRAW command in FlexV or other defined viewer	6.2.13
ERASE <graphic object>	Deletes the object with the next execution of display	6.2.14

There are also some menu specific commands. The following can only be called from the root menu.

Command	Explanation	Chapter in FlexX Manual
SCRIPT	Execute a bat script	6.3.3
DELALL	Deletes everything from the memory of TrixX BMI	6.3.1

There are four submenus descending from the root menu: The LIGAND-, PHARM-, RECEPTOR-, and SCREENIN menu. These can be called by simply typing their name. In the following, the commands that are unique for each menu will be described.

4.2.3 Submenu LIGAND

The submenu LIGAND is useful for reading a ligand, displaying it in a viewer, and comparing it to a reference ligand structure. The ligand or the reference ligand have to be in one of the following formats: SYBYL MOL2, MOL, PDB, SDF, SMILES. For converting SMILES to 3D coordinates, a program like CORINA is needed and has to be supplied in the `config.dat` file. The following commands can be called from the LIGAND submenu:

Command	**Explanation**	**Chapter in FlexX Manual**
READ	Read a ligand into TrixX BMI workspace	6.4.1
MINIMIZE	Minimize the ligand fix coordinates	6.4.23
SELINIT	Adjust the initialization levels	6.4.4
REINIT	Clean up the structure after TRANSFORM command	6.4.5
FROMPDB	Extract a ligand from a PDB file	6.4.3
READREF	Read a reference ligand structure for comparison	6.4.12
SETREF	Set reference coordinates to the coordinates from the input file that was read with the READ command	6.4.14
MAPREF	Match the reference structure to the ligand structure	6.4.13
WRITE	Write a set of ligand placements in a file	6.4.10
DELETE	Delete the loaded ligand from the TrixX BMI workspace	6.4.11
INFO	Display main characteristics of the loaded ligand	6.4.6
MOLINF	Display detailed information about the ligand	6.4.6
EDIT	Call editor with loaded ligand	6.4.7
SELADM	Specify the graphics object number and determine save modus of graphic files	6.4.16
SELGRA	Set specific default values for drawing ligands	6.4.17
SELCOL	Set color modes for molecule, molecular surface, and interaction geometries	6.4.18

Continued on next page

Command	Explanation	Chapter in FlexX Manual
SELLAB	Select labels for drawing the ligand	6.4.1
DRAW	Generate a graphic object of the ligand which can be displayed by a viewer	6.4.20
GRAINF	Output a list of all current graphic settings for the ligand	6.4.22

4.2.4 Submenu RECEPTOR

The submenu receptor is essential for the second step during virtual screening (3.2.2). Here, the receptor, which is supposed to be used for screening, is initialized. It has to be supplied as a receptor description (RDF) file, a PDB file, or a MOL2 file.

Command	Explanation	Chapter in FlexX Manual
PDBINFO	Display contents of a PDB-file (chains, ligands, metals)	6.5.2
READ	Read the receptor from a file	6.5.1
WRITE	Write a protein in a file	6.5.5
DELETE	Delete the receptor from the TrixX BMI workspace	6.5.6
INFO	Display information about the receptor	6.5.8
ATLIST	List all atoms of the active site and show the assigned properties	6.5.4
ACTIVE	Select the atoms that belong to the active site	6.5.3
EDIT	Edit the receptor input file	6.5.7
DEEPSITE	Form a subpocket containing the deep part of the active site	6.5.10
CSGRID	Build a clash-score info grid for the receptor	
SELADM	Specify the graphics object number and determine save modus of graphic files	6.5.12
SELGRA	Set specific default values for drawing	6.5.13
SELCOL	Select colors for drawing the pharmacophore	6.5.14
SELLAB	Selecting labels for drawing the receptor	6.5.15
DRAW	Generate a graphic object of the receptor	6.5.16
GRAINF	Output a list of all current graphic settings for the receptor	6.5.17

In the receptor menu you find another submenu: The SPOTS menu. This menu is used for the generation of interaction spots of the receptor. First, you have to generate these interaction spots before you can start the screening process.
In this menu you find the following commands:

Command	Explanation	Chapter in FlexX Manual
GENERATE	Generate the interaction spots for the receptor	7.6.2.3
DRAW	Draw the interaction spots to a graphic object	7.6.2.3
INFO	Display information about the generated interaction spots	7.6.2.3
SELADM	Specify the graphics object number and determine save modus of graphic files	7.6.2.3
SELGRA	Set specific default values for drawing	7.6.2.3
SELCOL	Select colors for drawing the pharmacophore	7.6.2.3

4.2.5 Submenu PHARM

The submenu pharm is especially useful for performing virtual screening with TrixX BMI. If you have a pharmacophore for your protein target, you load it here. It will later be used in the in the screening procedure (see 3.2.2). Its usage speeds up TrixX BMI enormously: Many ligands can be filtered and are thus excluded from time consuming docking calculations. Again, the submenu commands will be listed here. For further information have a look at the *FlexX* User Guide [7].

Command	Explanation	Chapter in FlexX Manual
READ	Read a set of pharmacophore constraints from a file	7.3.5
DELETE	Delete the pharmacophore from the TrixX BMIworkspace	7.3.5
INFO	Display information about the pharmacophore	7.3.5
EDIT	Edit the pharmacophore input file	7.3.5
FILTER	Test a set of docking solutions against the pharmacophore	7.3.5
DRAW	Generate a graphic object of the pharmacophore components	7.3.5
PICKPH	Launch FlexV with the pharmacophore manager in order to pick constraints	7.3.5
SELADM	Specify the graphics object number and determine save modus of graphic files	7.3.5
SELGRA	Set specific default values for drawing	7.3.5
SELCOL	Select colors for drawing the pharmacophore	7.3.5

4.2.6 Submenu SCREENIN

This submenu is used during preprocessing and virtual screening. In the following, the commands are explained in detail, since this submenu is unique for TrixX BMI, the commands cannot be found in the *FlexX* User Guide [7].

Catalog compound (CATALOG)

Syntax: `CATALOG` <id> <log_file>

Description: The command CATALOG generates a library with the <id> and writes information to the <log_file>. It can be used to create a compound index for all your library compounds by applying it in a loop using a script file.

Draw the results of virtual screening (DRAW)

Syntax: `DRAW` <nof_hits> <nof_poses>
Description: After virtual screening, you can display the results in a viewer. The first number specifies the number of the hits that are displayed. The second number specifies the number of poses to be drawn for each hit.

Draw the results of a specific solution (DRAWONE)

Syntax: `DRAWONE` <use_ID> <ID_rank> <nof_poses>
Description: Similar to DRAW: It displays <nof_poses> many results for a specific solution in a viewer. This solution can be specified by either identifier (use_ID $= y$) or rank (use_ID $= n$) and is supplied using <ID_rank>.

Delete a molecule from the index (DELMOL)

Syntax: `DELMOL` <id>
Description: Deletes <id> from the index.

Get site interaction triangles (GETSTRI)

Syntax: `GETSTRI`
Description: With this command site interaction triangles are created that are used as descriptors. As first step in the screening process, the receptor has to be read and the interaction spots have to be generated with the command GENERATE in the menu SPOTS. Otherwise, site triangles cannot be computed.

List the results of virtual screening (LISTALL)

Syntax: `LISTALL` <nof_hits> <nof_poses>
Description: After virtual screening you can display the results with this command. The results are rank ordered, the highest scoring solutions are listed first. The first number specifies how many solutions of the ranked hitlist are displayed. The second number specifies the number of poses to be drawn for each hit.

List the results of a specific solution (LISTONE)

Syntax: `LISTONE` <rank_number> <nof_poses>
Description: Similar to LISTALL: It displays <nof_poses> many poses for the solution on rank <rank_number>.

Optimize the physical index structure (PURGE)

Syntax: `PURGE`
Description: If many compounds have been deleted from the index, its physical structure should be reorganized. The associated routines are executed using this command.

Prepare conformational ensembles (SAMPLE)

Syntax: `SAMPLE` <id> <path> <ens_name>
Description: This command performs conformational sampling of the library compound <id>. It automatically fragments the compound if it has more than 10 rotatable bonds. The resulting ensemble(s) are stored in <path> and employ <ens_name> as file name.

Output statistical information (STATS)

Syntax: `STATS`
Description: This command lists the following information of the screening run: Name of the target, number of site descriptors used for querying, number of descriptor matches, number of clashes, number of poses, and timings (query-, match-, and total time).

Perform virtual screening (VHTS)

Syntax: `VHTS` <minimize> <filter_props>
Description: This command starts the actual virtual screening. The first parameter <minimize> specifies whether the resulting solutions should be subjected to FlexX minimization (<minimize> $= y$) or not (<minimize> $= n$). The second parameter <filter_props> allows the usage of a molecular property filter and can also be supplied as y or n value.

Index

Bibliography

[1] H.-J. Böhm. LUDI: rule-based automatic design of new substituents for enzyme inhibitor leads. *Journal of Computer-Aided Molecular Design*, 6:593–606, 1992. 149

[2] H.-J. Böhm. The development of a simple empirical scoring function to estimate the binding constant for a protein-ligand complex of known three-dimensional structure. *Journal of Computer-Aided Molecular Design*, 8:243–256, 1994. 149

[3] J. Gasteiger, C. Rudolph, and J. Sadowski. Automatic generation of 3d-atomic coordinates for organic molecules. *Tetrahedron Computer Methodology*, 3:537–547, 1990. 153

[4] G. Klebe and T. Mietzner. Correlation of crystal data to analyze and predict ligand/receptor interactions. In D. W. Jones, editor, *Organic Crystal Chemistry*. Oxford University Press, Oxford, UK, 1992. 149

[5] G. Klebe and T. Mietzner. A fast and efficient method to generate biologically relevant conformations. *Journal of Computer-Aided Molecular Design*, 8:583–606, 1994. 152

[6] B. Kramer, M. Rarey, and T. Lengauer. Casp-2 experiences with docking flexible ligands using flexx. *PROTEINS: Structure, Function and Genetics*, Suppl 1:1(1):221–225, 1997. 149

[7] M. Rarey. *FlexX Release 2.3 User Guide*. BioSolveIT GmbH, St. Augustin, Germany, 2007. 149, 150, 156, 163, 166

[8] M. Rarey, B. Kramer, and T. Lengauer. Multiple automatic base selection: Protein-ligand docking based on incremental construction without manual intervention. 1997. 149

[9] M. Rarey, B. Kramer, and T. Lengauer. Docking of hydrophobic ligands with interaction-based matching algorithms. *Bioinformatics*, 15:243–250, 1999. 149

[10] M. Rarey, B. Kramer, and T. Lengauer. The particle concept: Placing discrete water molecules during protein-ligand docking predictions. *PROTEINS: Structure, Function and Genetics*, 34(1):17–28, 1999. 149

[11] M. Rarey, S. Wefing, and T. Lengauer. Placement of medium-sized molecular fragments into active sites of proteins. *Journal of Computer-Aided Molecular Design*, 10:41–54, 1996. 149

[12] M. Rarey and M. Zimmermann. *FTrees 1.4.2 User Guide*. BioSolveIT GmbH, St. Augustin, Germany. 149

[13] J. Sadowski, J. Gasteiger, and G. Klebe. Comparison of automatic three-dimensional model builders using 639 x-ray structures. *Journal of Chemical Information and Computer Science*, 34:1000–1008, 1994. 153