New Approaches to Computer-Aided Drug Design

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von
Dipl. Inform. (Bioinformatik)
Marcel Schumann
aus Moers

Tübingen
2012

Tag der mündlichen Qualifikation:	05.06.2013
Dekan:	Prof. Dr. Wolfgang Rosenstiel
1. Berichterstatter:	Prof. Dr.-Ing. Oliver Kohlbacher
2. Berichterstatter:	Prof. Dr. Frank Böckler

Abstract

Computer-aided drug design is very important for modern drug discovery. Using a variety of different algorithms, approximations of the binding free energy of chemical compounds to a molecular target can be generated *in silico* in a very fast and very cheap way, without any need for physical availability of those compounds in this step. Computer-aided drug design thus allows to drastically speed up the task of developing new drugs, strongly reduces costs and enables the rapid testing of new, yet unsynthesized, classes of compounds.

In this dissertation, new approaches for computer-aided drug design are presented: a framework for Quantitative Structure-Activity Relationship (QSAR) modeling, a receptor-ligand scoring function and a docking algorithm, a three-dimensional target-specific rescoring procedure and CADDSuite, a software suite that contains all the aforementioned algorithms and a large set of additional, auxiliary tools and algorithms.

The QSAR framework provides all necessary steps to generate regression or classification models with high predictive quality: read input, generate molecular descriptors, generate a variety of different regression and classification models, automatically select relevant descriptors and evaluate the quality of models. Using several data sets, we will show that is easily possible to obtain high-quality QSAR models by using all the functionality in combination.

IMGDock, a deterministic receptor-ligand docking algorithm employing a specially designed empirical scoring function has been developed. Using the established DUD (Cross et al., J Med Chem, 2006, 49, 6789-6801) docking benchmark sets, we show that IMGDock yields results of high quality and in many cases outperforms other docking approaches. Furthermore, IMDock is fast, easily configurable and freely available as open source and can easily be deployed on compute clusters, clouds, or grids.

Target-Specific Grid-based Rescoring (TaGRes) employs three-dimensional information generated by docking and experimental binding free energy measurements for other compounds in order to rescore molecular interactions. Thereby, this approach takes into account receptor-ligand interactions, their three-dimensional locations and their target-specific importances. We will show that using this technique, the enrichment obtained by docking can be strongly enhanced.

CADDSuite (Computer-Aided Drug Design Suite), was created as a framework for computer-aided drug design, containing all the algorithms mentioned before, and a high number of auxiliary tools, for example for preparation or analysis purposes. Thus, CADDSuite provides flexibly combinable programs for all commonly required steps and can therefore make solving common drug design tasks much easier. To make

creation of pipelines even simpler, CADDSuite has also been integrated into the well-known workflow system Galaxy, thus essentially allowing users to create drug design workflows directly from a web browser, without any need for software installations on their local computer, and also to directly submit them to a compute cluster, grid, or cloud.

Last but not least, we will explain our work towards discovery of inhibitors for bacterial biofilm formation. We will describe how we found a number of very promising inhibitor candidates, using a combination of our computer-aided drug design tools and experimental validations.

Acknowledgements

First of all, I would like to thank Prof. Dr. Oliver Kohlbacher for the opportunity and freedom to work in the area I cared most for.

Many thanks to Prof. Dr. Frank Böckler for readily agreeing to serve as reviewer for this dissertation.

I am very grateful to Dr. Silvia Herbert for performing many wetlab experiments concerning IcaA. She performed the biofilm assays used to evaluate IcaA inhibitor candidates proposed by us.

Thanks a lot to all the members of the SFB766 for many interesting and inspiring discussions about biofilms, infections, cell walls as well as biology and science in general.

I would like to thank Dr. Alex Böhm for exciting discussions about IcaA and biofilms and possible future steps with respect to the biofilm formation inhibitors discovered by us.

I am grateful to Dr. Andreas Kämper for carefully and speedily proof-reading my manuscripts.

Last but not least, many thanks go to my former colleagues in the Kohlbacher lab for a nice and enjoyable time there.

In accordance with the standard scientific protocol, I will use the personal pronoun "we" to indicate the reader and the writer, or my scientific collaborators and myself.

Contents

1 Introduction

The development of medical drugs is a very important and exciting process, helping to finally alleviate or even cure many diseases. In former times, for hundreds and thousands of years until approximately the beginning of the 20^{th} century, finding new remedies just occurred by chance. However, as many of those discoveries are nowadays known as classical home remedies, we know that most of them probably had either a rather weak or nearly no effect at all. This may have been due to the use of the entire plant or animal parts instead of isolated chemical compounds serving as their active ingredients or simply due to the fact that for most diseases nature most likely does not present us with so easily available cures. Furthermore, relying on random chance is obviously a much too slow way to counter the spread of diseases, so that no cures for the great plagues having beset mankind - black death, cholera, typhus, tuberculosis, and many more - have been found in nature.

At the beginning of the 19^{th} century, chemists for the first times extracted and purified the chemical compounds bestowing a medically relevant effect on some plants. Examples for this are morphine, extracted from poppy plants in 1804 by Friedrich Sertürner [1], and salicin (pre-predecessor of acetylsalicylic acid, known today under its trade name Aspirin), isolated in 1828 by Johann Buchner [2].

Although this was a very important step, no completely new remedies could be generated and no existing ones could be chemically modified. This changed at the beginning of the 20^{th} century, when the latter was successfully attempted. Two of the first examples for this were the modification of morphine, yielding heroin (diacetylmorphine) and of salicylic acid to acetylsalicylic acid. Heroin acts as a much stronger analgesic than morphine and acetylsalicylic acid has much less severe side-effects than salicylic acid.

In the following decades, principles learned this way were extended and, step-by-step, procedures to check the effects of replacing moieties of known drugs by a number of fragments by *in vivo* or *in vitro* tests were established. Some more decades later, laboratory automations invented in the meantime allowed to speed up this process, so that the term high-throughput screening (HTS) was coined (HTS will be explained in more detail in the next chapter).

HTS was and still is an important tool for drug design, but still, it is very expensive, time-consuming and depends on the availability of both, protein and compounds to be tested in purified, high-quality form. Huge warehouses of compounds are necessary, leading to very high initial investments for setting up such a system and high costs for maintaining it. Although the speed with which experiments are performed is increased

by HTS in comparison to manual testing, screening a huge data bank containing hundreds of thousands or even millions of compounds would (depending on the size of the respective HTS infrastructure) currently probably still take several weeks. In addition, such large-scale testing would also be very expensive, since the test of each compound consumes a certain amount of the highly purified stockpile of this molecule. Also, HTS must work in a very fast way, so that only very quick experimental binding detection procedures, e.g. fluorescence-based ones, can be used. Those fast techniques however do not allow a measurement of the binding free energy and are less reliable than those that do allow this, which can be seen as the gold standard. (Explanations of binding free energy measurement techniques and descriptions of other issues with HTS will be given in Section 2.1.3.) Of course, evaluation of compounds that have not already been obtained from a natural source or synthesized, which are complicated and time-consuming efforts, is not possible by use of HTS.

Computer-aided drug design, established step-by-step over the course of the last few decades, makes it possible to eliminate many of the aforementioned problems. Using a variety of different algorithms, approximations of binding free energy of chemical compounds to a molecular target can be generated *in silico* in a very fast and very cheap way, without any need for physical availability of those compounds in this step. Computer-aided drug design thus allows to drastically speed up the task of developing new drugs, strongly reduces costs and enables the rapid testing of new, yet unsynthesized, classes of compounds. Furthermore, it can also be used to predict other chemical properties of molecules, like their absorption, distribution, metabolism, and excretion, and thereby speed up drug development by either removing compounds with undesired properties or by helping to optimize properties of discovered hits. In order to make use of all the advantages of modern computer-aided drug design, a plethora of different algorithms and preparation steps is necessary, which have be to utilized together in huge computational pipelines. However, this in practice is often difficult because of software tools that are incompatible, too slow, instable, produce too poor results, or are unavailable due to e.g. licensing issues. In addition, such software nearly always is not available as open source, so that it often cannot serve as a framework for further methological research by other scientists.

In this dissertation, we present algorithms and tools for most common applications of computer-aided drug design that are fast, scalable, stable, and efficient and can easily be used in conjunction with each other. Furthermore, all of them are publicly released as one open-source framework, CADDSuite, making it possible for everyone to use and even extend them free of any charge. Using a number of different data sets and also one drug design project with experimental validations, will we show that the created algorithms produce results of good quality and can be very helpful.

One of the areas of application of computer-aided drug design is quantitative structure-activity (QSAR) modeling. As will be described in Chapter 4, we implemented a variety of different regression and classification models, as well as input generation, data management, feature selection, and model validation techniques under one common framework, so that all of those procedures are easy to use in combination, are quickly extensible and flexibly usable.

For the field of structure-based drug design, we developed a fast receptor-ligand scoring function, a docking algorithm and a three-dimensional, target-specific rescoring approach. The scoring function and docking algorithm will be introduced in Chapter 5. The docking approach, utilizing this scoring function, is fast and scalable and easily and efficiently deployable on compute clusters, clouds or grids, and will be shown to achieve a quality better than the one obtained by other approaches for a significant number of data sets.

Furthermore, to allow for optimization of binding free energy estimates obtained by docking, we developed a new receptor-ligand rescoring technique, to be described in Chapter 6. It uses the three-dimensional informations (i.e., the so-called poses describing the putative ligands inside the binding pocket) generated by docking and experimental binding free energy measurements for other compounds in order to rescore the docking poses. Thereby, this approach, in contrast to all other ones known to us, takes into account receptor-ligand interactions, their three-dimensional locations and their target-specific importances. We will show that using this technique, the quality obtained by docking can be strongly enhanced.

Chapter 7 will then present an overview of our entire framework, called CADDSuite (Computer-Aided Drug Design Suite). This framework contains all the algorithms mentioned before, and a high number of auxiliary tools, e.g. for preparation or analysis purposes. The chapter will explain why a large number of auxiliary tools are necessary and shortly introduce those provided by CADDSuite. It will also contain a case study involving the virtual screening for carbonic anhydrase II inhibitors that will nicely visualize the practical usefulness of CADDSuite.

Chapter 8 will detail our efforts to find inhibitors for bacterial biofilm formation using a combination of our computer-aided drug design tools and experimental validations. We will describe how we found a number of very promising inhibitor candidates this way.

But first of all, Chapters 2 and 3 will now introduce you to the biochemical and computational background of all the topics mentioned above. Chapter 2 will give an overview of the long process of finding new medical drugs, as it is used in modern drug discovery projects and also mandated by the agencies responsible for approval of new drugs. This will include explanations about in which steps computer-aided drug design can help to speed up the overall process. Furthermore, Chapter 2 will also clarify the different means of obtaining binding free energy measurements and explain the biological and medical significance of carbonic anhydrase II, used a molecular target in Chapter 7, and bacterial biofilms. Chapter 3 will then give an overview of computer-aided drug design, introduce its different categories structure-based and ligand-based drug design, and describe machine learning techniques, quality statistics and other algorithms used by our approaches.

2 Biological Background

2.1 Overview of drug design pipelines

The development of a new drug is a lengthy, time-consuming, and very expensive process. It involves many steps to first generate candidates and then, step by step, filter out those that do not show the desired activity or exhibit adverse effects. In total, the entire process may well take more than a decade and its typical cost is currently estimated as 1.8 billion US dollars [3].

In order to show its complexity and point out at which points computational methods may aid and speed-up the process, we now give a short, generalized overview of this drug design pipeline. The succession of different steps that are usually employed and that are described in Sections 2.1.1-2.1.6 is furthermore visualized by Figure 2.1.

2.1.1 Disease selection

The fist step consists of the seemingly trivial but in practice far from simple task of selecting the disease for which a drug should be developed. While there are lots of illnesses for which no cure is known and no helpful drug is currently available, a number of criteria have to be satisfied. On the one hand, the degree and quality of existing medical, biological and biochemical knowledge about those diseases varies strongly. In some cases (animal) models of possible causes and courses may exist, while in others research is still in a very early stage. The more well-founded knowledge is available about a disease, the better. But, on the other hand, the question of the disease's complexity may have an important influence, too. If the possible causes of an illness are manifold or if its progress varies widely among different patients, it may be considered inappropriate for drug design if those complexities are not yet understood to a reasonable degree. Last but not least, economic considerations (unfortunately) usually also have to be taken into account. Diseases that do not affect a significant portion of the population but are either comparably rare or location-specific (like, for

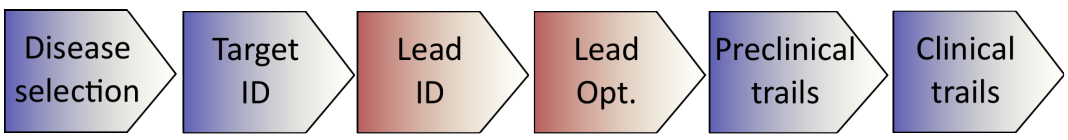


Figure 2.1: Overview of the usual process of drug development. The individual steps are be described in Sections 2.1.1-2.1.6.

example, tropical diseases) may often be neglected with respect to drug development, due to the decreased potential of financial benefits of potential drugs targeting such illnesses. Although this may not necessarily be true for non-commercial research institutes, the immense cost associated with developing new drugs shows that successfully executing all steps of the drug design pipeline may in most cases be impossible without involvement of pharmaceutical companies and (unfortunately) also explains, at least to a certain extent, the need for this economical constraint.

2.1.2 Target identification

If and when a disease has been selected, the next goal is to find an appropriate molecular target. For many ailments, many medical research results are available but no molecular source of either the illness or already known remedies, if any, are known. Thus, the progress of a disease might have been medically studied and its progress well documented, but the molecular structures, e.g. enzymes or receptors, that are its cause have not been identified. Even if some drugs exist, especially some that have been found more or less serendipitously without use of modern drug design, it is often unknown, which enzyme or other molecular structure they influence. Hence, identifying a list of possible target structures is often not easy. However, since the selection of a molecular target is vital for modern rational drug design, several techniques have been established in order to aid achieving this. One example are microarrays, chips containing single-stranded DNA probe segments that allow to bind complementary, fluorescently labeled sample gene transcripts. A second option is the use of proteomics, which can measure the amount of proteins of interest within a sample and thereby the degree of their expression by techniques like high-throughput liquid chromatography coupled to mass spectroscopy (HPLC-MS). Both, transcriptomics- and proteomics-based studies, then compare expression levels of respectively genes and proteins in cells obtained from healthy patients against the expression levels in cells obtained from patients affected by the illness in order to find proteins that are significantly up- or downregulated in the latter. In any case, differentially expressed proteins found this way just make up a list of possible targets and do not directly yield a definite molecular target. All candidate structures found in this, or any way, must be examined further to determine whether they meet a range of criteria.

First, an important point is how much biochemical information is already available about the respective structure, whether its function or, in the case of enzymes, its exact mode of action is already known. Candidates for which no or hardly any such information is attainable will most likely be rejected as target candidates. Furthermore, it is of interest to which class the respective target candidate belongs: enzymes, receptors, transporter or other kinds of molecular structures. While enzymes are the class from which most drug design targets have been chosen so far, receptors and transporters might be relevant as targets as well, if enough biochemical information is available about them. Other molecular structures, like non-enzyme proteins, DNA or RNA, are much less obvious choices and have only very rarely been chosen as potential drug targets. This is due to the point that for those molecules there is often no

well-defined structural area that might be targeted by a potential drug, as is the case with active or allosteric sites of enzymes.

Another factor influencing the selection of a target structure is the similarity of the respective target candidate to known antitargets, i.e. targets which may not be affected by the drug to be developed. If this similarity is too high, it may be prudent to reject the respective structure and instead prefer others that have lower similarities to the antitargets, thereby decreasing the chance of potential drug candidates to cause adverse effects. A special case of this is the development of drugs against enzymes of a pathogen, e.g. the search for antibiotics. Here, it thus has to be checked whether the same enzyme or one with a very high sequence similarity exists in humans. If an enzyme with exactly the same amino acid sequence is present in humans, the target candidate will have to be rejected. On the other hand, if that is not the case but there still is very high similarity, then the respective structure may or may not be dismissed, but its selection could well complicate efforts to find an effective drug without severe side effects.

If a long list of possible targets has been obtained by the above-stated methods and criteria, there any several points that can help to choose the most promising candidate. For one, the physiological or subcellular location of these structures may be examined. Depending on e.g. in which organs of the human body an enzyme is expressed or whether it is present in the cell wall or in the nucleus, the availability of different target candidates to potential drugs can vary strongly. Thus, structures that are more likely to be easily accessible to drugs can be preferred over others. Another option for reducing the set of candidates is to examine for each structure whether is known to be involved in diseases other than the one for which a drug is to be developed. If that is the case and the activity of the target candidate either needs to be up-regulated (e.g. by agonists in case enzymes) for all those illnesses or down-regulated (e.g. by an enzyme inhibitor) for all of them, the respective candidate might be especially interesting as a target structure for drug design. This way, several diseases could in the best case be alleviated by one and the same developed drug. Even if the drug candidate turns out not be a success for all targeted diseases, its chance of having an effect on at least one diseases may be higher, compared to targeting of only one disease.

2.1.3 Lead identification

After a molecular target structure has been agreed on, the next step is to find molecules, so called *leads*, that, at least to a low to moderate degree, show the desired effect on the target. Leads may bind relatively weakly to the target structure or have severe side-effects if they were administered to humans, because the goal here is just to obtain starting points, i.e. molecules that can be examined and in the next step be exchanged against similar ones or be chemically modified.

Traditionally, the quest for leads was and is often directed by high-throughput screening (HTS). HTS approaches aim to experimentally and as fast as possible check chem-

ical compounds for effects on the given target in an automated fashion. Many of the systems involved with this include fluorescence-based ligand binding detection techniques. One example for this is fluorescence resonance energy transfer (FRET) [4], for which receptor and (potential) ligands are endowed with different fluorescent tags. If and when target and ligand are close enough to each other, i.e. if the latter is bound to the former, energy is transferred from one fluorescent tag (donor) to the other one (acceptor). This emission of light, at a wavelength known for the chosen donor-acceptor pair, can be measured and thus help to evaluate whether molecules bind the target structure of interest. Since it may be impractically to fluorescently label all compounds to be screened, in several cases the substrate (or other binding partner), if any, of the target structure can be labeled instead. Thus, if leads for the development of inhibitors of enzymes are searched, decreased fluorescent light emission after addition of a compound will show that this molecule is able to act in the desired way.

HTS can be run completely automated, with robots extracting prepared proteins and compounds from storage facilities, mixing them and analyzing the resulting images of fluorescence, so that a high number of compounds can be screened per day this way. However, HTS also has a number of disadvantages. First, no information about (approximate) binding strength of detected hits is obtained by HTS. Compounds are just classified as active or inactive, based on the detected level of fluorescence. The measuring of fluorescence furthermore needs to be very accurate, so that in practice many false positives and false negatives might occur, especially when ligand candidates are small and bind relatively weakly, if at all, thus yielding only tiny changes in the level of emitted fluorescence. This problem is aggravated by the fact that during lead identification most molecules will indeed not bind strongly to the target. Furthermore, it can be hard to attach the tags required for HTS (e.g. fluorescent ones) to the target structure and the ligand candidates, or this may not work at all. These tags can also influence the properties and behavior especially of small molecules, so that ligands might not bind to their receptor any more after having been tagged. This may be due, among others, to sterical constraints of the binding pocket, induced electrostatic repulsion, compounds with attached tags binding (unspecifically) to other areas of the target structure or even to other molecular structures, or aggregation of the ligands caused by the tags. Additionally, no information about (putative) binding-modes is obtained by HTS and the procedure requires huge storage facilities in order to be able to test a high number of compounds automatically. Although HTS can be run in an automated way, it still necessitates immense financial investments for setting up and maintaining such a system and considerable costs in terms of time and money to screen a high number of compounds. Last, but not least, since HTS is done *in vitro* and not *in silico*, each and every compound to be tested has to have been synthesized, purified and added to the storage system. It is thus not possible to screen compounds before their synthesis could be achieved. Since synthesis of chemical compounds can often be a complicated procedure requiring a long time and often very many attempts, this is a significant constraint. Also, even with huge compound storage facilities, the number of molecules that can be stored and therefore screened is of course limited, whereas the number of theoretically possible molecules can be considered to be prac-

tically unlimited (e.g., an estimated 10^{40} molecules with a molecular weight less than 750 g/mol [5]). This also means that most compound libraries used for HTS will either have been created to be as chemically diverse as possible or it will be focussed towards a specific target (family).

As a complement to traditional *in vitro* high-throughput screening, virtual high-throughput screening (VHTS) can help to alleviate many these problems. VHTS in principle uses *in silico* methods to filter huge compound libraries for potential binders. This might involve various filtering steps, in order to remove molecules with undesired properties, and the application of a variety of algorithms that in essence try to predict the binding free energy between chemical compounds and the target structure. Which algorithm or which class of algorithms is applicable or most promising, depends on the kind and degree of chemical information that is available about the target and already known ligands, if any. The most important techniques towards this end include Quantitative Structure-Activity Relationship (QSAR) modeling, receptor-ligand docking and rescoring. An overview of computer-aided drug design methods, their key differences and their prerequisites will be given in Section 3.1. Furthermore, QSAR, along with the approach and software developed by us, will be described in more detail in Chapter 4, while docking, including our own docking algorithm, is covered in Chapter 5 and rescoring, along with our algorithm, will be introduced in Chapter 6. When using those VHTS procedures, only compounds for which a high affinity to the target was predicted by the respective approach will be regarded as hits (analogously to hits by HTS) and examined further. All other compounds will be rejected as ligand candidates. Since all this is done *in silico*, VHTS can drastically reduce cost and time requirements for lead discovery and also allow to quickly screen new compounds or entire classes of compounds, without needing to wait for their synthesis.

In any case, whether HTS or VHTS is used, hits, i.e. molecule that seem to some extent show the desired activity, have to be experimentally validated before they are classified as leads. This is done by *in vitro* measurement of the actual binding free energy between each hit and the target structure. Various techniques exists for achieving this, which will be described in Section 2.2. Only those compounds for which at least a low to moderate strength of binding is determined by one of those approaches will be regarded as leads. All other molecules, i.e. those having been determined to most likely not bind to the target at all, will be cast aside.

2.1.4 Lead optimization

If leads have been found by the aforementioned steps, the next important goal then is to try to modify those compounds in such a way that their strength of binding to the target structure of interest is enhanced. A secondary aim may be the optimization of absorption, distribution, metabolism, excretion and toxicity (ADMET).

Historically, this was done in a mostly trial-and-error fashion. Thus, compounds that were very similar to the obtained leads were more or less arbitrarily conceived and then synthesized and subjected to binding affinity measurements as mentioned above

and explained in more detail in Section 2.2. Newer approaches included computer programs that allow a scientist to interactively try to manually place molecules into the binding pocket of the target. Repeating this procedure for different derivatives of one molecule could then make it possible to determine beneficial substitutions of individual moieties.

As extensions of those ideas, current computer-aided drug design procedures allow to automatically search for derivatives with enhanced binding affinity. QSAR approaches can thus be used be predict the binding free energy of derivatives by regression techniques based on previously experimentally determined affinities of similar compounds. Receptor-ligand docking can, either alternatively to QSAR or subsequently, dock those derivatives into the target of interest and obtain estimates of the binding free energies by scoring the obtaining compound poses. For an overview of computer-aided drug design approaches, please see Section 3.1; for more detailed descriptions of QSAR and docking, please refer to Chapters 4 and 5, respectively. Furthermore, it can be very helpful to rescore receptor-compound complexes attained by docking in such a way that the three-dimensional information of the pose and previously determined free energy measurements for other compounds can be used. This target-specific, three-dimensional rescoring was, at least to our knowledge, not possible until now. However, we developed an approach that works in this way, which is described in Chapter 6.

If computer-aided drug design approaches are used for lead optimization, the best candidates, i.e. the best-scored derivatives of the original leads, will be subjected to experimental determination of their binding free energies. As explained previously, this reduction in the number of compounds to be experimentally tested is of huge importance due to large savings in both time and money. Again, as is the case during lead discovery, the loss of a need to synthesize all compounds before their affinity can be predicted is a tremendous advantage of computer-aided drug design techniques.

In the end of the lead optimization step, only those derivatives with the very best, experimentally confirmed, binding affinity will be retained and find their way into the next step.

2.1.5 Preclinical trials

All candidates, as generated by the drug design pipeline described so far, will be subjected to testing in animals during preclinical trials. The focus here lies on filtering out drug candidates that have severe side-effects or exhibit very unfavorable pharmacokinetical or pharmacodynamical behavior. Thus, the different candidates are administered in varying concentrations to mostly mice or rats in order to check for toxic, carcinogenic or teratogenic effects. In addition, tests studying the pharmacokinetics and pharmacodynamics are performed. As results of pharmacokinetic experiments, information about, among others, how fast the drug candidate is absorbed and distributed throughout the body, and how quickly it is metabolized and excreted is collected. Pharmacodynamic studies reveal, for example, the relationship of the concentration of the administered compound and its desired effect on the body.

Hence, if toxic, carcinogenic or teratogenic side-effects are observed with doses that are not at least by several orders of magnitude larger than the minimal effective dose determined during pharmacodynamics experiments, then the respective drug candidate will most likely be dismissed. Alternatively, such a compound could also be transferred to another iteration of lead optimization, so that chemical modifications that do hopefully remove these adverse properties might be obtained. If, on the other hand, several candidates show no significant side-effects, only the ones with the best pharmacokinetical and pharmacodynamical properties can be selected.

2.1.6 Clinical trials

If a drug candidate has been found by the above-mentioned steps, it is then extensively studied during clinical trials. Clinical trials are divided into several phases (I-IV), during which the compound, in contrast to preclinical trials, will be tested in humans.

In phase I of the clinical trials, the drug candidate is administered in various doses to a couple dozen healthy subjects. The aim is to study the pharmacodynamics and pharmacokinetics of the compounds and to check for adverse effects, all of which might differ considerably from those determined in animals during preclinical trials. Typically, the maximal applied dose is only a fraction of the minimal dose found to cause adverse side-effects in animals.

If a candidate passed phase I tests, it is subsequently evaluated during phase II using a larger group of several hundred patients affected with the disease for which a drug is to be developed. Now the goal is to ascertain that the drug candidate actually has the desired effect on the illness and to establish a suggestion for doses of the putative drug for use in humans. Also, attention is again paid to possible side-effects, which would, if severe or frequently occurring, lead to the adandonment of drug design efforts for the current candidate.

Phase III of the clinical trials then consists of a large double-blind study, usually involving several different clinics and several thousand patients, conducted over the course of several years. Here it has to be shown that the proposed drug is safe, i.e. does not have frequent or severe side-effects, that it is effective and has advantages over existing drugs targeting the same illness. Only if those criteria have been fulfilled, approval for general use and marketing as a drug might be obtained from the appropriate national regulatory agencies (FDA, the Food and Drug Administration, in the US and the EMA, European Medicines Agency, in the EU). However, approval is always limited to the drug's application to the specific illness for which is was developed and to the exact doses and dosage form established in the clinical trials.

After approval for a pharmaceutical drug has been granted, its safety is continuously monitored in phase IV. The primary aim is to check for potential severe side-effects that however occur so seldom that they were not experienced during phase I-III. Causes for the latter may be, among others, interactions of the new drug with other drugs used by

the general public but not by patients of phases I-III, or preexisting medical conditions that were also not encountered in earlier trial stages.

2.2 Binding-Affinity measurements

As explained previously, the determination of binding affinities of compounds to the target structure of interest is vital during both lead discovery and lead optimization. Several methods exists to achieve this and the two perhaps most prominent and most reliable procedures will be shortly explained in the following. However, the principle prerequisites are similar for all approaches: the target structure of interest has to have been produced in relatively high amounts using (mostly) bacterial expression systems and purified to a very high degree. Furthermore, all compounds whose binding affinity to this target is to be determined must have been synthesized and be available in pure form.

The binding affinity of a compound is then commonly calculated in form of the dissociation constant K_D, the concentration at which half of the molecules are excepted to be bound to the target structure, with the other half remaining in solution. The dissociation constant is defined as

$$K_D = \frac{[T] \cdot [C]}{[TC]}$$

where $[T]$ denotes the concentration of the molecular target (e.g. an enzyme), $[C]$ the concentration of the solvated compound to be investigated and $[RC]$ the one of the complex of target and compound.

If the dissociation constant has been obtained for compound, its binding free energy can be calculated as

$$\Delta G = R \cdot T \cdot ln(K_D) \tag{2.1}$$

with R being the gas constant (approximately $8.314 \frac{J}{K \cdot mol}$) and T the temperature in degrees Kelvin.

2.2.1 Surface plasmon resonance

One way to measure binding affinities is by use of the surface plasmon resonance (SPR) phenomenon. This phenomenon can be observed in an electrically conducting gold layer at the interface of buffer and glass. Under conditions of total internal reflection, light reflected by this gold layer will generate electromagnetic waves that oscillate parallel to the interface (evanescent waves). Only at a certain angle of incident, light excites plasmons in the gold layer, leading to absorption of energy by the evanescent waves and a reduction of the intensity of reflected light. The exact angle (SPR angle) at which this surface plasmon resonance phenomenon occurs depends on the refractive index of the medium and thereby the amount of protein or ligand near the surface

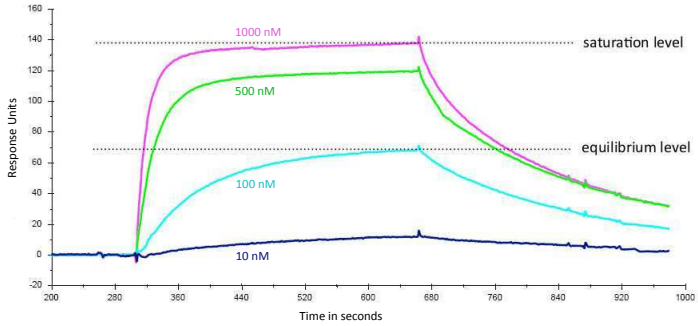


Figure 2.2: Example of an SPR diagram showing measurement for one compound done for 4 different concentrations (adapted from [6]). Equilibrium level is obtained with 100 nM. Thus, K_D = 100 nM.

of the SPR chip. A change in the mass concentration near the SPR chip surface is proportional to the resulting change in the SPR angle (SPR response).

The surface plasmon resonance phenomenon is thus utilized to obtain binding affinities by attaching target structure molecules to SPR chips and measuring the change of the SPR angle, during and after a compound is injected across the surfaces. If the compound binds to the target, the generated SPR response increases. On the other hand, if the ligand does not bind at all, no significant change in the SPR response occurs. In case binding does occur, it is possible to find the concentration at which the target is saturated with ligands (saturation level) and the one at which half the targets have ligands bound to them (equilibrium level) by repeating the experiment several times with varying concentrations of the compound. An example is shown in Figure 2.2.

Alternatively, it is also possible to calculate the dissociation constant by use of association rate k_{on} and dissociation rate k_{off}, both of which can be computed from the generated response curves, as $K_D = k_{off}/k_{on}$.

2.2.2 Isothermal titration calorimetry

Another frequently used approach for obtaining the binding affinity between a ligand and a target structure is isothermal titration calorimetry.

This approach works by detecting the temperature change generated by a binding between molecules. Therefore, two reaction cells, placed in an adiabatic container, are used. The temperature in one of those (sample cell) is always held at an identical level to the one in the other cell (reference cell) via a feedback mechanism that automatically powers a heater in the sample cell up or down, as needed. The change in energy consumption by the heater, which can then be directly monitored, is then equivalent to the enthalpic energy taken up or generated by ligand binding. A schematic overview of an isothermal titration calorimetry device in shown in Figure 2.3.

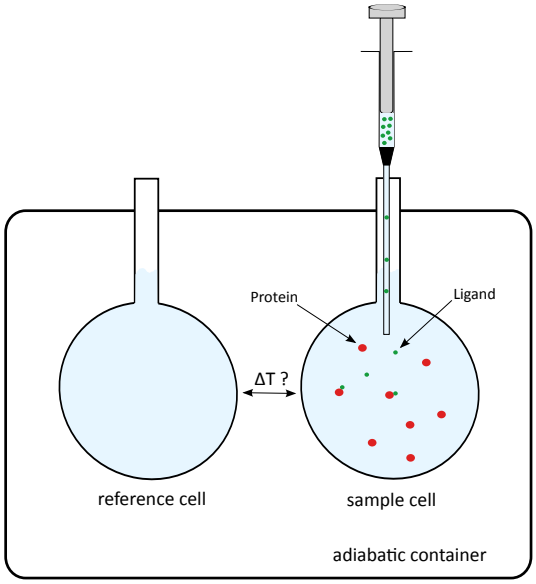


Figure 2.3: Schematic of an isothermal titration caliometry device.

Thus, for studying the binding of a compound to a target structure, the latter is placed into the sample cell and the former is subsequently injected into the chamber. The decrease and speed of decrease of energy consumption of the heater, due to the increased temperature in the sample cell, are then used to calculate the binding-free-energy (ΔG) of this receptor-ligand pair. The dissociation constant, can then be calculated in analogy to (2.1) as

$$K_D = e^{\Delta G/(R \cdot T)}.$$

The examination of the change in free energy over time furthermore allows to calculate the association and dissociation rates.

2.3 Carbonic Anhydrase II

Carbonic anhydrase II will be used as molecular target of a computer-aided drug design pipeline described in Chapter 7. We will thus shortly describe the function of this enzyme here and explain its medical significance.

In mammals, red blood cells transport oxygen through the entire body and release it fastest where it is needed most, i.e. in tissue that either has a lack of oxygen or an excess of carbon dioxide. In the former case, a low oxygen partial pressure prompts hemoglobin to do so, but in the latter case it is due to an increased acidity within the red blood cells. This increase in acidity is generated by carbon dioxide (CO_2) that has been converted to carbonic acid (H_2CO_3), which in turn dissolves into HCO_3^-, a strong acid, and H^+. However, the spontaneous conversion of CO_2 into H_2CO_3 is much too slow for cellular respiration to be able to work appropriately. The enzyme carbonic

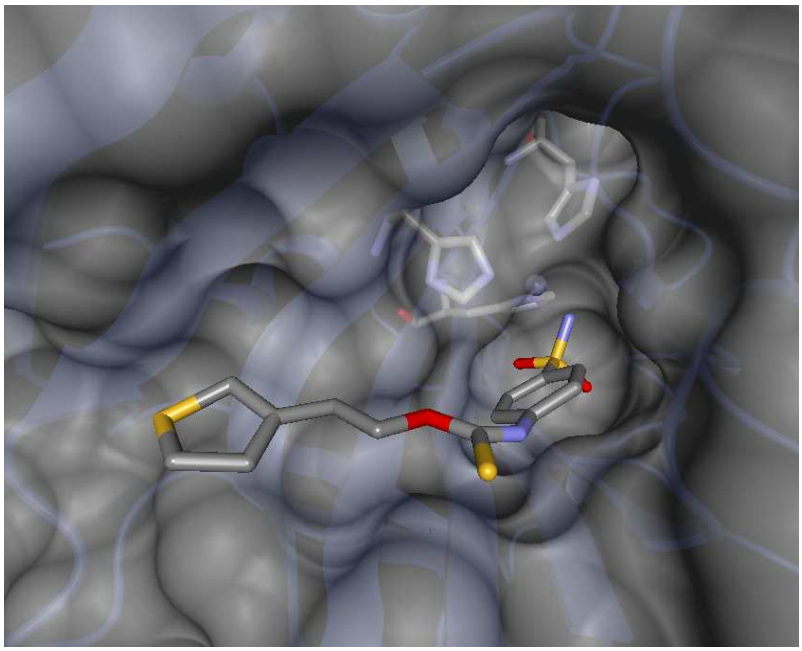


Figure 2.4: Crystal structure of carbonic anhydrase II (PDB ID: 3k34). Shown is the active site with three histidines coordinating a zinc ion and a bound inhibitor.

anhydrase II (and other carbonic anhydrases) catalyzes this reversible reaction and speeds it up by two orders of magnitude [5].

Thus, carbonic anhydrase II is vital for cellular respiration and is ubiquitous in mammals. On the other hand, it has been shown to be involved in the development of glaucoma [7]. Hence, a variety of carbonic anhydrase II inhibitors has been developed for intraocular application.

These inhibitors work by binding near the active site of carbonic anhydrase II and thus preventing a water molecule from associating with the enzyme in this position. The active side consists of three histidines that together with a water molecule mediate one zinc ion. This water molecule is, by doing so, converted to an hydroxide ion. Carbon dioxide, the substrate of carbonic anhydrase II, will then be the target of a nucleophilic attack by the zinc ion, creating a bond between the hydroxide ion and the carbon dioxide and thereby resulting in carbonic acid [5]. Hence, molecules that bind near the active site and block the binding of this water molecule can inhibit carbonic anhydrase II. Figure 2.4 shows a picture of the active site of carbonic anhydrase II with a bound inhibitor as an example.

2.4 Biofilms

Several bacterial families have be ability to attach themselves to each other and to polymer surfaces, thus creating so-called biofilms. Inside these biofilms, very many

layers of bacteria grow on top of each other, while those aggregations are firmly attached to (mostly) polymer materials and well protected from external influences like antibiotics.

During the development of biofilms, bacteria in a first step attach to a surface (Figure 2.5a). Several different chemical compounds produced by these microorganisms make this possible, one of which will be discussed later. Subsequently, bacteria grow layer-by-layer on top of each other (Figure 2.5b). By production of extracellular polymeric substances that form an extracellular matrix around each colony, the bacteria then protect themselves from attack (Figure 2.5c) by the host's immune system or drugs. Finally, bacterial cells are released from the colonies, disperse and form new colonies (Figure 2.5d).

One of the medically most problematic surfaces to be susceptible to biofilms are implants like artificial joints or catheters. There, bacteria can grow quickly and, due to the protective property of biofilms, be virtually immune to all antibiotics. The consequences of this are frequent infections that require the immediate removal of the respective medical implant and, even beyond this, have serious and often even life-threatening effects on the patient. Hence, biofilms are one of the most commonplace cause for nosocomial diseases [8]. A picture of a biofilm on a catheter is shown in Figure 2.6 as an example. Efforts to limit the growth of biofilms have so far mostly failed. Either a variety of antibiotics were tried or the surface of medical implants was in experimental versions changed in such a way as to be less susceptible to bacterial attachment. However, antibiotics did not work due to the self-protective property of biofilm colonies and the changed implant surfaces did not help either because lots of host cells attached to those new surfaces, so that the latter again was well suitable for biofilm formation [8].

The bacterial species that is the most frequent cause of medically problematic biofilms is *Staphyloccus epidermidis* [9]. This microorganism is ubiquitous on human skin and in the respiratory tract. A less frequent initiator of biofilms is *Staphyloccus aureus*. However, it is the most virulent strain, leading to the most severe medical complications for affected patients. In *S. epidermidis* and *S. aureus*, a group of proteins called intercellular adhesion proteins (Ica) has been shown to make up the machin-

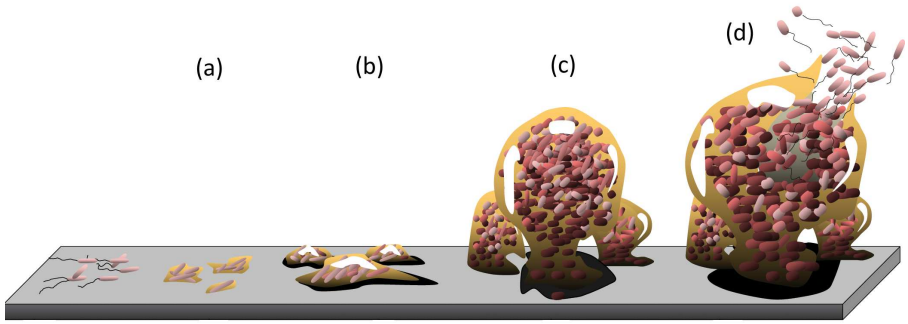


Figure 2.5: Development stages of biofilm formation (adapted from [8]).

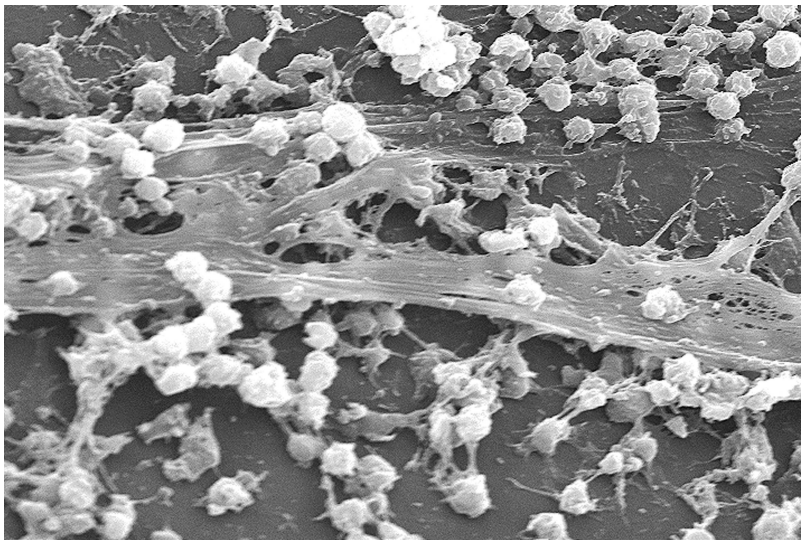


Figure 2.6: Biofilm on a catheter removed from a patient [8].

ery producing the adhesion that firmly attaches cells to a surface in the first step of biofilm development [10, 11]. This adhesion is made of many molecules of N-acetyl-glucosamine (GlcNAc) that are connected to each other to form complex networks. The protein IcaA catalyzes the polymerization and might therefore be considered the most important Ica protein. Is has in fact been shown that with an erased IcaA gene, no biofilm at all is being produced by Staphylococci [12]. Other Ica proteins (B, C and D) are assumed to be responsible for optimization of production and transport of polymerized GlcNAc and its cross-linking [11].

The substate of IcaA is UDP-N-Acetyl-Glucosamine, the UDP of which binds in the glycosyl-donor site of the enzyme. N-Acetyl-Glucosamine is then transferred to a mono- or polymer of N-Acetyl-Glucosamine in the glycosyl-acceptor site of IcaA [13, 10]. Thus, a chain of GlcNAc molecules (Figure 2.7) is created and elongated by IcaA.

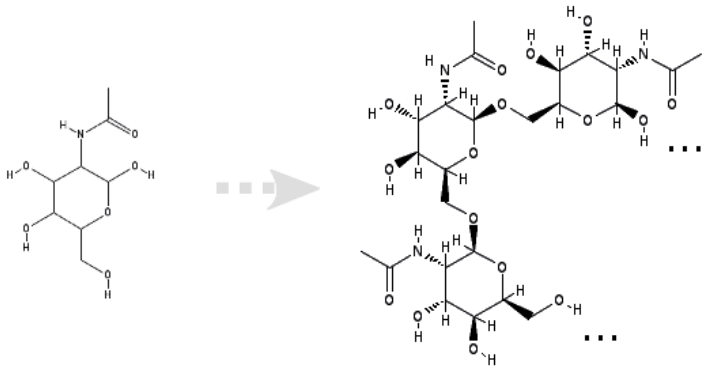


Figure 2.7: N-Acetyl-Glucosamine (GlcNAc; left), one moiety of the substrate of IcaA, and polymerized GlcNAc as the product of IcaA (right).

All this makes IcaA a good, although not perfect, target for drug design, according to the criteria explained in Section 2.1.2: It is an enzym that exists only in bacteria and not in humans and also has no orthologue in humans. Its significance for the disease, as explained above, has been well established and its mode of action is known. Problematic on the other hand is that IcaA is expected to be a membrane-bound protein (see Chaper 8 for details), and as such probably hard to express, purify and crystallize, and the fact that no crystal structure of IcaA exists, yet.

In Chapter 8 we describe how we searched for IcaA inhibitors using computer-aided drug design and experimental validation procedures. Such inhibitors would, due to the described function of IcaA, be very helpful in preventing bacterial biofilms and could hence alleviate the common problem of nosocomial infections. The most likely mode of application for them would be a coating of medical implants, so that these potential drugs would be available in the appropriate location in the body without need for systemic application. Chapter 8 will also show that we obtained a number of experimentally confirmed hits that may even serve as leads for such inhibitors.

3 Computational Background

3.1 Overview of computer-aided drug design

Computer-aided drug design (CADD) is, as explained in Section 2.1, very important for the development of new drugs. Especially during lead discovery and lead optimization steps it can help to drastically reduce the search-space, i.e. the number of molecules to be experimentally tested, thus strongly lowering financial costs and the amount of time necessary to develop a new drug. The point that traditionally, i.e. without the help of computer-aided drug design, identifying a drug candidate and performing all optimization, test and clinical trail steps (as detailed in Section 2.1) can easily take ten to 15 years and cost more than a billion US dollars [3], clearly visualizes the significance of reducing the development time by potentially several years. Furthermore, computer-aided drug design may also establish molecules as promising drug candidates that would never have been tested without computer-based methods, due to either the huge search-space or their initial unavailability in synthesized form. A special case of the latter reason is the *in silico* construction of new molecules, i.e. compounds that have not been observed in nature but were constructed on a computer manually or by an algorithm designed for this purpose.

In essence, computer-aided drug design approaches try to predict properties and actions of chemical compounds by a variety of techniques, so that molecules that are unlikely to experience the desired effect on the chosen molecular target (see Section 2.1.2 for target identification) can be cast aside. Examples of such molecular properties are absorption, distribution, metabolism, excretion and toxicity (ADMET). The expected effect on the molecular target, on the other hand, is usually evaluated by a prediction of the binding free energy (or binding affinity) of the compound to the target structure.

Computer-aided drug design can be divided into two major categories: ligand-based drug design and structure-based drug design. Ligand-based drug design uses information about known ligands of the target of interest, and, if available, their binding free energies to model the affinities of chemical compounds by a linear or non-linear function of their properties, particularly their topology. Structure-based drug design, on the other hand, tries to automatically place compounds into a previously determined three-dimensional structure (model) of the molecular target and scores the resulting complexes by use of various techniques.

The following two sections will explain the concepts of ligand-based and structure-based drug design, respectively, in some more detail. However, first there is the

question when to select which approach. The answer to this mainly depends on the type and quality of the available input data. If a number of compounds is already known to bind to the target of interest, and perhaps also a list of molecules experimentally verified not to bind to it is available, then ligand-based drug design might be helpful. For details, please see the next section. However, one rule is obvious so far: ligand-based drug design is not applicable to the prediction of binding affinities for a new molecular target, for which no known ligands (or only very few) exist, yet. This in practice of course is a considerable disadvantage of ligand-based drug design. Structure-based drug design, on the other hand, does not depend on the availability of known ligands but on the existence of a molecular structure for the target. The latter may have been obtained by x-ray crystallography, nuclear magnetic resonance (NMR), or homology modeling. Therefore, structure-based drug design procedures in practice may be applicable either if and when protein crystallography or NMR succeeds or if a crystal or NMR structure of a close homologue is available.

Beyond the above-mentioned different prerequisites, there are some more advantages or disadvantages of ligand- and structure-based drug design. In general, ligand-based approaches need much less time to predict the activity of each molecule (although the preceding creation of a QSAR model can take considerable time). They can also be

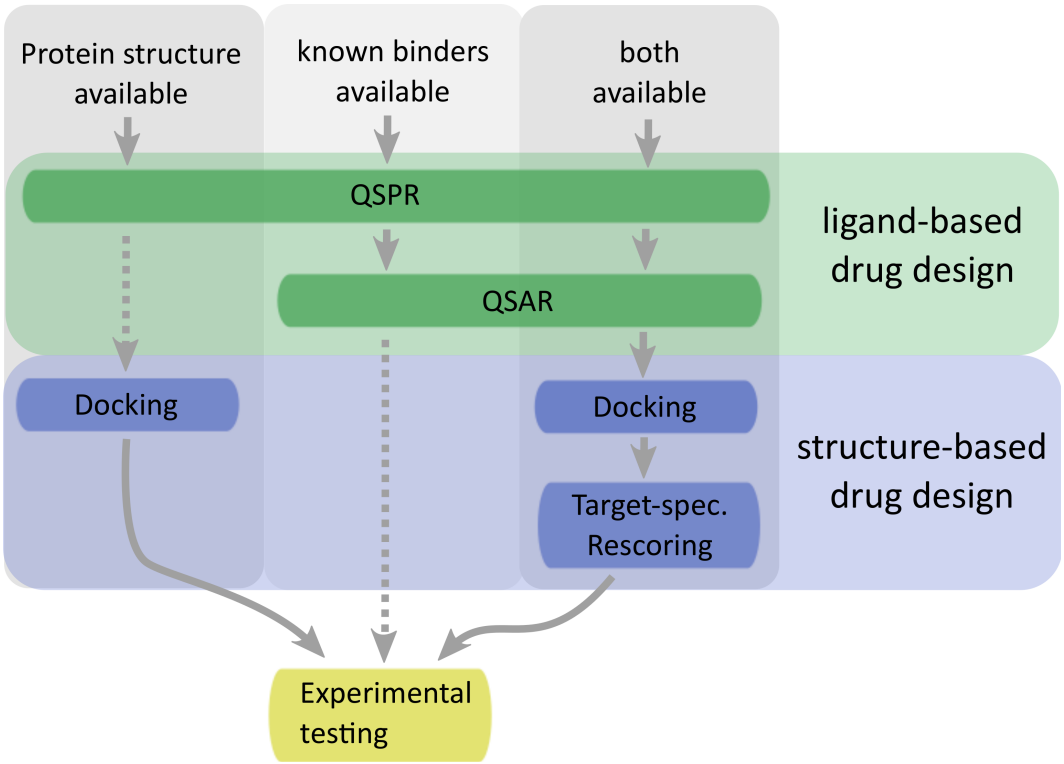


Figure 3.1: Flow chart of generally recommended application of the various computer-aided drug design techniques in dependence of the available input data. Note however, that the applicability of each step in a concrete case can significantly depend on the quality of the input data (see text for details).

used to model molecular properties, e.g. ADMET, which is not currently possible with most structure-based procedures. However, ligand-based techniques in principle are prone to problems with non-applicability due to low chemical similarity between training and prediction data set. Structure-based procedures mostly do not use any training steps and are thus immune to these troubles. In addition, it, in contrast to ligand-based approaches, generates three-dimensional information about chemical compounds inside the binding pocket that can be analyzed in order to evaluate the credibility of the obtained binding affinity estimate. Due to the direct modeling of interactions between target and putative ligand, structure-based procedures can also be considered to be better suited to differentiate between small changes in a molecule's topology. This point is particularly important for lead optimization.

Accordingly, if the necessary input data is available for both, it is often desirable to combine ligand- and structure-based techniques within one virtual screening pipeline. Especially if the dimension of the input data is very huge (i.e. millions of compounds), QSAR can thus be used in a first step to filter out molecules that are unlikely to bind to the target and subsequent docking and, if desired, rescoring can analyze the potential interactions between target structure and chemical compounds, step by step reducing the number of apt molecules. An overview of generally possible combinations of various ligand- and structure-based drug design techniques, in dependence of the type of available input data, is shown in Figure 3.1. The concepts of the different steps contained in it will now be explained in the following two sections.

3.2 Ligand-based drug design

Ligand-based drug design approaches, as previously mentioned, aim to model the action of a property of chemical compounds by a function of their topology or structure. The two most frequently used ligand-based drug design techniques can be considered to be Quantitative Structure-Property Relationship (QSPR) and Quantitative Structure-Activity Relationship (QSAR) modeling.

Properties modeled by the former include absorption, distribution, metabolism, excretion and toxicity (ADMET) or the logP, the octanol-water partitioning coefficient, of a compound. The logP estimate is commonly used to evaluate whether a molecule will be able to permeate biological membranes and thus reach the tissue or subcellular compartment where its molecular target resides. Since those properties are usually not dependent on a specific biological target, QSPR methods can be applicable even if no ligands for a target are known, yet (as indicated in Figure 3.1). However, a good, reliable and generalizable QSPR model for the property to be described is necessary for this. Problems with missing generalizability will be discussed below. The activity modeled by Quantitative Structure-Activity Relationship approaches usually is the compounds' binding free energy to the target of interest. The underlying machine learning technique are nevertheless the same for both QSPR and QSAR. Their difference lies only in the response (property vs. binding affinity) to be described by the respective approach.

Commonly, input for QSPR or QSAR consists of files describing the topology of chemical compounds and the experimentally determined response value for each molecule. Using this input, a number of so-called features or descriptors is then usually calculated. These features will be used to model the compounds' property respectively its activity. Frequently used examples are counts of different atoms types or of functional groups and indices that describe the topological complexity of a molecule or its three-dimensional structure. Anyhow, the latter is generally not that important or helpful for ligand-based drug design, since the actual three-dimensional structure of the compounds when bound to the receptor usually is not available to the QSPR/QSAR method, as docking (if any) is performed after the QSPR/QSAR analysis (as depicted in Figure 3.1).

The generated descriptors, whose number might vary from only a few dozen to several thousands, can then by used by a plethora of machine learning techniques to model the compounds' property, respectively its activity. If no binding free energy measurements are available but just information about known binders and non-binders of the target structure is at hand, classification algorithms can try to learn the characteristics that separate them. On the other hand, if the binding free energies of the compounds in the training data set are known, regression techniques may be able to model the response variable by, depending on the respective approach, either a linear or non-linear combination of the previously created features. The different classification and regression models implemented, evaluated and used by us in drug design pipelines are described in Section 3.6.1 and 3.6.2, respectively. In order to select the model type best suited for the data set at hand, a nested evaluation (see Section 3.5.1) should be done for each available and applicable type. A model of the type that achieved the best quality this way can then be trained and used to predict the property, respectively the binding free energy, of compounds for the target of interest.

Out of all generated features, usually just a relatively small number is sufficient to describe the response variable. Furthermore, a lower number of descriptors makes the created model more interpretable, which is very important in order to be able to manually verify it and to have the chance to infer general distinct rules that appear to govern the compounds' property, respectively their activity. This reduction in the number of descriptors can be achieved by a variety of feature selection procedures, as described in Section 3.6.3. Often several such techniques are applied in succession, which can strongly enhance the quality of the resulting model and can also significantly reduce run-time. Note that during a nested evaluation, these feature selection steps are done for each sample.

After a model has been created, its predictive quality should be evaluated using data on which this model was not trained. The sampling techniques described in Section 3.5 can be used for this. All of those approaches generate samples that each consist of one training and one test set. For each sample, the model is trained on the training set and the response values predicted by it thereafter for the compounds of the test set are compared to the experimentally determined values. The quality of the prediction for the current sample is then calculated according to one of the statistics described

in Section 3.4. The average over the qualities obtained for each sample is then used as estimate of the model's predictive quality.

However, if feature selections that themselves use a sampling technique have been used to obtain the model at hand (which is the case for most feature selection; for details see Section 3.6.3), then a nested evaluation has to be performed in order to obtain an unbiased approximation of the model's predictive quality. During a nested evaluation, the input data is split into multiple samples using one of the techniques described in Section 3.5. Each of these samples consists of a training and a test partition. For each sample, a model is generated using only the training partition. If feature selections are to be performed for this, they also utilize only the training partition (and not the test partition). The quality of the obtained model of each sample is then evaluated by use of the respective sample's test partition. The average over all quality values (using statistics described in Section 3.4) obtained for the various samples is reported as the nested model quality.

If a model has been created, evaluated and judged to have sufficient quality, it can be used to quickly predict the binding free energy (respectively the property in case of QSPR) of a different data set. Thus, for example, a QSAR model can be created for compounds whose binding free energies to the target of interest have been experimentally determined and this model can later be employed to swiftly predict the binding free energy of other compounds, for which no experimental measurements are available.

3.3 Structure-based drug design

Structure-based drug design uses the three-dimensional structure of the target of interest in order to find compounds that likely bind to its binding pocket and could thus be good drug candidates. The three-dimensional structure can be obtained by either x-ray protein crystallography, nuclear magnetic resonance (NMR), or by homology modeling. The latter technique employs the structure of a homologous protein that has been determined by one of the former procedures.

Algorithms for the field of structure-based drug design mainly consist of receptor-ligand docking and rescoring approaches.

The goal of receptor-ligand docking is to predict the pose of a ligand in the binding pocket of a receptor, given only the 3D coordinates of the latter and the topology (or input conformation) of the former. Therefore, docking approaches usually consist of a scoring function that evaluates the interaction energy of each (intermediate) pose and an algorithm that generates many different poses to be evaluated by the scoring function. Scoring functions can generally be divided into knowledge-based and empirical ones. While the former use an inversion of the Boltzmann factor [14] to calculate scores from the frequency of different observations, the latter employ a number of (often physically motivated) terms whose coefficients are optimized using a specific data set with known binding free energies.

A well-known example for knowledge-based scoring functions is the Potential of Mean Force (PMF) [15]. Using the assumption that chemically favorable receptor-ligand interactions appear more often in co-crystal structures than unfavorable interactions, a statistical analysis of the frequencies of receptor-ligand atom pairs within different distances is performed. Thereby, a probability p for a ligand atom of type i to appear within a given distance of a receptor-atom of type j is derived. An approximation of the binding free energy is then obtained by evaluating each receptor-ligand atom pair, $\Delta G = \sum_{i,j} -k_B T \ln(p_{ij}(r))$. DrugScore, a modification of this approach, was later developed by Gohlke et al [16], adding a knowledge-based solvation term to the scoring function. The solvation contribution is attained by calculating the probability of each ligand atom type to be solvated via comparison of the average solvated and desolvated fraction of the solvent-accessible surface of atoms of the respective type in the training data set. One of the first empirical scoring functions was LUDI [17], which uses a linear combination of terms evaluating hydrogen bonds, lipophilic contact surface area and the number of rotatable bonds of the ligand. Modifications of this scoring function later resulted in, among others, ChemScore [18], FlexXScore [19] and Glide [20]. ChemScore in effect adds a term for metal interactions to the LUDI function. Glide and FlexXScore then modify the ChemScore function in different ways. While FlexXScore adds an evaluation of aromatic interactions, Glide includes terms for polar-hydrophobic and van der Waals interactions and for solvation.

Docking algorithms, on the other hand, can be separated into those that use a stochastic sampling (e.g., genetic algorithms) and deterministic approaches. An example of the former category is AutoDock [21], which docks compounds using a Lamarckian genetic algorithm. FlexX [19] and Glide [20] are well-known representatives of the second group.

The goal of rescoring is to generate an estimate of the binding free energy that is more accurate than one provided by the score produced by docking, using the docking result as a start point. Most available rescoring methods can usually be classified into one of the following three groups: Members of the first group scale the score generated by docking in some fashion. A frequently used example for this is the scaling with respect to the number of heavy atoms of the ligand [22]. While this might seem to be a tantalizing approach, it of course only makes sense if the scoring function does not already penalize very large ligand molecules. Besides alleviating such possible shortcomings of a scoring function, no other enhancement of the score is possible this way. Another group of rescoring procedures tries to enhance the results by using consensus-scoring [23–25]. Therefore, several scoring functions are employed and the scores are averaged in some way. This approach is sometimes used because weaknesses of one scoring function can be mitigated by the scores computed by the remaining functions. At the same time, though, the result of the scoring function that performs best on a particular target is deteriorated. Furthermore, if the applied scoring functions are significantly collinear, even using the consensus might result in no quality increase. The third group of rescoring approaches modifies the scoring function used during docking in order to (hopefully) attain higher quality. Usually new, computationally expensive scoring terms are added or existing ones are replaced by more

complex ones [26, 27]. Although the average approximation of binding free energy may be enhanced by those new or modified scoring terms, no target-specific rescoring can be done this way. For different protein target families, the importance of different contributions to the overall binding free energy might vary significantly and, even more important, it may depend on the three-dimensional location of the interactions between receptor and ligand. Therefore, for example, for one target the existence of hydrogen bonds and their strengths in exactly defined regions may be important, while for another target the electrostatic interactions between charged groups of ligand and receptor play a predominant role. One approach that alleviates those problems and allows for target-specific rescoring will be described in Chapter 6.

3.4 Quality statistics

3.4.1 Coefficient of determination

As a quality statistic for regression approaches, we use the coefficient of determination, commonly abbreviated as Q^2, to assess the model's predictive quality,

$$Q^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

where n is the number of compounds in a test data set, y_i the expected, $\hat{y}_i$ the predicted activity value for compound i, and $\bar{y}$ the mean of the activity.

If identical data sets are used for training and testing, the above statistic will be called R^2, measuring the model's quality of fit to the training data instead of its prediction quality.

3.4.2 Quality of classifications

For classification approaches, we make use of the following statistics in order to evaluate their prediction quality, where TP is the number of true positives, FP the false positives, TN the true negatives, FN the false negatives and c the number of classes:

Sensitivity (SE) = $\frac{TP}{TP+FN}$
Specificity (SP) = $\frac{TN}{TN+FP}$
average Accuracy (ACC) = $\frac{1}{c}\sum_{i=1}^{c}\frac{TP_i}{TP_i+FP_i+FN_i}$
overall Accuracy (oACC) = $\frac{1}{n}\sum_{i=1}^{c}TP_i$
average Matthew's Correlation Coefficient (MCC) =

$$\frac{1}{c}\sum_{i=1}^{c}\frac{TP_i \cdot TN_i - FP_i \cdot FN_i}{(TP_i+FP_i)(TP_i+FN_i)(TN_i+FP_i)(TN_i+FN_i)}$$

overall MCC = $\frac{TP \cdot TN - FP \cdot FN}{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}$

3.4.3 Receiver operating characteristics curves

Receiver operating characteristics (ROC) curves make it possible to evaluate the performance of an algorithm in dependence of a variable parameter with respect to both sensitivity and specificity. Therefore, they consist of a plot of sensitivity (usually on the Y-axis) against 1-specificity (commonly on the X-axis).

ROC curves are in general created by either modifying one (or several) parameters of the algorithm under investigation or by using many different data sets and in each step calculating sensitivity and specificity. In the context of computer-aided drug design, we will modify the score threshold by which compounds are classified as putative binders or non-binders. The score, to which this threshold is applied, is generated by most computer-aided drug design algorithms as an estimate of the compound's binding free energy to the target structure. Thus, our algorithms will only be run once for each data set and a varying threshold is applied to their output.

The goal of using ROC curves for our computer-aided drug design algorithms is to analyze the fraction of binders (sensitivity) in dependence to the fraction of non-binders (specificity) that would be selected as putative ligands (according to their score) with a varying score threshold. ROC curves thus, among others, show whether the classification between binders and non-binders obtainable with our algorithms deviated significantly from random performance, which can be visualized by a diagonal line. Furthermore, they allow to see in which part of the rank-list (i.e. the list of compounds sorted ascendingly according to the score assigned by the algorithm) the highest enrichment of binders was achieved.

In order to be able to more easily compare ROC curves or compute the average quality of an algorithm over several data sets, several metrics can be used to compute one value for an entire ROC plot. The perhaps most commonly used metric is the AUC, the area under the ROC curve, which we will use to evaluate the quality of our docking and rescoring algorithms. In the context of drug design, the AUC is calculated as the average relative rank of binders [28]

$$AUC = \sum_{i}^{n} 1 - r(i)$$

where i is the index of the respective binder and $r(i)$ is its relative rank. The rank of a molecule is defined as its position within the list of all compounds (including non-binders) of the data set, sorted ascendingly according to their scores assigned by the CADD algorithm. The relative rank, on the other hand, is the fraction of non-binders that has been assigned a better score than the current binder.

Thus, in the best case, i.e. if a CADD algorithm could perfectly distinguish binders from non-binders, an AUC of 1.0 would be obtained. In case of random results the value of the AUC would be 0.5.

3.4.4 Enrichment factors

Enrichment factors compare the number of active compounds ($actives_{obs}$) observed within a certain top-scored fraction of a docking result against the number of actives expected in this fraction due to random distribution

$$enrichment(x) = \frac{actives_{obs}}{x \cdot actives_{tot}}$$

where x is the fraction of the docking result, sorted ascendingly according to assigned scores, and $actives_{tot}$ is the total number of active compounds in the given data set.

3.5 Sampling techniques

3.5.1 Cross-validation

Cross-validation partitions the input data into k evenly sized samples, where k is the user-defined number of folds. Each of those folds thereafter contains $\frac{n}{k}$ compounds, where n is the number of compounds in the input data set.

In total, the model is then trained k times. Each time one fold serves as test data set and the remaining folds together make up the training data set. The average quality of the k predictions is used to describe the prediction quality of the model (Figure 3.2).

3.5.2 Boot strapping

In contrast to cross-validation, bootstrapping [29] does not partition the input data, but creates groups by randomly drawing compounds with replacement from the input

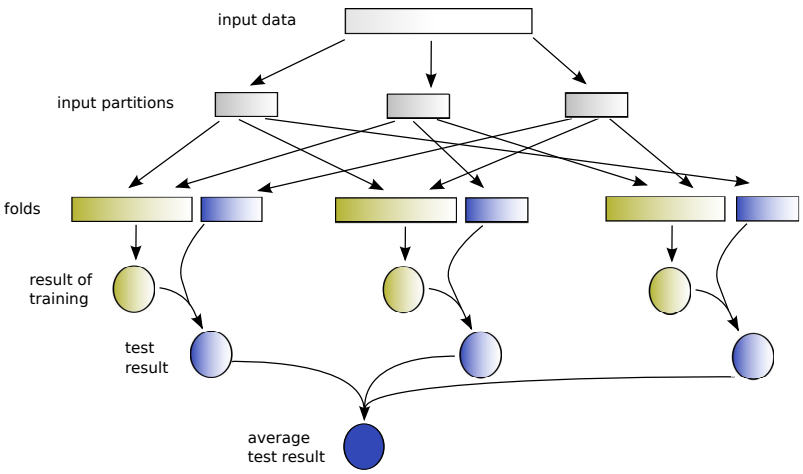


Figure 3.2: Schematic illustration of three-fold cross-validation. Training partitions are shown in yellow, test partitions and results in blue.

data set. The groups are usually called bootstrap samples and each of them has the same size (n) as the input data set. Since the drawing is done with replacement, a high number of bootstrap samples can be created, even for a relatively small input data set.

The predictive power of the model is evaluated successively for each bootstrap sample. Therefore, the model is trained each time on a bootstrap sample and then tested with those compounds on which it was not trained, i.e. which are not part of the current bootstrap sample.

However, since the probability of a compound i to be chosen for a bootstrap sample b is [29]

$$Pr(i \in b) = 1 - \left(1 - \frac{1}{n}\right)^n \approx 1 - e^{-1} \approx 0.632$$

the effective size of the bootstrap samples is reduced in comparison to the input data set by this factor, leading to underestimation of the model's predictive power.

To alleviate this problem, we use the "0.632 estimator" [29] (Figure 3.3), which weights the test results as described above by 0.632 and then adds the average quality of fit to the training data weighted by 0.368.

3.5.3 Response Permutation Testing

Cross-validation and bootstrapping, as described above, provide an estimate of the predictive power of the model but they do not assess the statistical significance of this predictive power [30, 31]. Response permutation testing thus repeatedly and

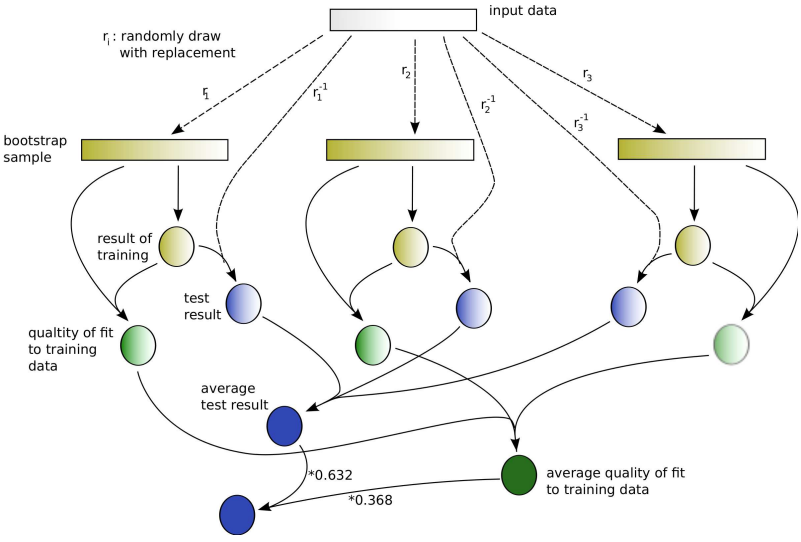


Figure 3.3: Schematic illustration of bootstrapping with three samples and the "0.632 estimator". r_i denotes a random drawing with replacement from the input data set, while function r_i^{-1} returns those compounds that have not been drawn by r_i.

randomly permutates the response values and checks the predictive quality after each permutation by use of cross-validation.

However, it might happen that sometimes relatively high predictive qualities are obtained during this procedure as a result of structural redundancy of the data set or chance correlation [32]. Nevertheless, if the predictions of the model are statistically significant, the prediction qualities obtained with randomized response variables should in most cases be much lower than those obtained with unchanged response variables. If, on the other hand, nearly all results of the response permutation tests are similar to those attained with unchanged response variables, this indicates that this model does not have significant predictive power for the current data set.

3.6 Machine learning

3.6.1 Classification approaches

In those cases where appropriate prediction of an activity cannot be achieved by either linear or non-linear regression, a classification might be a useful resort. The response variable could thus be discretized so that the applied classification approaches would predict, e.g., whether a compound binds strongly, relatively weakly, or not at all to the target structure of interest.

Furthermore, classification approaches are also helpful if no measurements of affinities (IC_{50}, K_i, etc.) are available for a given data set but results indicating whether or not each examined compound shows the desired activity.

Thus our software package also provides several classification approaches, which we describe in the following.

Naïve Bayes

Naïve Bayes [29] is a simple probabilistic classification approach that considers all features to be independent of each other. It calculates for each compound the probability that it is part of a certain class as the product over the probabilities of all feature values to belong to this class.

Since chemical properties are basically continuous values, they need to be discretized first, e.g., using equal-width discretization:

$$r(x_i) = \frac{(x_i - min_i)s}{max_i - min_i}$$

where s is the number of discretization steps to be used, min_i the minimal and max_i the maximal value of feature i within the training data set.

Naïve Bayes thus assigns the class k to a given compound x that maximizes the product over the probabilities of all m features as

$$class(x) = arg\ max_k\ p(k) \cdot \prod_{i=1}^{m} p(x_i|k)\ .$$

Simple Naïve Bayes

Simple naïve Bayes is a modification of naïve Bayes that does not need to discretize the features. Instead, it uses the normal distribution as a probability density function (pdf) in order to obtain a score for a given value x_i of a feature i to be derived from class k

$$pdf(k, i, x) = \frac{1}{\sigma_{i,k}\sqrt{2\pi}} \cdot exp\left(-\frac{(x_i - \mu_{i,k})^2}{2\sigma_{i,k}}\right)$$

where $\mu_{i,k}$ denotes the mean and $\sigma_{i,k}$ the standard deviation of feature i found within class k of the training data set.

This score can be converted into a probability by dividing it by the sum of all pdf values for the same feature value so that a given compound is assigned to class k for which the product over the probabilities of all m features is maximized,

$$class(x) = arg\ max_k\ p(k) \cdot \prod_{i=1}^{m} \frac{pdf(k, i, x)}{\sum_{j=1}^{c} pdf(j, i, x)}$$

where c is the number of classes.

Linear Discriminant Analysis

For a given compound, linear discriminant analysis (LDA) [33] assigns the class that is nearest to this input, taking into account the different covariances of descriptors,

$$class(x) = arg\ min_k\ (x - \mu_k)\Sigma^{-1}(x - \mu_k)^{-1}$$

where μ_k is the vector of the mean values of the descriptors for class k and Σ^{-1} is the inverse of the covariance matrix of descriptors.

Thus, linear discriminant analysis minimizes the weighted distance between a test compound and the mean of its predicted class; it is weighted in such a way that descriptors with large variance/covariance contribute less to the distance than those with small variance/covariance.

3.6.2 Regression approaches

Linear Regression

Linear regression techniques try to model one or more continuous response variables as a linear function of features. In the case of QSAR applications, the biological activity of a compound is approximated by a linear function of descriptor values x_i and coefficients b_i

$$\hat{y} = b_0 x_0 + b_1 x_1 + ... + b_m x_m \tag{3.1}$$

where m is the number of descriptors. The coefficients are derived by different linear regression techniques described below and each conveys a weight to a specific descriptor. Thus, interpreting linear regression models is relatively easy, which is a great advantage over nonlinear models as long as the number of descriptors is not too large. If a very large number of descriptors is available, it is thus very important to select them carefully, as will be explained later.

Examples of well-known linear regression models include:

Multiple Linear Regression (MLR) minimizes the residual sum of squares, i.e. the sum over all squared errors of activity predictions.

Ridge Regression (RR) [34] is an extension of Multiple Linear Regression which is better able to cope with multicollinearity of descriptors. It therefore adds a constant λ (usually 0.01 or less) to the diagonal of the variance-covariance matrix of descriptors.

Principle Component Regression (PCR) [35] uses singular value decomposition to reduce the dimensionality via calculation of principle components that explain most of the variance of the given data set, followed by projection of input data onto these principle components. Multiple Linear Regression is then applied to the principle components.

Partial Least Squares (PLS) [36] reduces the dimensionality through latent variables computed in such a way that their covariance with observed activities is maximized. We use the non-linear iterative partial least squares algorithm (NIPALS) [36] for this purpose.

Orthogonal Partial Least Squares (OPLS) [37] is an extension of PLS that eases the analysis of the regression result if large variation uncorrelated to the response variables (orthogonal variation) is present in the descriptor matrix. Therefore, a modified NIPALS algorithm [37], which computes PLS components with minimal covariance to the response variables and subtracts them from the descriptor matrix, is applied prior to the unmodified NIPALS algorithm.

Although in many cases the desired activity can be modeled quite well by a linear function of descriptors after selection of necessary features (see below), sometimes this cannot be achieved. This may either be due to chemical features important for

binding for which no descriptors have been computed or to the activity of interest depending on a more complex function of the structure and topology of the ligands.

There are two different general approaches for approximating higher-dimensional relationships: locally weighted linear regression and kernel-based regression techniques, both of which will be described in the following.

Locally Weighted Linear Regression

To predict activity, locally weighted linear regression approaches use a linear function of descriptors that is weighted with respect to the chemical similarity of the compounds of the training data set to the compound whose activity is to be predicted.

Thus, these models minimize the locally weighted residual sum of squares instead of the residual sum of squares

$$E = \sum_{i=1}^{n} w_i^2 (y_i - \hat{y}_i)^2$$

where w_i is a weight factor based on the chemical similarity of the compound whose activity is to be predicted to the i'th compound of the training data set. Different ways of obtaining the weights are shown below.

This minimization is then achieved through a modification of ridge regression that includes a diagonal matrix W comprised of the weights, so that a coefficient-vector containing one coefficient for each feature can be obtained as

$$\vec{b} = (X^T W X + I\lambda)^{-1} X^T W \vec{y} \; . \tag{3.2}$$

It is noteworthy that due to the use of these weights, there is no training of a locally weighted linear regression model that is separate from the prediction of activities, i.e. an individual regression is done each time an activity is to be predicted. Thus, there also is no single resulting vector of coefficients, as is the case for linear regression approaches (as described above), which could be analyzed in order to find chemical properties especially important for the modeled activity.

Automated Lazy Learning (ALL) [38] first calculates distances between the compound t whose activity is to be predicted and all compounds i of the training data set in terms of Euclidean distances of their descriptor values

$$d_i = \sum_{j=1}^{m} (X_{ij} - t_j)^2,$$

where m is the number of features.

These distances are then transformed into weights by use of a Gaussian

$$w_i = exp\left(\frac{-d_k^2}{2K^2}\right).$$

Here, K is the kernel width, determining how fast the weight decreases with increasing distance. This parameter can be optimized for each data set using, e.g., cross-validation or bootstrapping. The activity of the compound for whom the weights have thus been calculated can then be predicted as shown in Eq. 3.2.

k-Nearest Neighbor (KNN) regression uses only the k compounds that are chemically most similar to the compound whose activity is to be predicted.
It can thus be seen as a specialization of ALL, generating weights which are either 0 or 1, so that

$$w_i = \begin{cases} 0, \ if \ d_i < s_k \\ 1, \ else \end{cases}$$

where s contains the distances, computed as shown above, sorted in descending order.

Kernel-Based Non-linear Regression

Kernel-based non-linear regression methods do not directly use the descriptor matrix X to perform a regression, but a kernel matrix K that constitutes a mapping of X into higher dimensional space. The basic reason for doing this is that linearly non-separable data is often linearly separable after having been mapped into a higher dimension.

Therefore, descriptor matrix X is mapped to the implicitly high-dimensional kernel matrix K

$$\kappa : X \rightarrow K$$

by way of a non-linear kernel function κ calculating one value for each combination of rows of X

$$K_{ij} = \kappa(X_i, X_j).$$

We provide three different kernel functions that can be used for non-linear regression models:

1. polynomial kernel function $\kappa(X_i, X_j) = (X_i \cdot X_j^T)^d$, where $d > 1$.
2. radial basis kernel function $\kappa(X_i, X_j) = e^{-\gamma ||X_i - X_j||^2}$, with $\gamma > 0$.
3. sigmoid kernel function $\kappa(x_i, x_j) = tanh(cx_i x_j^T + d)$, where $c > 0$, $d < 0$ and $tanh$ denotes the hyperbolic tangent.

The different kernel-based regression approaches supplied with our programs are described in the following. Training results in coefficients for all compounds of the training data set, instead of coefficients for the descriptors, as is the case with linear regression approaches (see above).

Regardless of which kernel-based non-linear regression model is used, its quality substantially depends on the choice of kernel and kernel parameters. Only when a kernel

that adequately approximates the overall relationship between activity and descriptor values is chosen, optimal results might be achieved.

Kernel-Based Ridge Regression

Kernel-based ridge regression (KRR) first transforms the descriptor matrix X into a kernel matrix K by use of one of the kernel functions described above.

The result of the regression in the form of one coefficient for each compound of the training data set can then be obtained by

$$\vec{b} = (K + I * \lambda)^{-1}\vec{y} \; .$$

Kernel-Based Principle Component Regression

After transforming X into a kernel matrix K, kernel-based principle component regression (KPCR) reduces the dimensions of K by the application of singular value decomposition and creation of latent variables, so that most of the variance of the kernel matrix is explained by the latter.

Kernel-Based Partial Least Squares

Kernel-based partial least squares (KPLS) transforms the descriptor matrix into a kernel matrix K and calculates PLS components from K in such a way that their covariance with the response variables is maximized. This is done by use of the NIPALS algorithm [36] as mentioned above.

3.6.3 Feature selection

To receive easily interpretable training results and to reduce overfitting, it is very important to reduce the number of descriptors used by the model.

The goal therefore is to combine only those descriptors that result in the highest predictive power of the model. In theory one might try to assess all possible combinations of descriptors in order to find the one that yields the highest possible prediction power. But since the number of combinations to be evaluated this way is given by

$$\sum_{i=1}^{m} \binom{m}{i} = 2^m - 1$$

where m is the number of descriptors of the data set, we can clearly see that this is in practice infeasible due to its immense complexity.
Instead, a number of heuristics, which will be described in the following, can be utilized.

Forward Selection

Forward selection [39, 40] starts with no features, and in each step adds the feature that results in the largest increase of predictive quality, which is estimated by use of cross-validation.
Feature selection is stopped if either the predictive quality cannot be increased by more than a given minimum or all features have been selected.

Backward Selection

Backward selection [39, 40] starts with all features and in each step removes the feature whose removal results in the largest increase of predictive quality as estimated by cross-validation.
The procedure terminates once the predictive quality cannot be increased by more than a given minimum or only one feature remains selected.

Stepwise Selection

Stepwise selection [39, 40] can be used to combine forward and backward selection. In our software, it consists of a forward selection that, after adding a feature, always applies backward selection so that the properties that have become unimportant can be removed.

Removal of collinear features

Highly collinear features can be removed directly from the given model without need for assessment with cross-validation or bootstrapping.
Thus all features i that have a correlation coefficient $cor(i,j)$ to at least one other feature j, for which $|cor(i,j)| > t$, where t is a given threshold, are removed from the model.

TwinScan

TwinScan performs a simple check consisting of two successive scans of all features. In the first scan, the best single feature to start with is searched. Cross-validation is therefore used to assess the predictive quality of each feature. In the second scan, it is checked for each remaining descriptor whether it can increase the prediction quality. Here, the features are tested in descending order according to their predictive quality as determined in the first scan.

Thus, this method is particularly suitable as a fast heuristic for very large data sets or as a first filtering step in the case of several feature selection procedures being employed successively.

Removal of insignificant coefficients

For linear regression models it may be desirable to remove features with insignificant, i.e. very small or highly variable, coefficients. To evaluate the variability of coefficients, a bootstrapping is done. All features whose absolute coefficient value is smaller than d times its standard deviation are removed, with d being a user-defined threshold.

Since features are discarded whose coefficients vary a lot for slightly different training data sets, the resulting model is conceivably more stable, i.e. less prone to overfitting.

Removal of low response correlation

In the case of very large data sets, especially those with very many features, it can be helpful to remove those features that show very low correlation with response variables.

Since all described regression approaches do not consider features to be independent of each other, only a very small threshold (e.g. ≤ 0.1) can be chosen without considerably decreasing the quality of the resulting model. Nevertheless, this still often results in removal of a substantial number of features, thus decreasing run-time of further analyses.

3.7 Multi-greedy heuristics

Multi-greedy algorithms [41] are heuristical optimization techniques that are very useful if a high number of conditions has to be evaluated. Instead of enumerating all conditions, multi-greedy approaches can then used.

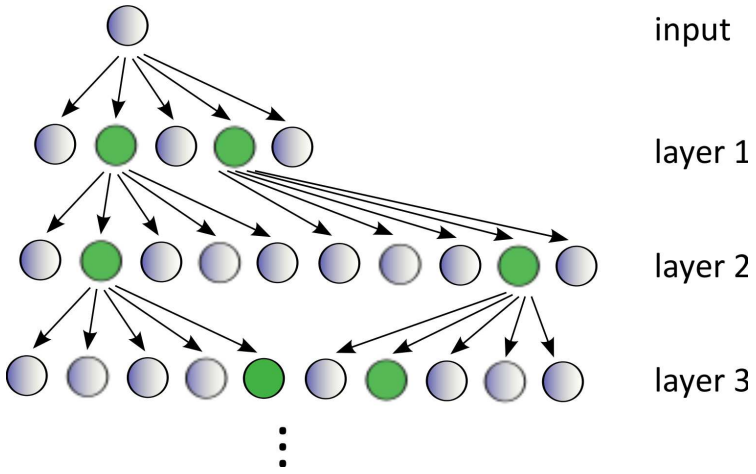


Figure 3.4: Schematic visualization of a multi-greedy heuristic. The best conditions of each layer are highlighted in green.

Multi-greedy heuristics work in several layers. In each layer, an external algorithm is used to generate a set of new conditions (e.g., ligand poses), along with a score for each of them, for the set of input conditions. Then, only the k best conditions (with k being a user-defined constant) are retained for use in the next layer and all others are not examined further. Usually, one multi-greedy layer is used for each variable to be optimized (e.g., the rotation angle of one bond in case of docking).

This way, the total number of generated conditions depends only linearly instead of, as in the case of trivial enumeration, exponentially on the number of employed layers. A schematic visualization of this procedure is shown in Figure 3.4.

4 QSAR approaches for ligand-based drug design

4.1 Introduction

Quantitative structure-activity relationship (QSAR) modeling is, as has been detailed in Section 3.2, very important for computer-aided drug design and can help reduce a list of putative drug candidates by predicting the binding free energy of those compounds to the molecular target of interest by a function of molecular properties.

A large number of machine learning techniques exist that can be used to derive a linear or non-linear function for this. Furthermore, there is a variety of feature selection and model validation approaches. Only by evaluating different modeling techniques on a given data set and applying several feature selection and model validation procedures can the probability of obtaining a good, interpretable and stable model be maximized.

However, while there are several software packages [42–45] available for QSAR research, most of them provide just a small range of the necessary functionalities and are also not extendable. Often just one (or very few) types of regression models are made available by those programs, so a comprehensive comparison of many different approaches is impossible. Furthermore, feature selection is often ignored even though it is absolutely vital for obtaining interpretable and stable models. Another major problem is the common lack of advanced model validation methods. A simple cross-validation does not suffice to estimate over-fitting and is unable to detect chance correlation. Nested validation (as described below) is probably the most reliable method to achieve the former but is also not available in most QSAR software packages. Additionally, most of these programs are not made available as open source, so that extending their functionality or adapting them to specific needs is not possible for anyone but their initial authors. Last but not least, robustness and speed of performed computations are seldom a focus of attention, making many applications unsuitable for use with large data sets, which would however be essential for lead identification projects since they often require the screening of huge data bases of chemical compounds (more details about lead identification can be found in Section 2.1.3).

Thus, we implemented a large set of well-established regression and classification models, feature selection and model validation techniques into one framework, which is presented in this chapter, that makes all techniques easy to use, open source, fast and efficient, numerically stable, and easily extendable. Furthermore, methods for reading of input and generation of molecular descriptors are provided, so that this

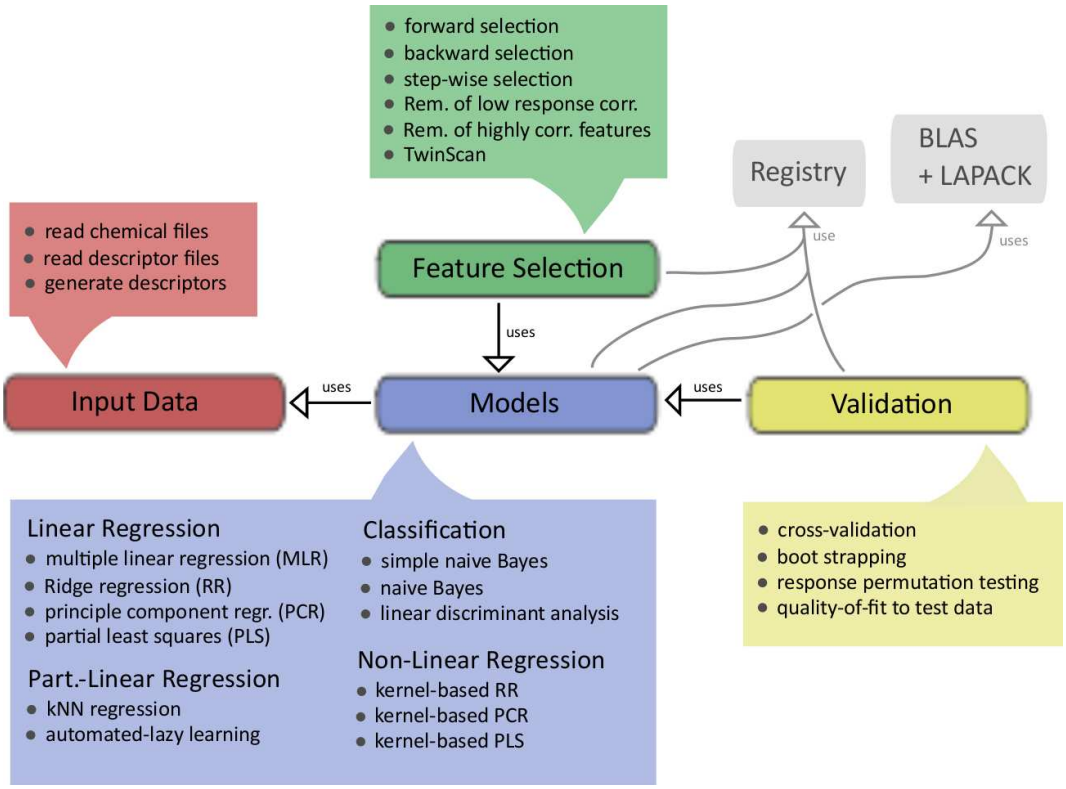


Figure 4.1: Overview over the modules provided by our QSAR framework.

framework contains all functionality necessary for reliable large-scale application of QSAR analysis, as required in the context of computer-aided drug design. Since, in addition to a library, a set of tools is supplied, these tasks can be directly solved, without any programming on the user's part, by using these programs in combination.

The tools provided by this framework are applied to a number of standard QSAR benchmarking data sets and thereby shown to achieve good modeling quality with high interpretability, i.e., with a low number of descriptors. In addition, this framework is used and shown to be very helpful in a large drug-design pipeline presented in Chapter 7.

4.2 Design & Implementation

The created QSAR framework is divided into four modules (represented by baseclasses in the actual implementation) for the following tasks: Creation and handling of input data, QSAR models, feature selection and model validation.

4.2.1 Input data module

The input data module allows reading input from files, preprocess it in the desired way and store the resulting data consisting of a set of descriptor values and one or more response values (e.g., a binding free energy) for each molecule. Input in the form of topologies and conformations can be read from files. Molecular descriptors can be automatically computed for all contained molecules. These descriptors can be divided into the following categories:

- 40 atom and bond count descriptors
- 2 connectivity indices (Balaban and Zagreb index)
- 4 partial charge descriptors
- 14 surface descriptors
- 133 topological descriptors (functional group counts)

A detailed list of all descriptors is provided in the Appendix in Table A1. Additionally, descriptors computed externally by some other program can be used by reading them from simple, comma-separated files.

Data, as read and generated by the input data module, can furthermore be centered (to a mean of zero and a standard deviation of one for each descriptor) and saved to files in order to facilitate reuse of this data by different tools, thus facilitating quick creation of workflows. Also, data can be split, randomly or evenly with respect to response values, into several data sets. This is useful for setting aside data for testing purposes or for nested validation.

4.2.2 Models module

This module supplies all implemented models under one common interface. It is divided into classification and regression models and the latter is furthermore subdivided into linear, non-linear and kernel-based non-linear models. Supplied models are furthermore listed in Figure 4.1 and their underlying principles have been explained in Section 3.6. For kernel-based methods, polynomial, radial-basis function (rbf) and sigmoidal kernel functions are available (as described in Section 3.6.2).

All models are kept track of by a registry created for this purpose, so that newly implemented models can be used by all tools without actually changing the tools themselves. Furthermore, the only significant functionality usually implemented in the childclasses representing the QSAR models is the actual training algorithm. Fetching of data, transforming, storing and loading it, etc. is performed by the respective parent classes, so that new models are easy to implement and add to this infrastructure.

The focus of the implementation of all models rested on fast, efficient and numerically stable calculations. In order to ensure the last point, vector and matrix operations have been delegated to the well-established BLAS [46] and LAPACK [47] packages. Since the popular BLAS interface is used to do this, those packages can, if desired,

be exchanged for other ones, e.g. ones with hardware-dependent optimization like the Automatically Tuned Linear Algebra Software (ATLAS) [48].

4.2.3 Feature Selection

A range of different feature selection techniques is provided by the feature selection module. All techniques operate on a connected QSAR model and, depending on the respective method, either evaluate its input data or repetitively train the model with a varying set of descriptors and evaluate its performance with cross validation. In each case, as a result of feature selection, information about which descriptors are to be used is set in the respective model object. Thus, no copying of large amounts of data is necessary and different feature selection techniques can easily be used in succession. Furthermore, all feature selection techniques are also registered in a registry, so that new ones can be implemented and later used by tools without any need to modify the latter.

Successive application of different feature selection approaches is especially important in cases of huge input data sets (containing thousands of molecules and/or features). By first utilizing computationally less demanding approaches, overall run-time can be considerably reduced and at the same time the quality of the obtained reduced models can be strongly enhanced.

Feature selection methods currently provided by this module are, listed in ascending order according to their computational complexity:

- Removal of low response correlation
- Removal of highly correlated features
- Removal of insignificant coefficients
- TwinScan
- Forward selection
- Stepwise selection
- Backward selection

The underlying theory of each of those approaches has been explained in Section 3.6.3.

4.2.4 Model Validation

The model validation module supplies a number of different evaluation techniques that can be used to assess the predictive power of a QSAR model. All approaches (repetitively) train a connected QSAR model using (parts of) the data stored in input data object bound to the latter. The response values (e.g. a binding free energy) predicted by the QSAR model are then compared by the validation technique to the expected (i.e. in most cases experimentally determined) response values annotated in the input data. The quality measure used to describe the similarity of those two sets

of values is the coefficient of determination (see Section 3.4.1) in case of regression models and one of the measures detailed in Section 3.4.2, as chosen by the user.

Model Validation techniques currently provides are cross validation, boot strapping, response permutation and evaluation of quality of fit to input data. All of them have been explained in Section 3.5. To allow for easy integration of newly implemented validation procedures, they are also tracked by a registry, just as has been explained for QSAR models and feature selections above.

4.3 Results & Discussion

We use six different data sets in order to evaluate our QSAR framework, which are described by Table 4.1.

For each data set, molecular descriptors are generated as explained above, but the last block of descriptors (topological features) has been replaced with 3,000 descriptors computed externally by *Dragon* [44]. All descriptors as well as the response variables were furthermore scaled to a mean of zero and a standard deviation of one.

The following steps are then performed for each data set and each model type:

1. Features having a correlation to another feature that is larger than 0.97 or smaller than -0.97 are removed. If applicable, model and kernel parameters are consecutively optimized by a grid search.
2. Forward selection is applied to the reduced model, using 5-fold cross-validation for estimation of predictive quality. Again, model and kernel parameters are optimized thereafter.
3. In case of linear regression models, a filtering of insignificant coefficients (see Section 3.6.3) is carried out, removing all features whose absolute coefficient value is smaller than three times its standard deviation. All parameters are again automatically optimized after this step.
4. The model obtained this way is assessed by a 4-fold nested validation. Thus, for each of the four folds, 25% of the compounds of the input data set are randomly selected as test data. The remaining compounds make up the training data of

name	abbrev.	source	cpds	min	max
Angiotensin-converting enzyme	ACE	[49]	114	-9.9	-2.1
Benzodiazepines	Benzo.	[49]	163	-8.9	-5.5
Carbonic Anhydrase II	CarbAn2	[50]	75	-6.0	0.2
Cyclooxygenase II	COX2	[49]	322	-9.0	-4.0
Dihydrofolate reductase	DHFR	[49]	397	-9.8	-3.3
Heat shot protein 90	HSP90	[50]	108	-5.3	-0.3

Table 4.1: Data sets used to test our QSAR framework. The number of compounds (cpds), as well as the minimal pIC_{50} (min) and the maximal pIC_{50} (max) of inhibitors contained in each of those data sets is indicated.

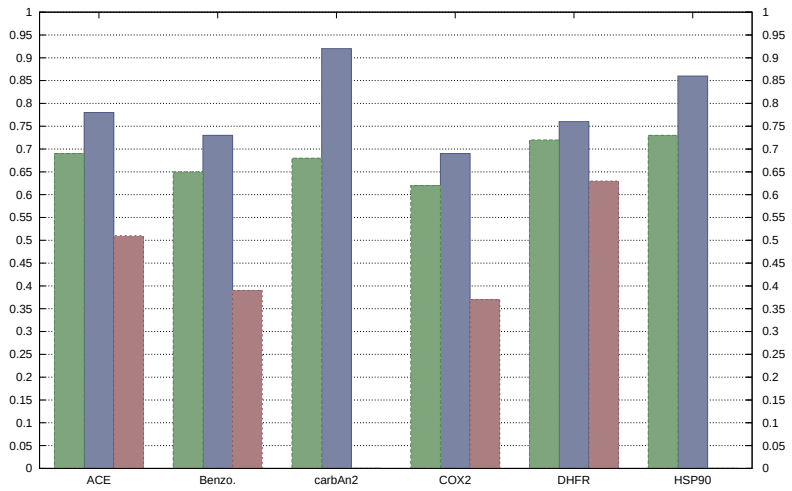


Figure 4.2: Nested validation (shown in green) and non-nested cross-validation Q^2 (blue) of our best model for each of the data sets in comparison to results obtained by Sutherland et al. [49] (where available; in red)

	ACE	Benzo.	CarbAn2	COX2	DHFR	HSP90
no. of selected features	23	31	13	19	49	16

Table 4.2: The number of selected features of our best model for each of the data sets.

the respective fold. Steps 1-3 are repeated for each nested validation fold. The average coefficient of determination over all four folds is then used as description of the predictive power of the respective model type for the current data set.

The results of our best model for each of the ACE, BZR, COX2 and DHFR data sets are qualitatively compared to the regression correlation achieved by the best respective model of Sutherland et al. [49] on their test set.

The best quality obtained by one of our modeling techniques for each of the six data sets, determined as described above, is shown in Figure 4.2. As the latter shows, it is possible to easily create QSAR models with high predictive quality, in all cases tested here outperforming those models that were published by the creators of the respective data set. Since our models were validated using multi-fold nested validation, a high observed quality is not simply due to the composition of a single test partition. Furthermore, as is detailed in Table 4.2, we utilized a relatively small number of features, which greatly facilitates interpretability of the models.

Table 4.3 shows the necessary runtimes for analyzing a data set of average size by use of the different regression procedures on a 1GHz AMD Opteron machine with use of Lapack [47] and ATLAS [48]. Theses numbers take into account all applied feature selections and nested validation as described above. As can be seen from Table 4.3, the data sets described above can be analyzed in the described way in a matter of minutes by linear regression approaches or at most in a few hours by non-linear approaches.

	MLR/RR	PCR	PLS/OPLS	ALL	KNN	GP	KPCR	KPLS
w/o opt	8	10	6	48	50	17	25	20
w opt	–	–	8	160	170	50	40	75

Table 4.3: Runtime performances (with and without model or kernel parameter optimization) in minutes for the different model types for a data set of average size (111 compounds and approximately 3,400 features). These numbers take into account all applied feature selections and nested validation as described in Section 4.3.

Since the predictive quality of a particular type of regression or classification model depends on the relationship between activity and selected features of the examined data set, it is, in general, advisable to try several different approaches. Thus, for each type of model to be evaluated, the entire pipeline including all desired feature selection steps as well as a multi-fold nested validation has to be utilized, to which our framework is ideally suited.

However, on average, the model that performed best on all data sets is PLS. It shows a high overall predictive quality and a relatively low amount of overfitting, compared to other types of models. An estimate for the latter property can be observed as the difference between the predictive qualities obtained by nested and non-nested validation, as shown in Figure 4.3. Partially weighted linear regression approaches (especially ALL) in our study exhibit very good modeling capability, which they achieve after feature selection if tested in a non-nested way. Still, they are also very prone to overfitting, displayed by much lower predictive qualities obtained by nested validation. Kernel-based regression models, on average, showed predictive quality similar to our partially weighted linear regressions. Nevertheless, both types of approaches have the disadvantage of their results being much harder to interpret than those of linear regressions.

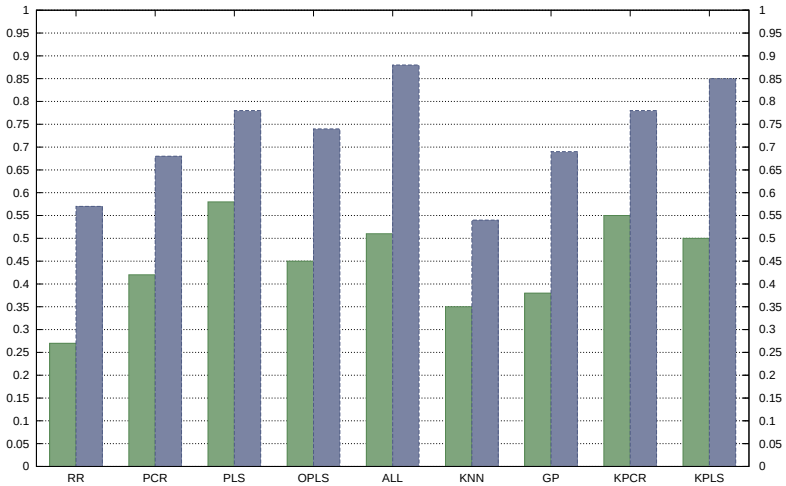


Figure 4.3: Average nested validation (shown in green) and non-nested cross-validation Q^2 (blue) of each type of regression model

Due to the smaller risk of overfitting, as well as the higher interpretability, it is thus recommended to try to create a model as simple and stable as possible, i.e. preferably using a linear regression approach and as few features as possible. As we showed here, it is often possible to obtain models with high predictive accuracy by employing this strategy. Thus, it is advisable to resort to more complex approaches only if simpler ones fail to achieve a sufficient prediction quality.

In order to make the described functionality of this framework useable in an easy way and to provide access to a number of preprocessing tools, this framework has been integrated into our computer-aided drug design suite, CADDSuite. More about this will be described in Chapter 7.

5 Receptor-Ligand Docking for structure based drug design: IMGDock

5.1 Introduction

As explained in Chapter 3, molecular docking is of great importance to computer-aided drug design since it allows to predict the binding pose and binding free energy of chemical compounds to the receptor of interest. Thus, it finds frequent use in lead identification (Section 2.1.3) and lead optimization (Section 2.1.4) steps of drug discovery projects.

In this chapter we present IMGDock, a docking approach developed with the goal of fast, scalable, interpretable, reproducible, universally applicable, and easily configurable docking.

A high speed of docking is especially important for applications in lead discovery where often hundreds of thousands or even millions of chemical compounds are to be screened *in silico*. Scalability allows to efficiently apply the algorithm to a high number of molecules and distribute the calculations to compute clusters, clouds or grids. An empirical scoring function is developed in order to ensure chemical interpretability of obtained results. The contribution of all modeled terms of interaction, like van der Waals or electrostatic, to the binding free energy can be examined separately and the pair-wise interaction between ligand and selected residues can be evaluated. Furthermore, the created pose generation algorithm is a deterministic one, so that obtained results are always reproducible, i.e. it always returns the same pose for the same pair of receptor and ligand, something seemingly trivial but of huge practical importance since otherwise exact reproduction of experiments is not possible. To ensure that the docking approach is universally applicable to a wide range of target (protein) families, the scoring function has been parameterized on a large and diverse set of protein-ligand complexes and validated on a separate, even larger and also very diverse data set. Easy configurability ensures that, if desired and the necessary biochemical information is available, the quality of docking can be enhanced by adapting it to a specific target. To this end, various scoring constraints can be generated, either manually or automatically, and used by our docking approach, thus guiding the latter towards ligands poses that are chemically more senseful with respect to the specific target.

Hence, major advantages of IMGDock over other docking programs (see 3.3) are the robustness and high speed of its algorithm, the ability to easily add different scoring

constraints, its availability free of charge as open-source software, high quality of obtained results and its simple deployment on compute clusters, clouds or grids (HPC, high-performance computing systems).

The quality of IMDGDock is evaluated using the DUD [51] library, a well-established benchmark set for receptor-ligand docking. We will show that IMGDock, even when used in a completely automatic fashion, in many cases outperforms state-of-the-art programs and even on average performs as well or better than most of them.

To ease the deployment on HPC systems, IMGDock has been integrated into our Computer-Aided Drug-Design Suite (CADDSuite) and thereby into the workflow-system Galaxy [52]. The inclusion into CADDSuite makes it possible to use IMGDock in combination with all data retrieval, preparation, checking and analysis tools provided by the former, making the usually very tedious setup of *in silico* drug design pipelines very easy and fast. All those tools together cover most areas of commonly used applications for computer-aided drug design, so that no external (commercial) software packages are required (but can be used if desired) within such pipelines. The integration into Galaxy enhances the user-friendliness even more, in effect adding a graphical user-interface and the ability to run individual tools or generate or execute workflows directly from a web browser, without any need for software installation on part of the user. It also allows to directly submit jobs to a cluster, grid or cloud and tracks all executions of jobs and their results, which makes work involving large workflows much easier and much more reproducible. Since furthermore all our tools use a modular concept and are available free of charge, eliminating the need for separate licenses for each process, IMGDock, together with CADDSuite, is therefore optimally suited for high-performance computing in the field of computer-aided drug design.

IMGDock is, as part of CADDSuite, licensed under the GPL and is available from `http://www.ball-project.org/caddsuite`. For more information about CADDSuite, please refer to Chapter 7.

5.2 Methods

5.2.1 Scoring function

We developed an empirical scoring function that contains terms for van der Waals interactions, electrostatic contributions, desolvation of the ligand, hydrogen bonds between receptor and ligand and rotational entropy.

$$
\begin{aligned}
score &= \Delta G_{vdW} + \Delta G_{ES} + \Delta G_{Solv} + \Delta G_{HB} + \Delta G_{ent} + \Delta G_{penalties} \\
&= K_{vdW} \sum_{i,j \in vdw} \frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{6}} \\
&+ K_{ES} \sum_{i,j \in ES} \frac{q_i q_j}{r_{ij} \cdot \varepsilon_{ij}} \\
&+ K_{Solv} \sum_{i \in ligand} p_i \cdot svol_i \cdot sfrac_i \\
&+ K_{HB} \sum_{i,j \in HB} f(r_{ij}, r_0) \cdot g(\alpha_{i,H,j}, \alpha_0) \\
&+ n \cdot \Delta K_{ent} \\
&+ \Delta G_{penalties}
\end{aligned}
\tag{5.1}
$$

The contribution of van der Waals interaction is computed by a standard Lennard-Jones potential as calculated by an AMBER [53] force field. For calculation of electrostatic interaction energy, a Coulomb potential, scaled by an approximation of the relative dielectric constant ϵ_{ij}, is used. The latter is obtained as the average between dielectric constant of the protein (4.0) and the dielectric constant of the solvent (80.0), weighted by the fraction fr_{ij} of the space between ligand i and receptor atom j that is taken up by solvent

$$\epsilon_{ij} = fr_{ij} \cdot \epsilon_{solvent} + (1 - fr_{ij}) \cdot \epsilon_{protein}. \tag{5.2}$$

For each ligand atom, the desolvation term computes a score as a product of the atom's solvation parameter (p_i), its solvation volume ($svol_i$) and the fraction of its volume that is desolvated ($sfrac_i$). Values for solvation parameters and solvation volumes are taken from AutoDock [54]. Hydrogen bonds are scored using a distance- and angle-dependend term (adapted from SLICK [55]). Deviations from the optimal hydrogen bond length (1.85 Å) or the optimal angle (180°) are penalized using a sigmoidal base function f in such a way that very unfavorable hydrogen bond geometries result in a contribution of zero. Very large and flexible ligands are furthermore penalized by a rotational-entropy term, so that $1 * K_{ent}$ is added for each rotational bond of the ligand exceeding the average number of rotational bonds in our calibration data set (14). A further scoring term adds a penalty value for each clash between non-bonded atoms. A pair of atoms, ligand intramolecular or receptor-ligand intermolecular, are considered to overlap if their distance is smaller than the sum of their van der Waals radii minus a predefined threshold (1.0 Å). The last term of our scoring function computes a penalty score for each constraint that has been added to be the scoring function (if any). Please see below for the description of our scoring constraints. The coefficients K of Equation 5.1 were optimized using the data set of AIScore [56]. The resulting values are shown in the Appendix in Table A2.

As an empirical scoring function, our function has several advantages over commonly used knowledge-based potentials. For its calibration, experimental data in form of binding free energy measurements could be used, whereas knowledge-based approach-

es take into account only the number of different atom types around each ligand atom. Furthermore, empirical functions are also chemically interpretable, so that the individual contribution of the terms to the binding free energy can be examined or pairwise interactions visualized. Our scoring function is also nearly completely precalculateable, allowing for score-grids to be computed and saved to a file, leading to a strong speed-up for successive docking. Only the part of the hydrogen term that evaluates interactions between hydrogen donor groups in the receptor and acceptor groups in the ligand cannot be precalculated (due to the angle-dependence of the term), so that it is automatically computed during docking and the resulting score added to the one obtained from the score-grids.

5.2.2 Preparation algorithms

Before docking ligands into a receptor it is often useful and advisable to remove all irrelevant water molecules from the receptor structure and try to find target-specific constraints that can later help to guide the docking algorithm to chemically meaningful optima.

Water placement

Crystal structures of proteins often contain hundreds or even thousands of water molecules, although only a small number of them is actually relevant for receptor-ligand docking. Water molecules that are either not bound to the receptor at all or are bound somewhere far outside of the ligand binding pocket are thus of no interest. On the other hand, water molecules that bind strongly to the receptor within the ligand binding pocket and/or to the (reference) ligand might play a significant role for establishing receptor-ligand binding.

We thus created an algorithm that searches water molecules within an input protein structure according to this criterion. Since water molecules found in (most) crystal structures contain no hydrogens (due to the resolution of the structure), waters are protonated in a first step. Water molecules that are far apart from all atoms of the reference ligand (more than 5 Å) are discarded. Then, all remaining protonated water molecules are iteratively rotationally optimized. In this step, the scoring function described in Equation 5.1 is used to evaluate the binding of each water molecule to the receptor and to other molecules. Finally, the binding of each water molecule to the receptor and to the reference ligand is evaluated using the same scoring function. All waters that either interact very strongly with the receptor (score < -5) or strongly with receptor and reference ligand (scores < -2 and < -1.5 resp.) are retained, while all others are deleted. Also, networks of water molecules (i.e. water molecules that are bound to each other) are retained if this condition is fulfilled for at least one their members.

Residue-specific interaction-constraints

Often the interaction of ligands with specific residues of the receptor is vital for its binding and the function of the enzyme. We therefore allow to set residue-specific constraints that can be used during docking in order to enhance its specificity. Those constraints can either be defined manually or created automatically. In any case, any such constraint contains information about the residues with which a strong interaction should take place, the minimal desired strength of this interaction (d), types of interactions (vdW, electrostatic, etc., according to Equation 5.1) to be considered, and a penalty factor (f). Thus, for each of those constraints a penalty score p is calculated for a ligand pose after computing the interaction energy e between the ligand and the desired residues as follows:

$$p = \begin{cases} |(e-d)/d| \cdot f & \text{if } e > d, \\ 0 & \text{else.} \end{cases} \tag{5.3}$$

To automatically find residue-specific constraints, we evaluate the binding of the reference ligand observed in the crystal structure to each of the residues of the protein. For each residue for which the sum of vdW, electrostatic and hydrogen bond interactions is larger than a predefined threshold (1.5), a constraint is created. If there are more than a maximal number of residues (3) that fulfill this criterion, constraints are only generated for those whose interaction with the reference ligand is strongest.

Pocket description

Another helpful way to enhance the specificity of docking is to describe the ligand binding pocket of the protein of interest. Our pocket descriptions are simple rectangular cuboids plus information about the number or fraction of ligand atoms (a_0) that should be located inside this area and a penalty factor (f). For each ligand pose, a penalty score p is calculated after computing a, the fraction (respectively the number) of ligand atoms inside the pocket description, as follows:

$$p = \begin{cases} (a_0 - a) \cdot f & \text{if } a < a_0, \\ 0 & \text{else.} \end{cases} \tag{5.4}$$

This pocket description can be created manually or automatically. In the latter case, we use a sphere based approach. Non-clashing spheres that have a significant number of neighboring receptor atoms, i.e. that are buried within a pocket, are placed onto the receptor surface. These spheres are then sorted ascendingly according to their distance to the geometrical center of the reference ligand. Starting with the sphere nearest the center, spheres are marked as part of the binding pocket if they are nearer than 1.5 Å to least one sphere that has already been selected. A cuboid is then placed

around all selected spheres, resulting in a pocket description constraint as described above.

5.2.3 Docking algorithm

Our docking algorithm uses the scoring function described in Equation 5.1 and an iterative multi-greedy approach to dock compounds into the binding pocket of a receptor. In the first step, the ligand to be docked is moved into the binding pocket and (optionally) superimposed with the reference ligand. Then, rotatable bonds of the ligand are detected. As rotatable, we regard all bonds that have an order of one, are not part of a ring and are not a peptide-bond.

During subsequent docking steps, each pose will be represented by a compact form of $2n+3$ angles of rotation. For each rotational bond, two degrees of freedom are added, since we allow to independently rotate both sides of the molecule that are connected by this bond around the latter. In addition to this, there are three degrees of freedom for global rotation of the entire ligand. The use of two degrees of freedom per bond, as opposed to one degree, allows to always optimize the binding of a compound while keeping its more strongly bound part (e.g., its headgroup) in place, instead of testing only poses that force its removal from the current binding position.

During docking, already created poses, represented in the above-mentioned way, are stored together with their binding free energy estimate in a list (pose list). Starting with the pose that was obtained by moving the ligand into the active site, each angle of rotation of each entry in the pose list is permuted with a predefined level of discretizaton (e.g., 10°). In each of these permutation steps, the pose as defined by the compact representation of the entry of the pose list is applied to the ligand and the scoring function is used to obtain an estimate of the binding free energy. After this has been done for all entries of the pose list for one angle of rotation, the k best poses (where k is the multi-greedy step-size; e.g. 100) found so far are retained, constituting the pose list for the next step of the multi-greedy optimization, i.e. the permutation of the next angle of rotation.

When all angles of rotation have been permutated in this way, one iterative application of the multi-greedy minimization is finished. If the score for the best obtained pose is not better than the one obtained during the last iteration (if any), the entire docking algorithm will abort here. If, on the other hand, an enhancement was achieved, the best ligand pose will be translationally optimized and the multi-greedy heuristic will be started again with the optimized pose. In any case, if the maximal number of iterations have been performed, the algorithm stops and returns the best obtained pose. A schematic overview of IMGDock is shown in Figure 5.1.

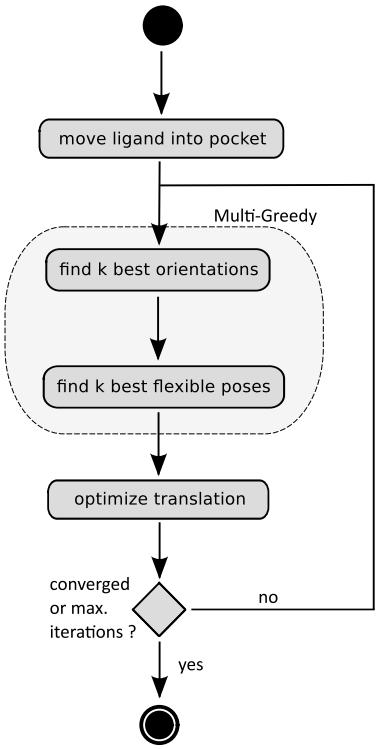


Figure 5.1: Schematic overview of IMGDock.

5.3 Results & Discussion

5.3.1 Scoring function

We evaluate our scoring function on two data sets, the one used for AIScore [56], and PDBbind [57]. The former consists of 101 receptor-ligand complexes and was used to develop our scoring function. PDBbind currently contains about 1,300 co-crystal structures, i.e. all complexes available in the PDB which were deemed to have sufficient quality and for which binding free energies are known. The scoring function is successively applied to all receptor-ligand complexes without any modification of receptor or ligand conformations in order to be able to the evaluate the scoring function without any docking. Complexes in which clashes between heavy atoms of receptor and ligand are detected, are excluded from the following correlation analysis.

The scores obtained by our scoring function for the AIScore set are shown in Figure 5.2a in comparison to experimentally determined binding free energies. On this data set, we achieve a correlation of 0.70. This result is better than the one obtained by Raub et al. with a version of the FlexX scoring function that was specially optimized on this data (correlation of 0.65) and lower than the correlation of AIScore (0.87).

However, since all these scoring functions have been optimized on the AIScore data set, we use the much larger PDBbind set in order to get a better estimate of the usefulness of the scoring function for predicting binding free energies. On the PDBbind data

set, our scoring function achieves a correlation of 0.48, as shown in Figure 5.2b. This is better than the correlation obtained by Raub et al. [56] with both the optimized Version of FlexScore and AIScore (0.43 and 0.46, respectively). Note however, that the difference to the publicly available version of FlexScore (i.e. the one not optimized by Raub et al [56]) is even greater, since it attained a correlation of only 0.17. Furthermore, a study by Wang et al. [58] found that only four out of 14 scoring functions achieved a correlation at least as high as the one obtained by our function on PDBbind.

5.3.2 Docking

In order to evaluate our docking algorithm, we use the "Directory of Useful Decoys" (DUD) [51], an established benchmark data set for molecular receptor-ligand docking. DUD contains 40 subsets, each of which consists of a co-crystal structure (with one exemption of an homology model), a set of compounds known to bind to the respective target, and a set of decoys. While Huang et al. collected the known binders from various published experimental results, decoys were generated *in silico* in such a way that they have "similar physical properties but dissimilar topology" to the ligands.

To assess how well our scoring function and our docking algorithm agree with experimentally observed ligand poses, we dock all ligands appearing in the co-crystal structures of DUD into their respective target. We then calculate the root-mean-square deviation (RMSD) between the pose generated by our docking and the crystal structure ligand pose. The distribution of RMSD values is shown in Figure 5.3. Hence, for 60% of the targets, the resulting RMSD is smaller than 1.1 Å, showing that reference ligand binding-poses can be approximated suitably by our approach.

Although the evaluation of RMSDs of docked reference ligands is a first step towards this end, it alone does that suffice to asses the quality of a docking approach. Potential problems like bad placement of known binders not appearing in the co-crystal structure or very indiscriminate docking, resulting in high scores for non-binders, could

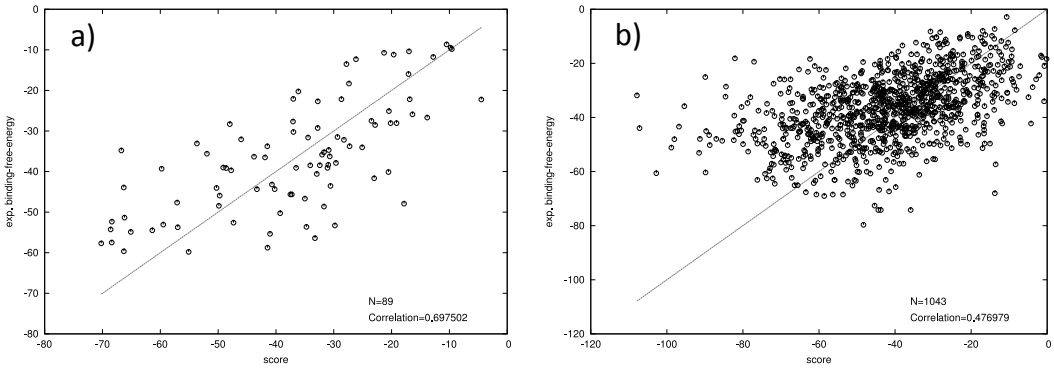


Figure 5.2: Correlation between experimentally determined binding free energies and scores computed by our scoring function for the AIScore [56] (left) and PDBbind [57] (right) data sets.

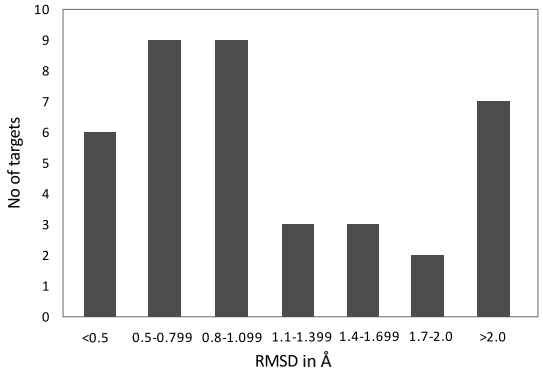


Figure 5.3: RMSDs for docking the reference ligand of all 40 DUD targets.

not be detected by docking of reference ligands. Since both, known ligands and decoys can be chemically very different to the reference ligand, this problem is actually profoundly aggravated.

Therefore, we evaluate our docking algorithm using the ligand and decoy sets for all 40 DUD targets. Molecules of both subsets are of course chemically different from the reference ligand, so that a more realistic estimate of the usefulness of docking than by simple examination of the RMSD of the docked reference ligand can be obtained. However, please note that the similarity between ligand molecules as well as between ligands and decoys unfortunately varies greatly between DUD data sets for different targets, complicating comparative analysis.

In a first step, water molecules observed in the PDB structure of the respective target that are tightly bound to the receptor are detected as described above and retained, while all other water molecules are removed. Then, residue-specific interaction-constraints and a binding-pocket description are automatically generated according to the above explanations. A score-grid is then precalculated for the target and the ligands and decoys supplied by DUD are docked into the pocket.

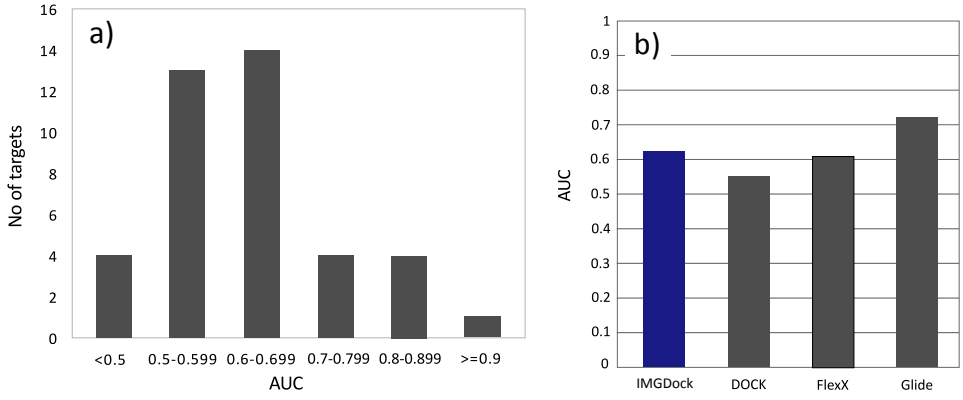


Figure 5.4: Analysis of AUCs for all 40 DUD targets: distribution of AUCs of IMGDock (left) and average AUC of IMGDock in comparison to other approches [59] (right).

In order to assess the results, we compute a receiver operating characteristic (ROC) curve for each target, which shows the fraction of decoys in comparison to the fraction of ligands that are observed within the docking result with a score smaller than a varying threshold. For 57.5% of all data sets, an AUC larger than or equal to 0.6 could be achieved (see Figure 5.4a), showing that in the majority of all examined cases the applied docking resulted in good separation of binders and decoys. ROC plots for all 40 DUD targets are furthermore shown in the Appendix in Figure A1.

For six of the DUD targets, our approach was observed to strongly outperform all other algorithms evaluated by Cross et al. [59]. The ROC plots for these data sets are in Figure 5.5. Therein, note especially the very high enrichments at the top of the rank-lists (e.g. for 0-10% of selected decoys).

Comparison of the average AUC over all 40 DUD data sets by our approach and several well-known docking algorithms (Figure 5.4b) shows that even on average our approach performs very competitively and also outperforms several other algorithms. The average AUC of 0.623 achieved by our approach is significantly higher than the results for FlexX (0.61) and DOCK (0.55) obtained by Cross et al. [59].

These results, obtained in a completely automatic fashion (i.e. without any manual tuning), together with the high speed of docking (approx. 30 sec for averaged-sized ligands with 15 rotatable bonds) and its availability free of charge make IMGDock very interesting for high-throughput screening applications. The easy way to manually define additional scoring constraints furthermore points out the usefulness of IMGDock for even more complicated task, e.g. during lead optimization.

However, on average Glide performs better (average AUC of 0.72) than our approach, although in eleven cases (DUD data sets for ACE, AChE, ADA, AmpC, GPB, HSP90, InhA, NA, PR, SAHH, TK) our approach yielded significantly better results. In order to thus enhance the scores obtained by docking, we developed TaGRes and also applied it the docking results of the DUD data sets. Please see the next Chapter for a detailed description of TaGRes.

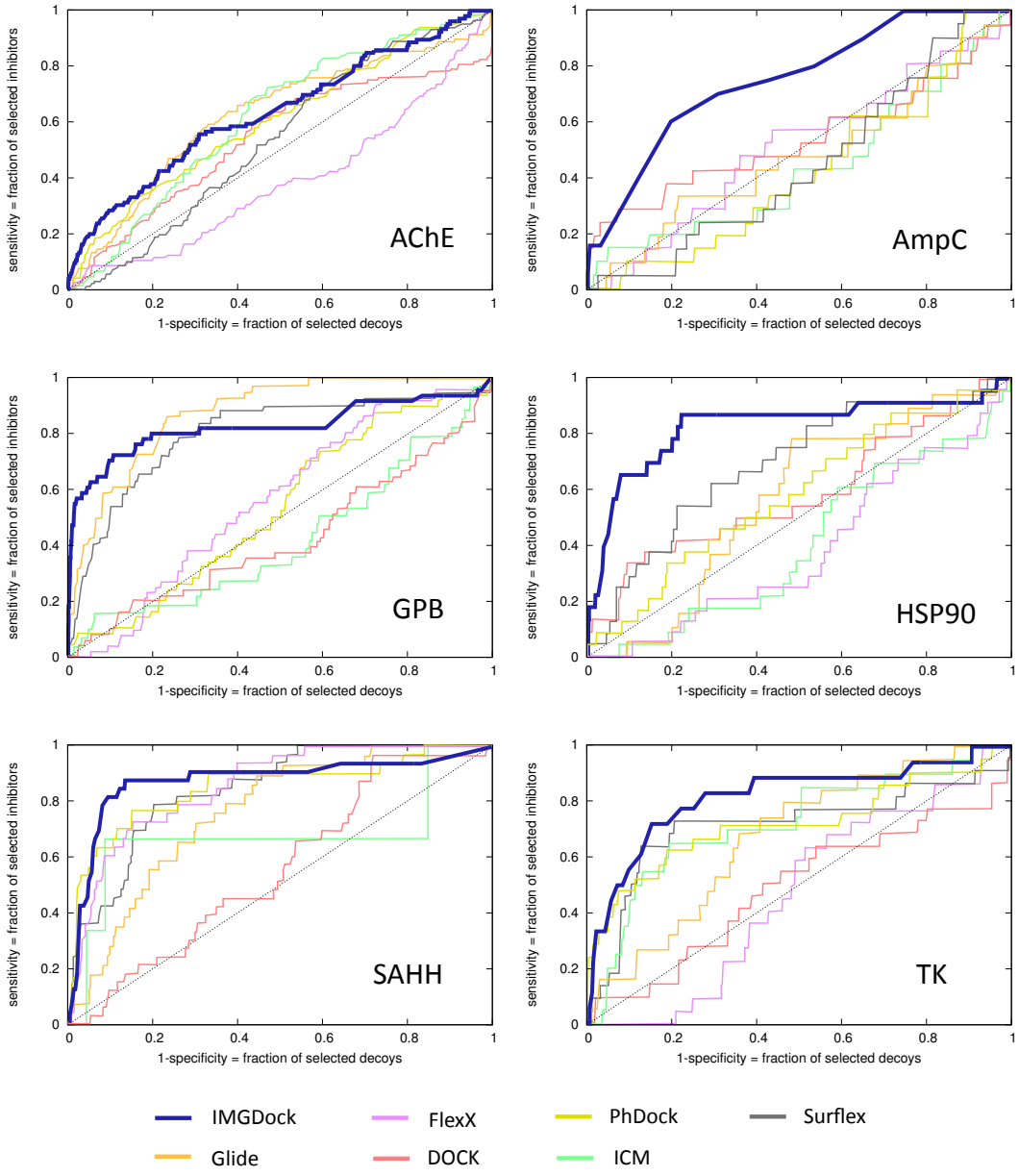


Figure 5.5: ROC plots for selected DUD data sets. In all of those cases, IMGDock significantly outperforms all other approaches evaluated by Cross et al. [59].

6 Receptor-Ligand Rescoring for structure based drug design: TaGRes

6.1 Introduction

As explained in Section 3.3, there exist several different strategies for rescoring receptor-ligand docking results. However, no target-specific rescoring can be done by most of the existing approaches. For different protein target families, the importance of different contributions to the overall binding free energy might vary significantly and, even more important, it may depend on the three-dimensional location of the interactions between receptor and ligand. Therefore, for example, for one target the existence of hydrogen bonds and their strengths in exactly defined regions may be important, while for another target the electrostatic interactions between charged groups of ligand and receptor play a predominant role. Furthermore, inclusion of available experimental binding free measurements for improvement of rescoring is also not possible with most of the established procedures.

We therefore developed Target-Specific Grid-Based Rescoring (TaGRes), which is presented in this Chapter. TaGRes allows to rescore docking results based on their three-dimensional pose in the binding pocket and employs experimentally determined binding free energy measurements for the respective target. Using docking results obtained with IMGDock, we will show that TaGRes can strongly enhance the binding free energy approximations (scores), leading to better separation of active and inactive ligand candidates and higher enrichment.

TaGRes has furthermore been integrated into our Computer-Aided Drug-Design Suite (CADDSuite) and thus into the workflow-system Galaxy [52]. This allows to easily and efficiently use TaGRes in conjunction with all other tools supplied by CADDSuite. Rescoring by use of TaGRes can thus, if desired, be started directly from a web browser and submitted to a compute cluster, cloud, or grid. Furthermore, even huge rescoring workflows can be easily created, used and shared with other users. A more detailed description of CADDSuite will be given in Chapter 7.

TaGRes is, as part of CADDSuite, licensed under the GPL and are available from `http://www.ball-project.org/caddsuite`.

6.2 Methods

6.2.1 TaGRes model generation

After having docked compounds into the binding pocket of the protein of interest, rescoring approaches can help to enhance the obtained estimate of the binding free energy. If experimental binding free energy measurements for the respective target are available, a training-based rescoring, which would be able to make use of this data, could be employed.

To this end, we developed Target-Specific Grid-Based Rescoring (TaGRes). As input we need a training data set consisting of compound poses in the binding-pocket of the target of interest and experimentally determined binding free energies for those compounds. Poses generated by any docking approach can be used for this. If, on the other hand, receptor-ligand co-crystal structures are available, they should be preferred over docking results.

TaGRes scores each input ligand pose with the function described in Equation 5.1 and generates an interaction grid for which each cell contains the sum over the score contributions of all ligand atoms located inside this cell (TaGRes-3D). By default, the binding pocket is therefore discretized into grid cells with 3 Å side length. Ligands that could not be successfully placed by the docking approach into the binding pocket without any heavy atom clashes are skipped. Molecules for which the input file contains no or an invalid binding free energy (e.g., one larger than zero) are ignored as well and do not contribute to the training data.

The procedure described above is repeated for each ligand and each resulting interaction grid is linearized to one interaction vector. The interaction vectors of all ligands are then joined into one interaction matrix. The latter, together with the experimentally determined binding free energies, is then used to find a regression model that can suitably model the latter. Therefore, all linear as well as non-linear regression techniques shown in Figure 4.1 are automatically evaluated on the given training data set. For each of those regression approaches, all model and kernel parameters (if any) are automatically optimized, feature selection is performed and the quality of the model obtained this way is evaluated using a nested cross-validation procedure. The nested evaluation in each step splits the given training data into two subsets covering approximately equal binding free energy range, executes all the previously mentioned steps with use of the first subset and then predicts the binding free energies of the second subset. This way, we are able to obtain an estimate of how useful each model could be for subsequent rescoring of a different data set. However, this estimate relies on the assumption that the latter will not be chemically completely dissimilar to the training set (please see Section 6.3 for a discussion of this).

After this evaluation has been done for each regression approach, the one with the best nested cross validation quality will be selected. If, however, all regression approaches achieved only a very low quality (by default a nested Q^2 below 0.2), then all models are rejected and the rescoring is aborted. This check is necessary in order to try to

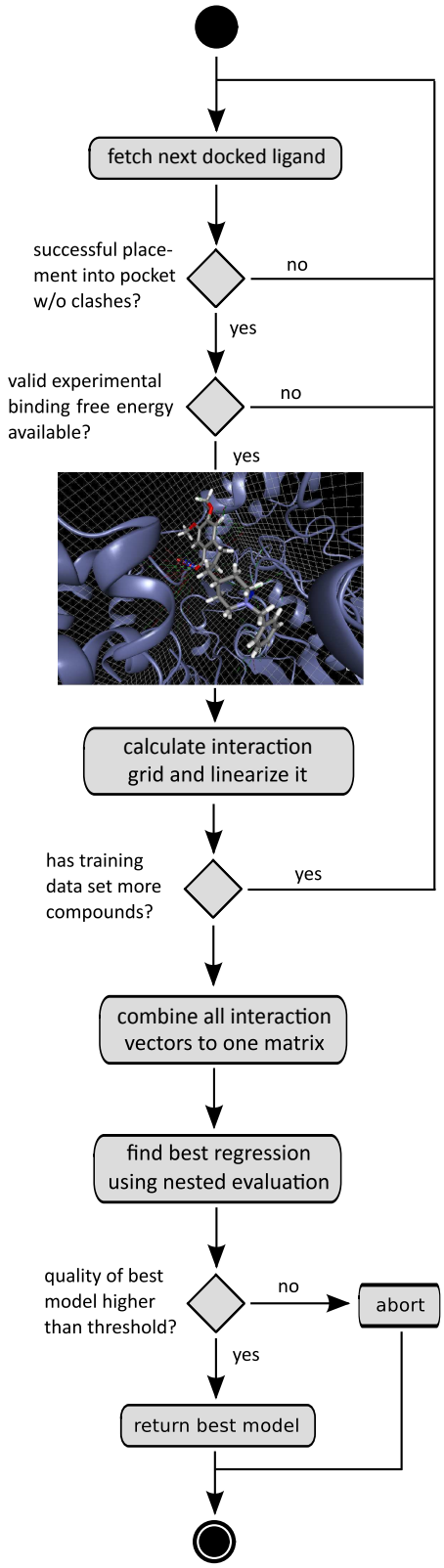


Figure 6.1: Schematic overview of the TaGRes model generation process.

prevent creation of a rescoring model that will result in deterioration of the binding free energy estimates when applied to a docking output.

For those cases in which TaGRes-3D could not find a suitable model, we developed an extension (TaGRes-4D) that allows for independent modeling of all scoring terms. Thus, TaGRes-4D generates the interaction grid for each given ligand pose in such a way that each cell contains the sum over the score contributions of one scoring term (e.g., vdW, ES) of all atoms located inside this cell. In this way, one interaction grid is created for each scoring term. TaGRes-4D therefore can be beneficial if a specific type of molecular interaction (in specific spatial areas of the binding pocket) is important for strong receptor-ligand binding.

An overview of the training process of TaGRes, as described above, in shown in Figure 6.1.

6.2.2 Rescoring of docking results

After a TaGRes model has been generated as described above, it can be used to predict the binding free energies of compounds based on their three-dimensional pose in the binding pocket.

Therefore, compounds to be rescored should first of all be docked into the binding pocket of same molecular structure that was used be generate the TaGRes model. TaGRes then fetches each docked compound from an input file and applies the scoring function described in Equation 5.1 to it. The interaction scores of all ligand atoms within each discretized area of space are then summed up, yielding an interaction grid grid, which is afterwards linearized to an interaction vector. Compounds whose pose generated by docking contains heavy atom clashes or was assigned a very bad score due to nonfulfillment of scoring constraints are skipped. This way, the number of false positives (i.e. non-binders/decoys obtaining a very good score) created by TaGRes is reduced, and the specificity of the latter enhanced.

The previously created TaGRes model is then applied to the interaction vector. In case of linear regression models, whose training results in coefficient for each grid cell, this is achieved by a simple vector product of interaction vector and coefficient vector. Non-linear models, on the other hand, first apply a kernel function to the interaction vector. This function evaluates the similarity of the interaction vector of the compound to be rescored to the interaction vector of each compound of the training data set and thus transforms the former into kernel-space. After this, a vector product of transformation and training result yields an estimate of the binding free energy of the molecule.

Figure 6.2 shows a schematic of the process used by TaGRes to rescore molecules.

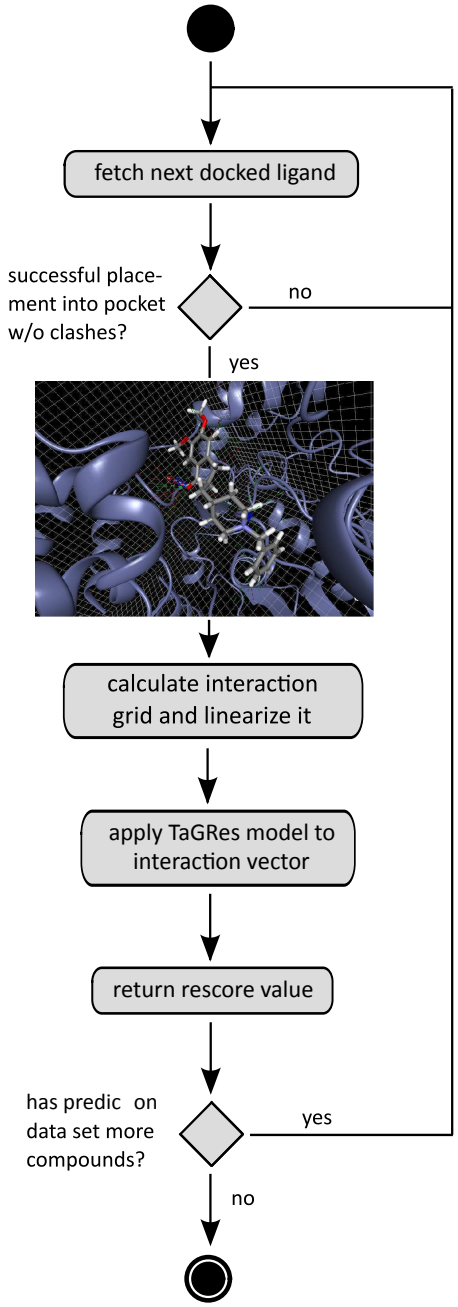


Figure 6.2: Schematic overview of the process used by TaGRes to rescore molecules.

6.3 Results & Discussion

For each DUD target for which data could be obtained from BindingDB [60], TaGRes is employed in order to try to enhance the docking results by rescoring. Therefore, for each of those targets, the BindingDB set, containing compounds that bind to the respective target and their experimentally determined binding free energies, is downloaded and a set of decoys is generated. The latter is obtained by searching the Zinc data base [61] for compounds that exhibit a moderate similarity to the molecules in the BindingDB set. Hence, Zinc compounds that have a maximal Tanimoto coefficient, calculated on binary pathway-based fingerprints, between 0.35 and 0.55 to molecules of the BindingDB set are chosen. BindingDB set and decoy set are then separately docked into the receptor and the top-scored 25% of both docking results together make up the training data set for TagRes.

Note that this use of decoys is important here, since we ultimately want to rescore DUD docking results with the created TaGRes models, i.e. we want to achieve a better separation between binders and non-binders (decoys), whereas BindingDB data sets contain only binders. On the other hand, if the final goal is to obtain a more accurate binding free energy estimate of only binders, as is the case for example during lead optimization, generation of the decoy data set should be skipped.

Since at least a moderate similarity between training and rescoring data set should be present in order for TaGRes to be able to do a helpful rescoring, we evaluate the similarity between those two sets for each target using the median of all pairwise Tanimoto coefficients on binary, pathway-based fingerprints. DUD targets for which the obtained similarity value is smaller than 0.5 are skipped, i.e. no rescoring is performed on them.

The training data set is then used to train a TaGRes-3D model. If this fails because no model with significant predictive quality could be found, a TaGRes-4D model is generated where possible.

If a model with significant predictive quality (as evaluated automatically by nested cross-validation) is returned, it is used to rescore the docking results for the respective DUD data set. In those cases were no such model could be generated, rescoring cannot be done. However, by use of this quality check, we prevent rescoring that would most likely result in a deterioration of binding free energy estimates (compared to the scores assigned by the docking algorithm).

The performance of TaGRes is evaluated using the AUC criterion, whose value is then compared to the one for the docking result for the same target. As Figure 6.3 shows, in six out of eight cases, a significant to large improvement of AUC was achieved by use of TaGRes. In only two cases, a slight decrease in AUC was observed. Figure 6.4 furthermore visualized the huge differences in quality between docking and rescoring results using ROC curves.

This shows that TaGRes can be very helpful for rescoring docking results. The enhanced binding free energy estimates obtained this way allow a better separation of

binders and non-binders (as shown above), which may therefore result in higher hit rates in lead discovery projects. Experimental binding free energy measurements for the target of interest must be available for TaGRes to be applicable, but on the other hand TaGRes enables integration of this experimental knowledge, something which is not possible with other rescoring approaches.

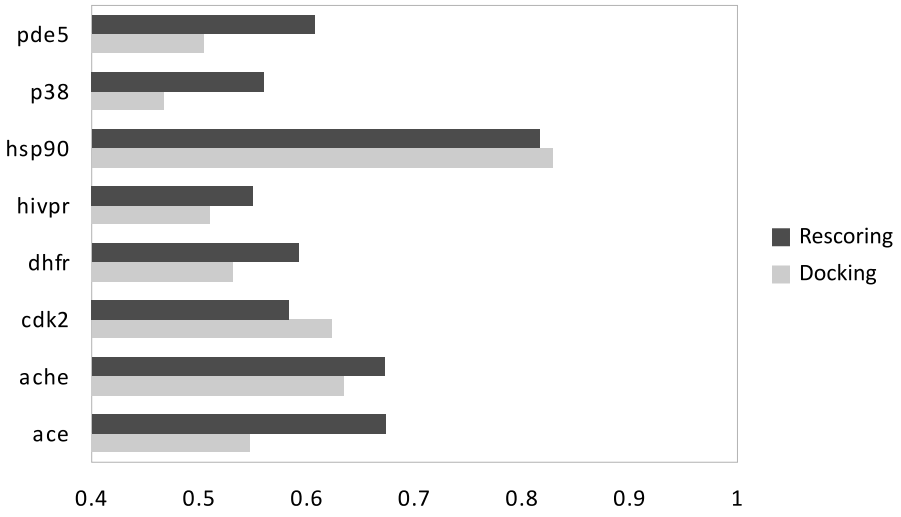


Figure 6.3: AUCs of rescoring and docking results in comparison.

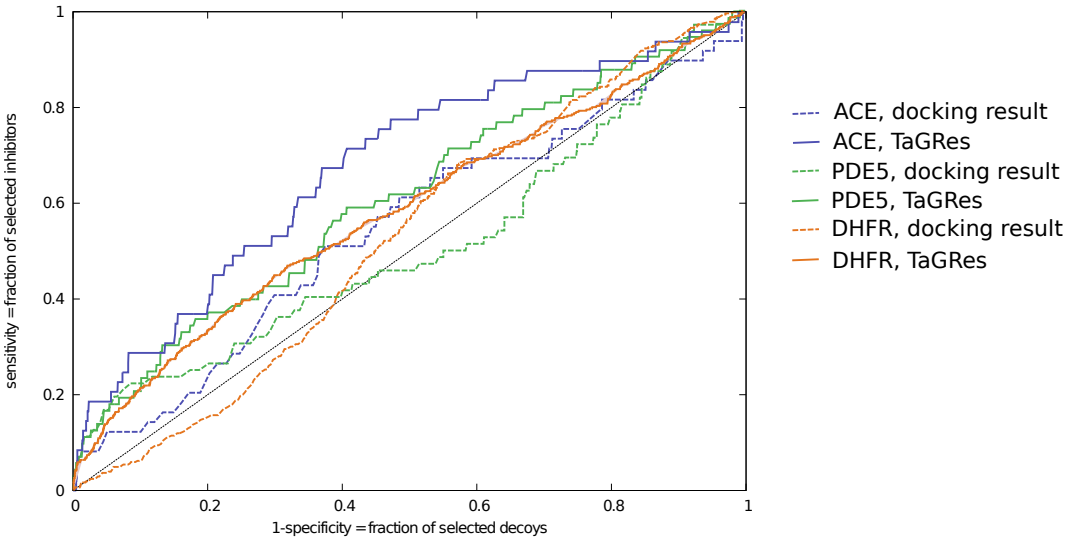


Figure 6.4: ROC curves for TaGRes in comparison to the ROCs for the docking results.

7 Consolidation of approaches into modular, workflow-enabled package: CADDSuite

7.1 Introduction

Computer-aided drug design, as explained in Chapter 3, aims to speed up lead discovery or lead optimization projects that make up the first steps of the development of new drugs. In essence, this is done by trying to predict the binding free energy of chemical compounds to the target of interest (e.g., an enzyme) using various *in silico* approaches. This way, a usually huge database of candidate compounds can be screened for a set of first hits. Depending on the available data, either ligand- or structure-based approaches, or a combination of both, are possible.

In a ligand-based approach, we are given only a set of active compounds and their experimentally determined binding free energies and maybe also a set of non-active compounds. We can then generate quantitative structure-activity relationship (QSAR) models that employ machine learning approaches, e.g. regression techniques, to describe the compounds' biological activities by a function of their properties, particularly their topologies [62–64]. Using these models we can screen a database for new putatively active compounds. For a more detailed introduction to ligand-based drug design, please refer to Section 3.2 and for a description of our QSAR software, see Chapter 4.

For a structure-based approach, on the other hand, we need obtain the three-dimensional structure of the receptor of interest, e.g., by protein crystallography or homology modeling. Thus, we can make use of molecular receptor-ligand docking, where sophisticated search-algorithms try to find optimal placements of the compounds within the binding pocket of a receptor. The interaction energy of the generated poses is evaluated using a scoring function that needs to be as fast as possible, since it is used very many times during each docking run, and consequently needs to be a simplified approximation. To get more accurate estimates of the binding free energy, rescoring methods are often used which try to achieve this by reevaluating the pose generated by docking in a different, usually computationally more complex, way [26, 27]. For more details about structure-based drug design in general, see Section 3.3 and for descriptions of our docking and rescoring algorithms, please refer to Chapters 5 and 6, respectively.

However, utilizing computer-aided drug design approaches is far from trivial. Each of them requires a number of pre- and postprocessing steps. Usually, input data needs to be generated in some way, for example by exporting compounds from a database or by creation of combinatorial libraries. Then, data commonly have to be prepared before they can be used by QSAR, docking or rescoring algorithms. This includes for example creation of three-dimensional coordinates for compounds and protonation of the receptor. In a next step, compounds as well as the receptor should be checked for chemical errors or dubiousness. Furthermore, the chosen modeling technique (QSAR, docking or rescoring) might require additional, specific preparation tasks (like those explained in the following section). Last but not least, it is often desirable to analyze the final output, in order to show the quality of a generated model or to visualize the distribution of compounds' properties or scores, or to convert it to different file formats or import into a database. On top of this plethora of tools, it is often prudent to combine QSAR, docking and rescoring approaches within one drug-design pipeline.

All this illustrates the need for a framework that provides modular tools for all the aforementioned tasks, makes them useable in a simple and consistent manner, and allows to easily create and reuse workflows employing those tools. Moreover, the often computationally demanding modeling steps and huge data sets make support for easily using those programs on a compute cluster or cloud very desirable.

Still, all software packages for computer-aided drug design we are aware of do not fulfill at least some of those specifications: Most of them provide tools for only a very small subset of the tasks mentioned above, thereby leading to the problem of having to use many different software packages in conjunction. Examples for this are FlexX [19] or AutoDock [21], which provide tools for docking but not for structure preparation, QSAR, rescoring or analysis of results. Then, there are some system like Accelrys' Pipeline Pilot [65] or Schrödingers' KNIME extension [66], that offer some more functionality but are commerical products, thus not being available free of charge, and furthermore often depend on third party tools to fulfill various tasks. Availability as only commercial products also means that, apart from programs in practice not being available to many researches, deployment of those tools on compute clusters or clouds is, if at all supported, practically impossible, since most times a separate license for each and every parallel process would have to be bought. Last but not least, most of the available software packages offer no integration into any workflow system, making creation of huge pipelines hard and reducing reproducibility.

Even using several different software packages in conjunction as a fall-back does not solve these problems and is tedious and error-prone: Many different software packages would have to be bought, installed, and maintained. The outputs of different tools are often incompatible with each other or require many conversions, usually involving loss of information. Furthermore, tools often have to be used in very different ways, confusing the user and complicating application of computer-aided drug design techniques. Hence, if tools are either not available or not easily usable in conjunction, users, at least less experienced ones, will often skip steps that would however be prudent or even vital to use. Thus, compounds or receptors might not be checked for errors,

only a QSAR analysis is performed, without successive docking (or vice versa), selection of important features during QSAR is skipped or generation of important docking constraints is not done and so forth.

Here, we present CADDSuite, a flexible framework that provides tools for all commonly required steps and can therefore make solving computer-aided drug design tasks much easier. It contains tools for data retrieval, preparation, checking receptor and ligand files, import of compounds into a database and own algorithms for QSAR, docking and rescoring. All tools can be used in a simple and consistent way, making it easy to use them in combination in order to solve all of the above mentioned tasks. CADDSuite furthermore is available free of charge, licensed under the GNU GPL. In order to improve its ease of use even more, CADDSuite has been integrated into the workflow system Galaxy [67, 52]. This way, a graphical interface can be used to run individual jobs or generate or execute workflows directly in a web browser, without any need for software installations by the end user. Furthermore, it also adds support for using a compute cluster, grid or cloud, so that jobs can be run and tracked on those. This, together with the point that all our tools use a modular concept and are available free of charge, eliminating the need for separate licenses for each process, makes CADDSuite optimally suited for high-performance computing (HPC) in the field of computer-aided drug design.

7.2 Methods

The tools provided by our package cover most areas of application of computed-aided drug design. There are tools for obtaining and preprocessing input from various sources, for quantitative structure-activity relationship (QSAR) modeling, receptor-ligand docking and rescoring and postprocessing and analysis of results. See Figure 7.1 for an overview over all packages. All of these tools have a similar interface and can operate on standard chemical structure files, thus allowing to easily solve complex tasks by using them in conjunction. In the following we will describe each package and explain its benefit for computer-aided drug design.

7.2.1 Data input

tool name	description
File upload	import molecules, receptors
CombiLibGenerator	generate combinatorial library
PDBDownload	retrieve pdb-file from pdb.org
DBExporter	fetch molecules from database

Table 7.1: Names and descriptions for tools of the *Input* module.

As a first step for computer-aided drug design, import of input data is necessary. We provide several tools in order to facilitate this. Aside from programs for uploading files and downloading files from publicly available web servers, combinatorial libraries can

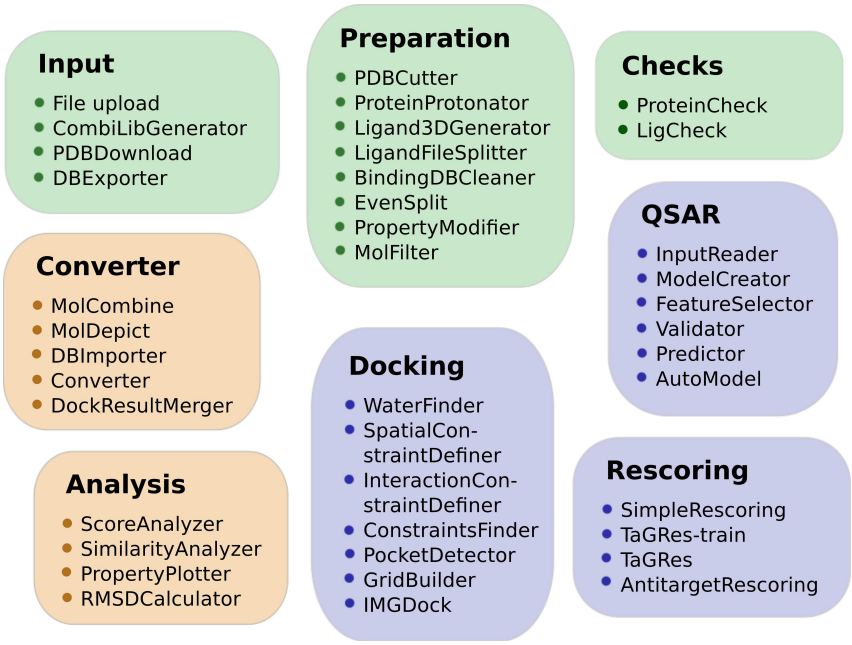


Figure 7.1: Overview over all provided packages.

be generated by enumeration of all possible combinations of molecule scaffolds and moieties, where the latter two are specified by the user using SMARTS expressions.

Furthermore, we created a database schema which allows to store molecules in a binary form and automatically creates, saves and tracks all additional data that is required for efficient storing and searching of molecules. After a such a database has been created (see below for *DBImporter*), compounds can be filtered from this database using a wide range of criteria, for example by specification of canonical smiles, logP, molecular weight, SMARTS [68] expressions, or by similarity to given query molecules.

7.2.2 Preparation

tool name	description
PDBCutter	separate ligand and receptor
ProteinProtonator	protonate protein
Ligand3DGenerator	generate 3D conformations
BindingDBCleaner	fix data from bindingdb.org
EvenSplit	generate splits w/ equal property range
PropertyModifier	modify property tags

Table 7.2: Names and descriptions for tools of the *Preparation* module.

Although the previous tool package provides tools for obtaining input files, in the field of computed-aided drug design those input files often require several preprocessing steps before they can be used by the core algorithms of interest (e.g., QSAR, docking

or rescoring). Performing all steps manually is often tedious, time consuming, error-prone and often leads to irreproducible results.

Protein co-crystal structure files (as obtained from, e.g., the Protein Data Bank [69]) need to be split into receptor and ligand files (*PDBCutter*) and the receptor structure needs to be protonated (*ProteinProtonator*), since in most cases hydrogens are not detected by x-ray crystallography due to a resolution that is too low for this. If compounds that are not observed in the co-crystal structure are to be docked, the assignment of initial 3D coordinates for those compounds is necessary as a starting point for docking (*Ligand3DGenerator*). In case of QSAR analysis, 3D conformations are also required for the input molecules since several 3D features will be calculated by the respective tools (see below for QSAR tools). Furthermore, it is often helpful to be able to quickly rename or add molecule property tags, which could store information about e.g. identifiers, scores, experimental results (*PropertyModifier*). If the user wants to validate or assess his drug design pipeline, splitting of the input data into subsets that cover equal property (e.g. binding free energy) range is often desirable (*EvenSplit*).

7.2.3 Checks

tool name	description
ProteinCheck	evaluate protein quality
LigCheck	chemical sanity check for ligands

Table 7.3: Names and descriptions for tools of the *Checks* module.

After input files have been obtained and prepared as described above, they could be used for computed-aided drug design algorithms. However, many molecules in publicly available sources contain problems, chemical errors, or have dubious quality. Especially for small molecule files, information about the elements of the molecule's atoms are sometimes missing, bonds sometimes have an incorrect length or are missing altogether, in effect resulting in two disconnected molecules, and hydrogens or 3D coordinates have not been assigned. Those simple but important checks are done by our tools for ligand files as well as for proteins. For the latter, we also generate a report in portable document format (PDF) containing information that can help to assess the quality the protein, like a secondary structure plot, a Ramachandran plot or visualization of temperature factors.

7.2.4 QSAR

Quantitative structure-activity relationship (QSAR) models allow to predict the binding free energy of molecules to a given drug target by use of regression techniques if an appropriate training data set is available. The training data set should therefore contain compounds and information about their experimentally determined binding free energy. Furthermore, the compounds whose activity is to be predicted need to have some similarity (with respect to both feature and activity space) to the compounds in

tool name	description
InputReader	read molecules and generate features
ModelCreator	create a QSAR model
FeatureSelector	automatically select features of a QSAR model
Validator	evaluate quality of a QSAR model
MolPredictor	predict molecule activities with QSAR model
AutoModel	automatically find best QSAR model

Table 7.4: Names and descriptions for tools of the *QSAR* module.

the training data set (although details depend on the employed model). Therefore, the availability of an appropriate training set can in practice be problematic. However, if suitable training data is available, QSAR models can be used in a drug discovery pipeline to quickly and strongly reduce the number of compounds to be considered for further examination. QSAR predictions are computationally much less expensive than for example receptor-ligand docking (see next package description), so that even very large data sets can be processed very quickly this way. An example for this is shown in the following section.

This module provides all necessary tools to perform fast and efficient QSAR modeling. Input can supplied in form of standard chemical structures files and a set of 193 descriptors (see Table A1 in the Appendix) is then automatically created for each molecule (*InputReader*). However, input files should be checked for chemical correctness and uniqueness of contained molecules beforehand (see subsection above for these checks), since the existence of identical molecules would obviously distort the created models. A variety of different QSAR models can then be created (*ModelCreator*), relevant features selected (*FeatureSelector*), QSAR models validated (*ModelCreator*) and the activities of molecules predicted (*MolPredictor*). Using the *EvenSplit* tool (Section 7.2.2) to split the input data, nested validation workflows can be created. Furthermore, *AutoModel* can be employed to automatically find the most appropriate QSAR model for a given data set. This tool performs nested validations of all available model types, including several successive feature selection steps and model and kernel parameter optimizations. For more information about our QSAR software, please see Chapter 4.

7.2.5 Docking

tool name	description
WaterFinder	find strongly bound water molecules
SpatialConstraintDefiner	define spatial constraint
InteractionConstraintDefiner	define interaction constraint
ConstraintsFinder	find strongly interacting residues
PocketDetector	detect ligand binding pocket
GridBuilder	precalculate grids for docking
IMGDock	Iterative Multi-Greedy Docking

Table 7.5: Names and descriptions for tools of the *Docking* package.

Molecular docking can be used to predict the binding poses and binding free energies of ligands to a drug target. Although it is computationally more expensive than QSAR predictions, it allows direct three-dimensional modeling of the interaction between receptor and ligand. Unsuitable ligand candidates can thus be rejected due to either a bad binding free energy estimate or a bad binding pose. In order to automate the latter, the creation of several kinds of constraints if often helpful. These include constraints for receptor residues with which a strong interaction should take place (e.g., residues in the active site; *InteractionConstraintDefiner* and *ConstraintsFinder*) and a spatial description of the binding pocket (*SpatialConstraintDefiner*, *PocketDetector*). Those constraints also help to enhance docking results since they can be used during docking, so that they can guide the algorithms towards favorable poses. Furthermore, important for docking are water molecules. While protein structures often include hundreds of water molecules, in most cases only very few of them (if any) have any effect on the binding of the ligands to the receptor. Therefore, it is helpful to try to distinguish water molecules that are strongly bound to the receptor and/or the reference ligand from unbound ones and use only the former during molecular docking (*WaterFinder*). In any case, before a docking is attempted, the receptor structure as well as all compounds that are to be docked should be checked for chemical errors and 3D start conformations for the latter should have been generated (see description of packages *Checks* and *Preparation* for this).

Our *Docking* module contains a program (*IMGDock*) with our own docking approach as well as tools for automatic search or manual definition of the above mentioned constraints, detection of strongly bound water molecules and a tool for precalculation of score-grids to be used during docking. For more detailed information about IMGDock and our other tools mentioned above, please see Chapter 5.

7.2.6 Rescoring

tool name	description
SimpleRescoring	use scoring function to rescore
TaGRes-train	Target-specific Grid-Rescoring, training
TaGRes	Target-specific Grid-Rescoring
AntitargetRescoring	rescore w/ respect to antitarget

Table 7.6: Names and descriptions for tools of the *Rescoring* module.

After compounds have been docked into a target it is often desirable to try to enhance the estimate of the binding free energy and thereby optimize the specificity of docking results by use of rescoring.

In our *Rescoring* module we therefore offer several different rescoring procedures. One possibility is to reevaluate all poses generated by docking with a scoring function (which may be similar to the one used during docking). Another way to rescore docking poses is by statistical analysis of the interaction fields of known binders (*TaGRes-train* and *TaGRes*). For this approach, a set of compounds whose binding free energies to

the receptor are known is docked into the binding pocket and a regression resulting in a contribution factor for each discretized area of space of the binding pocket is performed. For more information about TaGRes, please refer to Chapter 6. Furthermore, an alternative for enhancing the specificity of docking results (but not the correlation with actual binding free energies) is to compare the score (*ts*) each compound was assigned after being docked into the target to the score (*as*) it got after docking into an anti-target (*AntitargetRescoring*). A new score is thus calculated as

$$antitarget\_rescore = \begin{cases} ts + (ts - as) \cdot p & \text{if } as < ts, \\ ts & \text{else} \end{cases}$$

where p is a penalty factor (100 by default). Molecules that received a good anti-target score (indicating strong binding) that is even better than the score obtained on the target, are penalized, so that the probability of false-positives can be reduced.

7.2.7 Analysis

tool name	description
ScoreAnalyzer	generate ROC or enrichment plots
SimilarityAnalyzer	analyze similarity between two molecule sets
PropertyPlotter	plot molecule properties
RMSDCalculator	calculate RMSD between conformations

Table 7.7: Names and descriptions for tools of the *Analysis* module.

After *in silico* drug design experiments, like those described in the previous sections, have been performed, it is often desirable to analyze the results and their quality. If experimental binding free energy measurements or binder/non-binder classifications are available, generation of receiver operating characteristics (ROC) curves or enrichment plots, is often helpful (*ScoreAnalyzer*). While ROC plots allow to analyze the sensitivity in comparison to the specificity of the applied technique, enrichment plots visualize the increase in the number of binders between a top-scored subset of a molecules (e.g., by docking or QSAR) and random sampling. Furthermore, analyzing the similarity between sets of molecules can help to interpret or enhance the performance of a drug design workflow (*SimilarityAnalyzer*). This is of course particularly important for QSAR experiments, where the reliability of binding free energy predictions strongly depends on the existence of at least a moderate chemical similarity of the molecule whose activity is to be predicted to compounds in the training data set.

Also included in this module are tools for plotting molecule properties (e.g., scores obtained by docking or QSAR) and calculation or root-mean-square deviations (RMSDs). The former thus allows to, e.g., visualize the distribution of scores or to plot the correlation between scores and experimentally determined binding free energies. The calculation of RMSDs on the other hand is interesting if compounds taken from a co-crystal structure are docked into the binding pocket, so that the deviation of the pose

generated by docking from the experimentally determined pose can be evaluated.

7.2.8 Converter

tool name	description
Converter	interconvert molecular file-formats
MolCombine	combine molecular files
DockResultMerger	merge and sort docking output files
Mol2Picture	generate structure diagrams
DBImporter	import molecules into data base
VendorFinder	search vendors for compounds

Table 7.8: Names and descriptions for tools of the *Converter* module.

After executing a drug design workflow, converting the output to a format of the user's choice or storing results in a database is often necessary. This last package contains all tools to easily achieve this. Files can be converted between several chemical file formats (*Converter*), report documents containing structure diagrams for all compounds in the given data set can be generated (*Mol2Picture*) and molecule files can be sorted and filtered according to the scores of contained compounds (*DockResultMerger*). Furthermore, compounds can be imported into a database (*DBImporter*). Information necessary to enable fast searching of molecules in the database, like canonical smiles, logP, molecular fingerprints, molecular weight and functional group counts are automatically generated for each compound during import and stored in the database. In addition to this, scores obtained by docking or QSAR analysis will be automatically saved in the database as well.

7.3 Results & Discussion

7.3.1 Integration into Galaxy

To make the use of all described tools even more convenient, we integrated CADDSuite into the workflow system Galaxy [52, 67]. It is thus possible for users to start individual tools (see Figure 7.2), create or execute entire workflows (Figure 7.3) directly and easily from a web browser, without any need to install any programs on their local computers. Furthermore, data sets, results and workflows can easily be shared between different users or exported and downloaded.

This makes combining all CADDSuite tools even easier and the results more reproducible. Galaxy also offers compute cluster and cloud support, so that jobs will automatically be started and monitored on available nodes. This, together with the point that all our tools use a modular concept and are available free of charge, eliminating the need for separate licenses for each process, makes the combination of Galaxy of CADDSuite optimally suited for high-performance computing (HPC) in the field of computer-aided drug design.

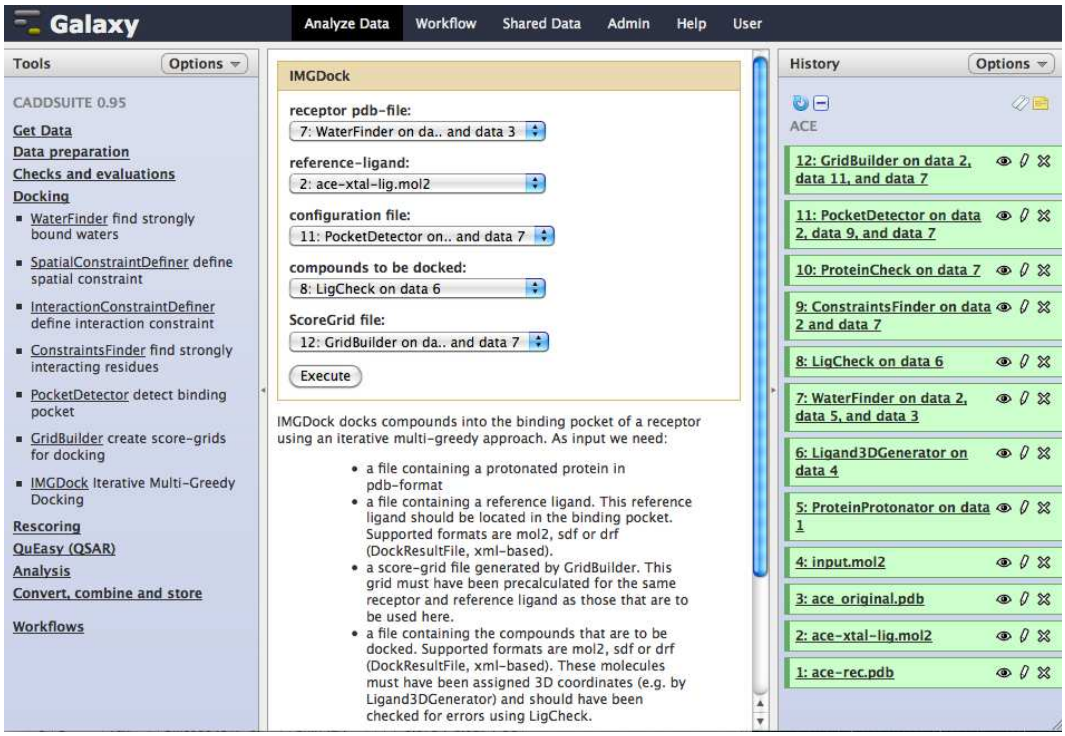


Figure 7.2: Screenshot of the Galaxy-CADDSuite web-interface.

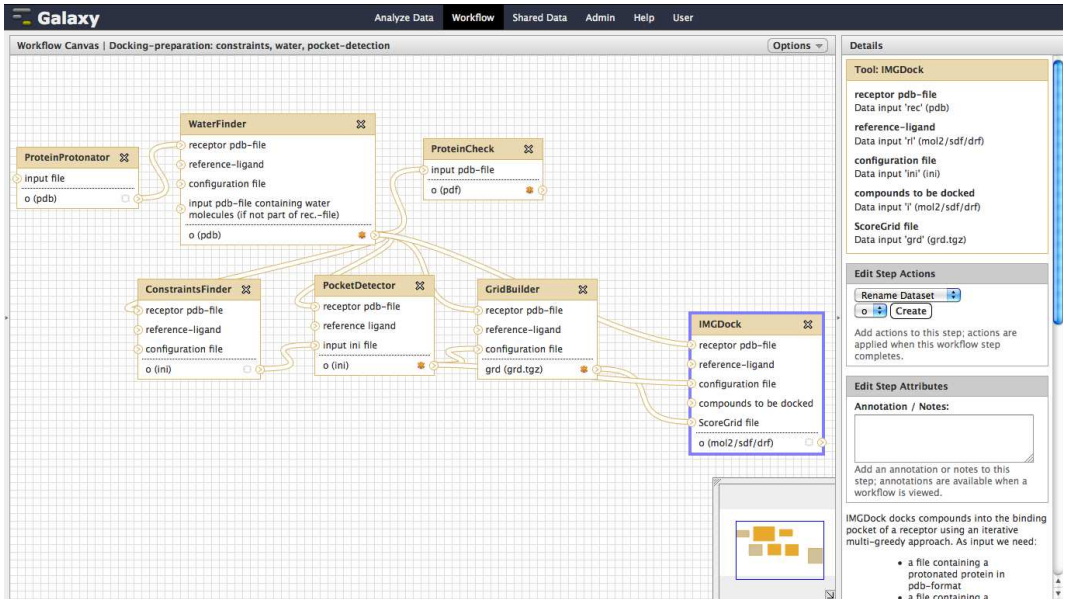


Figure 7.3: Screenshot of the Galaxy-CADDSuite workflow editor.

A release of CADDSuite, including the integration into Galaxy, can be obtained from `http://www.ball-project.org/caddsuite`.

7.3.2 Carbonic anhydrase II virtual screening workflow

In order to illustrate the usefulness of the CADDSuite, we generate a virtual screening workflow for the target carbonic anhydrase II in analogy to the well-known pipeline devised by Klebe et al. [70] and evaluate the quality of its result. As a hypothetical lead discovery pipeline, the focus here lies on achieving a high enrichment. Thus, the final output of a helpful pipeline should contain a higher ratio of binders to non-binders (or decoys) than the input data.

We will now shortly describe the individual steps involved in creating the described pipeline in order to show which preparation steps are necessary, which modeling procedures can be used, and which of our tools can be employed to easily solve each of those assignments. Figure 7.4 shows a schematic overview of this pipeline.

First, we download the data set for carbonic anhydrase II from bindingdb.org [60], convert the contained IC_{50} values to binding free energies and remove compounds without annotated activity data (*BindingDBCleaner*). The resulting set of compounds is filtered according to SMARTS expressions defined by Klebe et al. [70]. All compounds that match at least one of those expression are selected, while all others are rejected (*MolFilter*). This set of molecules will serve as binders for our pipeline and is then divided into two subsets (*EvenSplit*); one will be used for training a QSAR model (see below) and one will be used as part the prediction data set, i.e. the one containing the compounds whose activities are to be predicted by our pipeline.

Next, two sets of decoys are generated. One of those sets (training decoys) will be used during training and the other (prediction decoys) as part of the prediction set.

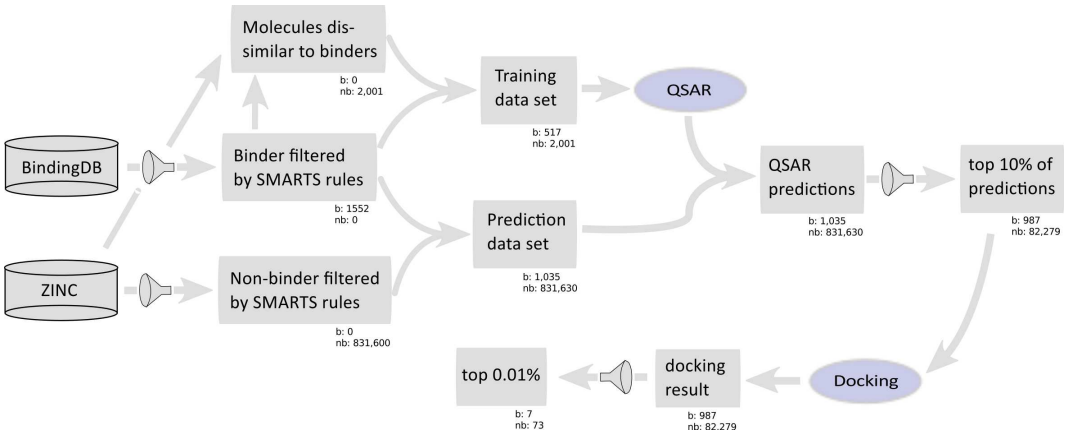


Figure 7.4: Schematic overview of the created virtual screening pipeline for carbonic anhydrase II. The number of binders (b) and non-binders (nb) are indicated for the output of each step.

Training decoys are obtained by similarity filtering of the ZINC [61] data base (*DBExporter*). Compounds that have a low to medium topological similarity to the binders are selected. Prediction decoys are found by filtering the ZINC data base for the SMARTS expressions defined by Klebe et al. [70]. We hereby assume that all compounds that match one of those expressions but were not part of the BindingDB data set are non-binders.

Subsequently, the training data set is created by combining the training decoys with the training binders and the prediction set by combination of prediction binders with prediction decoys (*MolCombine*).

A QSAR model is then generated by first reading the training data and generating features for it (*InputReader*), creating an initial model (ModelCreator) and applying feature selection techniques to it (*FeatureSelector*). The resulting final QSAR model is then used to predict the binding free energies of the compounds in the prediction data set.

The top 10% of compounds with respect to the predicted binding free energy are then filtered, all other molecules are discarded as likely non-binders. The resulting set contains approximately 1.2% binders (987/82,279 molecules), which in comparison to the 0.125% binders (1035/831,630 molecules) in the input data set is an enrichment by a factor of 9.6.

This reduced set is then be docked into the binding pocket of carbonic anhydrase II in order to enhance the enrichment even more. Therefore, the crystal structure for carbonic anhydrase II is downloaded from the Protein Data Bank [69] and prepared (*PDBCutter*, *ProteinProtonator*). Three-dimensional coordinates are generated for all

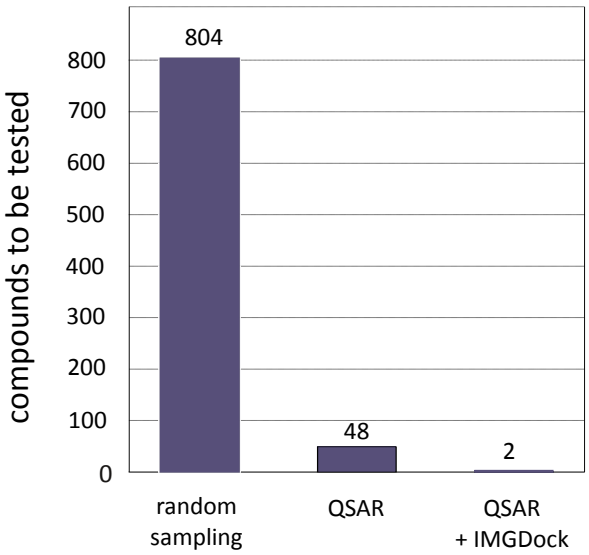


Figure 7.5: The number of molecules that would have to be experimentally tested in order to find just *one* carbonic anhydrase II inhibitor: random sampling in comparison to the output of QSAR analysis alone and QSAR combined with IMGDock (as explained in text).

compounds to be docked (*Ligand3DGenerator*) and a spatial constraint for use during docking is created using the available reference ligand (*SpatialConstraintDefiner*). All compounds are then docked into the binding-pocket by IMGDock and afterwards sorted ascendingly according to their assigned score (*DockResultMerger*). The top-scoring 0.01%, i.e. 80, molecules are filtered and make up the final result of our pipeline.

Analysis of this final output shows that it contains seven molecules known to inhibit carbonic anhydrase II. The resulting inhibitor-concentration of 8.75% (7/80 molecules) is thus equal to a total enrichment factor of 70. Thus, when, as part of a drug discovery project, a set of several hundred randomly chosen molecules would have been selected for experimental testing, most likely none of them would have turned out to inhibit carbonic anhydrase II. However, if the set of 80 molecules proposed by our pipeline were tested, seven inhibitors would have been found. Figure 7.5 furthermore visualizes the obtained enrichment.

We hope that this example illustrates the usefulness of CADDSuite and its contained tools. Note however, that no manual modification of either the target structure nor the pipeline setup was done. By specification of specific constraints (as explained in Section 5.2.2) and manual examination or correction of the target structure, results can often be improved even more.

8 Application: Virtual screening for biofilm-formation inhibitors

8.1 Introduction

Biofilm-mediated infections are a huge and common medical problem, leading to repulsion of medical implants, chronic infections and even death. Reasons for this, to be found in the biological function of such biofilms, have been explained in Section 2.4. In this chapter, we describe our work focused on searching inhibitors for *S. epidermidis* and *S. aureus*-mediated biofilm formation with the help of computer-aided drug design, including the methods presented in previous Chapters, and also present the hits that we found.

The chosen molecular target for inhibiting biofilm formation is Intercellular Adhesion A (IcaA). Its advantages as molecular target for drug discovery (as already mentioned in Section 2.4) are its lack of orthologues in humans, its already published mode of enzymatic action and the point that is has been shown to be essential for biofilm formation in staphylococci. However, no crystal structure of IcaA exists yet, which is of course a significant disadvantage for computer-aided drug design.

The only orthologue with known enzymatic function for which a crystal structure exists is SpsA, a protein relevant for spore-coat formation in *B. subtilis*. The substrate of IcaA and SpsA is identical (UDP-N-acetylglucosamine) and the function of both enzymes is the polymerization of glucosamine. Only a small difference exists in the exact way in which this polymerization is executed (β-1,6 connections created by IcaA, β-1,4 by SpsA). The sequence identity between IcaA and SpsA is rather low at 23%. Nevertheless, the crystal structure of SpsA does contain the UDP moiety of the substrate (which can be used as reference ligand during docking) in a relatively deep pocket, so that it is reasonable to assume that at least the UDP binding area is well conserved between SpsA and IcaA. Furthermore, there is a crystal structure of an unclassified protein in the Protein Data Bank (PDB identifier: 3BCV), which shows the same level of sequence identity to IcaA as SpsA does. Thus, the only way to immediately obtain a structure (model) of IcaA is to use homology modeling, for which the SpsA and 3BCV can serve as templates. This will be detailed in the next section.

The most likely way of delivery for any future biofilm inhibitors would be the coating of medical devices or implants. In this way, the hypothetical drug would be located in their target area (the surface of the implant or medical device) and would inhibit biofilm establishment by over time being slowly released from the coating. Thus, compounds' theoretical ability to reach target areas inside the human body are mostly

irrelevant, as are potential deactivations of molecules by metabolism that take place before the former can act as inhibitors (although metabolites might in principle still show toxic or other side-effects). For this reason, no predictions of absorption, distribution, metabolism, and excretion (ADME) are attempted. Toxicity on the other hand is very hard to predict, especially for arbitrary sets of molecules that do not necessarily show significant similarity to available training sets containing data about experimentally observed toxicity. Furthermore, the focus of this project is on finding leads for biofilm inhibitors, which, as explained in Section 2.1.4, can in a later step be optimized with respect to toxicity and other properties. Therefore, prediction of toxicity was judged to be unrealistic and not carried out.

The main goal of this project, as mentioned, is finding leads for biofilm formation inhibitors for *S. epidermidis* and *S. aureus*. The strategy we use for this employs homology modeling, computer-aided drug design, and experimental verification of hits by biofilm formation assays. An overview of the entire pipeline is given in Figure 8.1.

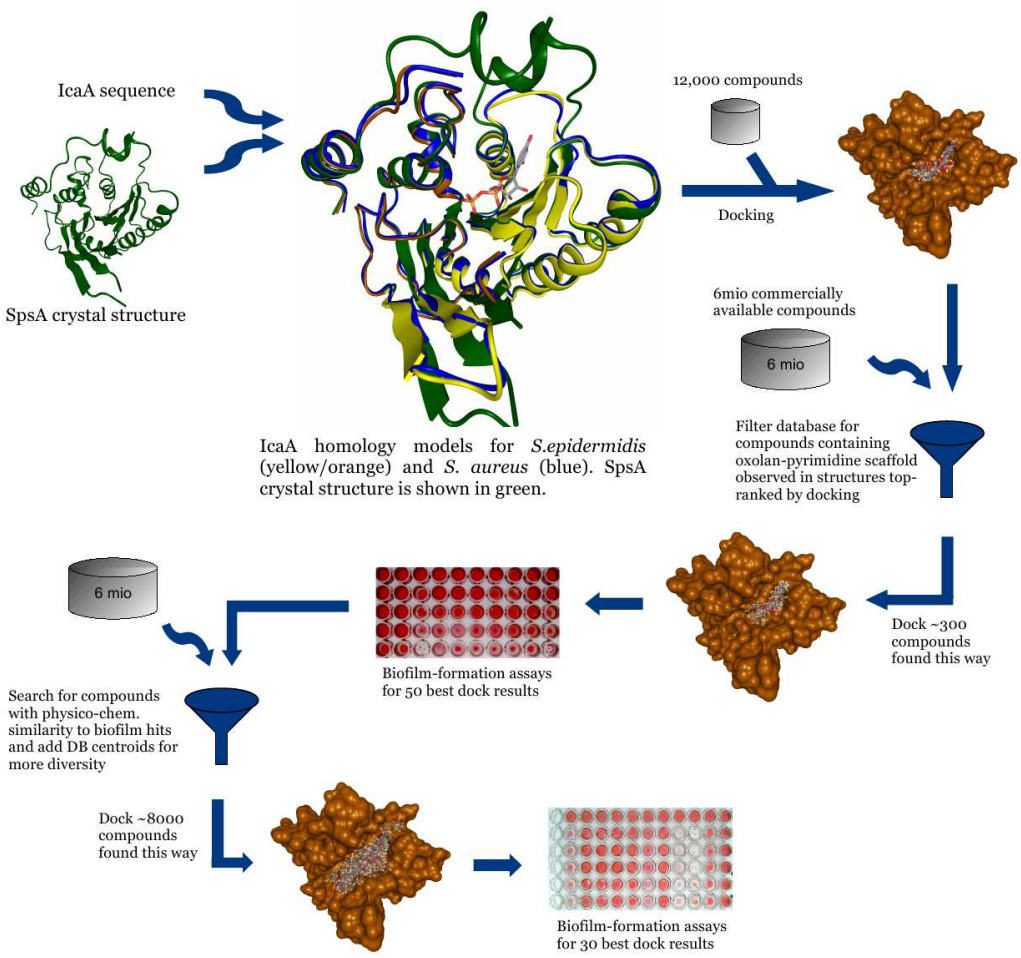


Figure 8.1: Schematic overview of our computational and experimental work towards inhibitors for IcaA.

The individual steps will be covered in more detail in the next sections. This chapter will also highlight several promising inhibitor candidates that have been found and experimentally validated by the pipeline described in the following sections.

A variety of CADDSuite tools (or their predecessors) are used for this project. Among them are *DBImporter*, *DBExporter*, *VendorFinder*, *ProteinProtonator*, *LigandFileSpliiter*, *Ligand3DGenerator*, *ProteinCheck*, *LigCheck*, *GridBuilder*, *IMGDock*, *DockResultMerger*, *Converter* and *MolDepict*. Those tools cover most modules offers by CADDSuite; the only modules not used are the ones for QSAR and Rescoring, which is only due to the lack of experimental binding free energy measurements for IcaA. If such measurements can be conducted in the future, use of our QSAR modeling and, particularly, TaGRes, might be highly interesting and helpful to find molecules with higher binding affinities than the hits described in this chapter. Hence, the applicability of a lot of CADDSuite tools to this project, together with the results, i.e. the hits that are found, also nicely depicts the usefulness of CADDSuite for drug design projects.

8.2 Homology Modeling

In a first step, homology models are to be built for IcaA. An important requirement for this is a good multiple alignment of IcaA and the two chosen template proteins, SpsA and 3BCV.

Since separate homology models are to be generated for the two *Staphylococci* strains, this multiple alignment is generated for the sequence of SpsA, 3BCV and the sequence of IcaA of *S. epidermidis* and *S. aureus* by use of the program ClustalW [71] (Figure 8.2). Problematic hereby hereby is the low sequence identity between SpsA and IcaA (23%), and the point that the sequence of IcaA is much longer than the one of SpsA, leading to large fractions of IcaA that do not show any homology do SpsA (e.g., amino acids 204-251 in Figure 8.2). Nevertheless, the UPD binding-pocket (amino acids 47-204 in Figure 8.2) shows significantly stronger conservation than the overall sequence, so that homology models should contain reasonable reliability for this area.

The alignment, together with the crystal structure of SpsA, was then used to build homology models, one for IcaA of *S. epidermidis* and one for IcaA of *S. aureus*, by use of the program MODELLER [72]. A rough comparison of the resulting protein models and SpsA is shown in Figure 8.3).

Due to its mentioned higher conservation and distinct structural definition, the UDP binding site, as it appears in the derived homology models, is determined as the primary binding site. Thus, all following molecular docking steps aim to dock molecules into this area of IcaA.

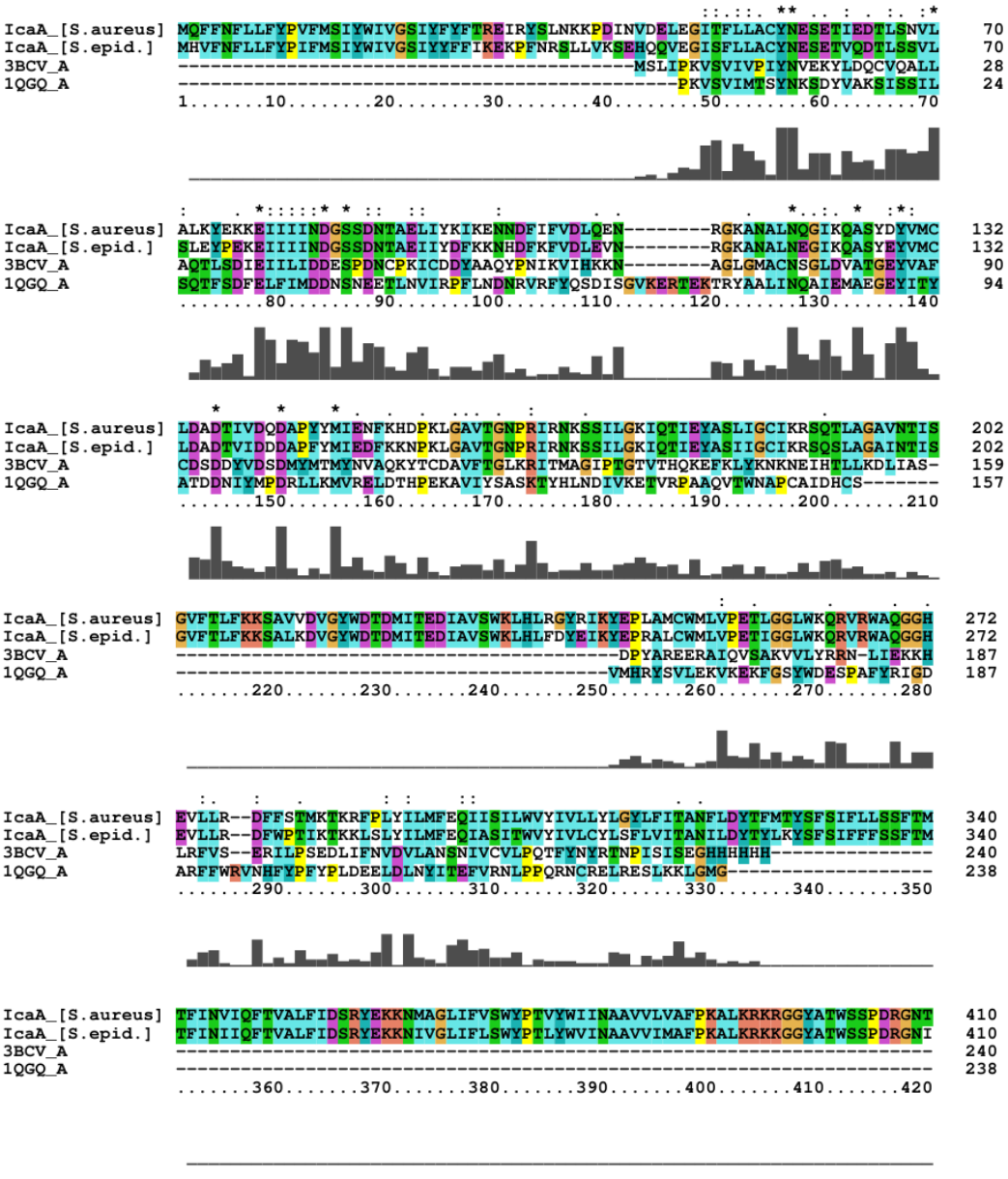


Figure 8.2: Mulitple alignment of SpsA (PDB ID 1QGQ), a related protein (PDB ID 3BCV) and IcaA of *S. epidermidis* and *S. aureus*.

Figure 8.3: SpsA crystal structure (green) and IcaA homology models for S.epidermidis (yellow/orange) and S. aureus (blue). Parts of IcaA marked in yellow indicate the sequence area with higher conversation.

8.3 Scaffold finding

Since no inhibitors are known, yet, for IcaA, and therefore knowledge about putative pharmacophores cannot be extracted from those, we employ a docking of a diverse set of molecules in order to investigate whether any particular molecular fragments might be especially important for the compounds' binding in the UPD pocket of IcaA.

Thus, PubChem [73] is filtered for compounds that contain the linear fragment of UDP-N-acetylglucosamine that was determined to bind most strongly to IcaA, as shown in Figure 8.4. The resulting 12,000 diverse compounds obtained this way are then docked into the crystal structure of SpsA and into the homology models. The top scored compounds in these results (nearly) all contained an pyrimidine-oxolan scaffold (as is also the case with the reference ligand UDP-N-acetylglucosamine), so that it is used a constraint in the following step. An example of an pyrimidine-oxolan scaffold in shown in Figure 8.5.

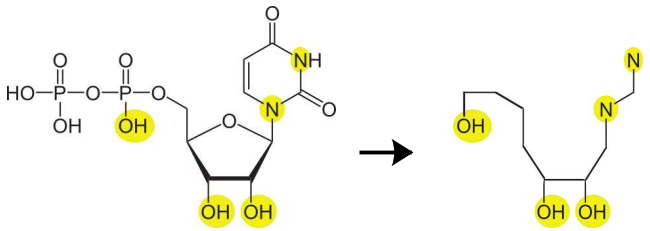


Figure 8.4: Left: UDP-N-acetylglucosamine, with atoms predicted to interact most strongly with IcaA highlighted in yellow. Right: extracted linear fragment.

8.4 Virtual screening I

In the next step, a virtual screening with compounds containing an pyrimidine-oxolan scaffold is carried out. The homology model for IcaA of *S. epidermidis* is used primarily during docking, and the one of *S. aureus* is utilized for docking for comparison purposes.

In order to be able to experimentally validate hits found by docking, a database of purchasable compounds is created from the libraries supplied by 15 vendors. In total, this database contains about 5.4 million compounds and can be filtered with the tools supplied by CADDSuite according to a variety of criteria. Next, this database is searched for structures containing an oxolan-pyrimidine scaffold, resulting in approximately 300 molecules.

Those molecule are docked into the homology models and the output is sorted ascendingly according to the score (i.e. the binding free energy estimate) assigned by the docking algorithm.

8.5 Hit verification I

50 molecules top-scored by docking are purchased and tested in a biofilm assay in the Götz lab. Therefore, *S. aureus* SA113 and *S. epidermidis* RP62A strains as well as the compounds that are to be tested are dissolved in dimethyl sulfoxide (DMSO) and added to microtiter polystyrene wells containg TSB (tryptic soy broth) buffer with 0.25% glucose. After incubation for 24 hours, biofilm-forming cells adhere to the plates and are detected, after a washing step, by safranin staining.

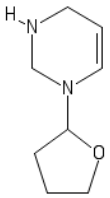


Figure 8.5: Example of an pyrimidine-oxolan scaffold.

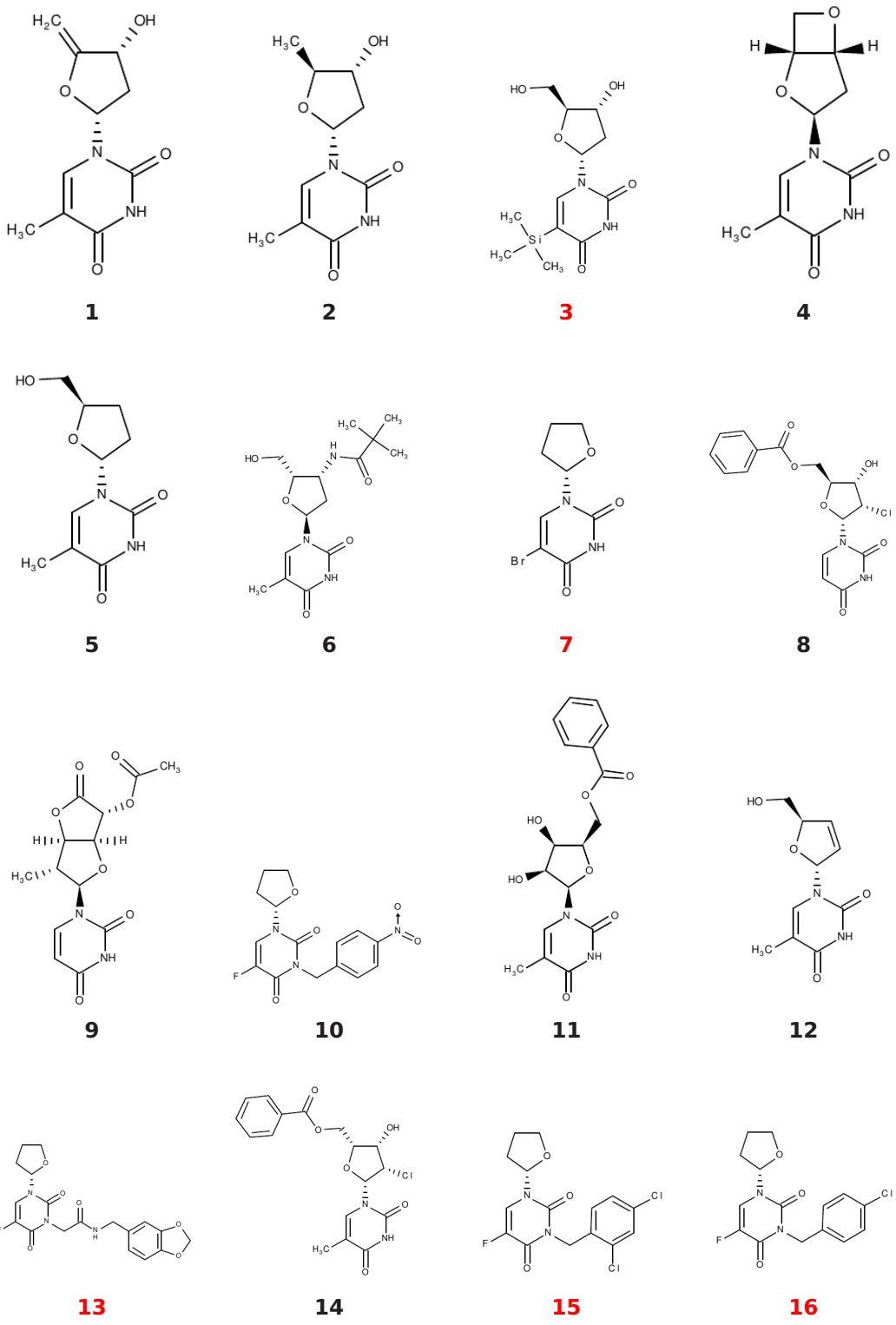


Figure 8.6: Hits found by virtual screening I. Hits confirmed by biofilm formation assays are highlighted with red numbering. Part 1 of 4.

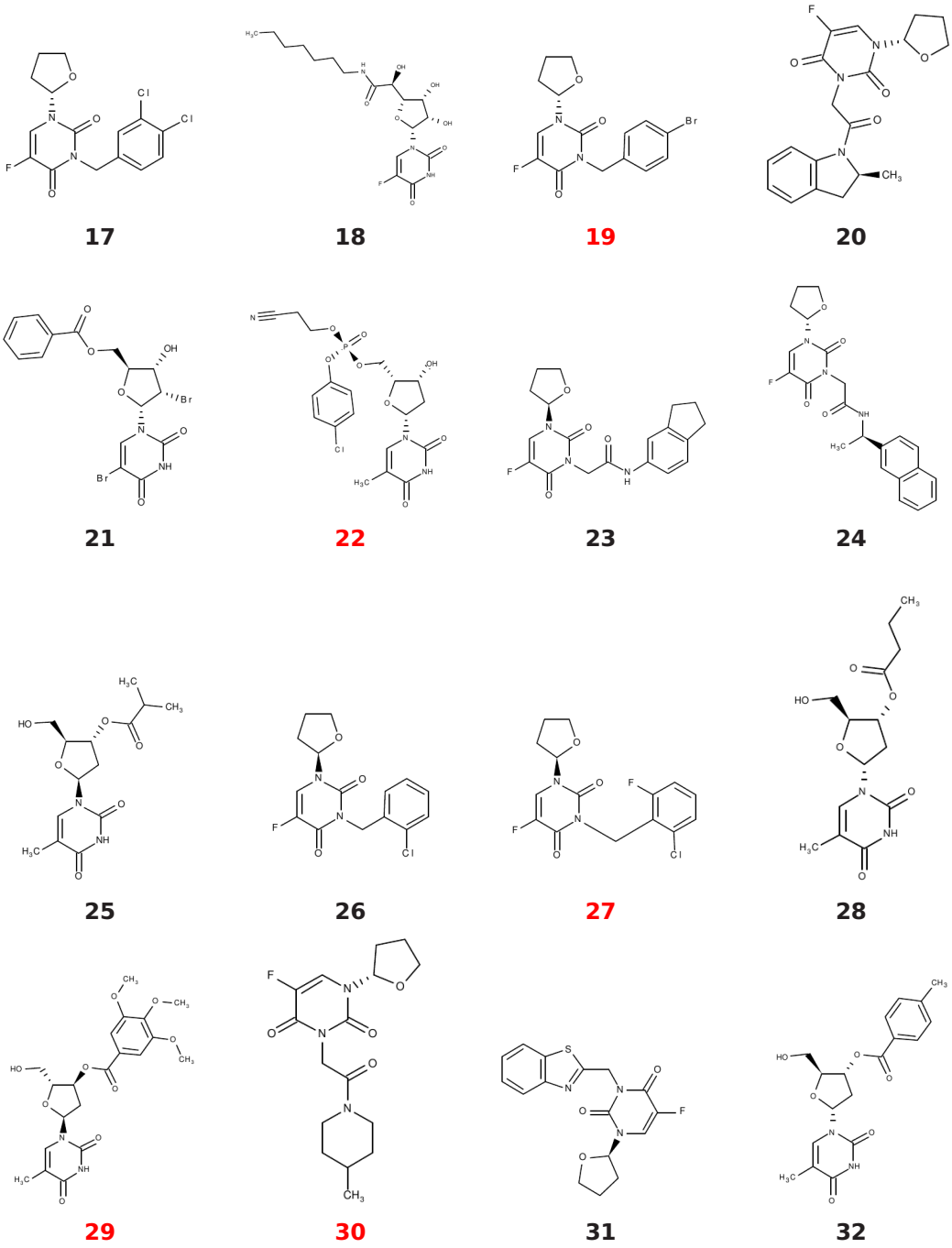


Figure 8.6: Part 2 of 4.

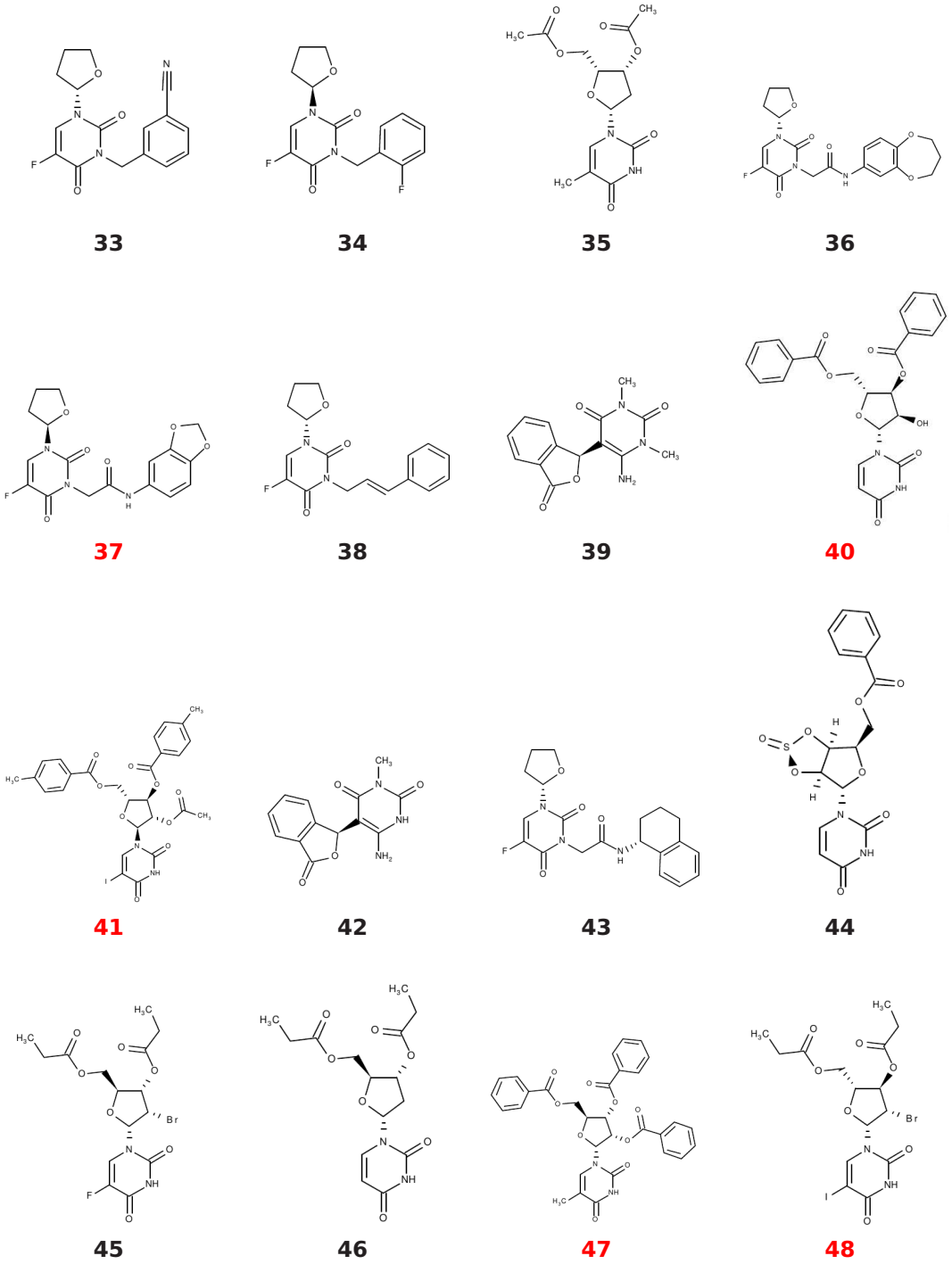
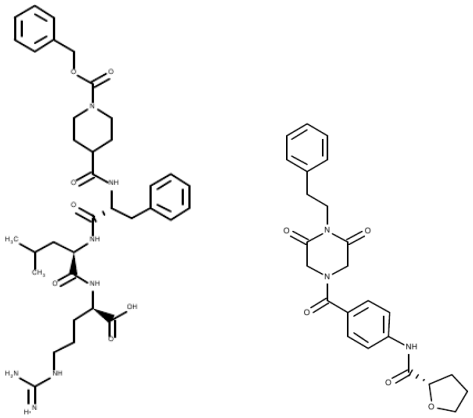


Figure 8.6: Part 3 of 4.



49 **50**

Figure 8.6: Part 4 of 4.

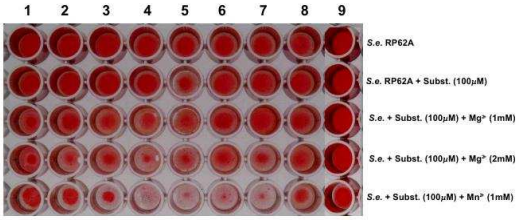


Figure 8.7: Biofilm-formation assay results for *S. epidermidis* RP62A for some of the compounds found during the virtual screening I.

This way, 15 compounds can be identified that show inhibition of biofilm formation in *S. aureus* and/or *S. epidermidis* at concentrations of 100 μM. All experimentally tested compounds are depicted in Figure 8.6, where molecule showing biofilm inhibition in either *S. epidermidis* or *S. aureus* have been highlighted with red numbering. Selected biofilm assays are shown in Figure 8.7 as examples.

8.6 Virtual screening II

After obtaining experimentally validated hits in the previous steps, we then try to find compounds inhibiting biofilm formation that, in total, show more diversity.

Thus, in contrast to virtual screening I, no scaffold constraints are used now. Instead, compounds having a moderately high similarity (i.e., a Tanimoto coefficient between 0.75 and 0.85) to validated hits obtained in virtual screening I are screened. In addition, in order to add even more diversity, our database containing the purchasable compounds is clustered (by k-means clustering employing the binary, fragment-based fingerprints) and 5,400 cluster centroids are retrieved.

The approximately 8,000 compounds obtained in these ways are docked into the UDP binding pocket of the IcaA homology model. The output is then sorted ascendingly

according to the score assigned by the docking algorithm.

8.7 Hit verification II

30 top-scoring compounds identified by the docking approach are purchased and tested experimentally by use of biofilm assays. The protocol used for experimental validation is the same as in Section 8.5.

As a result of these biofilm assays, six compounds were validated to possess the ability to inhibit biofilm formation in either *S. aureus* or *S. epidermidis* at a concentration of 100 μM. All compounds experimentally tested in this step are depicted in Figure 8.8. Selected biofilm assays are shown in Figure 8.9 as examples.

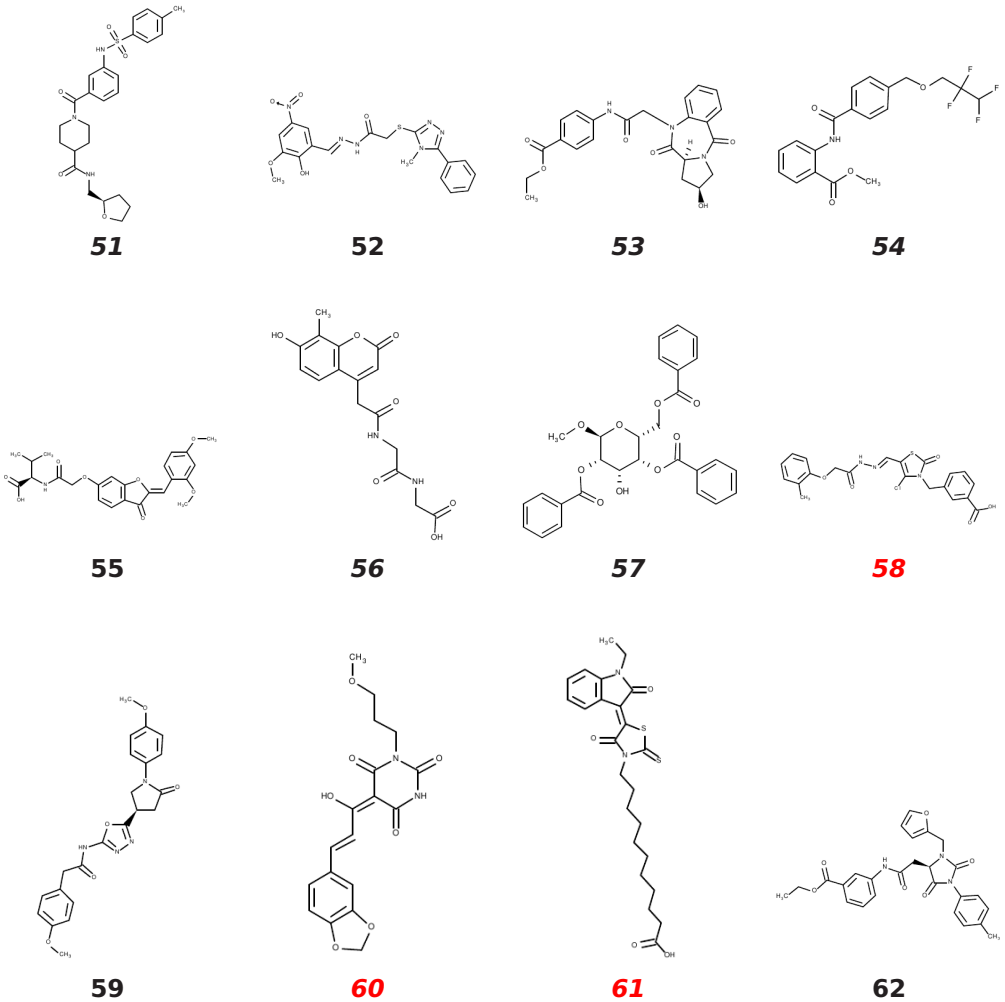


Figure 8.8: Hits found by virtual screening II. Hits confirmed by biofilm formation assays are highlighted with red, compounds obtained as database cluster centroids with italic numbering. Part 1 of 3.

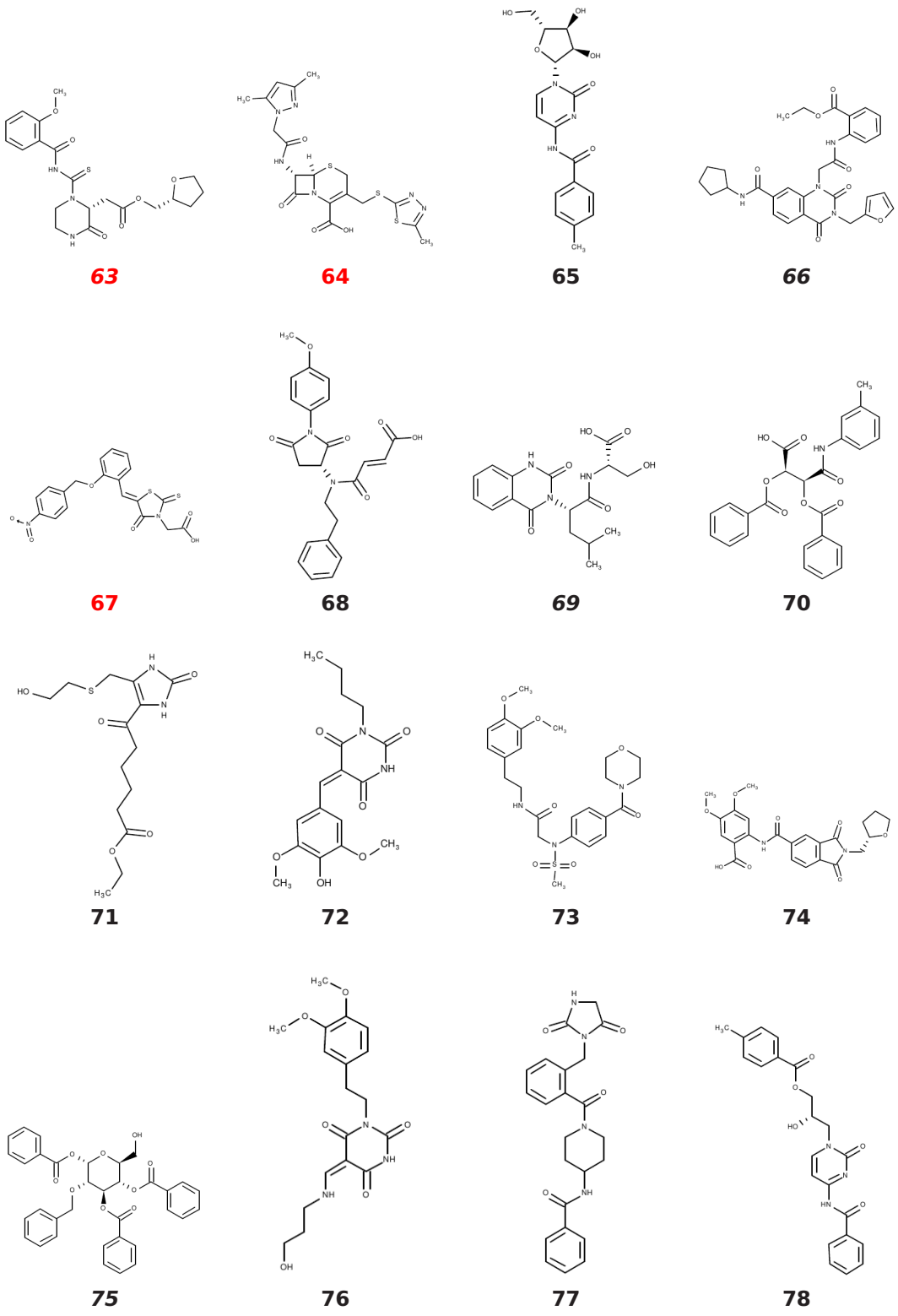
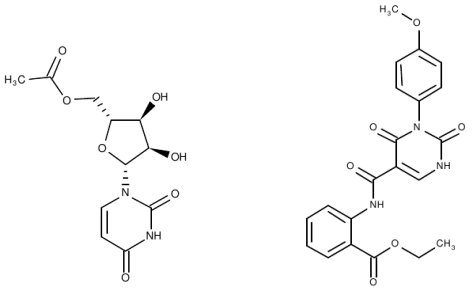


Figure 8.8: Part 2 of 3.



79 **80**

Figure 8.8: Part 3 of 3.

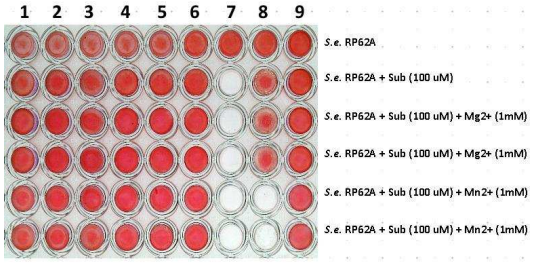


Figure 8.9: Biofilm-formation assay results for *S. epidermidis* RP62A for some of the compounds found during the virtual screening II.

8.8 Discussion

By a combination of virtual and experimental procedures, it has been possible to find 21 candidates for biofilm-formation inhibitors.

15 of those candidates have been found during the first screening. Since those compounds were obtained by filtering our database by use of a pyrimidine-oxolan scaffold constraint, the existence of this scaffold, which is also contained in the reference ligand, within the hits supports the hypothesis that those compounds do indeed bind to IcaA. However, final proof of this hypothesis can of course only be obtained by *in vitro* determination of inhibition constants (as described in Section 2.2). The performed *in vivo* biofilm assays, on the other hand, do not suffice to validate this. In principle, the observed biofilm formation inhibition might be due to compounds' influence on other molecular structures (other Ica proteins or other enzymes participating in the creation of biofilms) or could be caused by aggregation of the compounds [74–76]. Until now, *in vitro* tests could not be done since purification of IcaA by cooperation partners has not been not successful, yet.

Six more hits have been found and experimentally validated by biofilm-assays during the second screening. Interesting hereby is that none of those hits contains a pyrimidine-oxolan scaffold, although molecules containing one were not excluded in the database search. All those hits do however contain at least one pyrimidine and one 5-membered heterocycle, the latter similar to oxolan (tetrahydrofuran). These two

fragments are not direcly connected to each other in the obtained hits, but in topological (and thus spacial) neighborhood, indicating that the primidine-oxolan scaffold ligand can be replaced by similar scaffolds. Nevertheless, since no compounds showing biofilm inhibition have been found that do not contain at least a similar scaffold similar, this constraint might quite likely be essential for binding to IcaA.

Another interesting point observed in the results of the second screening is that four out of the six observed and experimentally confirmed hits are compounds that have been obtained as database centroids (as described above). This shows that our virtual screening worked quite well, even for a set of very diverse molecules.

In any case, while the results obtained so far are very promising and exciting, they need to be validated with *in vitro* measurements of binding free energies, for reasons already mentioned above. If and when IcaA can be enriched successfully with adequate quality, these experiments can be performed, which would allow more detailed analysis of chemical differences between binders and non-binders and thus might lead to possible optimizations of the current hits. If furthermore, the crystallization of the purified IcaA succeeds, a resulting crystal structure could be used for docking. While the use the homology models for docking seems to have worked reasonably well, the improved accuracy of a crystal structure would be very helpful, especially if the goal is to optimize binding to IcaA. Also, if binding affinity measurements and, optionally, the crystal structure are available, Target-Specific Rescoring (TaGRes), as described in Chapter 6, will be a very promising approach for determination of important contributions to the binding free energy and search or generation of compounds with higher binding affinity.

9 Discussion & Conclusion

In this dissertation, new approaches for computer-aided drug design were presented: a framework for Quantitative Structure-Activity (QSAR) modeling, a receptor-ligand scoring function and a docking algorithm, a three-dimensional target-specific rescoring procedure and CADDSuite, a software suite of contains all the aforementioned algorithms and a large set of additional, auxiliary tools and algorithms.

First, we presented a framework for QSAR modeling that includes all necessary capabilities to read input, generate molecular descriptors, generate a variety of different regression and classification models, automatically select relevant descriptors and evaluate the quality of QSAR models. Thus, all functionality for establishing even complex QSAR pipelines is available. Furthermore, all algorithms were implemented with focus on high speed and numerical stability and all functionalities are available as CADDSuite tools, making them easy to use separately and in combination. Using a number of different data sets, we showed that by use of our software, it is easily possible to obtain high-quality QSAR models, out-performing models published for the same data sets, while still only using a small number of descriptors and thus enhancing their interpretability. Hence, this QSAR framework is optimally suited for use in lead discovery or lead optimization steps in drug design projects.

Next, we presented IMGDock, an efficient deterministic docking algorithm, employing an empirical scoring function and an iterative multi-greedy pose-finding approach. In addition, algorithms and tools for automatic determination of spatial or interaction-based scoring constraints and for detection of water molecules tightly bound to the receptor structure have been developed, which for consistency use the same scoring function as the docking approach. The created scoring function was shown to result in good correlation with experimentally determined binding free energies by use of the a large data set containing more than 1,000 co-crystal structures. The docking approach was thoroughly evaluated using a well-established docking benchmark set (DUD) for 40 protein targets. For many of those targets our approach outperforms state-of-the-art programs and even on average performs as well or better than most of them. Furthermore, IMGDock is fast and stable, available free of charge as open-source and can easily be deployed on compute clusters, clouds, or grids.

In order to in the future enhance the results of IMGDock even more, there are primarily two options: The optimization of the specificity of the scoring function and the selective integration of relevant protein flexibility. The first option would create new scoring terms or modify existing ones in order to reduce the false positive rate. Thus, molecules that actually do not bind to the target structure would be more reliably predicted not to do so. While the enhancement of specificity is a perpetual goal in

the field of receptor-ligand scoring research, there are some promising approaches for our scoring function. One way would be a more detailed calculation of desolvation effects, although great care has to be taken to ensure that the modified scoring function would still be precalculatable as grids or otherwise fast enough for docking purposes. The second option would be to try to find types of molecular interactions that are currently not explicitly modeled and that render poses currently generated for non-binders chemically unfavorable. If any are found, special scoring terms could be developed for them, including possibly also cross-terms with the aforementioned solvation terms.

However, as always during the development of scoring functions, there are several caveats when trying to modify existing or create new scoring terms. The composition of data sets used to optimize and evaluate the function are two such issues. This data set of protein-ligand complex structures employed for optimization of coefficients needs to be large and diverse enough, and may not just contain data for certain classes of enzymes (at least as long as the goal still is to develop a universally applicable scoring function). The make-up and quality of the data set used to finally evaluate docking and scoring is another important issue. Data sets like the DUD library commonly used for this derive decoys by *in silico* procedures (e.g., similarity-based), i.e. decoys are not experimentally validated to be non-binders. In our opinion, it is important to keep this point in mind when evaluating docking and scoring algorithms. It may furthermore be desirable to devise procedures for checking the plausibility of decoys contained in such data sets. To name just one example, we observed decoys in DUD data sets that actually were nearly identical to experimentally determined binders, but just had one additional terminal moiety that, after the molecule was docked, turned out to be located outside the binding pocket and pose no sterical problems. Other examples in theory might include putative non-binders that contain significantly large fragment that are also observed in binders or that are highly similar to fragment found in binders. It may also be helpful to include three-dimensional information, so that high similarity or even identity of fragments that will probably be located near the active site is considered worse than similarity of ligand fragments far apart from the active site. Since in dubious cases it is of course not possible (without wet lab tests) to tell whether such molecules definitely do or do not bind to the target structure, such molecules should not be included into data sets intended to test the power of docking to discretize binders from non-binders.

One promising, recently published, approach to prevent false decoys is the one employed by DEKOIS [77]. Comparison of binary molecular fingerprints containing information about the occurrence of atoms with various functional roles are used to achieve this. However, it will certainly be necessary to investigate whether the use of, for example, functional group count fingerprints or the explicit search for maximal common substructures, are even more helpful before the data sets generated by DEKOIS might be established as generally accepted replacements for DUD. For this reason, we did not (yet) use DEKOIS to evaluate IMGDock, but DUD, which is currently widely regarded as the standard benchmark set for docking.

Integration of protein flexibility would be another way to try to enhance the quality

of docking results even more. However, this will of course only be helpful for protein targets that exhibit significant flexibility near the binding pocket. Also, great care has to be taken to make sure that only flexibility that is significant and relevant for ligand binding is integrated into the docking approach. If otherwise too many side chains are considered to be flexible, although they in reality are not flexible, or if too many backbone conformations are used that actually would never occur in vivo, then this will not only not improve docking results but will most likely significantly deteriorate them. Advanced analysis of molecular dynamics simulations or nuclear magnetic resonance (NMR) spectroscopy could help to study protein flexibility towards this end. Thus, integration of protein flexibility in a biologically sensible way is far from trivial, but offers significant enhancement of docking results for many protein targets.

We next presented TaGRes, Target-Specific Grid-based Rescoring. After docking a set of compounds whose binding affinities are known, the three-dimensional poses generated by docking are used together with the binding free energies to generate a regression model of the binding free energy as a function of molecular interaction scores in discretized areas of space. The obtained model can than be applied to other compounds (whose binding affinity is unknown) after they have been docked, thereby rescoring these compounds. Thus, this approach takes into account receptor-ligand interactions, their three-dimensional locations and their target-specific importances. We showed that by use of TaGRes, the quality of docking results with respect to their power to discretize between binders and decoys could be strongly enhanced. Specifically, in six out of eight cases docking results for DUD targets could be decisively improved by TaGRes, employing for the model generation step data for the same molecular target obtained from BindingDB [60]. Thus, TaGRes should be very interesting for both lead discovery and lead optimization steps and is applicable to any molecular target as long as at least several binding free energy measurements for this target are available. While, due to protein structure and ligand data set dependency, it is impossible to state a universal lower threshold for the number of required binding affinity measurements, it is, according to our experience, possible to achieve a very good rescoring with as few as about 20 measurements. If for a given training data set, TaGRes determines (by automatic use of nested validation) the obtained model to most likely have no significant predictive power, it automatically aborts. Thus, if a model is successfully generated, there is significant probability that a good rescoring will be possible for compounds that show at least a moderate chemical similarity to the molecules of the training data set. The only further check that, as described in Chapter 6, is advisable to perform is to test whether the median chemical similarity of these two data sets is at least as high as 0.5 (Tanimoto coefficient based on binary, pathway-based fingerprints). This comparison can however be easily done by use of the *SimilarityAnalyzer* tool that is part of CADDSuite.

Possible improvements in our opinion are primarily to be found in the generation of the training data sets. This involves the selection of compounds, as well as the reliability of binding affinity measurements. For example, a certain degree of diversity within the training data set should be present, with respect to both binding affinity and topology. Furthermore, there sometimes exist cases with drastically different binding free energy

measurements for one and the same compound in publicly available databases such as BindingDB. Reasons for this may either be due to experimental errors or due to experiments done (by different scientists) under strongly different conditions (e.g., pH). Enhancements of the specificity of the scoring function, as discussed above, would on the other hand most likely only have a very limited impact on the results obtained by TaGRes, since TaGRes achieves specificity by direct modeling (i.e. regression) of the differences of receptor-ligand interaction patterns between binders and non-binders (or between strong and week binders).

In this dissertation, we next presented CADDSuite, a flexible and open workflow-enabled framework for computer-aided drug design. CADDSuite makes solving common computer-aided drug design tasks considerably easier by providing all necessary tools for structure preparation and checking, QSAR, docking, rescoring, and analysis of results, all of which are useable in a simple and consistent manner. That is, all aforementioned approaches have been integrated into CADDSuite. Even complex assignments can be accomplished by combining the provided tools into a pipeline. To make creation of those pipelines even easier, CADDSuite has also been integrated into the well-known workflow system Galaxy. This essentially allows a user to create drug design workflows directly from a web browser, without any need for software installations on his local computer, and also makes it possible to directly submit workflows to a compute cluster, grid or cloud. Hence, CADDSuite is optimally suited for high-performance computing (HPC) in the field of computer-aided drug design. CADDSuite is an extensive framework, written as efficiently as possible and using the BALL [78] library for standard handling of chemical data (data structures for chemical systems, reading/writing of standard chemical files). CADDSuite encompasses about 100 classes, 50 tools, and approximately 65,000 lines of code. We furthermore illustrated the usefulness of CADDSuite by creating a virtual screening workflow for the target carbonic anhydrase II, which, as shown, resulted in a high enrichment of inhibitors in the final output. CADDSuite is available free of charge, licensed under the GNU GPL, from `http://www.ball-project.org/caddsuite`.

Last but not least, we recounted our efforts to find inhibitors for bacterial biofilm formation using a combination of computer-aided drug design techniques, provided by CADDSuite tools, and experimental validation steps. We found 21 compounds that exhibit biofilm formation inhibition in either *S. epidermidis* or *S. aureus* according to biofilm assay verifications. These are very encouraging results, which can hopefully, in the longer term, help to develop medical drugs that prevent infections due to biofilm creation on medical devices or implants. Since those kinds of infections are very frequent and often lead to severe medical problems for the affected patients, from the need to remove artificial joints to even death, the impact of such drugs would of course be considerable. Besides being encouraging in themselves, those results are also another nice example for the usefulness of CADDSuite.

The next steps necessary for continued drug design efforts in this field in our opinion would clearly consist of high-quality enrichment and purification of IcaA, determination of binding free energies for the hits obtained so far and the crystallization of IcaA. If and when this (or at the very least the first two steps) are achieved, it will be feasible

to apply computer-aided drug design procedures to try to find compounds that bind even stronger to IcaA or to modify existing hits towards this goal.

Appendix

ID	*descriptor name*	*ID*	*descriptor name*
1	AtomicPolarizabilities	50	PolarVdWSurface
2	AtomInformationContent	51	PositivePolarVdWSurface
3	BondPolarizabilities	52	PositiveVdWSurface
4	FormalCharge	53	RelHydrophobicVdWSurface
5	MeanAtomInformationContent	54	RelNegativePolarVdWSurface
6	MolecularWeight	55	RelNegativeVdWSurface
7	NumberOfAromaticAtoms	56	RelPolarVdWSurface
8	NumberOfAromaticBonds	57	RelPositivePolarVdWSurface
9	NumberOfAtoms	58	RelPositiveVdWSurface
10	NumberOfBonds	59	VdWSurface
11	NumberOfBoron	60	VdWVolume
12	NumberOfBromine	61	terminal primary C(sp3)
13	NumberOfCarbon	62	total secondary C(sp3)
14	NumberOfChlorine	63	total tertiary C(sp3)
15	NumberOfDoubleBond	64	total quaternary C(sp3)
16	NumberOfFlourine	65	ring secondary C(sp3)
17	NumberOfHeavyAtoms	66	ring tertiary C(sp3)
18	NumberOfHeavyBonds	67	ring quaternary C(sp3)
19	NumberOfHydrogen	68	aromatic C(sp2)
20	NumberOfHydrogenBondAcceptors	69	unsubstituted benzene C(sp2)
21	NumberOfHydrogenBondDonors	70	substituted benzene C(sp2)
22	NumberOfHydrophobicAtoms	71	non-aromatic conjugated C(sp2)
23	NumberOfIodine	72	terminal primary C(sp2)
24	NumberOfNitrogen	73	aliphatic secondary C(sp2)
25	NumberOfOxygen	74	aliphatic tertiary C(sp2)
26	NumberOfPhosphorus	75	allenes groups
27	NumberOfRotatableBonds	76	terminal C(sp)
28	NumberOfRotatableSingleBonds	77	non-terminal C(sp)
29	NumberOfSingleBonds	78	cyanates
30	NumberOfSulfur	79	isocyanates
31	NumberOfTripleBonds	80	thiocyanates
32	PrincipalMomentOfInertia	81	isothiocyanates
33	PrincipalMomentOfInertiaX	82	carboxylic acids
34	PrincipalMomentOfInertiaY	83	esters
35	PrincipalMomentOfInertiaZ	84	primary amides
36	RelNumberOfRotatableBonds	85	secondary amides
37	RelNumberOfRotatableSingleBonds	86	tertiary amides
38	SizeOfSSSR	87	(thio-)carbamates
39	VertexAdjacency	88	acyl halogenides
40	VertexAdjacencyEquality	89	thioacids
41	BalabanIndexJ	90	dithioacids
42	ZagrebIndex	91	thioesters
43	RelNegativePartialCharge	92	dithioesters
44	RelPositivePartialCharge	93	aldehydes
45	TotalNegativePartialCharge	94	ketones
46	TotalPositivePartialCharge	95	urea (-thio) derivatives
47	Density	96	carbonate (-thio) derivatives
48	HydrophobicVdWSurface	97	amidine derivatives
49	NegativePolarVdWSurface	98	guanidine derivatives

Table A1: List of molecular descriptors generated by our software. Part 1 of 2.

99	imines	147	R=CRX
100	oximes	148	R#CX
101	primary amines	149	CHRX2
102	secondary amines	150	CR2X2
103	tertiary amines	151	R=CX2
104	N hydrazines	152	CRX3
105	N azo-derivatives	153	halogene on aromatic ring
106	nitriles	154	X on ring C(sp3)
107	positive charged N	155	X on ring C(sp2)
108	quaternary N	156	halogene on exo-conjugated C
109	hydroxylamines	157	Aziridines
110	nitrosamine	158	Oxiranes
111	nitroso groups	159	Thiranes
112	nitro groups	160	Azetidines
113	imides	161	Oxetanes
114	hydrazones	162	Thioethanes
115	hydroxyl groups	163	Beta-Lactams
116	aromatic hydroxyls	164	Pyrrolidines
117	primary alcohols	165	Oxolanes
118	secondary alcohols	166	tetrahydro-Thiophenes
119	tertiary alcohols	167	Pyrroles
120	ethers	168	Pyrazoles
121	hypohalogenides	169	Imidazoles
122	anhydrides	170	Furanes
123	water	171	Thiophenes
124	thiols	172	Oxazoles
125	thioketones	173	Isoxazoles
126	sulfides	174	Thiazoles
127	disulfides	175	Isothiazoles
128	sulfoxides	176	Triazoles
129	sulfones	177	Pyridines
130	sulfenic acids	178	Pyridazines
131	sulfinic acids	179	Pyrimidines
132	sulfonic acids	180	Pyrazines
133	sulfuric acids	181	135-Triazines
134	sulfites	182	124-Triazines
135	sulfonates	183	Phenoles
136	sulfates	184	Phenyles
137	sulfonamides/sulfinamides/sulfenamides	185	Toluenes
138	phosphites/thiophosphites	186	Glucose
139	phosphates/thiophosphates	187	Fructose
140	phosphanes	188	Methyl
141	phosphonates/thiophosphonates	189	Halogenides
142	phosphoranes/thiophosphoranes	190	Propyl
143	CH2RX	191	Butyl
144	CHR2X	192	Pentyl
145	CR3X	193	Prenyl
146	R=CHX		

Table A1: Part 2 of 2.

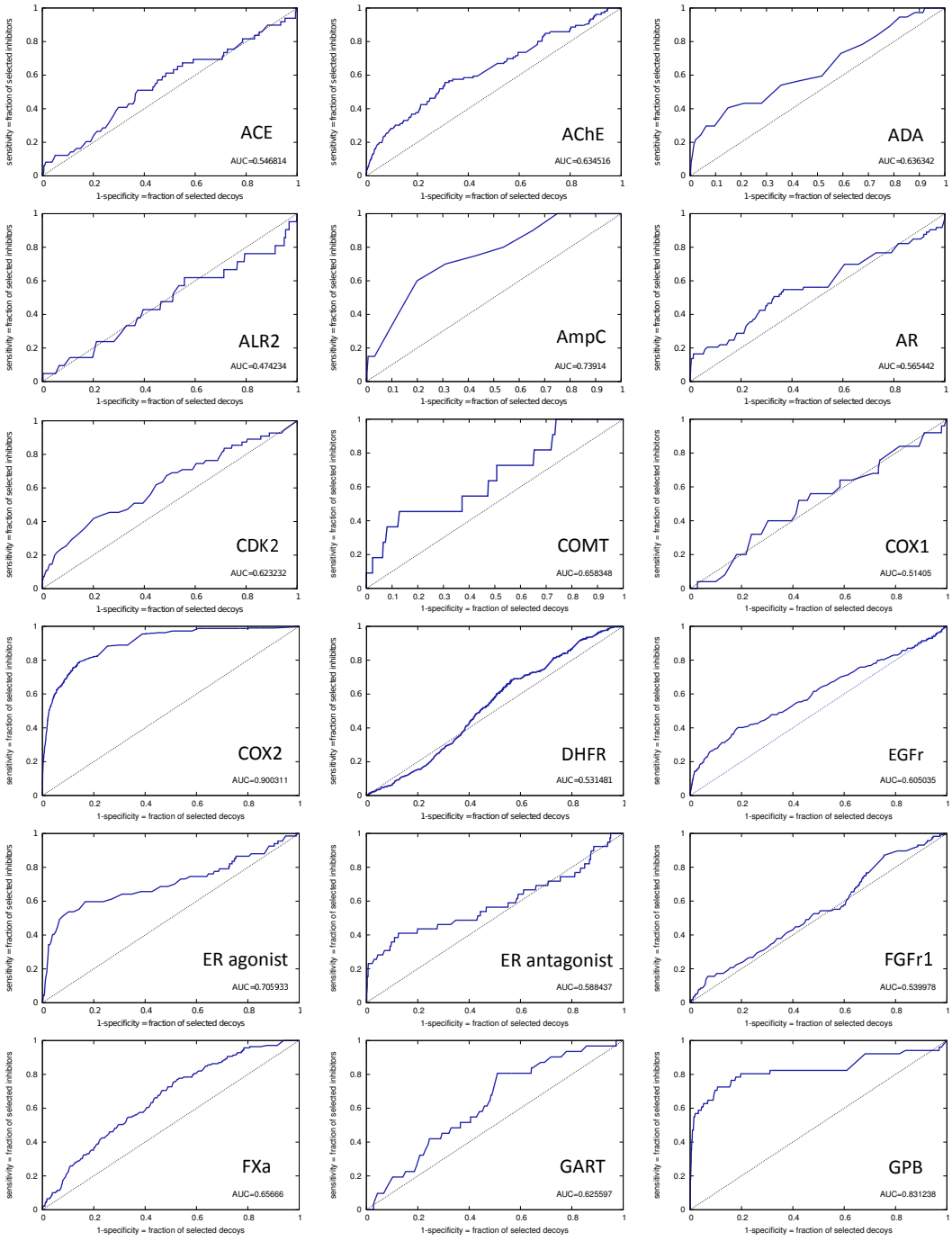


Figure A1: ROC plots for all DUD data sets. Part 1 of 3

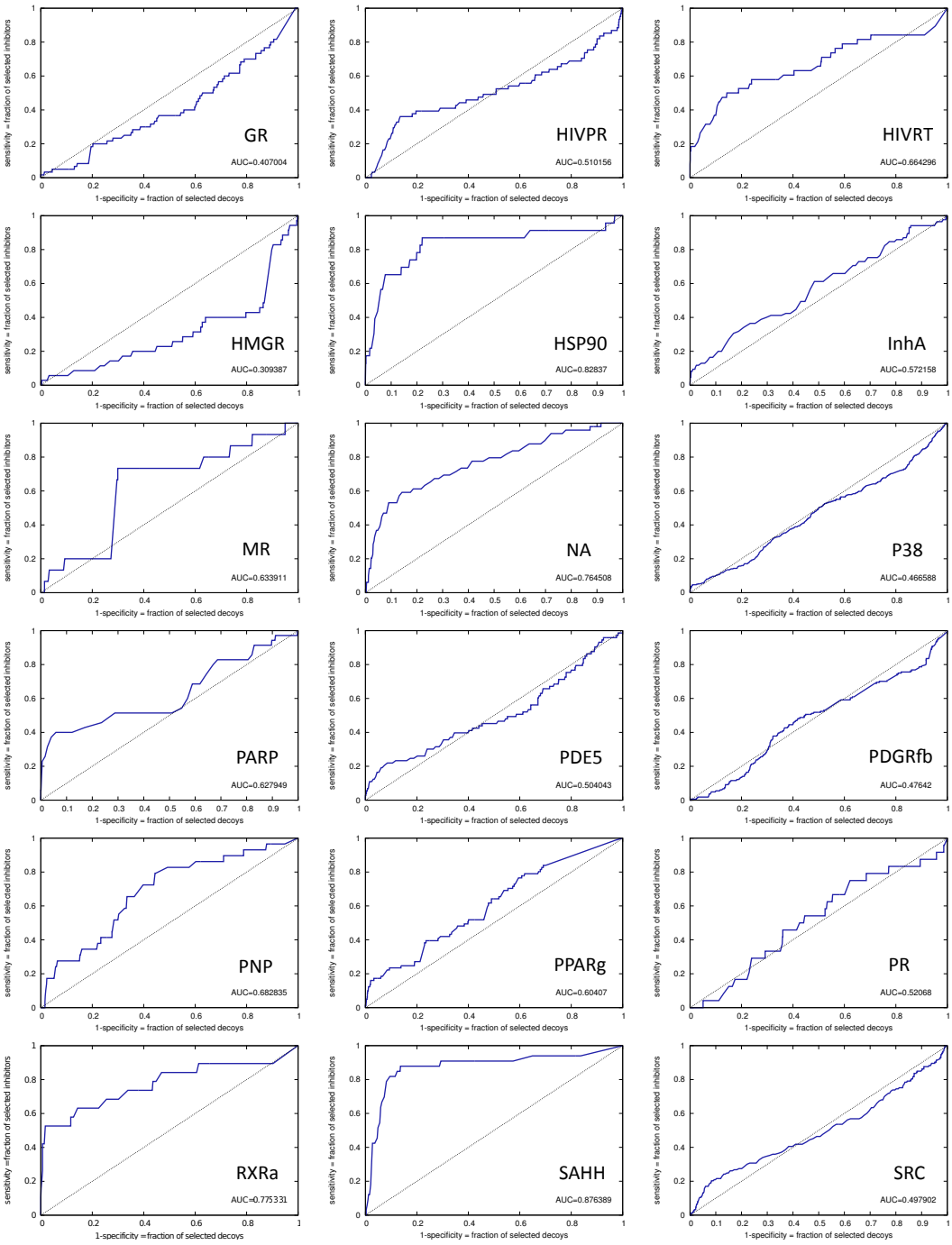


Figure A1: Part 2 of 3.

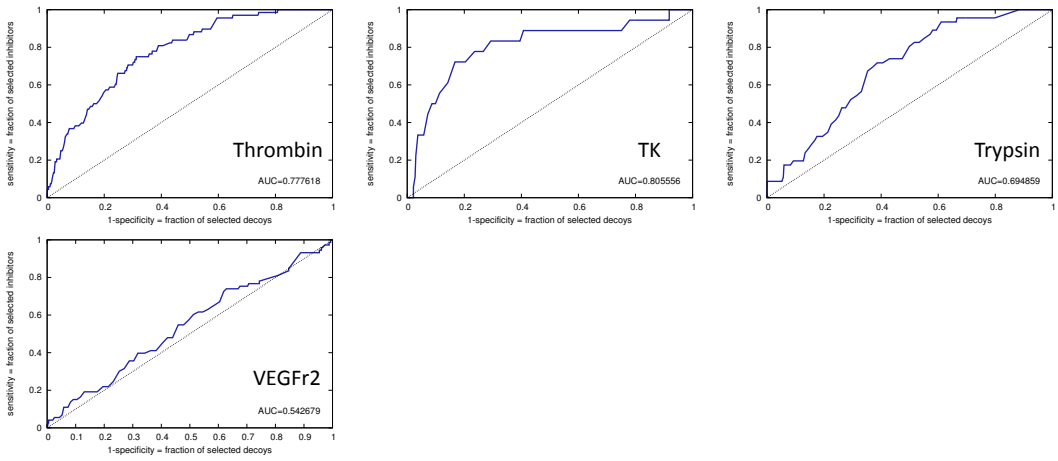


Figure A1: Part 3 of 3.

term	coefficients K
vdW	0.1
electrostatics	0.01
desolvation	19.82
hydrogen bonds	3.0
nRot	1.0

Table A2: Coefficients of our scoring function as show in Equation 5.1
.

Bibliography

[1] A. Luch. *Molecular, clinical and environmental toxicology*. Springer, Heidelberg, 2009.

[2] K. Schrör. *Acetylsalicylic Acid*. Wiley-Blackwell, Weinheim, 2008.

[3] S.M. Paul, D.S. Mytelka, C.T. Dunwiddie, C.C. Persinger, B.H. Munos, S.R. Lindborg, and A. L. Schacht. How to improve R&D productivity: the pharmaceutical industry's grand challenge. *Nat Rev Drug Discov*, 9(3):203–214, 2010.

[4] Th. Förster. Zwischenmolekulare Energiewanderung und Fluoreszenz. *Annalen der Physik*, 437(1-2):55–75, 1948.

[5] L. Stryer, J.M. Berg, and J.L. Tymoczko. *Biochemistry*. W. H. Freeman, New York, 2002.

[6] `http://en.wikipedia.org/wiki/File:Biacore_diagram.jpg`. accessed: 10.08.2011.

[7] M.F. Sugrue. Pharmacological and ocular hypotensive properties of topical carbonic anhydrase inhibitors. *Progress in Retinal and Eye Research*, 19(1):87 – 112, 2000.

[8] D. Monroe. Looking for Chinks in the Armor of Bacterial Biofilms. *PLoS Biol*, 5(11):e307, 11 2007.

[9] M. Otto. Staphylococcal Biofilms. *Curr Top Microbiol Immunol.*, (322):207–28, 2008.

[10] Ch. Gerke, A. Kraft, R. Süssmuth, O. Schweitzer, and F. Götz. Characterization of the N-Acetylglucosaminyltransferase Activity Involved in the Biosynthesis of the StaphylococcusepidermidisPolysaccharide Intercellular Adhesin. *Journal of Biological Chemistry*, 273(29):18586–18593, 1998.

[11] W. Ziebuhr, C. Heilmann, F. Götz, P. Meyer, K. Wilms, E. Straube, and J. Hacker. Detection of the intercellular adhesion gene cluster (ica) and phase variation in Staphylococcus epidermidis blood culture strains and mucosal isolates. *Infection and Immunity*, 65(3):890–6, 1997.

[12] S. E. Cramton, Ch. Gerke, N. F. Schnell, W. W. Nichols, and F. Götz. The Intercellular Adhesion (ica) Locus Is Present in Staphylococcus aureus and Is Required for Biofilm Formation. *Infection and Immunity*, 67(10):5427–5433, 1999.

[13] S.J. Charnock and G.J. Davies. Structure of the Nucleotide-Diphospho-Sugar Transferase, SpsA from Bacillus subtilis, in Native and Nucleotide-Complexed Forms. *Biochemistry*, 38(20):6380–6385, 1999.

[14] C. Kittel and H. Kroemer. *Thermal Physics*. W. H. Freeman, New York, 1980.

[15] I. Muegge and Y.C. Martin. A General and Fast Scoring Function for Protein-Ligand Interactions: A Simplified Potential Approach. *J Med Chem*, 42(5):791–804, 1999.

[16] H. Gohlke, M. Hendlich, and G. Klebe. Knowledge-based scoring-function to predict protein-ligand interactions. *J Mol Biol*, 295(2):337–356, 2000.

[17] H.J. Böhm. The development of a simple empirical scoring function to estimate the binding constant of a protein-ligand complex of known three-dimensional structure. *J Comput Aided Mol Des*, 8(3):243–256, 1994.

[18] M.D. Eldridge, Ch.W. Murray, T.R. Auton, G.V. Paolini, and R.P. Mee. Empirical scoring functions: I. The development of a fast empirical scoring function to estimate the binding affinity of ligands in receptor complexes. *J Comput Aided Mol Des*, 11(5):425–445, 1997.

[19] M. Rarey, B. Kramer, T. Lengauer, and G. Klebe. A Fast Flexible Docking Method using an Incremental Construction Algorithm. *J Mol Biol*, 261(3):470–489, 1996.

[20] R.A. Friesner, J.L. Banks, R.B. Murphy, T.A. Halgren, J.J. Klicic, D.T. Mainz, M.P. Repasky, E.H. Knoll, M. Shelley, J.K. Perry, D.E. Shaw, P. Francis, , and P.S. Shenkin. Glide: A New Approach for Rapid, Accurate Docking and Scoring. 1. Method and Assessment of Docking Accuracy. *J Med Chem*, 47(7):1739–1749, 2004.

[21] G.M. Morris, D.S. Goodsell, R.S. Halliday, R. Huey, W.E. Hart, R.K. Belew, and A.J. Olson. Automated Docking Using a Lamarckian Genetic Algorithm and an Empirical Binding Free Energy Function. *J Comput Chem*, 19(14):1639–1662, 1998.

[22] Y. Pan, N. Huang, S. Cho, and A.D. Jr. MacKerell. Consideration of Molecular Weight during Compound Selection in Virtual Target-Based Database Screening. *J Chem Inf Comput Sci*, 43:267–272, 2003.

[23] R. Wang and S. Wang. How does consensus scoring work for virtual library screening? An idealized computer experiment. *J Chem Inf Comput Sci*, 41(5):1422–1426, 2001.

[24] A. Oda, K. Tsuchida, T. Takakura, N. Yamaotsu, and S. Hirono. Comparison of consensus scoring strategies for evaluating computational models of protein-ligand complexes. *J Chem Inf Model*, 46(1):380–91, 2006.

[25] N.M. O'Boyle, J.W. Liebeschuetz, and J.C. Cole. Testing assumptions and hypotheses for rescoring success in protein- ligand docking. *J Chem Inf Model*, 49(8):1871–1878, 2009.

[26] P.A. Kollman, I. Massova, C. Reyes, B. Kuhn, S. Huo, L. Chong, M. Lee, T. Lee, Y. Duan, W. Wang, O. Donini, P. Cieplak, J. Srinivasan, D.A. Case, and Th.E. Cheatham. Calculating Structures and Free Energies of Complex Molecules: Combining Molecular Mechanics and Continuum Models. *Acc Chem Res*, 33(12):889–897, 2000.

[27] B. Kuhn and P.A. Kollman. Binding of a Diverse Set of Ligands to Avidin and Streptavidin: An Accurate Quantitative Prediction of Their Relative Affinities by a Combination of Molecular Mechanics and Continuum Solvent Models. *J Med Chem*, 43(20):3786–3791, 2000.

[28] M.D. Mackey and J.L. Melville. Better than Random? The Chemotype Enrichment Problem. *J. Chem. Inf. Model*, 49:1154–1162, 2009.

[29] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer, New York, 2001.

[30] L. Eriksson, J. Jaworska, A.P. Worth, M.T.D. Cronin, and R.M. Mcdowell. Methods for Reliability and Uncertainty Assessment and for Applicability Evaluations of Classification-and Regression-Based QSARs. *Environmental Health Perspectives*, 111(10):1361–1376, 2003.

[31] A. Tropsha, P. Gramatica, and V.K. Gombar. The Importance of Being Earnest: Validation is the Absolute Essential for Successful Application and Interpretation of QSPR Models. *QSAR & Combinatorial Science*, 22(1):69–77, 2003.

[32] R.D. Clark, D.G. Sprous, and J.M. Leonard. Validating Models Based on Large Dataset. *Rational Approaches to Drug Design. Proceedings of the 13th European Symposium on Quantitative Structure-Activity Relationships*, pages 475–485, 2001.

[33] R.A. Fisher. The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, 7:179–188, 1936.

[34] A.N. Tychonoff and V.Y. Arsenin. *Solution of Ill-posed Problems*. Winston & Sons, Washington, DC, 1977.

[35] I.T. Jolliffe. *Principal Component Analysis*. Springer, New York, 2002.

[36] S. Wold, M. Sjostrom, and L. Eriksson. PLS-regression: a basic tool of chemometrics. *Chemometr Intell Lab*, 58(2):109–130, 2001.

[37] J. Trygg and S. Wold. Orthogonal projections to latent structures (O-PLS). *Journal of Chemometrics*, 16(3):119–128, 2002.

[38] S. Zhang, A. Golbraikh, S. Oloff, H. Kohn, and A. Tropsha. A Novel Automated Lazy Learning QSAR (ALL-QSAR) Approach: Method Development, Applications, and Virtual Screening of Chemical Databases Using Validated ALL-QSAR Models. *J Chem Inf Model*, 46(5):1984–1995, 2006.

[39] A.J. Miller. *Subset Selection in Regression*. Chapman and Hall, Boca Raton, 1990.

[40] R.L. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach*. Prentice-Hall, Upper Saddle River, 1982.

[41] J.B. Moon and W.J. Howe. Computer design of bioactive molecules: A method for receptor-based de novo ligand design. *Proteins: Struct. Funct. Genet.*, 11:314–328, 1991.

[42] Accelrys Discovery Studio QSAR, `http://accelrys.com/products/discovery-studio/qsar.html`. accessed: 16.07.2011.

[43] Tripos QSAR, `http://tripos.com/index.php?family=modules,SimplePage,,,&page=QSAR_CoMFA`. accessed: 15.07.2011.

[44] DRAGON for Linux (software for molecular descriptor calculations). Version 1.4, `http://www.talete.mi.it/`. accessed: 15.07.2011.

[45] Schroedinger Molecular Modeling Platform, `http://www.schroedinger.com`. accessed: 15.07.2011.

[46] J.J. Dongarra, J. Du Croz, I.S. Duff, and S. Hammarling. A set of Level 3 Basic Linear Algebra Subprograms. *ACM Trans. Math. Soft.*, 16:1–17, 1990.

[47] E. Anderson and Z. Bai. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, 1999.

[48] R.C. Whaley, A. Petitet, and J.J. Dongarra. Automated Empirical Optimization of Software and the ATLAS Project. *Parallel Computing*, 27(1–2):3–35, 2001.

[49] J.J. Sutherland, L.A. O'Brien, and D.F. Weaver. A comparison of methods for modeling quantitative structure-activity relationships. *J. Med. Chem*, 47(22):5541–5554, 2004.

[50] T. Liu, Y. Lin, X. Wen, R.N. Jorissen, and M.K. Gilson. BindingDB: a web-accessible database of experimentally determined protein-ligand binding affinities. *Nucleic Acids Research*, 35(Database issue):D198, 2007.

[51] N. Huang, B.K. Shoichet, and J.I. Irwin. Benchmarking sets for molecular docking. *J Med Chem*, 49(23):6789–6801, November 2006.

[52] J. Goecks, A. Nekrutenko, and J. Taylor. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biol*, 11(8):R86, 2010.

[53] W.D. Cornell, P. Cieplak, Ch.I. Bayly, I.R. Gould, K.M. Merz, D.M. Ferguson, D.C. Spellmeyer, Th. Fox, J.W. Caldwell, and P.A. Kollman. A Second Generation Force Field for the Simulation of Proteins, Nucleic Acids, and Organic Molecules. *J Am Chem Soc*, 117(19):5179–5197, 1995.

[54] G.M. Morris. Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function. *J Comput Chem*, 19(14):1639–1662, 1998.

[55] A. Kerzmann, D. Neumann, and O. Kohlbacher. SLICK–scoring and energy functions for protein-carbohydrate interactions. *J Chem Inf Model*, 46(4):1635–42, 2006.

[56] S. Raub, A. Steffen, A. Kämper, and Ch.M. Marian. AIScore chemically diverse empirical scoring function employing quantum chemical binding energies of hydrogen-bonded complexes. *J Chem Inf Model*, 48(7):1492–510, July 2008.

[57] R. Wang, X. Fang, Y. Lu, and S. Wang. The PDBbind database: collection of binding affinities for protein-ligand complexes with known three-dimensional structures. *J Med Chem*, 47(12):2977–80, June 2004.

[58] R. Wang, Y. Lu, X. Fang, and S. Wang. An extensive test of 14 scoring functions using the PDBbind refined set of 800 protein-ligand complexes. *J Chem Inf Comput Sci*, 44(6):2114–25, 2004.

[59] J.B. Cross, D.C. Thompson, B.K. Rai, J.C. Baber, K.Y. Fan, Y. Hu, and C. Humblet. Comparison of several molecular docking programs: pose prediction and virtual screening accuracy. *J Chem Inf Model*, 49(6):1455–1474, 2009.

[60] T. Liu, Y. Lin, X. Wen, R.N. Jorissen, and M.K. Gilson. BindingDB: a web-accessible database of experimentally determined protein-ligand binding affinities. *Nucleic Acids Research*, 35(suppl 1):D198–D201, 2007.

[61] J.J. Irwin and B.K. Shoichet. ZINC - A Free Database of Commercially Available Compounds for Virtual Screening. *J Chem Inf Model*, 45(1):177–182, 2005.

[62] J.J. Sutherland, L.A. O'Brien, and D.F. Weaver. A Comparison of Methods for Modeling Quantitative Structure-Activity Relationships. *J Med Chem*, 47(22):5541–5554, 2004.

[63] R.D. Cramer, D.E. Patterson, and J.D. Bunce. Comparative molecular field analysis (CoMFA). 1. Effect of shape on binding of steroids to carrier proteins. *J Am Chem Soc*, 110(18):5959–5967, 1988.

[64] G. Klebe and U. Abraham. Comparative molecular similarity index analysis (CoMSIA) to study hydrogen-bonding properties and to score combinatorial libraries. *J Comput Aided Mol Des*, 13(1):1–10, Jan 1999.

[65] Accelrys Pipeline Pilot, `http://accelrys.com/products/pipeline-pilot`. accessed: 15.08.2011.

[66] Schrödinger KNIME Extensions, `http://www.schrodinger.com/products/14/8`. accessed: 15.08.2011.

[67] D. Blankenberg, G. Von Kuster, N. Coraor, G. Ananda, R. Lazarus, M. Mangan, A. Nekrutenko, and J. Taylor. Galaxy: a web-based genome analysis tool for experimentalists. *Curr Protoc Mol Biol*, Chapter 19:Unit 19.10.1–Unit 19.1021, Jan 2010.

[68] SMILES arbitrary target specification, `http://www.daylight.com/dayhtml/doc/theory/theory.smarts.html`. accessed: 20.07.2011.

[69] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, and P.E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28(1):235–242, 2000.

[70] S. Grüneberg, M.T. Stubbs, and G. Klebe. Successful Virtual Screening for Novel Inhibitors of Human Carbonic Anhydrase: Strategy and Experimental Confirmation. *J Med Chem*, 45(17):3588–3602, 2002.

[71] M.A. Larkin, G. Blackshields, N.P. Brown, R. Chenna, P.A. McGettigan, H. McWilliam, F. Valentin, I.M. Wallace, A. Wilm, R. Lopez, J.D. Thompson, T.J. Gibson, and D.G. Higgins. ClustalW and ClustalX version 2. *Bioinformatics*, 23(21):2947–2948, 2007.

[72] M.A. Marti-Renom, A. Stuart, A. Fiser, R. Sánchez, F. Melo, and A. Sali. Comparative Protein Structure Modeling With MODELLER. *Annu. Rev. Biophys. Biomol. Struct.*, 29:291–325, 2000.

[73] E. Bolton, Y. Wang, P.A. Thiessen, and S.H. Bryant. *Annual Reports in Computational Chemistry*, volume 4. Washington, DC, American Chemical Society, 2008.

[74] Susan L. McGovern, Emilia Caselli, Nikolaus Grigorieff, and Brian K. Shoichet. A common mechanism underlying promiscuous inhibitors from virtual and high-throughput screening. *Journal of Medicinal Chemistry*, 45(8):1712–1722, 2002.

[75] Susan L. McGovern, Brian T. Helfand, Brian Feng, and Brian K. Shoichet. A specific mechanism of nonspecific inhibition. *Journal of Medicinal Chemistry*, 46(20):4265–4272, 2003.

[76] James Seidler, Susan L. McGovern, Thompson N. Doman, and Brian K. Shoichet. Identification and prediction of promiscuous aggregating inhibitors among known drugs. *Journal of Medicinal Chemistry*, 46(21):4477–4486, 2003.

[77] S.M. Vogel, M.R. Bauer, and F.M. Böckler. DEKOIS: Demanding Evaluation Kits for Objective in Silico Screening - A Versatile Tool for Benchmarking Docking Programs and Scoring Functions. *J. Chem. Inf. Model*, 51(10):2650–2665, 2011.

[78] A. Hildebrandt, A.K. Dehof, A. Rurainski, A. Bertsch, M. Schumann, N.C. Toussaint, A. Moll, D. Stöckel, S. Nickels, S.C. Mueller, H.P. Lenhof, and O. Kohlbacher. BALL–biochemical algorithms library 1.3. *BMC Bioinformatics*, 11:531, 2010.