Technische Universität München
Fakultät für Physik

Abschlussarbeit im Masterstudiengang Biophysik

On the Development of Problem-Oriented Knowledge-Based Scoring Functions for Protein-Protein Docking

Alexander Sasse

Monday 8th June, 2015

Erstgutachter (Themensteller): Prof. M. Zacharias
Zweitgutachter: Prof. F. Simmel

Danksagung

An dieser Stelle möchte ich Prof. Dr. Martin Zacharias dafür danken, dass er mir die Möglichkeit gegeben hat an dem Thema "Protein Docking" zu arbeiten. Seine freundliche Art und Weise, mit welcher er einem stets beratend zur Seite steht, hat es mir ermöglicht meinen eigenen Weg in dieser Arbeit erfolgreich zu beschreiten.

Besonderer Dank gilt Dr. Sjoerd de Vries für die viele Zeit, die er sich genommen hat, um mir bei meinem Projekt zu helfen und meine vielen Fragen zu beantworten. Auch danke ich ihm für die vielen inspirierenden Diskussionen, mit denen er mich stets auf neue Ideen gebracht hat. Seine positive Einstellung hat mich motiviert meinen eigenen Interessen zu folgen und mir dabei geholfen Fehlschläge zu verarbeiten.

Christina Schindler und Dr. Isaure Chauvot de Beauchêne danke ich für die geduldige Unterstützung bei allen meinen Fragen und Problemen.

Den Leuten in der Arbeitsgruppe möchte ich für die tolle Zeit danken, in der ich hier arbeiten durfte. Besonders die täglichen Lunches und die Winterschool werde ich vermissen.

Meinen Freunden und meinem Bruder zu Hause möchte ich danken, dass sie so viel Verständnis dafür aufgebracht haben, dass wir uns so selten sehen. Trotz alledem ist es jedes mal wieder so, als wäre ich niemals umgezogen.
Meinen Freunden in München danke ich für die schöne gemeinsame Zeit in dieser wunderbaren Stadt.

Der größte Dank gebührt meinen Eltern, die mir dieses Studium erst ermöglicht haben, indem sie alle meine Interessen und Ideen unterstützen. Nur durch die Freiheit, die sie mir durch ihre Unterstützung bei allen meinen Entscheidungen im Leben geben, konnte ich dieses Studium erfolgreich abschließen.

Contents

Chapter 1

Summary

Protein-protein interactions (PPI) play a key role in all biological processes inside cells. They are essential for signal-transduction and transport processes, as well as for the immune response [3, 19]. The knowledge about the 3D structure and the interface contacts of the formed complex give insights into kinetics, thermodynamics and organization as well as human diseases [47]. Furthermore, a highly resolved interface of the 3D structure is necessary for drug and protein design [45, 25].
Although more and more new PPI's are discovered by experimentalists, the number of known structures for these protein complexes lags far behind [60]. The experimental exploration of the complex suffers from its size and especially from its stability. However, experimental structure prediction methods such as nuclear magnetic resonance (NMR) spectroscopy, X-ray crystallography and electron microscopy (EM) have been used successfully to determine many of the unbound protein structures.

Protein docking protocols aim to predict the structure of the protein complex out of its unbound constituents. Although most of the state-of-the-art docking programs achieve quite satisfactory results, particularly for cases with no or little structural changes during the docking procedure, predicting the native complex structure and especially its native contacts remains an unsolved problem[30].
The docking protocols can be divided into two stages: first, sampling possible complex solutions and second, scoring them on their 3D structure to predict the native complex. In the sampling step, many possible solutions for the complex structure (also referred to as decoys) are generated out of the unbound forms of its constituents, for which the protein with the longer chain is referred to as receptor and the shorter to as ligand. The scoring step has to determine near-native solutions and separate them by its score from the incorrect structures.
Sampling the conformational space suffers mostly from the rigid-body assumption of the two proteins. Although the introduction of flexibility by normal modes, side-chain flexibility or ensembles of protein structures can improve sampling with the unbound structures, the creation of the native structure from that procedure stays a challenge for most complexes.

Scoring functions lean their predictions on characteristics of the interface between the receptor and the ligand. Thus, well matching interfaces seem to be necessary in order to make successful predictions for the decoys. Scoring functions suffer from the diversity of interfaces between the protein complexes and the simplified models which are used for scoring. Furthermore, the scoring of complex structures which are generated by sampling with the unbound forms, is highly affected by the insufficiently aligned interfaces. Due to these problems, scoring remains the bottleneck for most of the docking protocols. However, it becomes visible that the scoring problem cannot be tackled separately from the sampling protocol since it strongly depends on the sampling algorithm.

This work presents a methodology for the creation of knowledge-based scoring functions for protein docking. Knowledge-based scoring functions are optimized for a particular decoy set and hence are oriented on the quality of their interfaces which is dependent on the sampling algorithm. As results, several promising scoring functions for decoys from unbound rigid-body docking by ATTRACT are presented in this work. The methodology can be easily adapted to other scoring problems in structural biology, for example for the scoring of protein-peptide, protein-DNA or protein-RNA complexes.
This work is divided into a chapter 'Introduction' which defines the scoring problem, points out the general approach for the creation of scoring functions and gives an overview on all the components of protein docking. In addition, the methodology which is used in this work for the creation of several scoring functions is illustrated explicitly. In the following, the scoring performance of the generated knowledge-based scoring functions by the methodology is evaluated in the chapter 'Results' and a conclusion is drawn in chapter 'Conclusion and Outlook'.

In the chapter 'Introduction', the challenges of scoring functions for protein docking are illustrated. Due to the insufficient sampling algorithms and the subsequent necessity of a refinement step for a sufficient docking, the challenge for scoring functions can be divided into a big and small scoring problem. The probability to predict the native structure at a certain rank in the decoy sets is referred to as the big scoring problem. The small scoring problem refers to the scoring performance on near-native structures. Depending on the usage of a refinement step in the docking protocol, the small scoring problem can be further divided. To make a prediction from unbound docking, the probability to predict a near-native structure after rigid-body sampling is regarded (small scoring problem I). The fraction of near-native structures illustrates the concentration of near-natives in a subset of decoys for refinement and is referred to as the small scoring problem II.
In addition, a rough overview is given on common sampling and refinement protocols and the composition and the functioning of the docking protocol for ATTRACT are

being illuminated in more detailed way. Therefore, the coarse grained representation of the proteins which is used in ATTRACT for the intermolecular potential, is described in the section 'Structure representation'. Additionally, two atomistic representations for interactions between the proteins are presented, namely the grouped-all-atom (GAA) representation and the representation from Tobi et al. [61]. Moreover, possible potential forms for scoring functions are described and several approaches for the parametrization of scoring functions are presented roughly. Finally, the methodology for the creation of knowledge-based scoring functions from energy models is explained in detail. Knowledge-based scoring functions are always based on the choice of the protein representation and the energy model which in combination can be interpreted as the choice of structural features. Moreover, the character of scoring functions is based on the choice of the decoy set and the optimization algorithm for the estimation of their parameters.
Therefore, the 'ATTRACT benchmark' is generated, which consists of 164 decoy sets for different protein complexes for training and assessment of the functions. As optimization methods for the parameters of the scoring functions, the Monte Carlo protocol and the linear regression protocol are introduced. For the usage of these protocols, the computation of feature vectors for the different energy models is depicted.

In the chapter 'Results', knowledge-based scoring functions are presented which use for their scoring intermolecular step potentials, Lennard-Jones-like (LJ) potentials and potentials based on the atomistic buried surface area (BSA-potentials). The parameters of the scoring functions are optimized for the complexes in the 'ATTRACT benchmark' on decoys from unbound rigid-body docking by ATTRACT. For the determination of the parameters, linear regression and Monte Carlo Annealing are used. Additionally, linear combinations of different scores from these models are generated by linear regression and their capability to improve the total scoring performance is evaluated.
For the assessment of the created scoring functions, the probability to predict a near-native structure from unbound rigid-body docking with ATTRACT is determined. From that, the efficiency to predict at least a near-native structure without the usage of flexibilities or a refinement step can be seen. Furthermore, the small scoring problem II is evaluated on the fraction of near-native solutions in different subsets to regard the potential of a successful refinement step. In addition, the probability to predict the native structure, if it were generated by the sampling algorithm, is evaluated as well to estimate the success rate for the big scoring problem.

The various created knowledge-based scoring functions from different energy models and their combinations are evaluated in comparison to the established scoring functions from ATTRACT and Tobi et al. [69, 23, 61]. It is shown that all of them were able to cope with the scoring performance of ATTRACT and Tobi for

both assessments of near-native structures. Especially step potentials outperform all other types of potentials on the scoring of decoys from unbound rigid-body docking. Potentials which are based on the atomistic buried surface areas show a sufficient ability to improve the fraction of near-native structures in subsets for a possible refinement. Moreover, created Lennard-Jones-like potentials showed improvements for prediction of near-native structure compared to ATTRACT but also for the average fraction of near-natives. Hence, these potentials might be useful for sampling as well.
From the evaluation of the most contributing parameters to the scores of the step potentials and the atomistic BSA-potentials, insights into the interface composition of protein complexes and the character of the scoring functions can be gained. The results of this evaluation show that interfaces of protein complexes consist of hydrophobic residues, especially aromates. Scoring functions favour these hydrophobic contacts on the interface and also the presence of arginines. Scoring functions which are able to predict a variety of near-native structures for a complex from unbound rigid-body docking, namely the step potentials and the BSA-potentials, incorporate at least the knowledge about the general interface composition for their scoring.
For these reasons, this work does not only present a methodology to generate knowledge-based scoring functions, but does also illustrate possibilities for the future improvement of the protein docking protocol of ATTRACT by the approaches which are evaluated in this work.

Chapter 2

Introduction

2.1 Defining the Scoring Problem

2.1.1 The Physical Viewpoint

Scoring by the composition of energy terms is derived from the physical understanding of protein interactions in which the native complex structure is formed in the global minimum of the energy landscape. Therefore, it can be assumed that the native structure possesses a lower free energy than any incorrect decoy that can be generated by sampling methods [63].

In cells the two proteins are primarily surrounded by water molecules. The driving force for complex formation under real conditions corresponds to the change in free energy associated with the binding reaction [33]. The change in free energy involves electrostatic interactions, van der Waals interactions and hydrogen bonding interactions. All these interactions include intra- and intermolecular contributions, interactions between the protein and the surrounding solvent and also interactions between the solvent molecules themselves. Therefore, the binding energy involves not only enthalpic but also entropic contributions resulting from the interactions with the solvent and other surrounding molecules. The total binding free energy for a protein complex in solvent cannot be determined directly from the intermolecular interactions between the atoms of the receptor and the ligand. However, it can be separated into its energy constituents by the regard to an energy cycle shown in figure 2.1.

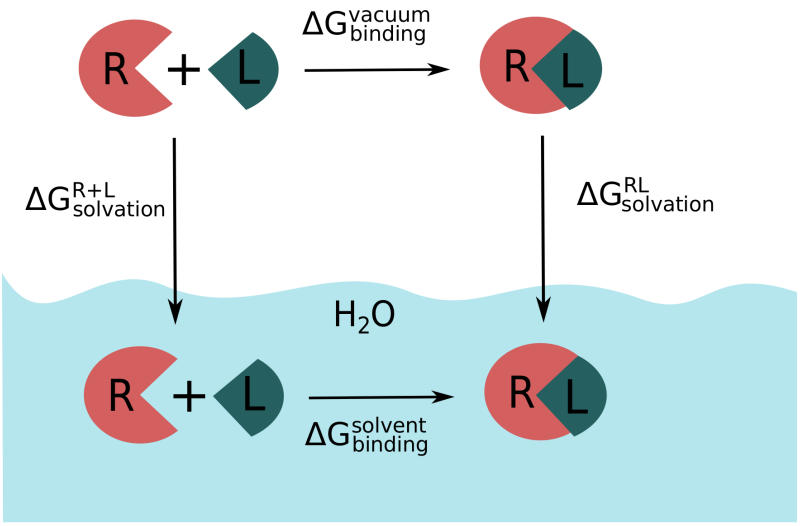


Figure 2.1: Binding free energy cycle.

The total binding free energy in solvent $\Delta G_{binding}^{solvent}$ consists of the difference between solvation energies of the complex and its separated constituents $\Delta\Delta G_{solvation}$, the binding energy in vacuum $\Delta G_{binding}^{vacuum}$, the internal energy $\Delta G_{Adaption}$ for each protein due to conformational changes and the conformational entropy $T\Delta S_{Conf}$ which is shown in equation 2.1. The term for the change in conformational entropy $T\Delta S_{Conf}$ represents all contributions due to changes in the mobility of the partners and solvent molecules during complex formation.

$$\Delta G_{binding}^{solvent} = \Delta G_{binding}^{vacuum} + \Delta\Delta G_{Solvation} + \Delta G_{Adaption} + T \cdot \Delta S_{Conf} \tag{2.1}$$

The solvation energy $\Delta\Delta G_{solvation}$ in equation 2.2 contains a term for the solvation of uncharged molecules $\Delta\Delta G_{surface}$ and an electrostatic contribution $\Delta\Delta G_{PB}$ which refers to the required energy to add charges into the solvated molecules.

$$\begin{aligned} \Delta\Delta G_{solvation} &= \Delta\Delta G_{surface} + \Delta\Delta G_{PB} \\ &= \Delta G_{surface}^{RL} - \Delta G_{surface}^{R+L} + \Delta G_{PB}^{RL} - \Delta G_{PB}^{R+L} \end{aligned} \tag{2.2}$$

The electrostatic contribution $\Delta\Delta G_{PB}$ can be calculated by solving the Poisson-Boltzmann (PB) equation for the complex and its constituents separately or approximating it by the generalized Born (GB) model [59, 27]. The energy for the solvation of uncharged molecules $\Delta G_{surface}$ is proportional to the solvent accessible surface

area SA_i and can be calculated using experimentally determined parameters σ_i for each atom type (equation 2.9).
The binding free energy in vacuum $\Delta G_{binding}^{vacuum}$ consists of the Coulomb energy $\Delta G_{Coulomb}$ and the van der Waals energy ΔG_{vdw} (equation 2.3). Van der Waals interactions between atoms of the receptor and the ligand in vacuum are usually represented by a distance dependent Lennard-Jones potential.

$$\Delta G_{binding}^{vacuum} = \Delta G_{vdw} + \Delta G_{coulomb} \tag{2.3}$$

Molecular Dynamics (MD) simulations attempt to determine the binding free energy for biological processes from long computational demanding simulations using explicit solvent molecules and detailed atomistic force fields between every atom-pair. However, these simulations fail due to the infeasible computational demand for huge systems and for processes with long time scales as they both occur for protein-protein interactions.
The energy models which are used for scoring in protein docking are based exclusively on intermolecular energy contributions between the proteins to obtain sufficient results in a moderate amount of computational time. Intramolecular contributions are neglected completely in rigid-body docking or mostly when using harmonic modes, as well as entropic contributions from interactions with the surrounding solvent. However, these terms might have a large influence on the formation of some protein complexes. Especially, solvation effects play a major role for many complexes and are often neglected. For that reason, a complete energy function may be necessary for a sufficient scoring by composite scoring functions. Simplified energy models might not be satisfactory in scoring for every protein complex due to differences between the influence of energy contributions which is a result from the diversity of their interfaces.

2.1.2 The Docking Viewpoint

The most problematic simplification in protein docking consists of the rigid-body assumption which leads to various steric problems during the sampling with the unbound forms of the receptor and the ligand. Although the usage of harmonic modes, ensembles or side-chain flexibility during sampling, as well as refinement afterwards can improve the quality of the generated structures, the native complex structure cannot be generated from docking with its unbound constituents due to conformational changes.
The Critical Assessment of Predicted Interactions (CAPRI) is a blind prediction experiment for protein-protein complexes organized by leading researchers in that area. CAPRI offers protein docking groups the possibility to test their docking protocols on proper challenges and hence contributes to the improvement of docking protocols.

Table 2.1: Criteria for the Critical Assessment of Predicted Interactions (CAPRI-criteria) serve for the qualitative evaluation of generated complex structures [37]. Based on a) the fraction of native contacts (fnat), b) the interface root mean square deviation (irmsd) and c) the ligand root mean square deviation (lrmsd) the predicted complex structures are divided into incorrect, acceptable, medium and high quality solutions.
a) The fraction of native contacts of a structure is determined by the comparison between the obtained contacts and the contacts of the native complex.
b) The interface root mean square deviation represents the deviation of backbone atoms on the interface between the native structure and the predicted structure by docking.
c) The ligand root mean square deviation represents the deviation of ligand backbone atoms between the its coordinates in the native structure and the predicted structure by docking.

Ranking	Stars	Conditions based on Capri computed conditions
High	***	fnat ≥ 0.5 **AND** [Lrmsd ≤ 1.0 **OR** Irmsd ≤ 1.0]
Medium	**	[$0.3 \leq$ fnat ≤ 0.5] **AND** [Lrmsd ≤ 5.0 **OR** Irmsd ≤ 2.0] **OR** fnat ≥ 0.5 **AND** [Lrmsd ≥ 1.0 **AND** Irmsd ≥ 1.0]
Acceptable	*	[$0.1 \leq$ fnat ≤ 0.3] **AND** [Lrmsd ≤ 10.0 **OR** Irmsd ≤ 4.0] **OR** fnat ≥ 0.3 **AND** [Lrmsd ≥ 5.0 **AND** Irmsd ≥ 2.0]
Incorrect		fnat ≤ 0.1 **OR** [Lrmsd ≥ 10.0 **AND** Irmsd ≥ 1.0]

CAPRI evaluates generated complex structures from docking qualitatively by dividing them into four categories: incorrect, acceptable, medium quality and high quality based on the interface root mean square deviation (Irmsd), the ligand root mean square deviation (Lrmsd) and the fraction of native contacts (fnat) seen in table 2.1. As can be seen in table 2.1, the categories are also referred to as zero-star, one-star (*), two-star (**) and three-star (***) solutions. Structures which are at least an acceptable solution will be referred to as 'near-native' in the following.
After rigid-body sampling by ATTRACT the majority of these near-native structures in the decoy set have only a fraction of native contacts between 0.1 and 0.3.

Only for 11 % of the complexes in the protein-docking benchmark it was possible to obtain a high quality solution with a fraction of native contacts larger than 0.5 (see appendix A "The ATTRACT Benchmark").

Due to the fact that conformational adjustments on the interface arise for almost every protein complex, steric barriers prevent the proteins from building up their native contacts. Thus, for most of the complexes the native structure cannot be obtained from unbound docking and the interface of these near-native solutions deviates a lot from the native structure. Figure 2.2 shows that even small changes of backbone atoms of the ligand between the bound and the unbound form cause clashes in the complex when placing the unbound form exactly at the position of the bound form. It can be seen that clashes with the unbound form result not only from side-chain displacements but also from changes in the backbone.
Therefore, even well performing scoring functions may assign a better score to a lot of incorrect decoys from the contacts between their constituents than to near-native structures. In an elaborated test of 115 scoring functions on a set of 500 decoys from 118 structures of the protein benchmark 4.0 [32], the best performing scoring functions rank a near-native complex structure for 27 % of the complexes on top 1, for 58 % in the top 2 % of all generated decoys and for 93 % in the top 20 % [47]. From these results, the success rate to predict a near-native structure from the scoring of just 500 unbound docking solutions seems to be insufficient.

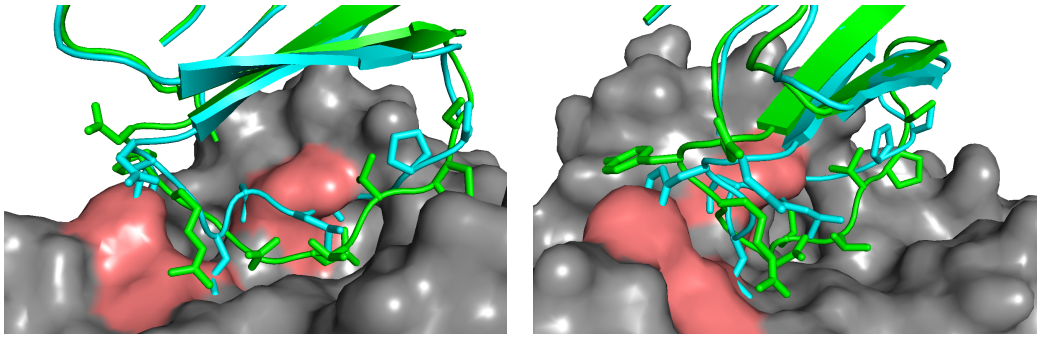


Figure 2.2: Interface of complex 1ACB from the bound structure (gray) of the receptor and the superposed unbound (cyan) bound (green) forms of the ligand. The light red parts on surface of the receptor highlight clashes of the ligand chain with the surface of the receptor.

How deviant acceptable, medium and high quality solutions can be from the native structure is illustrated in figure 2.3. All four ligand conformations differ extremely in their lrmsd (left figure) and their irmsd (right figure) resulting in different contacts

and distances between two atoms on the interface. None of the three proposed solutions matches perfectly the interface of the native structure and thus makes it difficult to estimate the quality of these near-native structures only based on the intermolecular contacts.

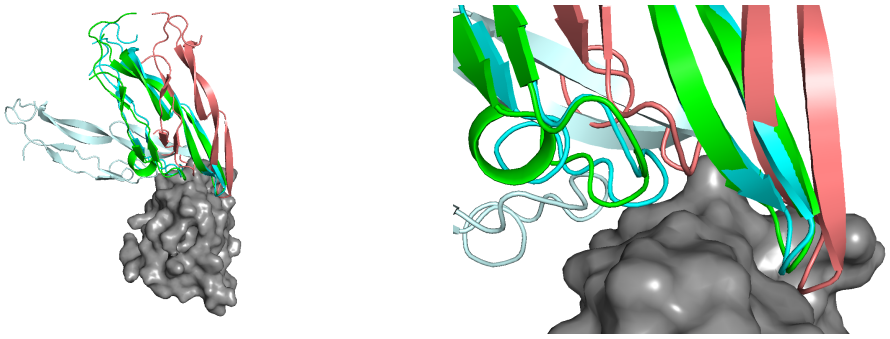


Figure 2.3: Comparison between the native structure (green) and acceptable (white), medium (red) and high quality (blue) docking solutions for complex 1KTZ from unbound docking.

In fact, the binding energy of these structures would deviate strongly from the native structure. The binding funnel for protein complexes is narrow towards the native complex which means that only slightly modified conformations would possess a lower energy than incorrect solutions [63]. Hence, even the use of a complete energy function for scoring might not be successful due to the steric problems in sampling and the resulting insufficient near-native structures.
That scoring functions are highly dependent on the resolution of the interfaces to make successful predictions can be seen from their performance on bound docking solutions. In bound docking, it is very likely that the native structure can be generated. Thus, scoring on these decoys serves for the prediction of the native structure, which is also referred to as the big scoring problem (general scoring problem). However, scoring of the native structure remains an artificial problem due to the fact that it cannot be generated out of the unbound forms of the protein.
Thus, the precondition for the usage of a complete energy function for the scoring of decoys from unbound docking, is the improvement of the sampling algorithm for the generation of well aligned interfaces of the near-native structures. For that reason, at least flexibilities for the side-chains and the backbone during the sampling have to be enabled and a refinement step which enables to account for changes on the interface is essential.
Scoring functions which are supposed to score all near-native structures from un-

bound docking well, have to cope additionally with a large diversity of interfaces between different near-native structures of one complex since sampling algorithms generate up to hundred near-native solutions. Furthermore, the interfaces of the protein complexes itself show also diverse compositions and sizes.
Usually, sampling algorithms generate tens to hundreds of thousands of decoys depending on the size of the protein partners and the sampling algorithm. A refinement cannot be performed for all of the sampled decoys due to its large computational demand. Hence, refinements are usually done for subsets of the best scoring hundreds of decoys. In order to increase the probability of a successful refinement, it is necessary that the subset of decoys is enriched with near-native solutions which can be brought closer towards a native structure. Furthermore, the enriched subset should be chosen as small as possible to safe computational time.
Due to the insufficient resolved structures from unbound docking, two small scoring problems ensue for proper docking: First to create a scoring function which places as many as possible near-native solutions in the subset for refinement (small scoring problem II), and secondly to create a scoring function which ranks a sampled or refined near-native structures on position 1 (small scoring problem I).

A scoring function which solves the big scoring problem is very selective towards the native structures of all protein complexes. From a physical point of view, a complete description of the total free energy represents such function and can be defined as a universal scoring function. Nevertheless, the usage of such an universal scoring function is limited in its practical value due to the fact that sampling algorithms mostly cannot not generate near-native structures from its unbound constituents which are very close to the native complex. A scoring function which deals with the small scoring problem II must be able to deal with the diversity of interfaces between near-native decoys and between complexes. Scoring functions for the small scoring problem I can be stronger selective to score a near-native structure on top, preferentially the best resolved. The better the generation of the decoys from sampling and refinement algorithms performs, this means the closer the generated near-native structures get to the native form, the more transforms the small scoring problem I into the general scoring problem.
This work focuses on approaches for the small scoring problem. Knowledge-based scoring functions are trained on decoys from ATTRACT unbound rigid-body docking. Primarily, the performance of all generated scoring functions is evaluated on the probability to predict at least one near-native structure (small scoring problem I) and the fraction of near-native structures in a subset for refinement (small scoring problem I). Furthermore, the performance for the big scoring problem is regarded by the insertion of the native structures into the decoy sets for the evaluation of its scoring.

2.2 Solving the Scoring Problem

Scoring functions build their score for the decoys on a set of features from their 3D structures like atomistic contacts, surface areas or other geometric descriptors of the surface without any further knowledge of the native complex. Scoring functions can be any function which assigns a value to a protein complex structure based on its coordinates.
For the development of knowledge-based scoring functions, first a representation of the two proteins has to be defined, which is used by the potential form for its calculations (see section 2.3). Coarse grained representations might be useful to safe computational power and to smooth the energy landscape in sampling and scoring. Atomistic representations have the advantage to account better for side-chain orientation and the chemical character on the interface. Second, the form of scoring functions has to be chosen (see section 2.5). Generally, any function that is able to use the protein representation for the computation of a score can be used. Especially, several potential forms of energy terms seem to be a sufficient choice for scoring functions.
Finally, each of the scoring functions determines their parameters from a set of training structures or training values by various optimization methods. For the generation of the training structures, a set of protein complexes and an algorithm for the generation of decoys has to be defined (see section 2.7.2). The methods for the parametrization vary from simple genetic algorithms, over inverse Boltzmann equation on contact densities to complex supervised machine learning approaches like non-linear classifiers or generalized linear models (see section 2.6.1). Hence, the construction of a knowledge-based scoring function is always composed of four steps:

1. Selection of protein representation
2. Selection of scoring potential form
3. Selection of structures for a training set
4. Selection of parametrization method

In the following, the methodology to develop problem-oriented knowledge-based scoring functions from physical energy models will be illustrated. To generate problem-oriented scoring functions which account for characteristics of the sampling method, the decoys in the training set need to be generated by the same algorithm as test cases afterwards.

2.3 Structure Representation

The three dimensional coordinates of the constituents of protein-protein interactions can be taken from the Protein Data Bank (PDB) (www.rcsb.org)[4]. Protein models from the PDB were usually determined by experimental groups using X-ray crystallography, nuclear magnetic resonance spectroscopy or electron microscopy. Currently, coordinates for over 100.000 protein structures are available. For reliability, the experimental method and the publication for each structure is quoted for every published structure.
For docking, the complex structure and the unbound structures of the receptor and the ligand must be available. Docking protocols use atomistic or coarse grained representations of the 3D structures. Atomistic representations use the coordinates of each atom, whereas coarse grained models try to reduce the three dimensional structure further by representing several atoms as a pseudo atom which is usually located at their center of mass. All-atom representations often use artificially inserted hydrogen atoms since they cannot be resolved in X-ray crystallography but might be important for docking goals.

2.3.1 Grouped-All-Atom Representation

To describe any interaction between atoms by a physical potential in this work the grouped-all-atom representation (GAA) is used. It possesses 27 atom types which are shown in table A.1. The 27 atom types are necessary because the chemical character of any atom is not just dependent on the chemical atom type but rather on its molecular bonding. In the atomistic representation of the proteins, the hydrogen atoms are inserted into the coordinate files for the backbone and the partially charged end groups of the residues arginine, asparagine, cysteine, glutamine, histidine, lysine, serine, threonine, tryptophan and tyrosine.
In this GAA representation, atom types were defined separately for five groups of amino acids which are nonpolar, polar, aromatic, positively charged and negatively charged. Tyrosine and tryptophan possess aromatic C-atoms but also polar atoms for their polar groups. Since, it was shown that the composition of residues on the interface of protein complexes is conserved [64, 3] this representation was chosen to account for the different chemical character of the residues in these groups. A full atomistic definition with partial charges for each amino acid is given in the appendix in table A.2.
The partial charges were taken from the atomistic optimized potential for liquid simulations (OPLS) [34], which derives them from quantum mechanical calculations.

2.3.2 Coarse Grained Representation of Attract

A coarse grained representation of the proteins is often used to smooth the energy landscape and to reduce the computational demand for sampling [69]. Furthermore, a well defined coarse grained representation might also be sufficient for scoring due to these reasons. A coarse grained representation can vary between models which represent a residue by one bead and models which use several beads for the side-chains to account for their orientation and their chemical character.
ATTRACT uses a coarse grained representation for which each amino acid is represented by a pseudo atom at the C_α position and up to two pseudo atoms for its side-chain [69]. Alanine, Serine, Tryptophan, Valine, Leucine, Isoleucine, Asparagine, Aspartate and Proline side-chains are represented by one pseudo-atom located at the center of geometry of the side-chain heavy atoms. For larger amino acids, the first pseudo atom is located between C_β and C_γ and the second is positioned at the center of geometry of the remaining heavy atoms. For a Coulomb interaction between pseudo atoms, full charges are assigned to side-chain ends of charged residues. Furthermore, asparagine and glutamine receive full negative charges respective to their highly polar character.

2.4 Protein Docking

2.4.1 Sampling

The sampling step aims to generate possible solutions for the protein complex from its constituents by an efficient search algorithm. The search algorithms can be divided into systematic search methods and guided search methods.
Systematic search methods include discrete sampling such as the Fast Fourier Transformation method (FFT) [41, 11] or geometric surface matching [49, 14, 18]. Both methods use geometric complementarity of the interface for their predictions. Geometric surface matching algorithms represent the receptor and the ligand as surface descriptors capturing the essential features of the surfaces in terms of concave and convex, their size, their depth, and their relative location on the surface. Finally, the search for pattern matching is performed by geometric hashing. FFT docking discretizes the coordinates of the two proteins on a 3D grid to calculate the correlation for various positions and orientations to each other. Therefore, it uses Fourier transformation to calculate the correlation $c_{\alpha,\beta,\chi}$ between the digitalized three dimensional grid-representation $a_{l,m,n}$ and $b_{l,m,n}$ of the coordinates of the receptor and the ligand atoms.

$$a_{l,m,n} = \begin{cases} 1 & \text{on the surface} \\ \rho & \text{inside the molecule} \\ 0 & \text{outside} \end{cases} \tag{2.4}$$

$$b_{l,m,n} = \begin{cases} 1 & \text{on the surface} \\ \delta & \text{inside the molecule} \\ 0 & \text{outside} \end{cases} \tag{2.5}$$

$$c_{\alpha,\beta,\chi} = \sum_{l=1}^{N} \sum_{m=1}^{N} \sum_{n=1}^{N} a_{l,m,n} \cdot b_{l+\alpha,m+\beta,n+\chi} \tag{2.6}$$

The interior parameters ρ and δ are used to discriminate overlapping areas with ρ being a large negative value and δ a small positive. Well aligned surfaces between the receptor and the ligand gain a high score due to many overlapping surface points of both proteins, whereas structures with clashes receive negative penalties for the penetration into the proteins interior. By calculating the Fourier transformation $C_{o,p,q}$ instead of $c_{\alpha,\beta,\chi}$ directly, computational time can be saved due to the avoidance of the product of $a_{l,m,n} \cdot b_{l+\alpha,m+\beta,n+\chi}$ at each grid point.

Guided search methods can be divided into data-driven methods which use experimental data directly from the complex that needs to be determined, and unconstrained methods which were trained on test complexes for their predictions.
Data-driven techniques use low resolution experimental data from NMR or cryo-electron microscopy of the complex that needs to be predicted to constrain its interface area while using common sampling algorithms to generate their decoys [17, 16, 55]. It is obvious that these methods usually achieve more satisfying results then docking methods which do not constrain the interface on external data.
Unconstrained guided sampling techniques include energy minimization (EM) [69], Monte Carlo simulations (MC) [57, 20] or genetic algorithms (GA) [29]. These methods base their sampling on physical potentials between the atoms or pseudo atoms of the two proteins which guide their way to the final structures. The parameters for these potentials are derived from experimentally determined protein complexes. The potentials are created on the knowledge of these complexes with the goal to assign lower energy to near-native solutions than to any incorrect structure.
For a guided search, the ligand is placed on various positions in different orientations around the receptor and an algorithm for optimization drives its way to an energetic minimum at their surfaces. ATTRACT, for example, uses an quasi-newton

minimizer to find the energetic minimum for the placement of the ligand around the receptor.

Sampling algorithms are supposed to generate their decoys out of the unbound structures of the receptor and the ligand. Due to conformational changes of the two protein partners, most of the docking protocols introduce the possibility to insert flexibilities during the sampling. Common approaches are the docking with ensembles of receptor and ligand conformations, the introduction of normal modes or the implementation of side-chain flexibility [43, 44, 26, 20]. The usage of ensembles generates more complex structures due to the different conformations for the protein partners. Normal modes are calculated and used as additional degrees of freedom for global conformational changes along them to account for opening of closing mechanisms during the docking. Side-chain flexibility accounts for the fact that most of the proteins adapt the position of their side-chains for a perfect alignment of their interface.
Usually, the described sampling algorithms generate tens to hundreds of thousands possible solutions depending on the algorithm and the size of the complex. These decoys need to be scored or further refined to make predictions about the protein complex.

2.4.2 Refinement

Most of the docking protocols perform a separate refinement step after the rigid-body sampling. Refinement protocols optimize a subset of decoys in a computationally more demanding procedure which introduces a greater portion of protein flexibility to increase interface complementarity and thus increase the number of native contacts [56]. The improvement of the interface contacts is important for the scoring of near-native solutions due to the fact that scoring functions make their predictions on the complementarity of the surfaces or the contacts between the two proteins. Normal rigid-body docking cannot deliver well matching interfaces due to steric clashes which result from the differences between the bound and the unbound form of the proteins.
Flexibility of the protein complex can be introduced in various ways on different scales. To account just for global conformational changes of the binding partners, the introduction of soft normal modes can be sufficient[43, 44].
Atomistic refinement on the protein complex can be performed by Molecular Dynamic simulations (MD) [15, 65]. MD simulations use intra- and intermolecular force fields between all atoms to achieve full flexibility of the complex. Furthermore, it is possible to introduce explicit water molecules in these simulations due to the fact that many protein complexes are driven towards their complex structure by solvation effect. Nevertheless, a refinement with MD is limited to a few structures and steps

due to the large computational demand. Therefore, applicability is limited by the accuracy of the force field and the ability to overcome energy barriers in the short simulation time.
Recent methods combine full flexibility of interface atoms with rigid-body movements of the ligand by the combination of intramolecular potentials between interface atoms and intermolecular potentials between the complex constituents [56].
Usually, refinements are performed on a subset of the best scoring decoys to safe computational time. This generally means hundreds out of tens of thousands decoys. So, refinement is also dependent on scoring due to the fact that the scoring function defines the structures which are placed in the subset for refinement. However, refinement generates near-native structures from which the scoring function should be able to predict a near-native solution with a high probability based on the well aligned interface. Hence, scoring has to be improved respective to the sampling and refinement protocols which are used for docking.

2.4.3 Rigid-Body Docking via ATTRACT

The ATTRACT docking program is one of the existing docking programs which has been used successfully in various rounds of CAPRI [70, 43, 66]. ATTRACT uses the previously described coarse grained representation for the two protein partners. Between the pseudo atoms a distance r_{ij} dependent modified Lennard-Jones potential (LJ) in combination with a Coulomb potential are used to describe the interaction. A soft Lennard-Jones potential is used to describe attractive interactions whereas repulsive interactions are introduced by a saddle point potential which consists of a reversed and a shifted LJ-potential. The attractive interactions are described in the form of equation 2.7 and the repulsive in form of equation 2.8.

attractive:

$$V = \epsilon_{AB} \left[(\frac{\sigma_{AB}}{r_{ij}})^8 - (\frac{\sigma_{AB}}{r_{ij}})^6 \right] + \frac{q_i q_j}{\varepsilon(r_{ij}) r_{ij}} \tag{2.7}$$

repulsive:

$$V = \begin{cases} -\epsilon_{AB} \left[\left(\frac{\sigma_{AB}}{r_{ij}}\right)^8 - \left(\frac{\sigma_{AB}}{r_{ij}}\right)^6 \right] + \frac{q_i q_j}{\varepsilon(r_{ij}) r_{ij}} & \text{if} \quad r_{ij} > r_{min} \\ 2e_{min} + \epsilon_{AB} \left[\left(\frac{\sigma_{AB}}{r_{ij}}\right)^8 - \left(\frac{\sigma_{AB}}{r_{ij}}\right)^6 \right] + \frac{q_i q_j}{\varepsilon(r_{ij}) r_{ij}} & \text{if} \quad r_{ij} \leq r_{min} \end{cases} \tag{2.8}$$

The parameter σ_{AB} and ϵ_{AB} define the effective pairwise radii and the strength of the attraction and the repulsion respectively, for each interaction between pseudo atoms of type A and B in contact. In the Coulomb potential, $\varepsilon(r_{ij})$ defines a linearly distance dependent dielectric constant and q_i and q_j define pseudo charges for the pseudo atoms i and j. At distance r_{min} the attractive term possesses its minimum at $-e_{min}$ and the repulsive term a saddle point of hight e_{min}. The values for σ_{AB} and ϵ_{AB} for each contact were iteratively optimized by improving the scoring of near-native against incorrect decoys [23].

A flowchart for the docking protocol of ATTRACT is shown in figure 2.4. For the docking run, ATTRACT places the ligand at various positions and in various orientations around the receptor. The maximum distance between the center of masses is slightly larger than the maximum distance of any atom from the ligands center. The orientation of the ligand is defined by its Euler angles φ, θ and ψ. For rigid-body docking, ATTRACT usually performs an energy minimization with a quasi-newton minimizer on the ligand's three translational (x,y,z) and three rotational (φ, θ, ψ) degrees of freedom. ATTRACT also possesses the possibility to use a Monte Carlo search for this purpose.
To account for conformational changes during the docking process, it is possible to perform docking with ensembles of receptor and ligand conformations. For larger conformational changes during the sampling, ATTRACT allows to introduce harmonic modes for both constituents which create additional degrees of freedom along the chosen normal modes [43, 44].
From a usual sampling, ATTRACT generates tens of thousands possible solutions for the protein complex. For most protein partners which knowingly do undergo no or little conformational changes, it is sufficient to score the determined structures with the original ATTRACT force field or another predictive scoring function to identify near-native structures.
For harder cases, it is often required to let a subset of structures undergo a further refinement of their 3D structure to get closer to the native form of the complex and to make them feasible for final scoring. Since refinement protocols are computationally more demanding due to calculations of intramolecular energy terms for full flexibility, a subset of structures which undergo refinement have to be defined. Refinement in ATTRACT is performed by interface ATTRACT (iATTRACT) [56] which combines full atomistic flexibility of interface atoms and rigid-body optimization at the same time. For that reason, iATTRACT uses an intramolecular force field for interface atoms and combines it with an intermolecular force field to allow large scale translational and rotational optimization simultaneously to smaller scale relative movements of interface atoms. The refinement protocol promises an improvement of native contacts up to 70 % by an average Δfnat of 0.189.

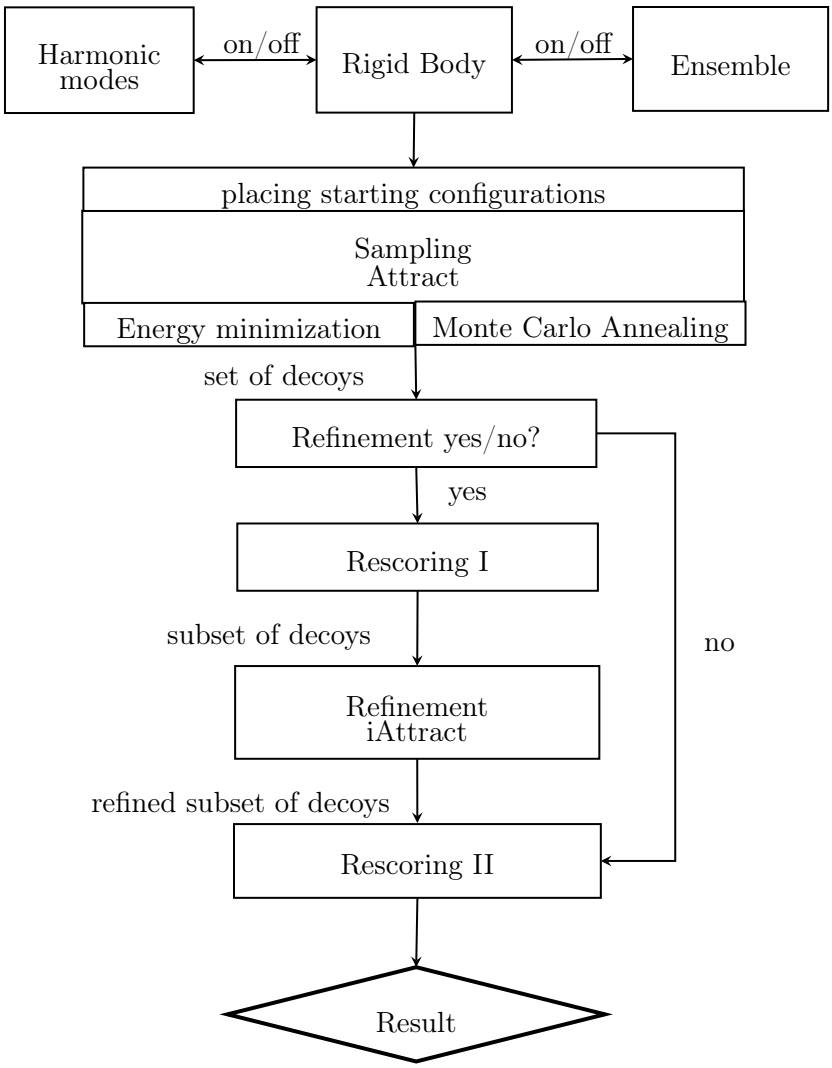


Figure 2.4: The ATTRACT protocol consists of the sampling with ATTRACT and refinement with interface Attact. To simulate flexibilities during the sampling, harmonic modes or ensembles of protein structures can be consulted.

To have as many as possible near-native solutions in the subset for a sufficient refinement, it is desirable to rescore with a scoring function which accounts for the diversity of interfaces from the near-native structures of the complex to place them in the subset. This scoring function might not necessarily determine the best structures on top but needs to place most of the near-natives in the subset. Currently, the scoring before the refinement is performed with the coarse grained potential of

ATTRACT and the final scoring after the refinement by the optimized potential for liquid simulations (OPLS) [34] due to its use for the refinement by iAttract.

2.5 Potential Forms for Scoring Functions

Scoring functions aim to predict the quality of decoys exclusively from the position of the proteins to each other. Therefore, they assign a score to the complex structures based on their coordinates. It has already been mentioned above, that a complete description of the free energy is supposed to be a perfect scoring function for the native structure of each protein complex. However, the native complex cannot be sampled from its unbound constituents. Nevertheless, the usage of energy constituents of the total free energy seems to be a reasonable approach for scoring.
Since many guided sampling methods use modified potentials based on energy functions, it seems also be appropriate to use their potentials for scoring. Nevertheless, potentials which are created for sampling purposes try to smooth the energy landscape to predict a variety of near-native structures for unbound docking. Hence, potentials which are trained explicitly for scoring purposes may provide more satisfactory results. In the following, potentials which use atomistic buried surface areas, step potentials, Lennard-Jones-like potentials and the Coulomb potential will be described to use their potential forms for the creation of knowledge-based scoring functions.

Potential based on Atomistic Buried Surface Area (BSA-potentials)

The solvation energy (equation 2.2) consists of a surface or cavity term for the solvation of uncharged molecules and an electrostatic contribution [59] to account for electrostatic interactions with the solvent. The surface term describes the energy difference from the solvation of uncharged atoms and the electrostatic part describes the energy which has to be expended for the addition of charges to the solvated molecule. The electrostatic contribution in combination with the Coulomb interaction yields the total electrostatic potential of the protein complex in the solvent. Thus, this term will be described later.
The cavity energy can be described by a potential which is proportional to the solvent accessible surface area (SASA). Thus, the difference of the cavity energies for the complex and its separated constituents is proportional to the buried surface area bsA_α of the atom types α [28]. The buried surface area (BSA) of the complex structure arises from the difference between the SASAs of the separated constituents and the SASA of the complex. This is shown schematically in figure 2.5. Hence, the cavity potential can be described as the sum over the product of the parameters σ_α with their buried surface areas 2.9.

$$\Delta\Delta G_{surface} = \sum_{\alpha} \sigma_{\alpha} bsA_{\alpha} \tag{2.9}$$

The calculation of the solvent accessible surface areas sA_{α} is performed by the rolling probe algorithm developed by Shrake and Rupley [58, 36]. The rolling probe algorithm tests at discrete positions around each heavy atom of the molecule whether a sphere with radius 1.4 Å, which is the assumed size of a water molecule, fits between itself and its neighbouring heavy atoms. The test is performed for many mesh grid-points around each atom and each uncovered point contributes a peace of surface area to the total protein surface. Furthermore, it is possible not only to determine the total SASA but also the SASA for different atom type. For this purpose, the chemical atom types as well as the assigned atom types from the GAA model or the ATTRACT model can be used.

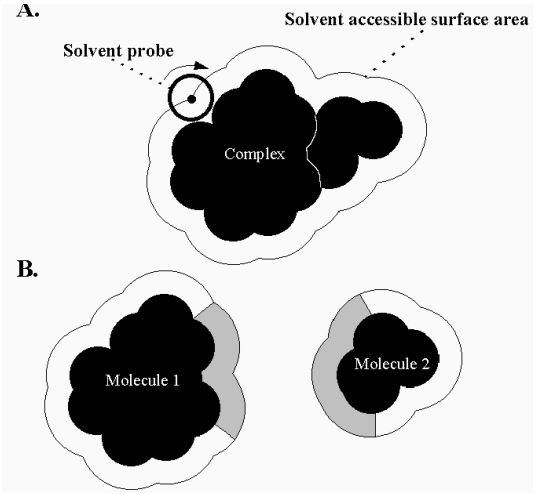


Figure 2.5: Computation of the buried surface area by the rolling probe algorithm. (image from: http://swift.cmbi.ru.nl/teach/theory/Theory_3.html)

Lennard-Jones Potential

Van der Waals interactions between atoms i and j can be described by a distance r_{ij} dependant Lennard-Jones (LJ) potential as shown in equation 2.10. The parameter ϵ_{AB} defines the depth of the minimum and σ_{AB} represents the Lennard-Jones radii

between atoms of type A and B. For distances smaller than the Lennard-Jones radii the potential becomes positive. Furthermore, the LJ radii defines the position of the minimum which located at $\left(2^{\frac{1}{6}}\sigma, \frac{1}{4}\epsilon\right)$.

$$V_{LJ} = \epsilon_{AB}\left[\left(\frac{\sigma_{AB}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{AB}}{r_{ij}}\right)^{6}\right] \tag{2.10}$$

$$V_{LJ} = \frac{\alpha_{AB}}{r_{ij}^{12}} - \frac{\beta_{AB}}{r_{ij}^{6}} \qquad \alpha_{AB}, \beta_{AB} > 0 \tag{2.11}$$

The Lennard-Jones potential can also be rewritten as a linear dependent potential on the parameter α_{AB} and β_{AB} shown in equation 2.11. For the equality between the two descriptions of the Lennard-Jones potential V_{LJ} the parameter α_{AB} and β_{AB} have to be larger than zero.

Saddle Point Potential

In ATTRACT the Lennard-Jones potential is used in a modified form. The attractive van der Waals interactions between two pseudo atoms are smoothed by taking the repulsive term to the power 8 instead of power 12. Additionally, a repulsive interaction is introduced which is described by a distance dependent saddle point potential. The saddle point potential is derived from the original LJ-potential shown in equation 2.12. Interactions between two atom types can either be attractive or repulsive. A binary variable $i_{vor} \in \{-1; 1\}$ defines the character of the interaction by using -1 for repulsive and 1 for attractive interactions.
At distances smaller than r_{min} the saddle point potential consists of a shifted LJ-potential by $2e_{min}$ to possess a saddle-point at r_{min} with the hight of e_{min}. For distances larger than r_{min} the potential is represented by an inverted LJ-potential which stays positive during its convergence to zero for infinity.

$$V_{repulsive} = \begin{cases} -\epsilon_{AB}\left[(\frac{\sigma_{AB}}{r_{ij}})^8 - (\frac{\sigma_{AB}}{r_{ij}})^6\right] & \text{if} \quad r_{ij} > r_{min} \\ 2e_{min} + \epsilon_{AB}\left[(\frac{\sigma_{AB}}{r_{ij}})^8 - (\frac{\sigma_{AB}}{r_{ij}})^6\right] & \text{if} \quad r_{ij} \leq r_{min} \end{cases} \tag{2.12}$$

Step Potential

Step potentials present a simple description for interactions between atoms of type A and B. They add a constant value e_{AB}^{D} to total score if the distance between two atoms is located in a certain range D. For scoring purposes, the range and number

of steps can be chosen freely. Hence, the score of the step potentials can be described as in equation 2.13.

$$E_{step} = \sum_{i}^{N} \sum_{j}^{M} e_{AB}^{D} \theta_D(r_{ij}^{AB}) \qquad \theta_D(r_{ij}) = \begin{cases} 1 & \text{if} \quad r_{ij} \in D \\ \\ 0 & \text{else} \end{cases} \tag{2.13}$$

Step potentials have already been used for the scoring of protein complexes in former works [62, 61, 51]. Tobi et al. developed a knowledge-based step potential which consists of two steps between 0-4 Å and 4-6 Å.
ZRank uses the atomic contact energy determined by Zhang et al. in their composite scoring function. This atomic contact energy (ACE) for proteins in a solvent is a simple energy function which accounts for desolvation effects in protein-protein interactions [71, 46]. The original parameters for the ACE potential used contacts between proteins in the range of 6 Å for their fit on experimentally determined values of the solvation energy.

Coulomb Potential

The simplest model to describe electrostatic interactions is given by a Coulomb potential between two charged atoms i and j in vacuum or in an dielectric medium with the dielectric constant ε (equation 2.14).

$$E_{coulomb} = \sum_{i}^{N} \sum_{j}^{M} \frac{q_i q_j}{\varepsilon r_{ij}} \tag{2.14}$$

The coulomb interaction is a long-range interaction due its dependency on the distances r^{-1}. To account for the influence of a surrounding dielectric medium the dielectric constant ε is used. Most solvents and especially water is a dipole and hence the description by a dielectric constant is inappropriate. A trivial way to account roughly for shielding effects of the water molecules is to define a distance dependent dielectric constant $\varepsilon = \varepsilon_0 r_{ij}$.
Both models represent a very rough description of the electrostatic contributions in protein interactions. Therefore, they might cause problems due to the overestimation of the coulomb interaction in combination with the Lennard-Jones potential. However, they might be moderate for insufficient resolved interfaces as they occur in bound docking.
Nevertheless, for structures of higher quality, a model which accounts for the effects of buried charges in the protein can be more satisfactory. Hence, the total electrostatic energy E_{es} would become the sum of the Coulomb energy $E_{coulomb}$ and the polarization energy E_{pol} (equation 2.15).

For the polarization term, the Poisson-Boltzmann equation can be solved analytically. Unfortunately, solving this equation for large systems is demanding very much computational time. To save computational time, an approximation for the polarization energy is given by the generalized Born (GB) model introduced by Still et al. [59]. The GB-model treats a molecule as a discrete medium of overlapping charged spheres in a polarizable dielectric medium. For this model, the polarization energy E_{pol} becomes dependent on the born radii α_i of each atom i (equation 2.16). The born radii depend on the neighbourhood of each atom and have to be calculated for each complex in advance [54, 27]. The function f_{GB} uses the born radii to compute an effective radii for electrostatic interactions which accounts for shielding effects by the protein environment (equation 2.17).
The consideration of polarization effects has been quite successful for MD simulations. However, it remains unsettled whether that term can contribute to the improvement of scoring, especially for the decoys from unbound docking. Nevertheless, the model for the polarization energy deserved to be mentioned for completeness of the total free energy contributions.

$$G_{es} = G_{coulomb} + G_{pol} \tag{2.15}$$

$$G_{pol} = -166\left(1 - \frac{1}{\varepsilon}\right)\sum_{i}^{N}\sum_{j}^{N}\frac{q_i q_j}{f_{GB}} \tag{2.16}$$

$$f_{GB} = \left(r_{ij}^2 + \alpha_{ij}^2 e^{-D_{ij}}\right)^{0.5}; \quad \alpha_{ij} = (\alpha_i \alpha_j)^{0.5}; \quad D_{ij} = r_{ij}^2/4\alpha_{ij}^2 \tag{2.17}$$

2.6 Parametrization

The estimation of the parameters for the presented potential forms can be divided into a physics-based parametrization and a knowledge-based parametrization. To estimate parameters which represent a physical energy term, they are determined by the fitting to experimentally determined values from physical processes [34, 21, 71]. On the other hand, knowledge-based scoring functions determine their set of parameters from the comparison between the scores of near-native or native structures and incorrect structures.

As mentioned above, physics-based energy models may represent reasonable approaches for the scoring of protein complexes, especially for the scoring of the native

structure. For a physics based parametrization, the form of the potential model must be chosen by reasonable assumptions on the dependency of the energy term. Usually regression methods are used to estimate a set of parameters which are able to determine energy values for biological processes.
For instance, the atomic contact energy (ACE) for proteins is a simple energy function which accounts for desolvation effects in protein-protein interactions [71, 46]. The ACE provides an approximation of solvation energy change when protein-water contacts dissolve and new inter-protein contacts are formed. Hence, the ACE includes van der Waals interactions between the proteins and accounts also for broken van der Waals interactions with the solvent which can seen as the difference in solvation energy. The atomic contact energy is defined by a simple step potential between a receptor atom i and ligand atom j of atom types A and B. In the original definition of the ACE by Zhang et al. two atoms are defined to be in contact if they are in a distance of 6 Å to each other.

ITScore-PP [31], Sipper [53], ProBinder [24], DECK [39], DARS [13], Tobi [61] and ATTRACT [69] belong to the group of knowledge-based potentials. They are determined from the comparison between the scores of a set of native structures or generated near-native decoys and incorrect solutions. Various techniques can be used to generate a knowledge-based scoring function from its training set including heuristic and gradient using optimization methods. These methods optimize a target function which aims to improve the scoring of the near-native structures against the incorrect.
The potential of Tobi et al. is a knowledge-based scoring function which uses 19 atom types for the description of heavy atoms of the receptor and the ligand. Between these atom types A and B, a constant energy value e^{sr}_{AB} is defined in the range of 0-4 Å and another e^{lr}_{AB} in the range of 4-6 Å. Thus, the scoring function is represented by a step potential consisting of two steps with 190 parameters for each of them.
Tobi's parameters were trained by a simplex algorithm on a decoy set which contained near-native and incorrect structures. The decoys were generated by a surface matching algorithm from the bound forms of their constituents. A solution was defined to be near-native if the ligand rmsd was less than 5 Å.

Composite scoring functions usually use a linear combination of physical potentials for the different energy contributions of protein interactions to score their results with an pseudo energy E_{score} as in equation 2.18. Famous docking programs as pyDock [12], RosettaDock [40], HADDOCK [17], ZDock [51, 52], FireDock [1], and FiberDock [42] use these composite functions for their scoring. ZRank for instance, uses a combination out of seven constituents consisting of Van der Waals, Electrostatics and desolvation terms including long and short range interactions and

the atomic contact energy model. The weights ω_i for the different contributions are usually estimated by machine learning algorithms to account for the different scales and impacts of the constituents to the total score. Hence, composite scoring functions are hybrid scoring functions since they use potentials which possess a physical meaning but combine them to improve scoring of a training set of protein decoys.

$$E_{score} = \omega_{vdw} E_{vdw} + \omega_{Coulomb} E_{Coulomb} + \omega_{Solvation} E_{Solvation} \tag{2.18}$$

A successful combination of scoring functions can be achieved by the combination of orthogonal scores. This means that bad scored near-native structures receive a good score from the combined scoring function. Furthermore, the false positive structures receive a bad score for a better separation from the near-native structures [47]. In other words, The scoring functions should account for different characteristics of the structures and hence are supposed to favour different types of structures.

2.6.1 Parametrization Techniques

To find a set of parameters for a knowledge-based scoring function from the presented potential forms, it is necessary to train them on a set of decoys. Heuristic and non heuristic optimization algorithms can be used to optimize self-defined target functions or the cost-functions of machine learning algorithms to predict a set of parameters for scoring purposes.
Currently developed methods use machine learning techniques like neural networks [9, 50], ROC based genetic algorithms [8, 5], nonlinear [2, 7] or linear classifiers [5, 22] on different sets of complex features to create their scoring functions. All these supervised machine learning algorithms train their underlying decision functions g on a set of given data $\{(\boldsymbol{x_0}, y_0), \ldots, (\boldsymbol{x_n}, y_n)\}$ to determine its parameters for the predictions on unknown cases. For the usage of classifiers, the structures must be divided into at least two classes $y \in \{-1, 1\}$ based on their quality. The decision functions can be probability models of the underlying data or more generally functions which assign an output value t for their classification y on a given feature vector $\boldsymbol{x}$. The classification by a support vector machine for instance, defines a hyperplane in the feature space for an optimal separation between the two classes. The distance from that hyperplane is used as a decision function for classification.

For the creation of a distance d dependent interaction potential, inverse Boltzmann equation can be used on the density distribution for the contacts between atoms of type A and B [68].

$$V_{AB} = -RT \cdot ln \left(\frac{N_{AB}^{observed}(d)}{N_{AB}^{expected}(d)} \right) \tag{2.19}$$

In equation 2.19 $N_{AB}^{observed}(d)$ corresponds to the number of observed contacts between atoms of type A and B at distance d, while $N_{AB}^{expected}(d)$ refers to the reference state which states the number of contacts if they were freely distributed. For scoring purposes RT can be set to 1.
The potentials from this method cannot only differ from the structures in the training set but also from the calculation of the reference states. For an infinite system the reference state would result in a uniform density of atoms. However, proteins are not infinite and their atoms are not uniformly distributed in surrounding shells around the center of mass. Thus, many groups try to account for the irregular distribution for their reference state by different approaches. While some correct the uniform distribution analytically, others try to take the information from the surface or the interface distribution of experimentally determined structures [39] or incorrect docking solutions.

For the estimation of the scoring parameters in this work, Monte Carlo Annealing and linear regression serve as training methods. Monte Carlo Annealing is a heuristic method which is applicable to complex problems in computational sciences. The advantage of Monte Carlo Annealing is that the target function is dependent on the ranking of each complex separately and not directly on their scores. For this reason, the scoring problem itself can be approached. Furthermore, the Monte Carlo approach is able to tackle different scoring problems directly by the definition of several problem-adapted target functions.
For Linear Regression the scoring problem must be redefined into a regression problem. Instead of regarding the rank of decoys for each complex, linear regression tries to estimate certain values for the structures from their feature vectors. Nevertheless, this description of the problem correlates with the target functions of the MC approach but might be less prone to overfitting due to its indirect description of the problem.

Monte Carlo Annealing

Monte Carlo Annealing is a heuristic optimization method which is often used to solve complex problems in computational sciences. The idea results from the annealing process in metallurgy in which atoms find their energetic minimum while they cool down slowly. Monte Carlo is supposed to be a good working algorithm if the problem is too complex to solve it in an analytical way. Furthermore, the algorithm is a useful approach to estimate the global minima of a target function due to the fact that

it can overcome barriers by accepting also worsening steps with a certain probability.

In order to find an ideal parameter set for a scoring function, it is possible to perform simulated annealing in the parameter space. The developed protocol changes a randomly chosen parameter p_i by a constant or temperature dependent step $\pm\Delta p(T)$ and re-calculates the score of the training decoys for each complex. To rescore all training decoys, the product of the changed parameter and the precalculated feature is just computed. Afterwards, the decoys are resorted for each complex and the target function t_f assigns a value to the new order. The sum over all target function values τ_i for the number of complexes in the training set is compared to the value at the previous step τ_{i-1} and the parameter change is accepted by a certain probability which is calculated from the Metropolis criteria (equation 2.20).

$$P_{accept} = \min\left(1, exp\left\{-\frac{\tau_i - \tau_{i-1}}{T}\right\}\right) \qquad \tau = \sum_{n}^{N_{complexes}} t_f(\overrightarrow{E}) \tag{2.20}$$

A target function can be any function that derives a value from the scores of the decoys for each complex. To tackle the scoring problem directly, the target functions determines their target scores from the ranking of the decoys of each complex in the set. To improve the scoring of near-native structures compared to incorrect structures for each complex, a qualitative evaluation of their complex structures, for example the fnat, the irmsd, the lrmsd or the Capri stars, is taken to assign a qualitative weight. Afterwards, the target function calculates a value based on the rank and the quality of the structures with the aim to place the structures of high quality further on top.

It can be seen from equation 2.20 that the temperature T affects the probability to accept a decreasing τ. While the probability of a decreasing τ is close to 0 for low temperatures it can become nearly 1 for high temperatures.
The procedure is repeated for a defined number of steps or until a convergence criteria is fulfilled. The temperature is cooled down after each step to decrease the possibility of worsening steps and hence to force the parameter to stay in the located maximum of the target function.

Generalized Linear Models for Regression

Generalized linear models (GLM) serve for linear regression to predict outcome values from its features. In generalized linear models each outcome $\widehat{y}_i$ is supposed to be linearly dependant on its feature vector $\boldsymbol{x}$. The observed outcome $\widehat{y}_i$ is supposed to be a combination of the vector product between the parameters and the features

$\boldsymbol{w} \cdot \boldsymbol{x}$ and some noise ϵ (equation 2.21).

$$\widehat{y}_i = \boldsymbol{w}\boldsymbol{x_i} + \epsilon_i \tag{2.21}$$

Each GLM is based on the assumption that the underlying distribution of the noise ϵ is a function of the exponential family with mean $\mu = 0$ and some variance σ^2 [6, 48].
For a Gaussian distribution $\mathcal{N}\left(0, \sigma^2\right)$, the distribution of the outcome values can be derived if the noise is independent from the input vector $\boldsymbol{x}$ (equation 2.22 and 2.23). Thus also the probability distribution of the outcomes is a Gaussian with variance σ^2.

$$p\left(\epsilon_i\right) = \frac{1}{2\pi\sigma}\exp\left(-\frac{\epsilon_i^2}{2\sigma^2}\right) \tag{2.22}$$

$$p\left(\widehat{y}_i|\boldsymbol{x_i};\boldsymbol{w}\right) = \frac{1}{2\pi\sigma}\exp\left(-\frac{\left(\widehat{y}_i - \boldsymbol{w}^T\boldsymbol{x_i}\right)^2}{2\sigma^2}\right) \tag{2.23}$$

Given a set of feature vectors $\boldsymbol{X} = (\boldsymbol{x_0}, \ldots, \boldsymbol{x_k})$ and outcomes $\widehat{\boldsymbol{y}} = (\widehat{y}_0, \ldots, \widehat{y}_k)$ the likelihood for the parameters $\boldsymbol{w}$ can be written as the product of all probabilities for the outcomes.

$$L\left(\boldsymbol{w}\right) = \prod_{i=0}^{k}\frac{1}{2\pi\sigma}\exp\left(-\frac{\left(\widehat{y}_i - \boldsymbol{w}^T\boldsymbol{x_i}\right)^2}{2\sigma^2}\right) \tag{2.24}$$

To estimate the most probable parameters under the given data $(\boldsymbol{X}/\boldsymbol{y})$, the logarithmic likelihood is maximized. The problem which derives, is equivalent to the problem of minimizing the well-known χ^2 cost-function for ordinary least squares regression.

$$\chi^2 = \sum_{i=0}^{k}\left(\widehat{y}_i - \boldsymbol{w}^T\boldsymbol{x_i}\right)^2 \tag{2.25}$$

From other assumptions of the distribution for the noise, other well-known regression methods are derived like ridge regression, Lasso regression, logistic regression or Least absolute deviation. Each method possess advantages respective to its cost-function. The cost-function is the equivalent to the target function of the MC approach and describes the regression problem as a minimization problem. Ordinary least squares fits are supposed to be prone to outliers and thus can lead to insufficient results for some data sets. A more robust regression method would be based on a

noise distribution which does not respect the outliers for their regression and hence might help to improve the parameters.

Each type of regression should be able to predict the parameters of a scoring function on a set of precalculated feature vectors and their outcome values. Due to the fact that scoring aims to predict the quality of the structures, the outcome values $\widehat{y}$ for the generation of a scoring function by linear regression are supposed to be a qualitative evaluation of their 3D structures. Thus, possible values are as in Monte Carlo Annealing, the Capri-stars, the Irmsd's, the Lrmsd's, the Fnat's and also the fractions of non-native contacts.
The estimated set of parameters $\boldsymbol{w}$ might not be able to serve for predictions of the absolute outcome values due to the diversity between the interfaces and hence the diversity between the feature vectors for structures with the same quality. However, the tendency for scoring purposes should be reflected due to the cost-function which respects each data point equally for its minimization.
For the minimization of the cost-functions, linear regression protocols use common optimization algorithms like the Levenberg-Marquardt method or other minimizers which use the gradient of the cost-function for their predictions. Due to the form of the cost-functions for linear regression, these methods are able to find its global minimum. However, this ideal solution for the regression problem might not be the ideal solution for the scoring problem due to the redefinition of the scoring problem for the usage of linear regression algorithms. Nevertheless, the regression on the values of qualitative evaluations aims to give all structures with good quality a good score and hence correlates much with the target function of the Monte Carlo Approach.

2.7 Methods

2.7.1 Creation of Feature Vectors for different Energy Models

As declared above, scoring functions assign scores to complex structures based on characteristics of their coordinates. Whenever these characteristics can be transformed into numbers that describe the complex, scoring functions can be used to calculate a score E_{score} out of these features f^i by its parameters p^i. For the estimation of the parameters different algorithms can be used depending on the form of the scoring function.
Most of the energy models for protein interactions can be described in a linearly dependent form on some features and their parameters. Therefore, linear machine learning methods like linear regression can be used to determine the parameters for an ideal scoring on feature vectors $\boldsymbol{f}$ from a set of training structures $\{(\boldsymbol{f_0}, y_0), \ldots, (\boldsymbol{f_n}, y_n)\}$.

Furthermore, the Monte Carlo Annealing algorithm uses these feature vectors to calculate the final score by a vector product between the parameters and the features. For contact dependent scoring functions, it avoids the huge sum over all atomistic contacts ij for the score and replaces it by a short sum over each interaction type AB. The general principle is shown in equation 2.26. The sum over the features f_{ij}^{AB} between all atoms i and j of the receptor and ligand is calculated in advance for each type of contact AB. Each energy potential possesses different features which are used for their scoring and will be explained in the following.

$$\begin{aligned} E_{score} &= \sum_{i}^{N}\sum_{j}^{M} p_{AB} f_{ij}^{AB} \\ &= \sum_{A}^{\Lambda}\sum_{B}^{\Lambda} p_{AB} F_{AB} \quad \text{with} \quad F_{AB} = \sum_{i}^{N}\sum_{j}^{M} f_{ij}^{AB} \quad \text{if} \quad f_{ij} \quad \text{of AB} \\ &= \begin{pmatrix} p_{11} \\ \vdots \\ p_{\Lambda\Lambda} \end{pmatrix} \cdot \begin{pmatrix} F_{11} \\ \vdots \\ F_{\Lambda\Lambda} \end{pmatrix} \end{aligned} \tag{2.26}$$

For the potentials based on the atomistic buried surface area, it is possible to precalculate for each decoy in the training set a vector consisting of the buried surface areas for each atom type. Hence, the BSAs serve as features for a scoring function with the parameters σ_α which is based on the model for the solvation energy.

$$E_{Solvation} = \begin{pmatrix} \sigma_1 \\ \vdots \\ \sigma_\Lambda \end{pmatrix} \cdot \begin{pmatrix} bsA_1 \\ \vdots \\ bsA_\Lambda \end{pmatrix} \tag{2.27}$$

For the Lennard Jones potential, a linear dependency on the exclusively positive parameters α and β can be depicted from equation 2.11. From that description, it is possible to calculate vectors containing the sum over distances to the power of -6 and -12 respectively for all contacts of contact type AB between the receptor atoms i and the ligand atoms j. Thus, the van der Waals energy can be described as the sum of two vector products between vectors of the length of the number of contact types as it is shown in equation 2.28.

$$E_{vdw} = \begin{pmatrix} \alpha_{11} \\ \vdots \\ \alpha_{\Lambda\Lambda} \end{pmatrix} \cdot \begin{pmatrix} \sum_{ij} \left(1/r^{(11)}\right)^{12} \\ \vdots \\ \sum_{ij} \left(1/r^{(\Lambda\Lambda)}\right)^{12} \end{pmatrix} - \begin{pmatrix} \beta_{11} \\ \vdots \\ \beta_{\Lambda\Lambda} \end{pmatrix} \cdot \begin{pmatrix} \sum_{ij} \left(1/r^{(11)}\right)^{6} \\ \vdots \\ \sum_{ij} \left(1/r^{(\Lambda\Lambda)}\right)^{6} \end{pmatrix} \tag{2.28}$$

For the modified Lennard-Jones potential which is used by ATTRACT, to have also the possibility of total repulsive interactions between pseudo atoms, no linear dependency to the parameters can be established. This results out of the dependency of r_{min} and e_{min} on the parameter ϵ and σ. Thus, the energy score from the saddle point potential cannot be represented by a vector multiplication between the parameter and feature vectors. Hence, linear regression or classifiers are not able to determine parameters for this type of potential. In the Monte Carlo Annealing protocol, it would be possible to rescore the decoys at each step by taking the sum over all atomistic contacts. Nevertheless, this procedure might be unable to perform a lot of steps for sufficient results due to long computations.
However, it is possible to use spline interpolation by Lagrange polynomials between pre-defined nodes $\{x_0, \ldots, x_k\}$ to make fast rescoring feasible. Thereby, only the potential values at these nodes have to be calculated and multiplied with the sum over the Lagrange basis polynomials at each step.
The potential value in between the defined nodes can be approximated by an interpolation of chosen degree k. The value of the potential at position x can be interpolated by the usage of polynomial Lagrange interpolation on a given set of k+1 pre-defined surrounding nodes and their re-calculated potential values $\{(x_0, y_0), \ldots, (x_k, y_k)\}$ as shown in equation 2.29.

$$f(x) \approx \sum_{j=0}^{k} y_j l_j(x) \tag{2.29}$$

$$l_j(x) = \prod_{m \neq j}^{k} \frac{x - x_m}{x_j - x_m} \tag{2.30}$$

Thereby, the potential value at position x becomes linearly dependent on all surrounding potential values y_i at the nodes x_i. The sum over all Lagrange basis polynomials (equation 2.30) is stored for each surrounding node $\{x_0, \ldots, x_k\}$ and contact type AB. Hence, only the potential values on the nodes for one contact type have to be re-calculated for the rescoring during the Monte Carlo procedure.
In order to keep the distances between the nodes small for an satisfactory interpolation, nodes are defined at uniform distances. For the calculation of the potential

value at the position x, only the k+1 surrounding nodes are taken into account which is referred to as a spline interpolation. Usually an interpolation by polynomials to the power of k=3 is performed which means that the nearest 4 surrounding nodes are taken for the interpolation. The total interpolated score of any potential function can then be determined by equation 2.31.

$$E_{interpolate} = \begin{pmatrix} y_0^{(11)} \\ \vdots \\ y_0^{(\Lambda\Lambda)} \end{pmatrix} \cdot \begin{pmatrix} \sum_{ij} l_0\left(r^{(11)}\right) \\ \vdots \\ \sum_{ij} l_0\left(r^{(\Lambda\Lambda)}\right) \end{pmatrix} + \cdots + \begin{pmatrix} y_k^{(11)} \\ \vdots \\ y_k^{(\Lambda\Lambda)} \end{pmatrix} \cdot \begin{pmatrix} \sum_{ij} l_k\left(r^{(11)}\right) \\ \vdots \\ \sum_{ij} l_k\left(r^{(\Lambda\Lambda)}\right) \end{pmatrix} \quad (2.31)$$

Vectors of the complex structures for step potentials contain the number of contacts n_{AB} for atoms of type A in contact with atoms of type B. A contact between an atom i and j can be defined on the distance to each other. Thus, two atoms are in contact if their distance is in between the ranges of the defined step. Finally, the score can be calculated by the vector product between the number of contacts for each type and the parameter, shown in equation 2.32.

$$E_{ACE} = \sum_{b}^{N_{bin}} \begin{pmatrix} e_{11}^b \\ \vdots \\ e_{\Lambda\Lambda}^b \end{pmatrix} \cdot \begin{pmatrix} n_{11}^b \\ \vdots \\ n_{\Lambda\Lambda}^b \end{pmatrix} \quad (2.32)$$

The size of the feature vectors depends on the number of parameters and on the number of atom types. An atomistic description by 27 atom types as for the grouped-all-atom model results already in 378 contact types. Hence, a potential based on the buried surface area gets along with 27 parameters, a step potential possessing one step uses 378 parameters and a normal Lennard-Jones potential already 756. In general, potentials which use more parameter tend to overfit more easily.

2.7.2 The ATTRACT Benchmark

The protein docking benchmark 4.0 [32] consists of 176 protein complexes. The benchmark which is used in this work for the generation and evaluation of scoring functions uses 164 complexes because of the fact that 12 complexes were sorted out due to several reasons.
The complexes 1OYV and 1QFW are duplicates, showing alternative binding modes.

The complex 1N2C is massive, so that cleaning up the structure failed. For the structures 1IRA, 1Y64, 1H1V, 1F6M and 1FAK, the rigid-body docking approach with unbound protein forms is inappropriate since they cannot be aligned due to the large conformational change with an Rmsd $\geq 6\,\text{Å}$. Two further structures 1DE4 and 2NZ8 cannot be handled by the default ATTRACT because 1DE4 is too big to use grids and 2NZ8 has a positive score in the bound form. For the complexes 1R8S and 1BGX ATTRACT was not able to generate atleast one acceptable solution and hence they are not used for development of scoring functions as well. In the following the 164 remaining complexes will be called the ATTRACT benchmark.

The complexes in the protein docking benchmark 4.0 are divided into rigid-body cases also called easy cases, medium and hard cases for docking protocols. The difficulty was defined by the degree of conformational change due to the interface Rmsd between the bound and the unbound form and the fraction of non-native contacts of the unbound form. Accordingly rigid-body structures are defined as complexes which have an irmsd $< 1.5\,\text{Å}$ and a fraction of non-native contacts fnon-nat < 0.4, difficult cases have an irmsd $> 2.2\,\text{Å}$ and medium cases are the remaining complexes. Finally, the ATTRACT benchmark consists of 121 rigid-body cases, 26 medium cases and 17 hard cases.
For the complexes in the ATTRACT benchmark an unbound rigid-body docking run was performed with ATTRACT. Afterwards, the generated decoys were sorted after their ATTRACT score which was calculated using a cut-off distance of $r_{cut}=\sqrt{50}\,\text{Å}$. For the final decoy-set redundant structures which were generated atleast twice in set were sorted out. The number of generated decoys for the complexes by this procedure varies between 6,000 to 60,000 depending on the molecular size of protein complex.
The quality of the generated decoys was evaluated on the CAPRI criteria (table 2.1) which is based on the fraction of native contacts, the interface and the ligand Rmsd. The complexes 1A2K, 1AKJ, 1BJ1, 1D6R, 1E4K, 1EZU, 1F51, 1FCC, 1HCF, 1I4D, 1JWH, 1JZD, 1KKL, 1ML0, 1OFU, 1OYV, 1RLB, 1RV6, 1WDW, 1XU1, 2B4J, 2HMI, 2HQS, 2OOR, 2VDB, 3BP8 possess symmetric binding modes. For that reason the Rmsd's and fractions of native contacts were also calculated for the alternative symmetric structures and the best values for a generated decoy were taken for its evaluation. The total number of generated structures, the number of acceptable, medium and high quality solutions and the type of difficulty for each complex is listed in table A.3 in the appendix.

Crossvalidation

For the training of the scoring parameters, the ATTRACT benchmark was divided into a test and a training set consisting of 140 complexes and 24 respectively. The

test set contains 14 rigid-body (2OOB, 1US7, 1MAH, 1JTG, 1E6E, 1TMQ, 1PPE, 1HCF, 1FFW, 1DQJ, 1OFU, 1AZS, 1PVH, 1EZU), 4 medium (1IB1, 2J7P, 1M10, 1HE8) , and 6 (2OT3, 2HMI, 1FQ1, 2C0L, 1IBR, 2I9B) hard cases. For the training of the scoring functions, the training set was further divided into a proper training set consisting of 112 structures and a validation set of 28 structures to perform 5-fold crossvalidation on 5 different sets of complexes. 5-fold crossvalidation is performed to avoid overfitting on complexes in the training set. Thereby, it is possible to evaluate the result during the training process in Monte Carlo Annealing on the validation set. After the training, the improvement of the scoring performance of the validation set, the increase of its target function output for Monte Carlo Annealing or the root mean square deviation of its predicted values for linear regression, can be taken to evaluate the training procedure.

2.7.3 Monte Carlo Annealing for Step Potentials and the BSA-Potentials

Potentials which were trained by Monte Carlo Annealing used a `'ziczac'` annealing scheme whereby the temperature fluctuates with a sinus^2 between two linearly decreasing boundaries. The starting temperature of the annealing was chosen to be 50 depending on the target function and their outcomes which are generated for an improvement. The temperature was cooled down in 100.000 steps until 0.1 % of the starting temperature was reached. The Monte Carlo search was stopped after a convergence criteria was fulfilled for 300 steps. The step size in parameter space was chosen `'adaptive'` to be dependent on the temperature with a maximal step size of 1 at the start of the Monte Carlo search.
As a target function `'positionlinear'` was chosen which sums up the product of all the structural weights $\omega_{quality}$ which is given by a qualitative evaluation of the structure and the weight for its rank ω_{pos} in the set of decoys, shown in equation 2.33. As structural weights, the Capri-stars were chosen to account for qualitative differences between acceptable, medium and high quality solutions.

$$t_f = \sum_{i}^{N_{trainset}} \omega^{(i)}_{quality} \cdot \omega^{(i)}_{pos}(rk[E_i]) \tag{2.33}$$

The implemented protocol is able to train a set of 378 parameters on 3000 decoys of the 140 training set complexes in 10,000 steps in about 5 minutes on a single core. This means that the program is able to execute 33.3 steps per second which is equal to rescore and resort about 15 million protein structures in that time. Further information on the adjustments of the training parameter and especially the potential target functions of the Monte Carlo Annealing protocol are given in section **training-MC.py** in the appendix.

2.7.4 Linear Regression for Step Potentials and the BSA-Potentials

To estimate the potential parameters by linear regression, an ordinary least squares fit was used on the number of contacts in the defined steps and the atomistic buried surface areas respectively. As target values for the regression process, the negative Capri-stars were taken. Thus, structures which possess a high quality receive a larger negative outcome value.
Linear regression was performed on enriched decoy sets which included all near-native structures for each complex. Due to the fact that each structure is weighted equally in the regression, an enriched decoy set accounts more for the features of the near-native structures in its regression. To eliminate the influence of the complex size and hence its average total number of contacts and size of the BSA, each number of the contact types AB was divided by the average number of contacts of its complex and each atomistic BSA was divided by the average total BSA for the complex.
Further information on the protocol for linear regression are given in the appendix in the section **training-glm.py**.

2.7.5 Monte Carlo Annealing for van der Waals Potentials

To estimate parameters for a pure Lennard-Jones potential, Monte Carlo Annealing is performed on feature vectors which use the sum over the distances to the power of -8 and the negative sum over the distances to the power of -6. Due to the fact, that the parameter α is larger than β in the LJ-potential, the parameters ϵ and σ are changed in the annealing and α and β are calculated on the run by equation 2.34:

$$\begin{aligned} \alpha_{AB} &= \epsilon_{AB} \cdot \sigma_{AB}^{8} \\ \beta_{AB} &= \epsilon_{AB} \cdot \sigma_{AB}^{6} \end{aligned} \tag{2.34}$$

By the usage of a repulsive potential form for van der Waals interactions by a saddle point description, spline interpolation has to be performed to use Monte Carlo Annealing for the estimation of the potential parameters. Therefore, the feature vectors are used which contain the sum of the Lagrange polynomials for each contact type at the predefined nodes for interpolation (section 2.7.1). At each step in the Monte Carlo procedure a parameter value for one contact type is changed and the van der Waals potential values are calculated for the predefined nodes of the spline interpolation. By a simple multiplication of the sums of the Lagrange polynomials in the feature vectors and the potential values the new score can be approximated.
Due to the fact that the parameter $i_{vor} \in \{-1, 1\}$ defines whether a LJ-potential or a repulsive saddle point potential is used for the description of the interaction between two atoms of type A and B, i_{vor}^{AB} is changed if ϵ_{AB} becomes smaller than zero.

Chapter 3

Results

This chapter describes the various scoring functions that were generated, and their performances in terms of the small and the big scoring problems (see section 2.1.2). As described, there are four choices to be made: structure representation, decoy set, potential shape and parametrization. Among all scoring functions, the decoy set is the same, namely the 'ATTRACT benchmark', coming from a rigid-body sampling by ATTRACT on 164 complexes of protein docking benchmark 4.0 (see section 2.7.2). The set of complexes were split in a training set and a test set. For the training of all potentials, 3000 structures of each training set complex were taken, containing the near-native solutions and the best scoring incorrect solutions after the scoring by ATTRACT. In some cases, the decoy set was enriched by adding the near-native structures outside the top 3000 structures (these potentials are marked with a star ('*')). Thus, the training sets which were enriched by this method do contain a larger diversity of near-native structures but do also contain near-native structures which must have unfavourable contacts or steric problems for the ATTRACT potential. As potential forms, step potentials, potentials based on the atomistic buried surface areas (BSA-potentials), Lennard-Jones potentials and the mixed van der Waals description of ATTRACT including saddle point potentials are used.
To prevent the potentials from overfitting on single complexes, 5 different training and validation sets were created out of the complex structures in the training set to perform 5-fold or leave-one-out-crossvalidation. Hence, the actual training was performed on 112 complexes while 28 complexes were used for training set validation. After the training on each set, the average for each interaction parameter was taken as the resulting set of parameters for the BSA-potentials and the step potentials. To account for the possibility of different scales in the parameter sets, the parameters were normalized by their standard deviation. For the van der Waals potentials, the final parameter set was chosen on the largest increase of the target function in the validation set.
Finally, the final potentials were evaluated on all decoys. To consider the possibility of overfitting on the training set, the evaluations are performed separately for the complexes in the training and the test set.

Two structure representations were tested: the all-atom GAA representation and the coarse-grained ATTRACT representation (see section 2.3). Two parametrization procedures (section 2.6.1) were tested: Monte Carlo (designated as MC) and Linear Regression (designated as LinReg). The sections of the chapters are organized by the potential shapes that were tested: step potentials (section 3.1), buried surface area (section 3.2) van der Waals potentials (section 3.3). In section 3.4, it is analyzed to what extent the scoring functions correlate with each other, and in section 3.5, a composite scoring function is created based the most successful scoring functions.
In all sections, the ATTRACT force field (section 2.4.3) and the Tobi step function (section 2.6) are used as baseline scoring functions, to which all others are compared.

The results for all evaluated scoring functions in this work can be found in the appendix in tables A.4, A.6 and A.8 for the training and in the tables A.5, A.7 and A.9 for the test set.

3.1 Step Potentials

Six step potentials were generated differently and evaluated: one potential which possesses one step to 10 Å in the ATTRACT coarse grained representation of the proteins called 'MC_attract_step_10', three potentials possessing one step to 10 Å in the grouped-all-atom (GAA) representation (section 2.3.1) and two further potentials in the GAA representation possessing two steps from 0-6 Å and from 6-10 Å. The potentials 'MC_gaa_10', MC_gaa_10*' and 'LinReg_gaa_10*' are potentials using contacts in the GAA representation and one step from 0-10 Å. The potentials 'MC_gaa_6_10' and LinReg_gaa_6_10*' do represent grouped-all-atom potentials with steps from 0-6 Å and 6-10 Å.
An overview of the training parameters of the generated scoring functions is given in table 3.1. The Monte Carlo Annealing and the linear regression protocol for the generation of each parameter set are described in detail in section 2.7.3 and 2.7.4 respectively.

Scoring of Near-Native Structures from Unbound Rigid-Body Docking

To evaluate the probability to predict a near-native structure from the scoring after an unbound rigid-body sampling, the rank of the best scoring near-native structure is regarded. In figure 3.1 the percentage of complexes in the training set (a) and the test set (b) for which a near-native structure can be predicted in the top 1, top 10, top 100 and top 1000 of the generated decoys is plotted.

Table 3.1: Parameters for development of six step potentials.

potential name	method	decoyset	ranges	representation
MC_gaa_10	MC	original	0-10	GAA
MC_gaa_10*	MC	resorted	0-10	GAA
MC_gaa_6_10	MC	original	0-6-10	GAA
MC_attract_10	MC	original	0-10	ATTRACT
LinReg_gaa_10*	LinReg	resorted	0-10	GAA
LinReg_gaa_6_10*	LinReg	resorted	0-6-10	GAA

In the figures 3.1 and 3.2, it is striking that the performance of Attract is much worse for the test set compared to the training set. In the test set, the Attract score finds an acceptable solution for only 50 % of the complexes in the top 1000 decoys compared to 100 % in the training set. Furthermore, Attract places only 25 % of the near-native complexes in the best 10 % of the decoys in the test set but 50 % in the training set. Therefore, it could be stated that the choice of complexes for the training and test set might not be moderate. On the other hand, the scores of the step potentials have to deal with the same structures in the test set and a quantitative comparison to Attract and Tobi is satisfactory for the evaluation of the generated potentials.

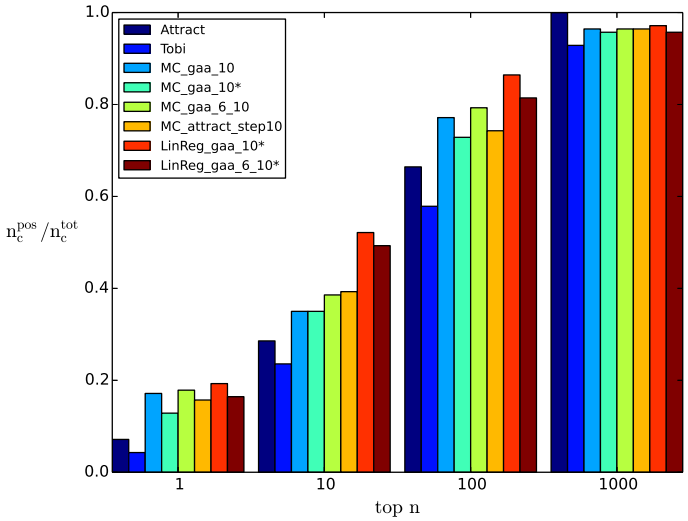
Overall, it can be seen that all step potentials find near-native structures for a higher portion of complexes than Attract and Tobi in both sets, except at rank 1 in the test set and rank 1000 in the training set. Compared to Attract and Tobi the performances of the generated potentials in the test set are closer to their performance in the training set.
In figure B.3 the performance respective to the estimation of at least one near-native structure is divided into the complexes difficulties after their classification in the benchmark 4.0 [32]. It can be seen that for the medium and hard cases the performance of the step potentials is much better than the performance of Attract in the test set and still little better for the training set. Furthermore, it can be regarded that mostly the performance for the medium and hard cases in the test set lags behind their performance in the training set. Due to the fact, that all step potentials show the same differences between the training and the test set for the medium and hard cases but not for the rigid-body cases, it might be concluded that these complexes in the test set are harder for prediction of a near-native structure from unbound docking. The close performance for the rigid-body cases in the training and test set can might show that the step potentials are not overfitted on the training set.

To gain further insight into the scoring of the different scoring functions and

to regard the performance for the small scoring problem, the average fraction of near-native structures in the decoy sets is plotted in figure 3.2 for the top 0,1 %, top 1 %, top 5 % and top 10 % of the generated decoys. All step potentials increase the average fraction of near-native structures by a factor 2 compared to Attract in the training set in the top 5 % of all decoys. For the test set this enrichment of near-native structures is even higher and up to a factor 3 in the top 5 %. Thus, up to 64 % of the near-native solutions will be on average in the top 5 % in a the rescored training set compared to 37 % if they are scored by Attract. In the test set on average 51 % stand against 13 % in the best scoring 5 %.
This average fraction of near-native structures is nearly equal for all difficulties in the training set as it can be seen in figure B.4 in the appendix. Also the average fraction of the rigid-body cases in the test set reaches the same performance as the these cases in the training set. Only the hard and medium cases in the test set lack far behind their counterpart in the training sets. Nevertheless the improvement compared to Attract and Tobi in the test set is immense also for these structures. Furthermore, the potentials 'LinReg_gaa_6_10*' and 'LinReg_gaa_10*' perform a little worse in the test set than the potentials generated by Monte Carlo annealing due to a lower fraction of near-natives for all three difficulties. This behaviour might indicate that linear regression tends to overfit a little on some structures in the training set.

In total, the creation of step potentials from the normalized average of parameter sets from 5-fold crossvalidation shows sufficient results. All six generated step potentials perform nearly equal to find a near-native structure in the sets. Furthermore, they enrich a subset of the top 5 % of decoys with near-native solutions by a factor 2 for the training set and a factor 3 for the test set in comparison to Attract. The step potentials showed a similar performances for the atomistic and the coarse grained representations. It seems as if the description of the proteins, the number of steps and the method to train the parameter does not influence the scoring performance for structures from unbound docking much.
Furthermore, it can be seen that the improvement for the ability to predict a near-native structure and the ability to estimate many near-native structures in the decoy set just correlate little. The generated step functions increase the fraction of near-native structures in a subset dramatically but increase the probability to predict a near-native structure slightly. Thus, it may be concluded that the scoring by the step potentials using a total range of 10 Å for their definition of contacts, is not very selective and hence it is possible to find a large number of diverse near-native solutions.

a: Training set



b: Test set

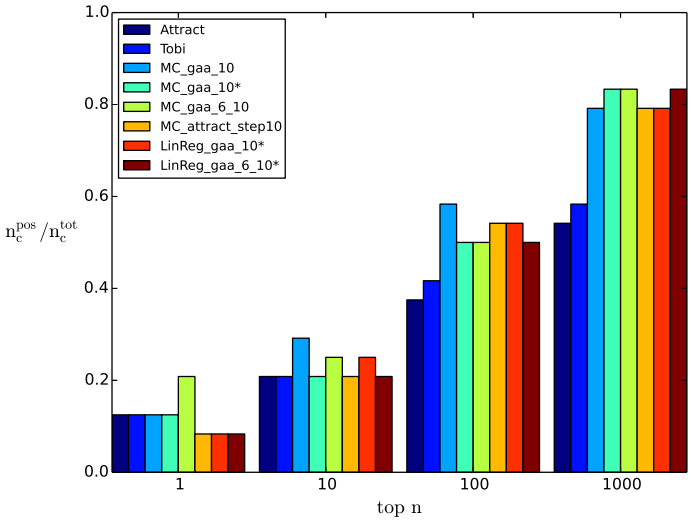
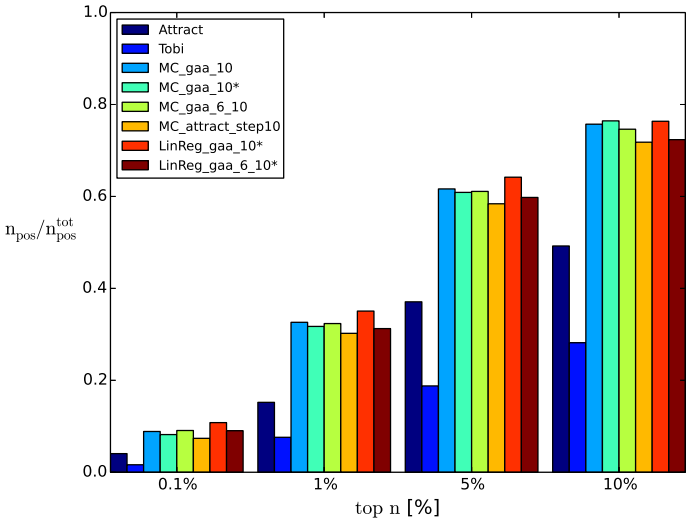


Figure 3.1: Small scoring Problem I: The probability to predict a near-native structure in a) the training and b) the test set at the given ranks is shown for the generated potentials and compared to the scoring by Attract and Tobi. On the ordinate the fraction of complexes for which a near-native structure can be observed is plotted for their position in the decoy set.

a: Training set



b: Test set

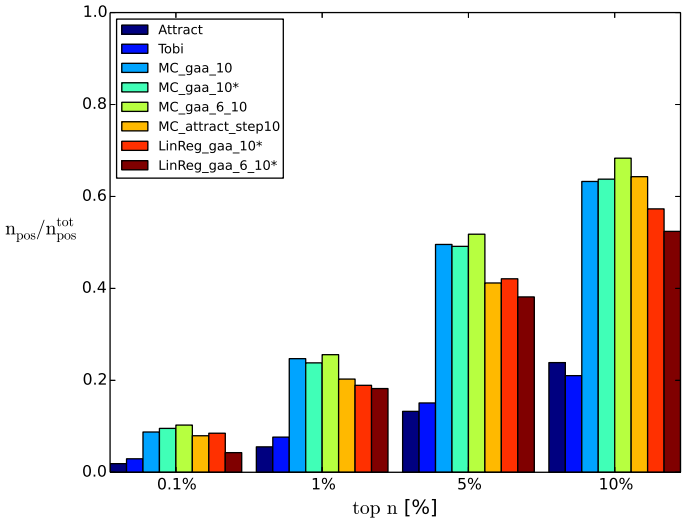


Figure 3.2: Small scoring Problem II: The average fraction of near-native structures in a subset is shown for the generated potentials in a) the training set and b) the test set and compared to the established scoring functions of Attract and Tobi. On the ordinate the average fraction of near-native structures in the decoy sets is plotted for the fraction of all decoys in the set.

Scoring of the Native Structure

In figure 3.3 the fraction of complexes for which the native structure was predicted in the top 1, top 10, top 100 and top 1000 decoys in (a) the training and (b) the test set is plotted. It is particularly striking that the scoring of the native solution with Tobi outperforms all other potentials. Tobi predicts the native structure for 96 % of all complexes in the training set and for 88 % of all complexes in the test set on rank 1. On the one hand, this might result due to the bad score of the decoys from unbound docking with Attract and on the other hand due to the very good scoring of the native structures.
As mentioned above, the step potential of Tobi uses two steps in between 0-4 Å and 4-6 Å. These ranges act on the assumption of well resolved interfaces which do have many close atomistic contacts which can just occur frequently in bound docking. Structures which suffer from little replacements cannot build up the native contacts in these ranges which would be necessary for a good scoring as for the native structure. Tobi was trained on docking solutions from FFT docking using only the bound forms of the constituents. Hence, it might be expected that Tobi performs very well for structures which do get very close to the native structure and build very close contacts.

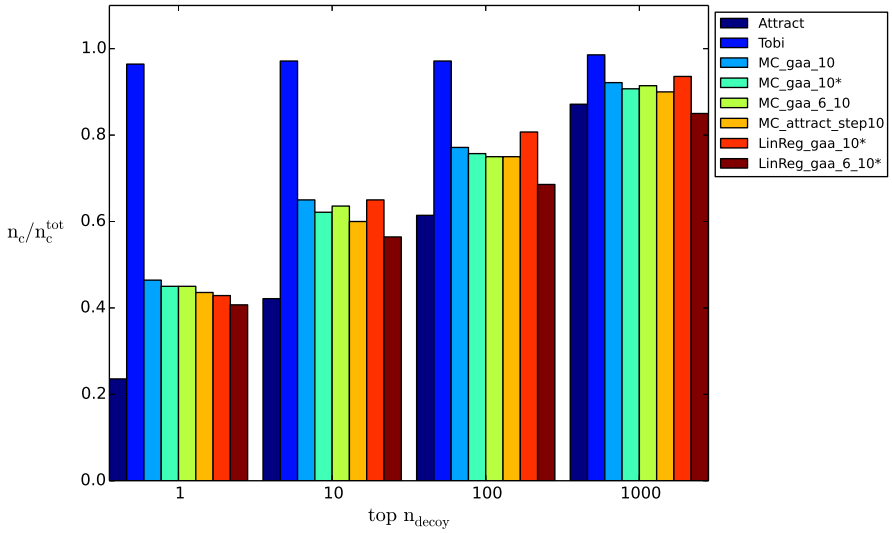
In comparison to Tobi, Attract ranks the native structure for only about 25 % of the complexes on 1 and for 42 % in the top 10. This might results due to clashes in the Attract representation which can occur by the usage of the Attract force-field without a previous energy minimization for the experimentally determined structures. Hence, its performance to predict the native structure in the top 1000 decoys is worse than finding a near-native solution. The parameters of the Attract potential are optimized for the sampling out of unbound structures. Usually, these structures cannot generate the same interfaces as bound forms and thus the parameters for the potential which describe the interaction between pseudo-atoms is not optimized for native structures but for structures which deviate a little from it.

For the step potentials it is observable that they place the native structure for 45 % of the complexes on top 1 and for about 60 % in the top 10 decoys. Also for this evaluation of the scoring performance for the native structure, the different step potentials do not show striking differences. In fact, the performance for the scoring of the native structure shows equal results for the test and the training set. This result might be interpreted as that the hard and medium complexes in the test set suffer from the sampling algorithm which is not sufficiently able to generate enough near-native structures with native contacts which can be predicted by a scoring function.
Although the generated step potentials were not trained on the native structures,

these potentials are able to distinguish them for a sufficient portion of complexes. However, the generated wide range step potentials are not as selective as the short range step potential by Tobi. This leads to the assumption that the contacts of the generated near-native solutions resemble the contacts of the native structures in the range of 10 Å. However, to obtain a scoring performance as with Tobi, the contacts in the range of 10 Å might be either to diverse between the complexes or the parameter of the step potentials do not account for particular native contacts strong enough.

The differences in scoring of near-native structures and the scoring of native structures between Tobi and the generated step potentials adumbrates that the scoring characteristics of step potentials is dependent on their range and on the structures which were used for training. Short range step potentials seem to have an advantages for the scoring of very close near-native solutions whereas long range step potentials account for the diversity of insufficient resolved interfaces for the versatile near-native solutions from unbound docking.

a: Training-set



b: Test-set

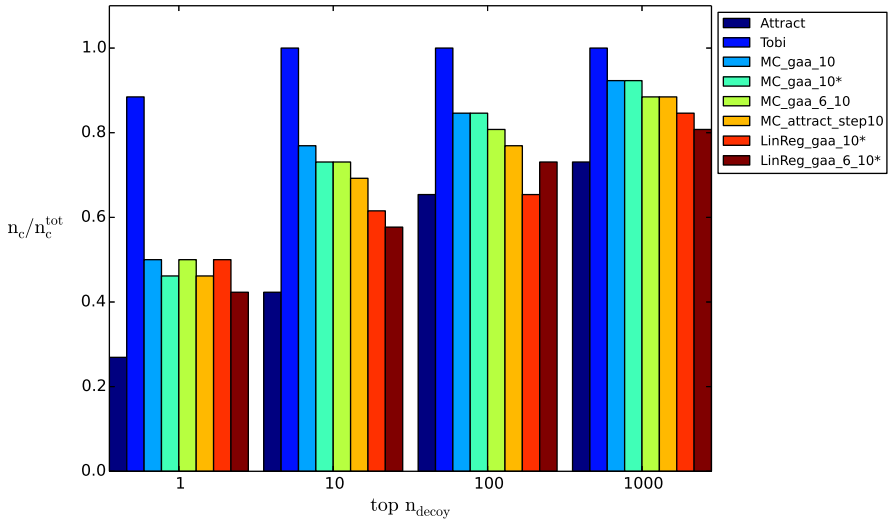


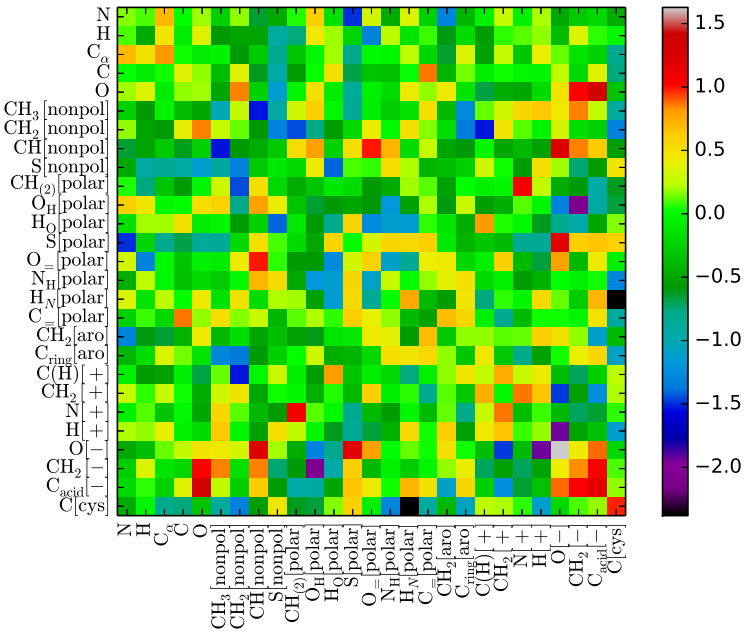
Figure 3.3: The big scoring problem: The probability to predict the native structure in a) the training set and b) the test set at the given positions is shown for the generated potentials and compared to the scores of Attract and Tobi. On the ordinate the fraction of complexes for which the native structure can be observed is plotted for their position in the decoy set.

Scoring Analysis of Step Potentials in the GAA Representation

To analyse which contacts contribute most to the difference between the scores of the native, the near-native and the incorrect complex structures, the parameters for each contact type are multiplied by the average number of contacts for each of the three types of structures in the benchmark. The 20 most negative and 20 most positive contributions are plotted in figure 3.6 and B.2 for the potentials 'MC_gaa_10*' and 'LinReg_gaa_10*' respectively. The average contributions are given for the native (blue), the near-native (green) and the incorrect (red) structures. They are sorted after the difference between the incorrect and the near-native contributions.
Large parameters with a low average number of contacts do not influence the final score strongly. On the other hand, contacts with backbone atoms are very frequent and already low negative parameters will seem to have a large impact on the score. However, these contacts might be present in incorrect structures to the same amount which can be seen as an offset to the scores.
The contribution of each contact for incorrect, near-native and native structures shows which contacts contribute most to differentiate between these structures. Thus, the contributions make it possible to determine the most important contacts for each scoring function. Thereby, the scoring performance of the scoring functions can be evaluated in detail and differences and equalities between different functions can be detected. From the contributions characteristics of scoring functions also insight into the composition of the interfaces of their training structures might be gained.
However, to distinguish the influence of the average number of contacts and the parameters to the contributions of the score, the parameters are shown for both potentials in the figures 3.4 and B.1 and the average number of the contacts for the native structures (blue), the near-native solutions (green) and the incorrect decoys is divided by the total average of each contact type and the 40 most deviant types of the near-native from the incorrect structure are shown in figure 3.5.

By the regard to both matrices of the parameter values for the two step potentials, it becomes visible that the parameters of 'LinReg_gaa_10*' deviate more between each other than of 'MC_gaa_10*'. The parameters of 'LinReg_gaa_10*' favour three to five times more strongly contacts between polar sulphides with themselves and disfavour contacts between nonpolar sulphides with themselves. Furthermore, contacts between nonpolar C-groups are preferred as well as contacts between nitrogens from polar residues with the sulphide of cysteine.

Figure 3.4: Parameter matrix 'MC_gaa_10*'.

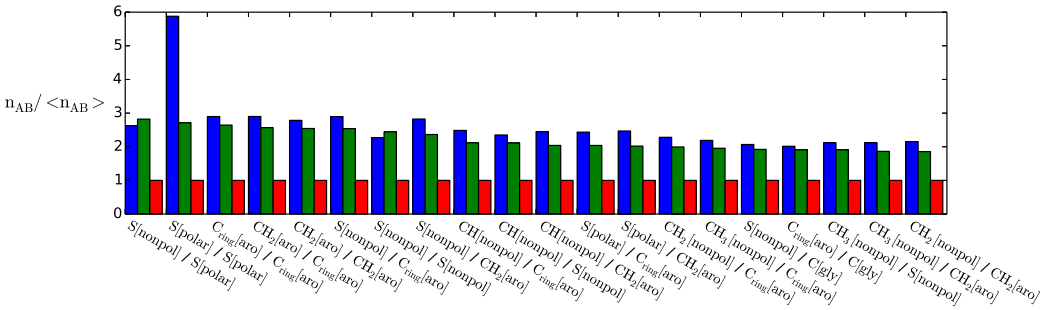


The parameters of 'MC_gaa_10*' seem to be more balanced. The most favoured contacts are between the oxygen of negative charged amino acids and the hydrogen of positive charged residues, the side chain atoms of negative charged with a oxygen of polar amino acids and the C_α of glycine with a polar NH_2 group. Disfavoured contacts seem to be contacts between amino acids of the same charge and contacts between the backbone oxygen with negative charged amino acids.

In figure 3.5, the average numbers of contacts of incorrect, near-native and native structures are divided by their total average of this contact type. It shows that especially nonpolar and aromatic contacts are more frequent on the interface of near-native and native structures than on average. No contact with the backbone can be found in the set of the 20 most positively deviant contacts. The difference of contacts between polar sulphides, that means sulphide bridges between cysteines, is significant for native structures and six times higher than for incorrect solutions. On the other side, contacts between residues with the same charge do occur more frequent in incorrect solutions than in the near-natives. Furthermore, contacts of the backbone with negatively charged residues show up less frequent in near-native than in incorrect structures. Due to the fact that backbone atoms are very frequent

on the interface and almost equally distributed, the number of contacts to backbone atoms in the range of 10 Å can be interpreted as a value which determines the general occurrence of a side chain atom on the interface. Therefore, it shows that negatively charged amino acids are generally disfavoured to be present on the interface.

a: top contacts



b: worst contacts

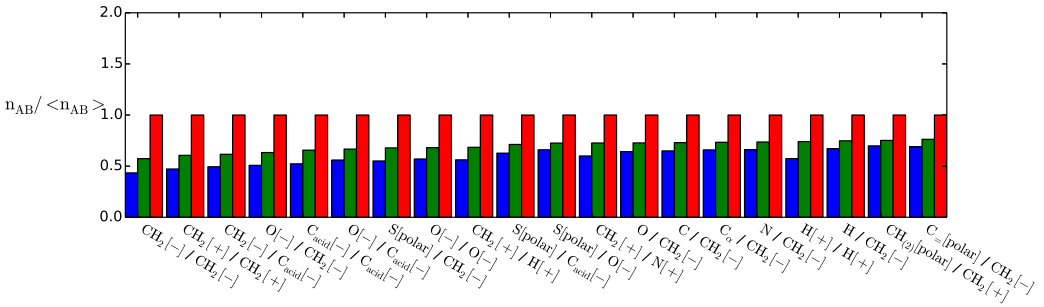


Figure 3.5: Average number of contacts in the benchmark in a range of 10 Å for native (blue), near-native (green) and incorrect structures (red) divided by their total average number.

Further insight about the scoring functions can be gained by the regard to the average contributions of each parameter to the score. The largest differences between the score of near-native and incorrect structures in both parameter sets results from contacts between parts of nonpolar amino acids with aromates and between each other.
Although the average numbers of contacts for most of the charged residues is larger for incorrect structures, some of these contacts possess a large negative contribution in both scores due to their parameters. For the scoring by 'LinReg_gaa_10*' this

difference between tendency of the average number of contacts and the sign of the parameters occurs even more frequent. One of the largest positive contributions was assigned to contacts between aromatic carbons and the backbone nitrogen although this contact occurs twice as much in near-native structures.
As mentioned above, contacts formed by backbone atoms might represent numbers which contain information about the interface composition. Although most of the contacts with the backbone do not show up in the figure of the most deviant contacts between near-natives and incorrect structures, the parameters of the scoring functions give some of them a large average contribution. Both scoring functions account positively for the presence of aromatic or nonpolar residues on the interface. Nevertheless, 'LinReg_gaa_10*' also classifies contacts of backbone atoms with aromates as a negative influence whereas these contacts are given a positive influence for 'MC_gaa_10*'. Furthermore, both scoring functions assign a large positive influence to contacts of charged residues with the backbone, although the average number of these contacts is higher for incorrect structures. This is especially true for atoms of positive residues. For these contributions, the average number of contacts might draw a false picture for these atom types. They might be important for some near-native structures or the the definition of the atom types might be insufficient and hence the training algorithms define negative parameters to them.
The contact between the aromatic ring atoms with the nitrogen of the positively charged residues plays also a main role in both parameter sets. Chemically this contact seems not be reasonable but due to the fact that both atom types seem to be favoured on the interfaces of near-native structures, a contact between these two residues in a range of 10 Å can be defined as a characteristic of near-native structures. On the other hand, disfavoured contacts of 'MC_gaa_10*' seem to consist mostly of at least one charged residue. The largest negative influence results mostly from backbone atoms with charged residues which were already shown to be most deviant between incorrect and near-native structures.
'LinReg_gaa_10*' on the other side, respects also some of these contacts between the backbone and atoms of charged residues but assigns also positive parameters to many contacts which occur more frequent in native structures. This might be due to the the creation by linear regression. This procedure targets not implicitly the scoring improvement of near-native structures but fits certain scores to the number of contacts. Thus, many structures with different outcomes have the same number of contacts of a certain type. In general, outliers are overestimated in an ordinary least squares fit and thus they might change the parameters more towards a direction which is not represented by the average of near-native structures.

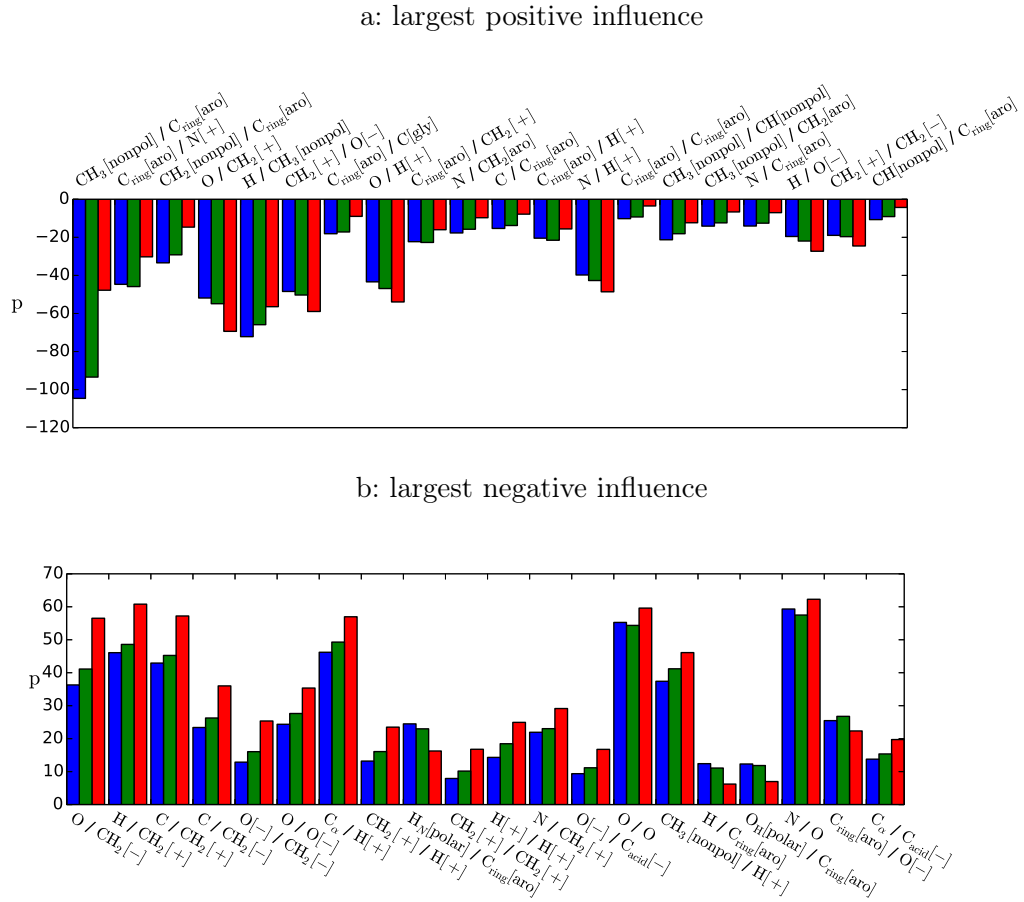


Figure 3.6: Contributions 'MC_gaa_10*': The parameters are multiplied with the average number of their contacts for the native (blue), the near-native (green) and the incorrect structures (red) to regard the most negative (bottom) and most positive (top) influencing contact types on the score.

To sum it up, the contacts of near-native and native structures are mainly formed between atoms of hydrophobic residues. Incorrect structures possess more often contacts between atoms of residues with the same charge or between negative charged residues and backbone atoms.
Both sets of parameter account favourable for contacts between atoms of hydrophobic residues, especially aromates. Furthermore, the large contributions of contacts with backbone atoms show that the scoring functions base their scoring also on the presence of some atoms and hence also on the composition of the interface.

In general, they favour the presence of atoms from hydrophobic residues and end groups of positively charged residues. 'MC_gaa_10*' disfavours mostly contacts between atoms from charged residues of the same charge or with the backbone. Also 'LinReg_gaa_10*' assigns positive parameters to some of these contacts but also to contacts which are more frequently represented in near-native structures.
The scoring by these long range step potentials seem to consist of a portion which accounts for the exact alignment of the protein surfaces to each other and a portion which regards their general composition based on backbone contacts.

3.2 Scoring based on Buried Surface Area

The total buried surface area (BSA, section 2.5) can be seen as a trivial scoring function which accounts for the geometric alignment of the interface by its size. On the assumption that native complexes are well aligned at the interface and thus tend to have larger buried surface areas than incorrect structures, the BSA can serve as a scoring function. [10, 3]. Alternatively, the BSA can be considered per atom type, which is justified by several studies that show that interfaces of protein complexes contain well conserved residues [38, 35, 64]. As mentioned before, this is essentially a description of the solvation energy, which is a weighted sum over the product of the BSA of each atom type bsA_α multiplied by a weight σ_α as shown in equation 2.9.

In this section, two unweighted and three weighted potentials based on BSA are evaluated. The two unweighted potentials are all-atom total BSAs calculated by different probe sizes. Usually, the BSA is calculated by the rolling probe algorithm which uses a sphere with a radius of 1.4 Å for its computation. To account better for slight interruptions on the interfaces of incorrect structures, the BSA is also calculated with a probe of radius 0.8 Å. The scoring by the total BSA calculated by a rolling probe of size 1.4 Å is named 'bsA_wrad_1.4' and by a probe of size 0.8 Å 'bsA_wrad_0.8'.
The weighted BSA potential 'MC_gaa_bsA' trained its parameters for the 27 atom types of the GAA representation (section 2.3.1) by Monte Carlo annealing and 'MC_attract_bsA' for the 32 pseudo atom types of ATTRACT. For the calculation of the BSA of the coarse grained model of ATTRACT, the van der Waals radii, which were mentioned in Zacharias 2003 [69], were used as the size of the pseudo atoms. Finally, to compare the training protocols, the potential parameters of 'LinReg_gaa_bsA*' for the atomistic BSA's of the GAA model were trained by linear regression. As well as for the step potentials from linear regression, the regression worked better for a training set which was enriched by all near-native structures from the decoy set, and thus '*' is appended to its name.

The Monte Carlo Annealing and the linear regression protocol for the generation of each parameter set are described in detail in section 2.7.3 and 2.7.4 respectively. The final parameter set was taken from the normalized average of the parameters from 5-fold crossvalidation.

Scoring of Near-Native Structures from Unbound Rigid-Body Docking

In figure 3.7 the portion of complexes for which a near-native structure can be predicted, is plotted against their rank to estimate the probability of the scoring functions to predict a near-native structure after rigid-body docking by ATTRACT. Figure B.6 shows this performance for each class of difficulty.
It can be seen that the two terms for the total buried surface area perform worse than the compared scoring potentials. They rank a near-native structure on rank 1 for only one complex in the whole benchmark. Nevertheless, 'bsA_wrad_0.8' finds near-native structures for the same portion of complexes as the other scoring functions in the top 1000 and 'bsA_wrad_1.4' gets close to that performance. In the test set, both BSAs predict near-natives for even more complexes in the top 1000 than Attract and Tobi. By the comparison between the total BSA terms, it seems that 'bsA_wrad_0.8' is more selective than 'bsA_wrad_1.4' due to the fact that it is able to predict near-natives for more complexes in the benchmark.

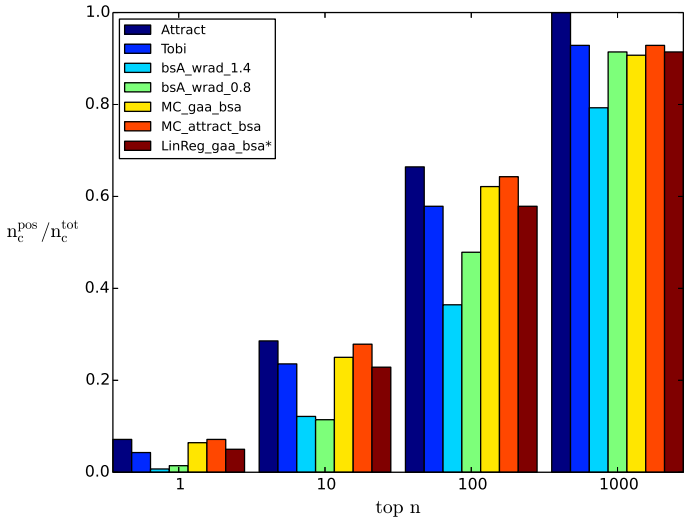
The BSA-potentials show about the same probability to predict a near-native structure as Attract and Tobi in the training set and show a better scoring in the top 1000 in the test set. Due to the separation of the buried surface areas they are more selective than the total BSAs.
For the rigid-body cases in figure B.6, the performance of the potentials in the test set reaches nearly the same probability as in the training set. For the hard and medium cases in the test set, the performance shows worse results but each of the BSA-potentials predicts near-natives for a larger portion of complexes than the scoring with Attract or Tobi.

The figures 3.8 a) and b) show the average fraction of near-native structures in the subsets of the best 0.1 %, 1 %, 5 % and 10 % of the decoy sets. The scoring by 'bsA_gaa_wrad1.4' and 'bsA_gaa_wrad0.8' predicts just about 35 % of all near-native structures in a subset of 10 % of all generated decoys in the test and the training set. Thereby, they estimate less near-native structures as Attract in the training set but more in the test set.
The BSA-potentials place on average about 60 % of all the near-native solutions for a complex under the top 10 % of all decoys in the training and the test set. Thus, these potentials find twice as many near-native structures as Attract and Tobi for each subset in the test set. Also in the training set they are able to increase the

fraction of near-native solutions of approximately 30 % compared to Attract.
Furthermore, an enrichment of near-native structures in comparison to Attract and Tobi can be observed for all classes of difficulties for the test and training set shown in figure B.7. Even in the test set, the differences between the average fractions for the medium and hard cases to the rigid-body cases is slight for the total BSA and the BSA-potentials. This cannot be taken for granted due to the fact that the buried surface areas for complexes with a large conformational change might show irregularities for decoys from unbound docking. Consequently, scoring for these cases might suffer due to problems from the estimation of the BSAs.

To sum it up, both terms for the total BSA are only able to place 35 % of the near-native solutions in the top 10 % in both sets. Nevertheless, it could be shown that the calculation of the total BSAs with a smaller probe size improves the selectivity for scoring, due to the fact that probably incorrect structures tend to have worse aligned interfaces which can be detected by the smaller probe.
The BSA-potentials are even able to place about 60 % of all near-natives in the best 10 % in both sets. The performance of the step potentials does not outnumber the performance of these potentials much in the test set but with an average fraction of up to 75 % in the training set. Due to the fact that the atomistic BSAs have only 27 and 32 parameters respectively, this result seems very satisfactory.
On the other hand, the scoring from the BSA-potentials shows no improvement for the identification of a near-native structure compared to Attract. Hence, it seems as if the exclusive scoring on the composition of the interface is very sufficient to enrich a subset of decoys with near-native structures but it is not sufficient for the prediction of a near-native solution. This might be due to the fact that the atomistic BSAs cannot separate the areas of the receptor and the ligand, neither distinguish their overlap, so that selectivity for a near-native structure will suffer from this impreciseness in the characterisation of the alignment. On the other side, they account well for the diversity of near-native structures due to these unspecific characteristics which they use for scoring.

a: Training-set



b: Test-set

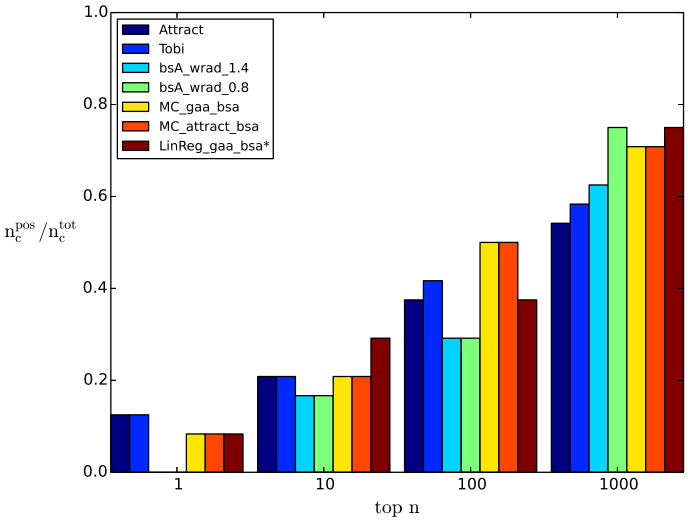
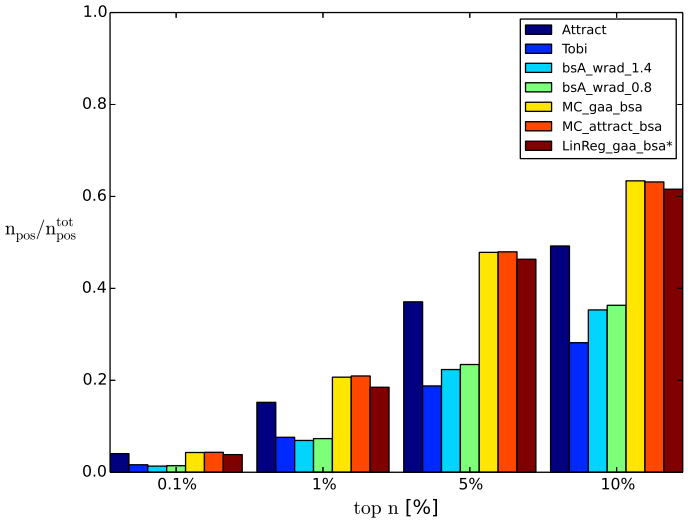


Figure 3.7: Small scoring Problem I: The probability to predict a near-native structure in a) the training and b) the test set at the given ranks is shown for the generated potentials and compared to the scoring by Attract and Tobi. On the ordinate the fraction of complexes for which a near-native structure can be observed is plotted for their position in the decoy set.

a: Training-set



b: Test-set

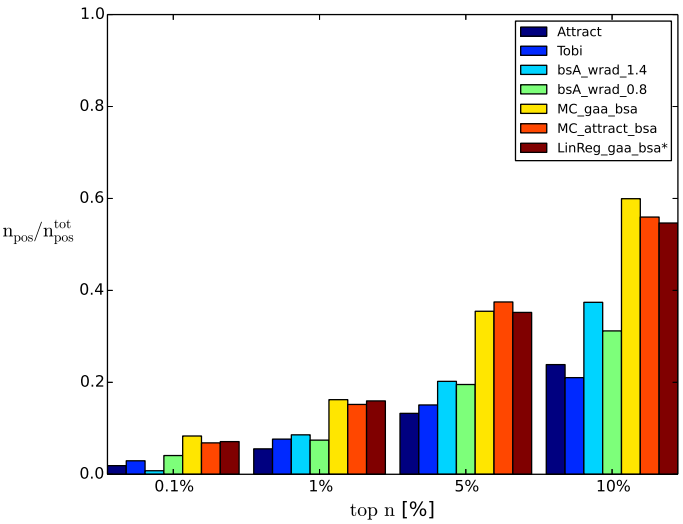


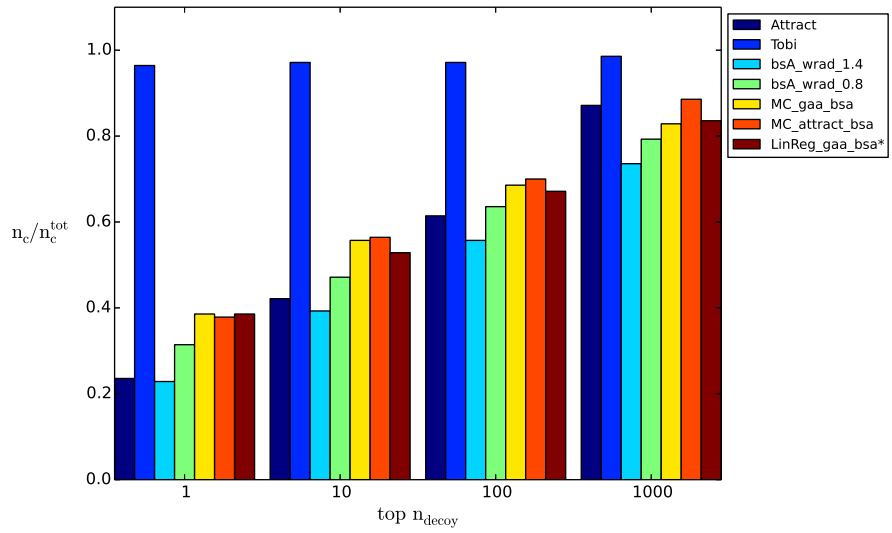
Figure 3.8: Small scoring Problem II: The average fraction of near-native structures in a subset is shown for the generated potentials in a) the training set and b) the test set and compared to the established scoring functions of Attract and Tobi. On the ordinate the average fraction of near-native structures in the decoy sets is plotted for the fraction of all decoys in the set.

Scoring of the Native Structure

The scoring of the native structure in the decoy set is regarded for the BSA-potentials in figure 3.9. The scoring by 'bsA_wrad1.4' does estimate the native structure on rank 1 in 22 % of the complexes in the training set and for 53 % in the test set. The BSAs calculated with a probe size of 0.8 Å score the native structures for 31 % of complexes in the training set on 1 and 50 % for the test set. The BSA calculated with the smaller probe performs better for the top 10, top 100 and top 1000 in both sets, as well. Furthermore, the portion of complexes for which the native structure can be found does not increase for the 'bsA_wrad1.4' in the test set up to the top 1000 decoys. Thus, the total BSA of 'bsA_wrad0.8' seems to score the native structure in general slightly better than 'bsA_wrad1.4'. In combination with the results on the near-native structures, it might become visible that 'bsA_wrad0.8' accounts better for not well resolved interfaces and thus provides better properties for a geometrical scoring.

The potentials 'MC_gaa_bsA', 'MC_attract_bsA' and 'LinReg_gaa_bsA*' do already place the native structures for 38 % of the complexes on top 1 in the training set and for 50 % in the test set. In the top 1000 decoys they predict the native pose for about 85 % of the complexes in both sets which is about the same number as Attract in the training set.
In spite the fact that all scoring functions based on the BSA did not show very sufficient results for the probability to predict a near-native structure, the result for the native structures seems to be better. Especially, the probability to predict the native structure from the total BSA, is much higher than for the near-native structures.
On the other side, the scoring performance of the total BSAs for the native solutions is very close to the performance of potentials from the atomistic description. Thus, the dictating part for the scoring of these potentials derives in all likelihood from the total size of the BSA. Especially the high probability to predict the native structures in the top 10 decoys of a set seems mostly be related to the total size of the BSA. Therefore, except for the probability in the top 10, the performance of the atomistic potentials on the native structures seems to resemble their performance for the near-native structures from unbound docking. Hence, the scoring on the atomistic BSA seems not to be sufficiently selective for native solutions but is very suitable to detect many near-native solutions based on the interface composition of the decoys.

a: Training-set



b: Test-set

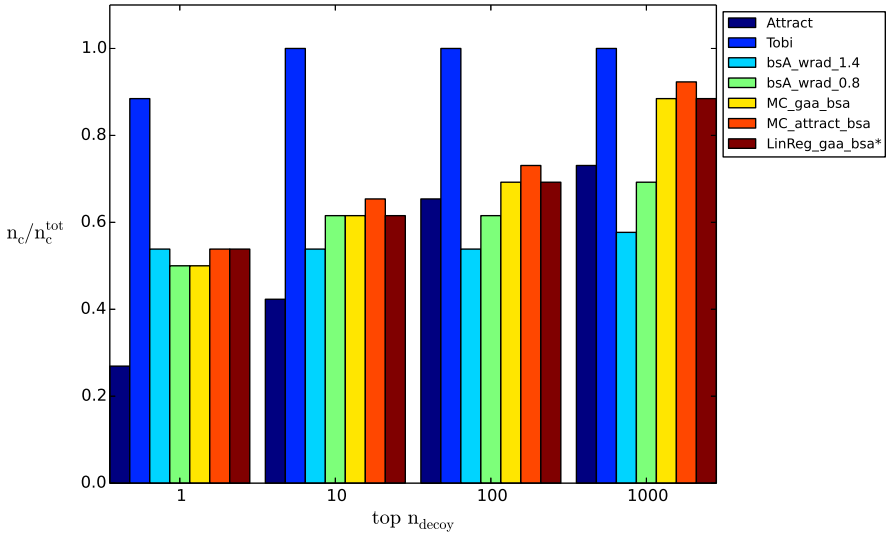


Figure 3.9: The big scoring problem: The probability to predict the native structure in a) the training set and b) the test set at the given positions is shown for the generated potentials and compared to the scores of Attract and Tobi. On the ordinate the fraction of complexes for which the native structure can be observed is plotted for their position in the decoy set.

Analysis of the Interface Composition and the Scoring of the Buried Surface Areas

The calculated atomistic BSAs for all generated decoys provide the possibility to gain more insights into the interface composition of incorrect, near-native and native complexes. The average size of the BSAs for native structures, near-native decoys and incorrect decoys is determined for the 164 complexes in the benchmark. For the error of the native structures, the standard deviation between the BSAs of the native complexes is given to estimate differences in the interface size. Instead of that, for the error of the incorrect or near-native structures the mean of the standard deviation for each complex is given to exclude mostly the influence of the different complex sizes.
It shows that native structures tend to have larger interface areas than incorrect or near-native decoys generated from unbound docking. This result might be expected due to the fact that the decoys do not possess as well aligned interfaces as the native structures due to steric barriers between the partners in the unbound form. However, the large standard deviation between the native structures of the complexes points out that the sizes for the BSAs of the complexes deviate much. Hence, the native structure might not possess for every complex a larger interface than incorrect decoys which can be build from its constituents.
However, it shows that even near-native structures from unbound docking tend to possess on average a larger BSA than incorrect solutions and thus it becomes possible to use the total BSA as a score. Nevertheless, their large mean standard deviations display as well that it might be insufficient for a distinction between them.

native:	$1682 \pm (522)\,\text{Å}^2$
near-native:	$1282 \pm 233\,\text{Å}^2$
incorrect:	$1025 \pm 268\,\text{Å}^2$

For further examinations, the average of the atomistic BSAs for the atomistic description in the GAA model for near-native, native and incorrect structures is regarded. For a better comparison, the atomistic areas are divided by their average of all decoys and sorted after the BSAs for near-native structures. In figure 3.10, the normalized average atomistic BSAs for the native structures are shown in blue, the near-native structures are shown in green and the incorrect in red. The hydrogen atoms of the GAA description do not possess surface areas and hence are excluded in the figure. The same figure is generated for the coarse grained representation of Attract which is shown in figure 3.11.

From the atomistic BSAs in the GAA model, it can be observed that the interface of near-native and native structures is enriched with aromatic and nonpolar amino acids. On the other side, native and near-native structures tend to have a smaller fraction of charged amino acids on the interface. This is not unexpected due to the fact that it was already shown above on the analysis of the contacts in the interfaces. Furthermore, the average area of sulphides from the cysteine is increased for native structures, although the areas of the near-native structures are nearly equal to the incorrect. For all the other BSAs the averages of the near-native structures are close to the averages of the native.

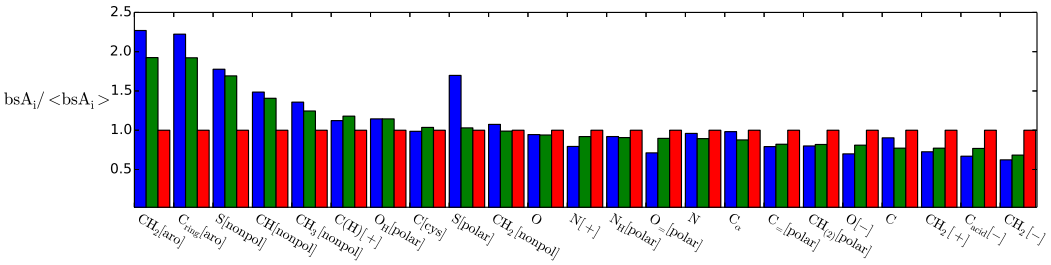


Figure 3.10: Average size of the BSAs for each atom type of the grouped-all-atom model for native (blue), near-native(green) and incorrect structures divided by their total average.

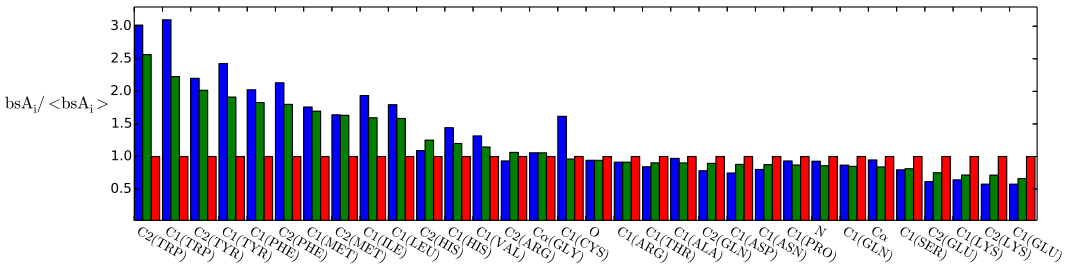


Figure 3.11: Average size of the BSAs for each type of pseudo atom from Attract for native (blue), near-native(green) and incorrect structures divided by their total average.

The results from the BSAs of ATTRACT's coarse grained representation show inevitable the same tendencies as the atomistic BSAs (figure 3.11). However, in the

coarse grained model further division between the residues can be regarded. It can be seen that the BSAs for the near-native and native structures of the three aromates tryptophane, tyrosine and phenylalanine are increased by a factor two to three compared to the incorrect structures.
Furthermore, it can be observed that not all nonpolar residues are enriched on the interface. Only leucine, isoleucine, methionine and valine are enriched whereas the average BSAs for alanine and proline resemble incorrect structures. The interface composition of the near-native structures resembles the composition of the native structures, except from the enrichment of cysteines which can only be found for native complexes. For the charged amino acids, it can be observed that histidine and arginine are little enriched in the near-native structures whereas lysine, glutamate and aspartate are disfavoured.

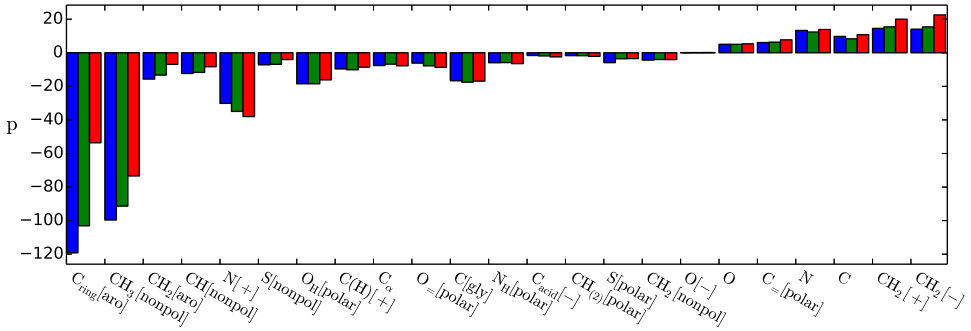


Figure 3.12: Contributions to 'Mc_gaa_bsa': The parameters are multiplied with the average size of the BSAs for each type of pseudo atom from Attract for native (blue), near-native(green) and incorrect structures.

By the regard to the most important contributions of the scoring functions which are based on the interface composition of the complexes, the values of the parameters σ_α were multiplied by the average atomistic BSAs for the native (blue), near-native (gree) and incorrect (red) structures and plotted in figure B.5 for 'LinReg_gaa_bsa*', figure 3.12 for 'Mc_gaa_bsa' and in figure 3.13 for the Attract representation.
By the comparison of the parameter contributions p from 'LinReg_gaa_bsa*' and 'Mc_gaa_bsa', it can be discovered that both potentials do mostly account for the atomistic areas of aromatic and nonolar side chain ends CH_3[nonpolar] and C_{ring}[aro]. Although the average fraction of nitrogen atoms from positive charged amino acids is higher for incorrect solutions, its parameter give it a high negative contribution. This might result from the fact that quite some native complexes contain an enriched

fraction of them whereas others do not show these atoms on the average. The contribution from polar endgroups of arparagine and glutamine (C$_=$[polar]) as well as side chain atoms from charged amino acids influences the score negative due to their positive parameter values.

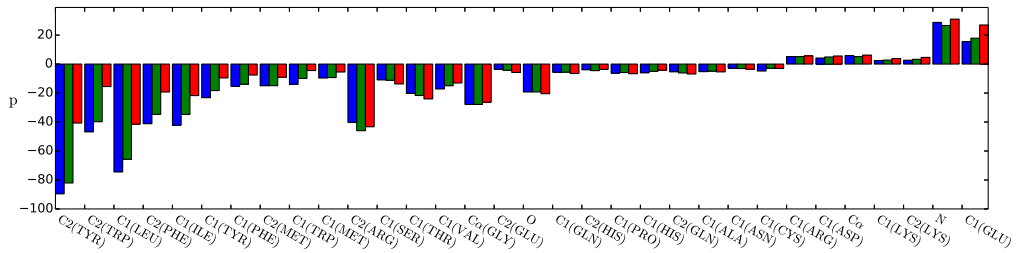


Figure 3.13: Contributions to 'MC_attract_bsa': The parameters are multiplied with the average size of the BSAs for each type of pseudo atom from Attract for native (blue), near-native(green) and incorrect structures.

The parameters for the potential in coarse grained representation of Attract base their score mostly on the contributions from tyrosine, tryptophane and phenylalanine, lecine and isoleucine. As it could already rudimentary be observed in the the atomistic contributions, the positive charged residue arginine receives a large negative contributions from its parameters although its average BSA is just a little enriched for near-native structures. On the other hand, disfavoured for the scoring are BSAs of glutamate, lysine, aspartarte, the backbone nitrogen and the C_α's.
From these similarities in the contributions to the scores of the atomistic and the coarse grained potential, the grouped-all-atom representation might be adapted. From the observation that lysine is disfavoured to be on the interface whereas histidine and arginine seem to be more preferred, it may be better for scoring to define an own group for lysine, or may group it together with the negatively charged amino acids. Also the nonpolar residues could be divided further into short and long chain amino acids with alanine, proline and valine beeing short chained amino acids. These residues seem to be neither favoured nor disfavoured and hence might also cause problems for scoring functions in the GAA representation.

To sum up the analysis of the interface composition, it could be shown that the score for near-native structures is mostly based on the size of aromatic and nonpolar BSAs. The negative contributions dictate mostly the total score whereas no strong positive counterpart can be detected. Thus, the total size of the BSA seem to be included in the score by the potentials from the atomistic and pseudo atomistic BSAs. Nevertheless, this analysis determines as well that the native structure can

only be differentiated from a near-native solution by the total size of its interface and not by the scoring based on the composition of their interfaces.

3.3 Van der Waals Potentials

Lennard-Jones (van der Waals) and Coulomb (electrostatic) potentials are popular in docking programs. They are differentiable and thus usable not only for scoring but also for sampling by energy minimization with a gradient using algorithm, such as ATTRACT.
ATTRACT uses two different Lennard-Jones potentials, a standard van der Waals form for attractive atom-atom interactions and a saddlepoint potential for repulsive interactions (section 2.5 and 2.5). Based on these two potentials, five scoring functions were created.
For the scoring functions 'MC_gaa_vdw' and 'MC_attract_vdw', based only on the van der Waals form, Monte Carlo Annealing was performed on the sum over the distances to the power of -8 plus the negative sum over the distance to the power of -6. The scoring functions 'MC_attract_saddle', 'MC_attract_saddle+' and 'MC_gaa_saddle' are based on both the van der Waals form and the saddle point form. These scoring functions were trained via spline interpolation by Monte-Carlo Annealing, and interactions were allowed to flip between the two forms (section 2.7.5). The potential 'MC_attract_saddle+' used the set of parameters from the original ATTRACT force field as its starting parameters for the annealing. All others started from random configurations.
The generated structures from sampling in the coarse grained representation can possess clashes if the proteins are represented in an atomistic form afterwards. Due to the introduction of the atomistic representation, the atoms get very close to each other, so that the distances might be located in the repulsive part of the LJ potential. For the estimation of the scoring function parameters, very close contacts of a few atoms would dominate the sums in the feature vectors and hence also the score. For that reason, all intermolecular atomic distances were set at least 2 Å before they were used for the sum in the feature vector. Also for the rescoring for the performance analysis, distances were set to be at least 2 Å to avoid scoring problems due to clashes.

It showed up that, although these potentials do not use more parameters than step potentials with two steps, they tend to overfit more easily. This might result from the fact that these potentials use the absolute distances between two atoms in contact to negative powers. Thus, the potentials might be prone to overfit on close contacts in near-native structures which dominate the score.
Therefore, the parameters of all potentials were constrained to prevent them from overfitting. Furthermore, as target function `'refine-positionlinear'` was used for

the potential 'MC_attract_saddle+' instead of `'positionlinear'` as for the other potentials. This target function multiplies the target value t_f of the target function `'positionlinear'` for each complex with the fraction of near-native structures in the top 300. For that reason, it might be able to account more for the diversity of near-native structures, due to the fact that the fraction in the top 300 has to be increased as well as the overall rank of each near-native structures. Further information on the Monte Carlo protocol for van der Waals potentials and the target functions can be found in the manual for **training-MC.py** in the appendix.

Table 3.2: Parameters for the training of van der Waals potentials: a) 'MC_attract_saddle+', b) 'MC_attract_vdw', c) 'MC_attract_saddle', d) 'MC_gaa_vdw', e) 'MC_gaa_saddle'

Training parameter	a)	b)	c)	d)	e)
power	8;6	8;6	8;6	8;6	8;6
ϵ_{AB}	[0,30]	[0,30]	[0,20]	[0,20]	[0,15]
σ_{AB}	[1,7]	[1,6]	[2,6]	[1,5]	[1,5]
T_f	refine-poslin	poslin	poslin	poslin	poslin
Grid	inter	dist	inter	dist	inter
Steps	500.000	300.000	500.000	300.000	300.000
Stepsize	0.025	0.1	0.05	0.05	0.05
Annealing	zizac	ziczac	exp	exp	exp
Start	ATTRACT	random	random	random	random

Furthermore, it was necessary to run the annealing for hundreds of thousands of steps for all the potentials to obtain an increase in the annealing curve of the validation set. In addition, for short Monte Carlo searches starting from a random set parameters, it was not possible to generate potential parameters which performed better in scoring than the original parameters from ATTRACT. In table 3.2 all the settings for the training by Monte Carlo annealing for the five van der Waals potentials are listed.

To avoid overfitting on the training set, 5-fold crossvalidation was performed. For these potentials it is not possible to build the normalized average of the parameters for their usage as scoring function due to high sensitivity of these potentials to the parameter σ_{AB} and ϵ_{AB}. Thus, it was implied that all annealing curves for the validation sets increased and the parameter set with the highest increase of the target function for the validation set was chosen as the final set.

Scoring of Near-Native Structures from Unbound Rigid-Body Docking

The scoring of the differentiable potentials is also evaluated on the probability to predict a near-native solution after a rigid-body sampling and the average fraction of near-natives in a subset. To make a fair comparison, the Attract score is divided into its score for the van der Waals interaction and its Coulomb score, named 'Vdw_Attract' and 'Elec_Attract' respectively. Additionally, the atomistic Coulomb score from the partial charges in the GAA representation are evaluated in this section, as well.
In figure 3.14 and figure 3.15, it is observable that the combination of 'Vdw_Attract' and 'Elec_Attract' in Attract improves the probability to predict a near-native structure in the decoy sets and also the average fraction of near-native structures. The combination of two scoring terms can improve the total scoring performance if both terms account for different characteristics of the structure.

It can be seen that the Coulomb term which is used in Attract from interactions between full charged side chains performs slightly worse than the score from the partial charges in the atomistic GAA representation. The probability to predict a near-native structure is a little better for the atomistic Coulomb score but it predicts on average about twice as many near-naitve structures. Nevertheless, it can be seen that both electrostatic scores do not distinguish well near-native structures on their own.

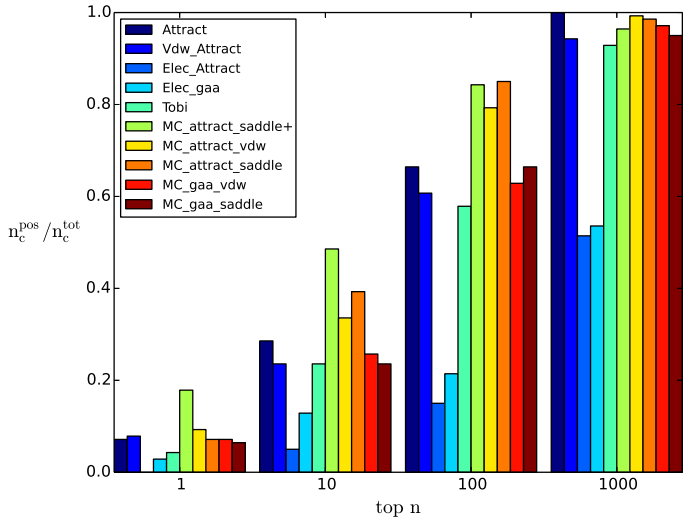
Evaluated on the probability to predict a near-native structure, the potentials from the coarse grained representation of Attract perform very well on the complexes in the training set for which they predict a near-native structure for about 80 % of the complexes in the top 100. On the other hand, the performance of the two saddle point potentials, 'MC_attract_saddle' and 'MC_attract_saddle+', does not exceed the original Attract score and is only 30 % for the top 100 in the test set. Just for the pure Lennard-Jones potential 'MC_attract_vdw' it was possible to train a scoring function which performs well for the training and the test set predicting near-natives for about 50 % of the complexes in the test set.
In figure B.8, it can be observed that the coarse grained LJ potential 'MC_attract_vdw' predicts near-native structures even for about 70 % of the rigid-body cases in the top 100 of the test set and thus seems not to be overfitted. The saddle point potentials 'MC_attract_saddle+' and 'MC_attract_saddle' on the other side, might be overfitted on the training structures due to the fact that they predict just near-native solutions for the rigid-body cases in the top 100 of the test set complexes and are not able to deal with the medium and hard complexes.
The performance of the atomistic saddle point potential 'MC_gaa_saddle' and the Lennard-Jones potential 'MC_gaa_vdw' is comparable to Attract for the training

set but slightly better for the prediction of a near-native structure in the top 100 and 1000 in the test set. The satisfactory performance in the top 100 in the test set results mostly from the good performance on the medium and hard complexes.

By the regard to the average fraction of near-native structures in a subset, it shows that the potentials in the coarse grained representation predict on average more near-natives in the training set than the atomistic potentials. Once again, only 'MC_attract_vdw' is able to enrich the test set with near-native structures compared to Attract. Nevertheless, this potential does also show slight differences between the average fraction in the test and the training set for the rigid-body cases but is able to predict a sufficient fraction of near-natives for the medium and hard cases.
The atomistic potentials predict just about the same average fraction of near-natives as Attract in the training set but places on average about 35 % of the near-natives in the test set under the top 10 % in the decoy set, so that they increase the average fraction of near-natives in the test set compared to Attract by a factor 2.5.

For all the presented types of differentiable potentials, the generation of a parameter set for a satisfactory scoring result was hard to obtain. The training parameter for Monte Carlo annealing had to be adjusted several times and constrains for the scoring parameters had to be set to avoid overfitting. Nevertheless, potentials which use saddle point description in the coarse grained form of Attract show the tendency to overfit whereas the Lennard Jones potential 'MC_attract_vdw' shows quite equal results in the test and the training set.
The overfitting might be a result from the extra parameter $i_{vor} \in [-1, 1]$ which classifies whether the potential is described by the attractive or the repulsive potential and therefore expends the parameter space further.
Sufficient atomistic potentials in the GAA representation could be generated in the pure Lennard-Jones form but also using the saddle point form. Furthermore, the Lennard-Jones potential in the coarse grained representation showed also a successful scoring performance for the training and the test set. However its scoring performance for the near-native structures in the training set seems to be even more satisfactory than of the atomistic potentials.
Due to the fact that these three potentials show quite the same fraction of near-native structures in the test and trainig set as the potentials based on the atomistic BSA, they seem to be able to account for the diversity of near-native solutions from unbound docking. Therefore, they might also represent a good choice for sampling in Attract or a refinement with iAttract. However, these potentials were not improved nor tested due to their sampling performance but might be alternatives to the original potentials due to the mentioned observations.

a: Training-set



b: Test-set

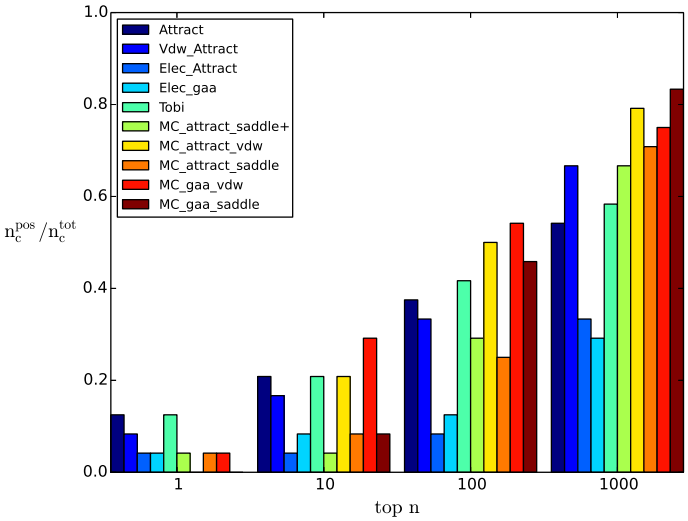
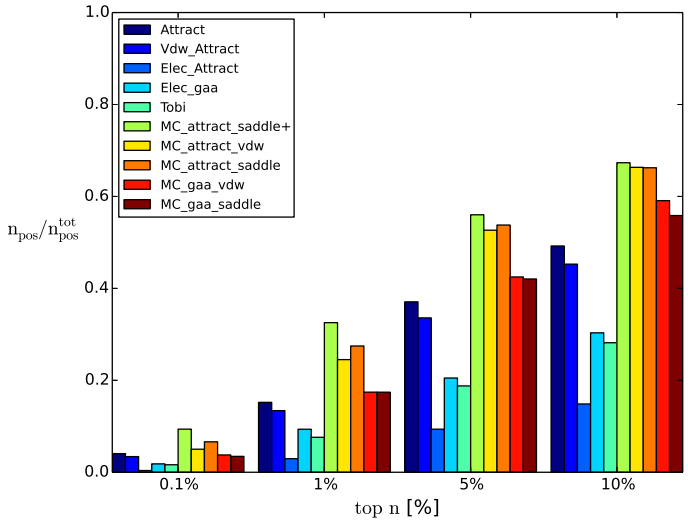


Figure 3.14: Small scoring Problem I: The probability to predict a near-native structure in a) the training and b) the test set at the given ranks is shown for the generated potentials and compared to the scoring by Attract and Tobi. On the ordinate the fraction of complexes for which a near-native structure can be observed is plotted for their position in the decoy set.

a: Training-set



b: Test-set

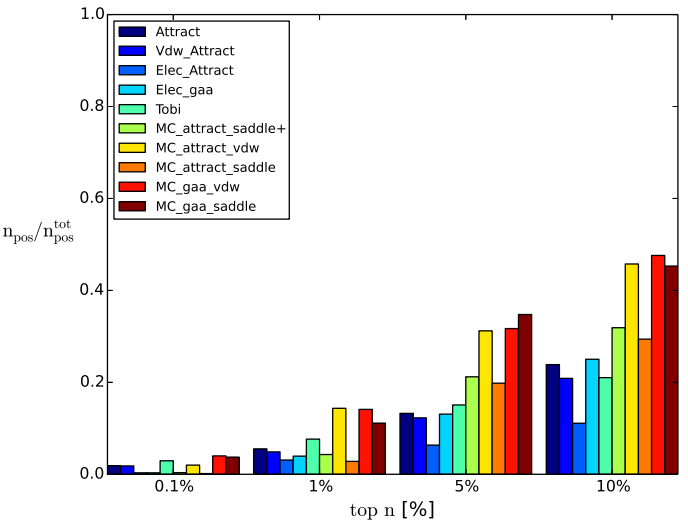


Figure 3.15: Small scoring Problem II: The average fraction of near-native structures in a subset is shown for the generated potentials in a) the training set and b) the test set and compared to the established scoring functions of Attract and Tobi. On the ordinate the average fraction of near-native structures in the decoy sets is plotted for the fraction of all decoys in the set.

Scoring of the Native Structure

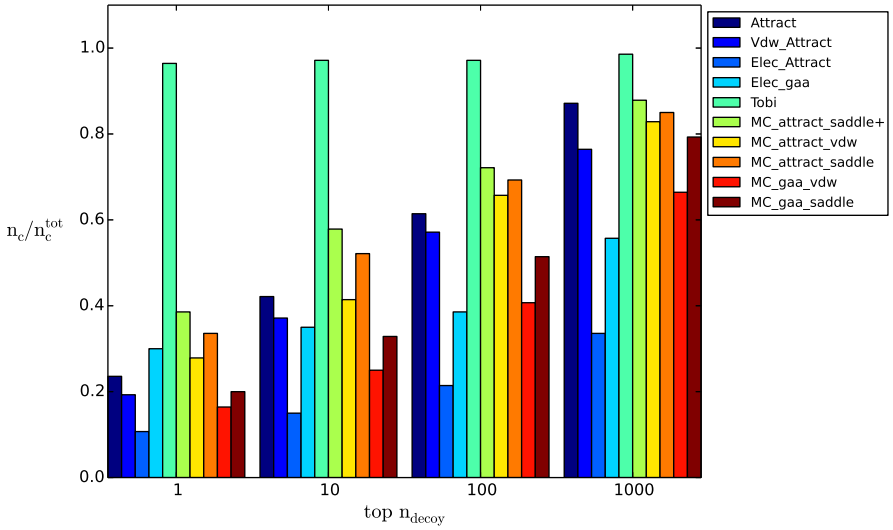
Especially for the differentiable potentials which were trained on structures from unbound docking in the coarse grained representation of Attract, it seems interesting to check the scoring of the native structures if they were generated by the sampling algorithm. Due to fact that they used structures from unbound docking, the contacts of the native structures might not be located in the minimum of the potentials for each contact type or even in the repulsive part which is referred to as a clash. Already one clash might lead to a huge shift of the score due to the very sharply bounded repulsive area of the Lennard Jones potential.

For the electrostatic scores it can be seen that the atomistic scoring performs better for the native structures than the pseudo atom representation. For the near-native structures, the scoring performance of the atomistic electrostatics was only slightly better for some complexes. Although the Coulomb interaction is a long-range potential, the better resolution of the interface seems to improve scoring of the atomistic electrostatics.
The generated potentials using a coarse-grained representation, predict the native structures with about an equal probability as the potential used by Attract. The saddle point potentials show also sufficient results for the complexes in the test set, whereas their performance deviated much between the test and the training set for the near-native structures.
The atomistic potentials struggle to identify the native structure in the training set. Their performance is nearly always slightly worse than the scoring by Attract. The generation of structures in the coarse grained representation might have caused unrealistic results for the atomistic representation. As mentioned above, the distances were set to be at least 2 Å for the training structures, which might be too short or too long for the native contacts and thus that distance is overestimated in the generation of the parameters. At least, it seems that the distances which occur on interfaces in native structures seem to deviate from the near-native structures and thus induce problems for their scoring.

Due to the scoring results of the atomistic potentials on the native structures, it can be adumbrated that saddle point and Lennard-Jones potentials become specific to frequently recurrent distances between atoms on interfaces. Especially, the change from structures in the coarse grained representation towards atomistic representation causes problems and generates unrealistic parameters from the distances at the interface. For the generation of these potentials which do not contain clashes for some type of structures, it becomes necessary to introduce a minimum distance between atoms which is also realistic for native structures or to generate the training decoys in by another algorithm. However, due to fact that these potentials favour

structures which possess contacts at certain close distances, a potential of these forms might be a sufficient approach for the prediction of near-native structures with a well aligned interface.

a: Training-set



b: Test-set

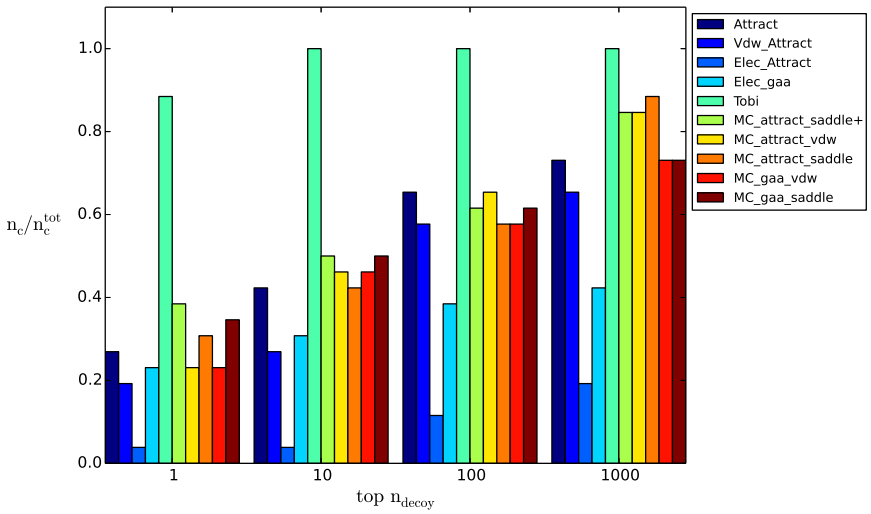


Figure 3.16: The big scoring problem: The probability to predict the native structure in a) the training set and b) the test set at the given positions is shown for the generated potentials and compared to the scores of Attract and Tobi. On the ordinate the fraction of complexes for which the native structure can be observed is plotted for their position in the decoy set.

3.4 Correlation between Scoring Functions

In the sections above it could be observed that potentials of the same type showed equal scoring performances. The protein representation and the way of training could be shown was less important. Furthermore, it might even be concluded each potential form posses a maximal scoring performance for the decoys from unbound rigid-body docking.
As it could be seen for the van der Waals interaction used by Attract and its Coulomb interaction, the combination of different scoring functions can improve the total scoring. Therefore, the combined scoring functions should account for different features of the 3D structures. This can be seen as being orthogonal which means that near-native structures having a bad score after one scoring function would be scored well by another. On the other side, it might also be helpful if false positive structures receive a bad score by another scoring function and thus their total score would decrease.
Further it should be kept in mind that the combination can improve the scoring successfully respective to small or to the big scoring problem. For the big scoring problem a combination is targeted which predicts one near-native structure for each complex at rank 1 whereas for the small scoring problem the combination which places many near-native structures in a subset is favoured.

To regard whether a combination of different scoring functions might place more near-native structures in a combined subset, an examination of the sets for the best scored near-native and best scored false positive structures can provide insights. For the examination of the sets, the union, the symmetric difference and the complement of the a) near-native structures and b) false positive structures in the best scored 10 % of the decoy set of the test set are shown in figure 3.17, figure 3.18 and figure 3.19. These analysis of the scoring functions by their sets of the structures in the best 10 % can detect similarities between scoring functions which serves as a basis for their combination.
In the matrix of the symmetric difference of these decoys sets, it can be regarded that the difference for the near-native structures between the same type of potentials is rather small. The symmetric differences of the false positive structures show the same correlations. Hence, it might be assumed that the form of the potential dictates its scoring character. On the other hand, slight differences in these subsets for each type of scoring function can still be seen in the matrix. It shows for instance, that the step potentials found by linear regression generate a slightly different subset of structures than the step potentials found by the Monte Carlo procedure. This supports the results from the analysis of most important contacts for the scoring by the step potentials. Also note worthy is that the symmetric difference between BSA-potentials, the differentiable potentials and the step potentials is not very

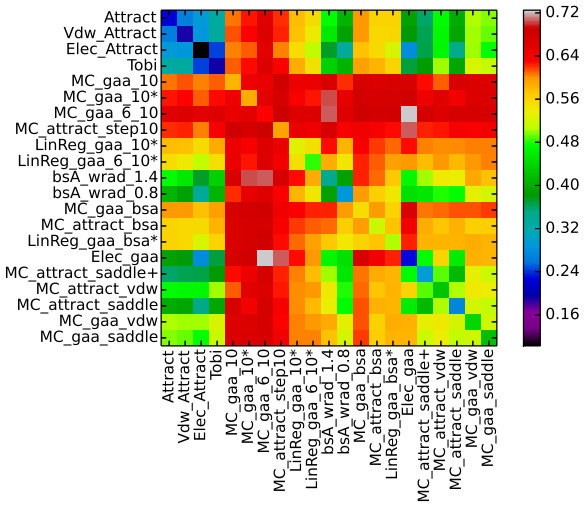
large. Thus, it seems as if these three types of potentials are slightly correlated and hence are not perfectly orthogonal.

By the regard to the union matrix of the near-native structures, possible enrichments of the subset of the top 10 % decoys with near-natives from combination can be adumbrated. In figure 3.17, it is striking that the combination of the step potentials with any other scoring function could result in finding about 65 % to 72 % of all the near-native structures in the top 10 % complexes of the test set. Also the combinations of the potential 'MC_gaa_bsA' with the differentiable potentials and atomistic electrostatics might show good results.
The union of the best 10 % incorrect structures can reach values up to 2 if the two scoring functions favour two completely different sets of incorrect structures and possesses at least 1 if they both predict the same false positives. By the regard to this union matrix it strikes out that the potentials used by Attract and the potential of Tobi determine different sets of false positives than all other potentials. Also the scoring with the atomistic electrostatic and both BSAs produce a different set of false positives. A combination with other scoring functions could increase the hit rate by setting scores of false positives down. Once again, it can be depicted that potentials of the same type seem to be correlated since they predict very equal sets of false positives. Also the determined correlation between the step potentials, the BSA-potentials and the differentiable potentials from the symmetric difference can be retrieved from the union of the incorrect structures.

The examination of the symmetric difference and the union of the decoy subsets showed some possibilities for combinations. Nevertheless, a large symmetric difference may also be the effect of only one of the two scoring functions. To regard the fraction of structures which each potential contributes to whole set, the complement matrix of the near-native solutions in the top 10 % is shown in figure 3.19. The entries of the matrix are defined as the set of estimated structures from the potential on the ordinate minus the determined structures from the potential on the abscissa. To obtain an ideal result for scoring by combinations, it is desirable that both functions contribute to the union with an equal number of structures. From that follows that the complement matrix is supposed to become symmetric with large fractions on both sides.
In the figure for the complement, it can be seen that any potential combined with a step potential does not contribute much to the overall union. The additional structures which are predicted by the combined potentials would represent just up to 10 % of all the near-native structures in the combined set. Quite the same can be shown for the BSA-potentials. From that observation, it can be derived that combinations which include one of the step potentials or the BSA-potentials might not improve the scoring successfully. However, it can also be seen that combinations

of differentiable functions with electrostatics and the geometrical BSA terms might probably come up with a larger set of near-native structures in the top 10 % of complexes.

a: near-native structures



b: incorrect structures

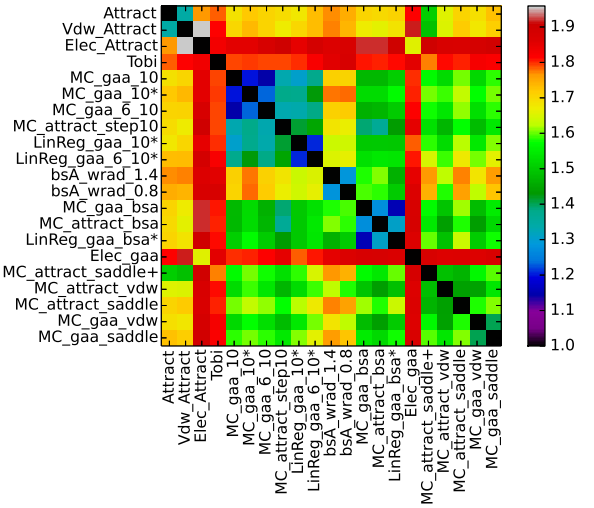
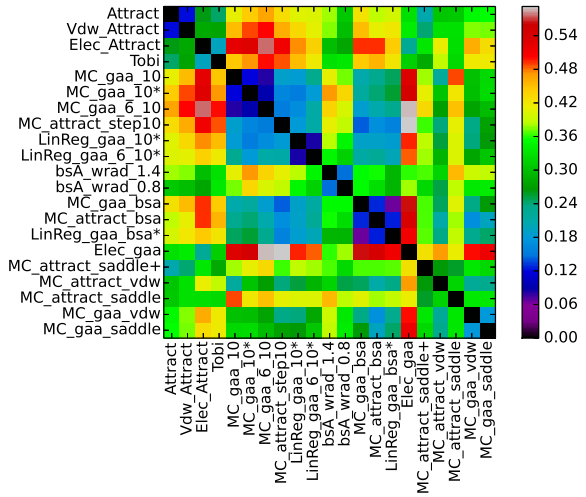


Figure 3.17: The union of the a) near-native and b) incorrect structures in the best scored 10 % of all decoys between all presented scoring functions: Shown are the unions of the sets of best scored 10 % structures of two scoring functions. A large union of near-natives illustrates promising combinations due to an enrichment of near-native structures, but also a large union of incorrect structures promises a scoring improvement due to the probable scoring functions orthogonality.

a: near-native structures



b: incorrect structures

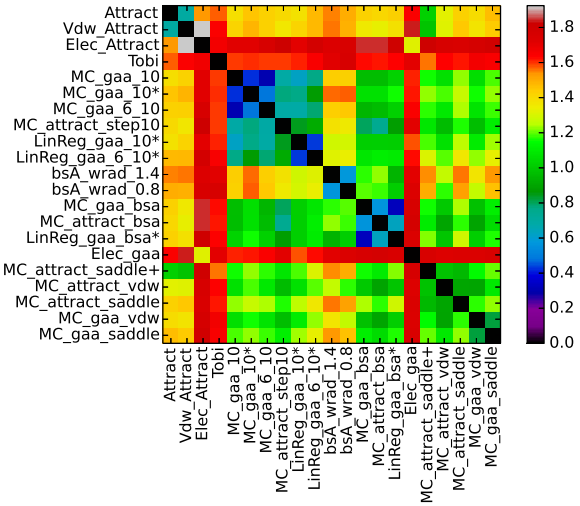
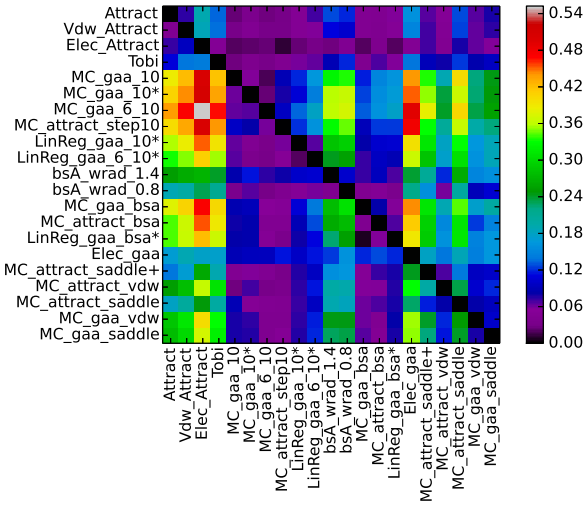


Figure 3.18: The symmetric difference of a) near-native and b) incorrect structures in the best scored 10 % of all decoys between all presented scoring functions: Shown are the symmetric difference of the sets of best scored 10 % structures of two scoring functions. A large symmetric difference illustrates that the two compared scoring functions favour different sets of decoys and so it promises a scoring improvement due to the probable scoring functions orthogonality.

a: near-native structures



b: incorrect structures

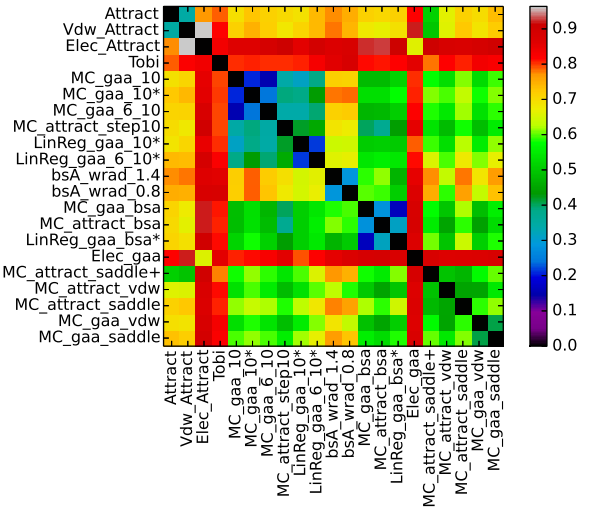


Figure 3.19: The complement of a) near-native and b) incorrect structures in the best scored 10 % of all decoys between all presented scoring functions: Shown are the complement of the sets of best scored 10 % structures of two scoring functions. The set of the scoring function on the abscissa is subtracted from the set on the ordinate. For a successful combination, the entries of the complement should be symmetric to the diagonal because then both scoring functions contribute equally to the increase in the union.

3.5 Combination of Scoring Functions

In the section above it could be shown that combinations of different scoring functions are able to improve the scoring performance if each scoring function favours different sets of structures.
A lot of works propose a linear combination of various scores for that purpose. For this reason, support vector machines are often used to train the weights between the scores based on a classification between near-native and incorrect structures. However, SVM's with a linear kernel show problems for the presented data due to the fact that the large set of structures seems not to be well linearly separable since the near-native structures from unbound docking receive very diverse scores due to their insufficiently resolved and diverse interfaces.
Earlier presented methods, that is linear regression and Monte Carlo Annealing, possess also the ability to predict weights for the combination of scores. In this work, the weights for the combination of different scores were found by an ordinary least squares fit on the Capri-stars as outcome values.
Problematical for linear regression is that due to the size and the chemical composition of the complexes, the scores of the their generated decoys possess different means and deviations. Therefore, the scores for each complex were normalized before they were used for the linear regression by setting their mean to zero and their standard deviation to one.
To avoid overfitting for the weights, 5-fold crossvalidation was performed and the average of the estimated weights for each training set were used as the final set of weights. Due to the normalization of the scores for training purposes by linear regression, the scores for each complex had to be normalized before rescoring, as well.

Respecting the results from the sections above, the combination of scoring functions from each potential type seems to be most successful, using 'MC_gaa_vdw' as atomistic differentiable model, 'MC_gaa_6_10' as step potential, 'MC_gaa_bsA' as BSA-potential, 'Elec_gaa' for the Coulomb interactions and 'bsA_wrad0.8' as an geometrical score. Out of these scores, three combinations were generated by linear regression: 'Combination_full' weights all the normalized contributions to find a final score. 'Combination_no_step' uses all scores except from the step potential to evaluate the influence of it to the complete combination. 'Combination_elec_vdw' uses just the differentiable potentials 'MC_gaa_vdw' and 'Elec_gaa' to gain further insight on the influence of the BSA-potentials. In table 3.3 the weights for the normalized contributions are shown.

Due to the normalization of the scores, these parameters represent directly the influence of each score to the combination. It can be seen that in each scoring combination

Table 3.3: Combination weights $\omega_i \times 10^3$: a) MC_gaa_vdw, b) MC_gaa_6_10, c) MC_gaa_bsA, d) Elec_gaa, e) bsA_wrad0.8

	a)	b)	c)	d)	e)
Combination_full	1.271	8.973	1.135	3.251	0.506
Combination_no_step	2.863	–	7.308	4.546	-0.732
Combination_elec_vdw	6.923	–	–	4.103	–

the best performing score influences the combinations the most. This means that for 'Combination_full' the step potential dominates, for 'Combination_no_step' the BSA-potentials and for 'Combination_elec_vdw' the LJ-potential. What is even more noteworthy is that the Coulomb interactions becomes in each combination the second most influencing term despite its insufficient scoring by itself. Although the scoring performance of the BSA is even better as the Coulomb interaction on its own, it receives very little weights and even a negative weight in 'Combination_no_step'. This might result from the fact that the BSA represents the size of the interface which is already incorporated in the total number of contacts or as been seen also in the size of the atomistic BSAs.

Scoring Performance

In figure 3.20, the probability to find a near-native structure for the presented combinations and its constituents is shown for the training and the test set. The average fraction of near-native structures for each subset is shown in figure 3.21. Looking at the probability to find a near-native structure by the scoring with 'Combination_full' in the tables A.4 and A.5, it can be seen that the performance for the training set rises slightly from 17.14 % to 18.57 % for the near-natives at top 1 and from 38.57 % to 40.0 % for the near-natives at top 10. On the other side, one can see no increase of the probability to predict a near-native structure in the test set. For the average fraction of near-native structures it is visible in tables A.6 and A.7 that the it increases slightly for the training set while it decreases slightly in the test set. Therefore, it seems as if the complete combination does not lead to a real improvement compared to the step potential 'MC_gaa_6_10'.

The combination 'Combination_no_step' excludes the step potential from its combination. Hence, from the combined scores the potentials 'MC_gaa_bsA' and 'MC_gaa_vdw' show the best scoring performances on their own. Regarding the weights for each constituent of the combination, the scoring performance of the combination mostly relies on the scoring of the BSA-potential and of the atomistic Coulomb potential. However, the probability by the combination to predict a near-

native structure ranked as top 1 increases for the training and the test set. For the training set, also the probability to find a near-native in the top 10 and top 100 is higher than the values for its constituents.
In addition, the combination increases the fraction of near-natives for each subset of the training decoys and also slightly for the best 1 % and 5 % decoys in the test set.

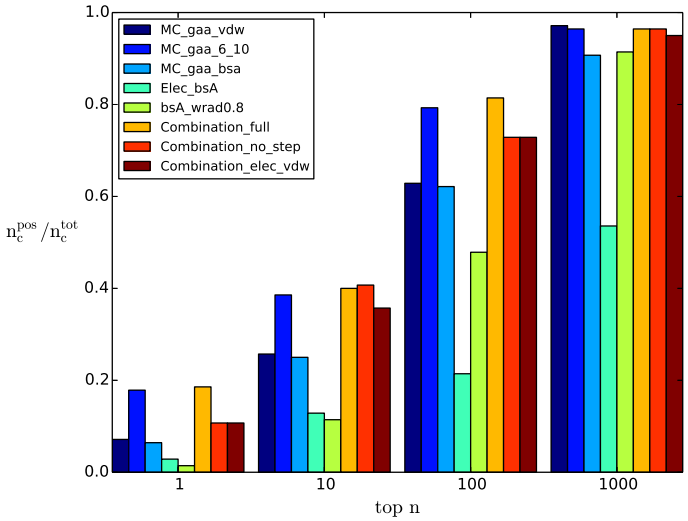
The combination of two scores from differentiable potentials, the atomistic Lennard-Jones potentials 'MC_gaa_vdw' and its Coulomb potential 'Elec_gaa', in 'Combination_elec_vdw' might also be usable for energy minimization algorithms which use gradients for their purpose. For scoring, the combination shows an increase in the probability to find a near-native structure up to the top 100 for the training set and the top 10 in the test set. Unfortunately, it shows a decrease for the probability to predict a near-native in the top 100 and 1000 in the test set compared to the LJ-potential on its own.
Furthermore, the combination showed an increase of the average fraction of near-natives in the training set and also slightly for the top 5 % in the test set.

It could be seen that the different combinations were able to improve scoring slightly. The overall improvement is not very large and mostly seen for certain subsets or positions in the sets. It seems as if the combination of scores by linear regression works a little better to improve the score of individual near-natives than for all near-natives in the decoy sets. For this reason, the effect of the combination to the scoring of the native structures is shown in figure 3.22.
For the probability to predict the native structures, the combinations show sufficient results for the test and the training set. Two out of three combinations could place native structures for more complexes as their constituents on rank 1 in the training and the test set. 'Combination_full' and 'Combination_no_step' receive a higher probability to find a near-native at rank 1 in the training set whereas 'Combination_full' and 'Combination_elec_vdw' in the test set. At rank 10, all three combinations increase the probability in both sets. Especially, the combination of the Lennard-Jones score with the atomistic Coulomb score improves the scoring performance very successfully for both sets.
To keep these results in perspective, the total gain by combination is only slight and dependent on the position in the decoy sets. Nevertheless, these combinations were trained on the scores of near-native structures but improve scoring for native structures even more. Hence, an optimization of the weights towards an increase of the probability to predict the native structure could improve this behaviour even more.

It was shown that in each combination the electrostatic contribution got the second highest weight. From this result and the observations on the sets of the

top scored 10 % decoys, it might be inferred that electrostatics account for different features of the decoys and thus the Coulomb score seems to be a reasonable addition to other types of potentials. On the other hand, the other potentials seem to be less orthogonal since they account for similar structures and hence for similar characteristics of the decoys. However, this result is only valuable for decoys from unbound docking and might be different for structures from a refinement.
Although the improvement of scoring for the near-native structures from the combination of scoring functions was little, especially for the enrichment of the fraction of near-native structures, the combination showed sufficient results for the prediction of the native structures.
As it was shown above, to place many near-native structures from unbound docking by scoring in a subset of decoys, the score is mostly related to the composition of the interface. For this purpose a scoring function has not to be very specific, in fact simple scoring functions show very satisfactory results.
Nevertheless, to receive a high probability to predict a well aligned near-native structure, scoring functions must be very specific and they have to be capable of dealing with the diversity of the interfaces from protein complexes. Due to the satisfactory results of the combinations for the probability to predict the native structures, linear combination seems to be an approach to increase the probability to predict a near-native structure in a refined decoy set.

a: Training-set



b: Test-set

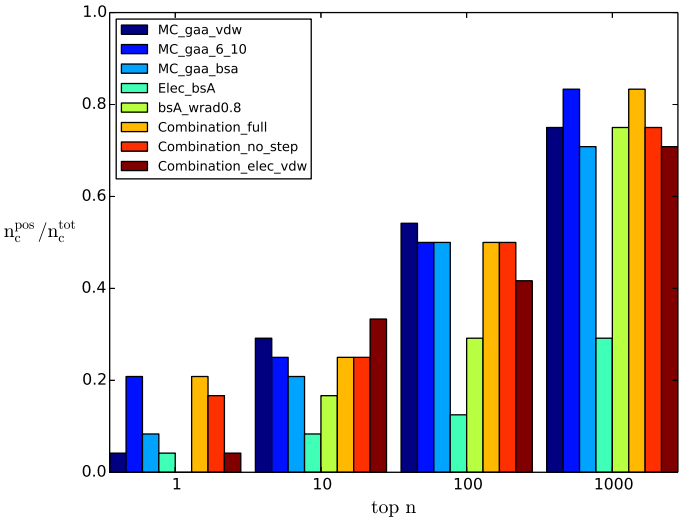
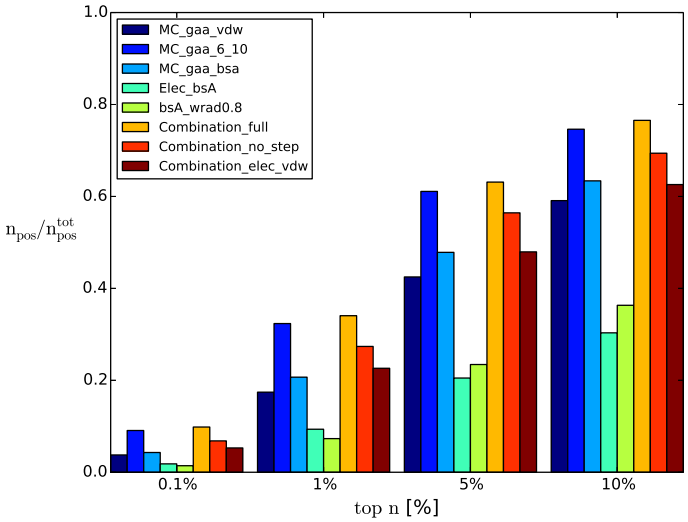


Figure 3.20: Small scoring Problem I: The probability to predict a near-native structure in a) the training and b) the test set at the given ranks is shown for the combinations of potentials and compared to their constituents. On the ordinate the fraction of complexes for which a near-native structure can be observed is plotted for their position in the decoy set.

a: Training set



b: Test set

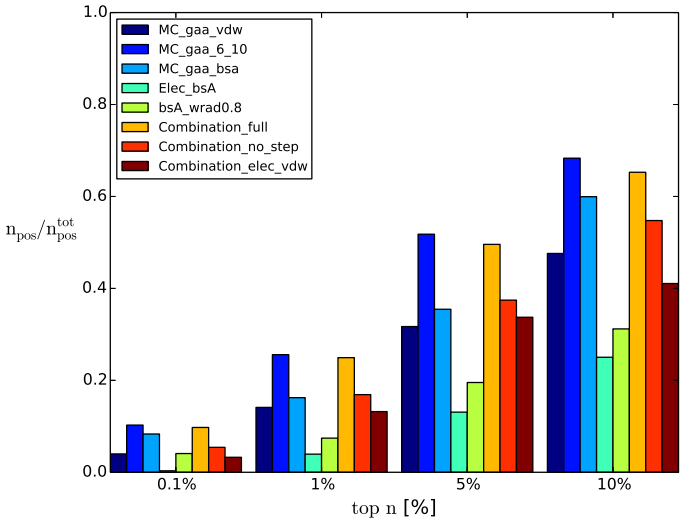
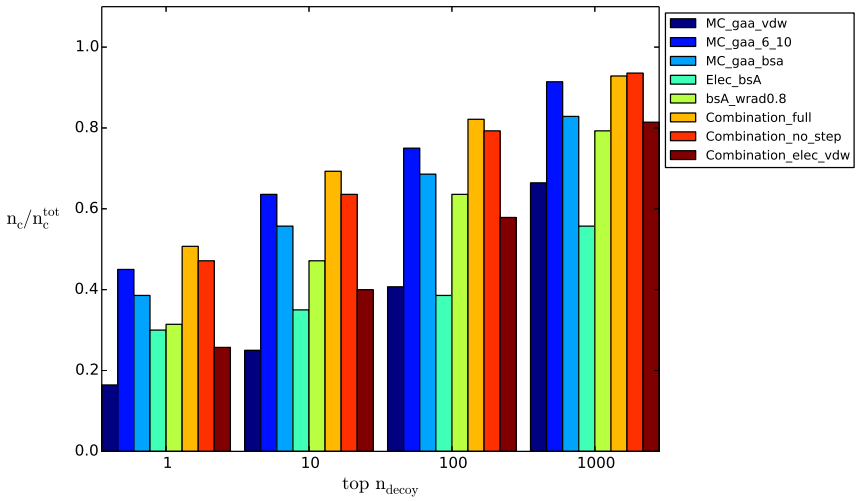


Figure 3.21: Small scoring Problem II: The average fraction of near-native structures in a subset is shown for the combinations of potentials in a) the training set and b) the test set and compared to their constituents. On the ordinate the average fraction of near-native structures in the decoy sets is plotted for the fraction of all decoys in the set.

a: Training set



b: Test set

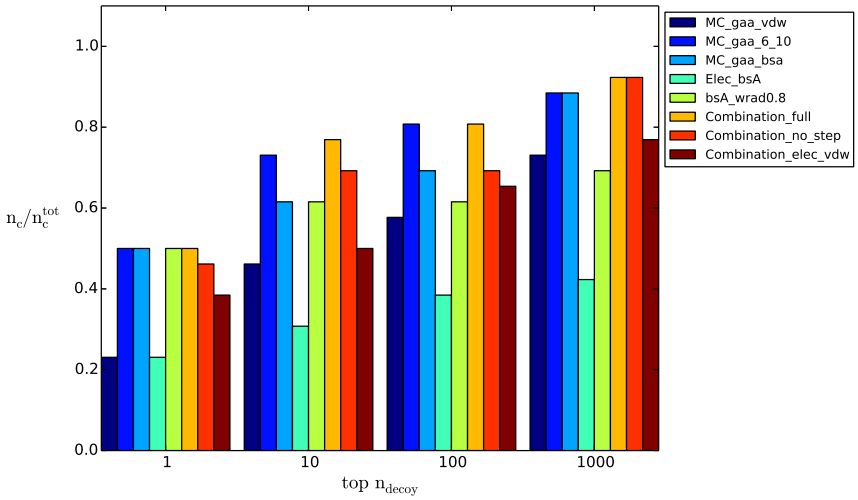


Figure 3.22: The big scoring problem: The probability to predict the native structure in a) the training set and b) the test set at the given positions is shown for the combinations of potentials and compared to their constituents. On the ordinate the fraction of complexes for which the native structure can be observed is plotted for their position in the decoy set.

Chapter 4

Conclusion and Outlook

The aim of this work was to present a methodology to create and improve scoring functions for protein-protein docking. Thus, scoring functions based on various physical potential models were created by Monte Carlo Annealing and linear regression on rigid-body docking solutions from the unbound forms of the proteins by ATTRACT. For each of the created knowledge-based scoring functions, the performance was evaluated on the probability to predict a near-native structure for a complex, on the average fraction of predicted near-natives and on the probability to predict the native structure in this decoy set. From the comparison to the scoring potentials of Attract and Tobi, it was shown that all the approaches showed sufficient results for the scoring of decoys from unbound docking and also for the native structures. All predicted scoring potentials were able to beat the Attract and Tobi score at least for the average fraction of near-natives or the probability to find near-natives in the decoy sets.

Nonetheless, simple step potentials defining contacts for distances between atom pairs up to 10 Å, outperformed all other forms of potentials. They were not only able to rank a near-native structure from unbound docking on 1 for about 20 % of the complexes in the training and the test set but also to increase the fraction of near-natives in the best 5 % of the decoys up to 60 % in the training set and up to 50 % in the test set.
It was shown that these potentials base their scoring on contacts between nonpolar and aromatic residues. Furthermore, they account for the presence of these residues on the interface by contacts to backbone atoms. On the other side, they disfavour atoms from charged residues to be on the interface except if positively charged residues form contacts with aromates.
The step potentials which were generated by linear regression showed similar characteristics as the potentials from Monte Carlo Annealing. Both methods seem to be able to estimate parameters for scoring functions with a sufficient scoring performance. Nevertheless, Monte Carlo Annealing tackles the scoring problem directly for each complex due to its target functions whereas the scoring problem has to be redrafted towards a regression problem for the usage of linear regression algorithms. However, both algorithms show correlations in their estimations due to their target

and cost-functions respectively.

BSA-potentials enrich the fraction of near-native solutions in subsets but do not show an increase of the probability to find a near-native structure compared to Attract in the training set. Nevertheless, these potentials were able to place on average 48 % of the near-native structures in the top 5 % of the training set and 36 % in the test set. The lag of specificity for a near-native structure in the decoy set might be due to the usage of the atomistic BSAs which do not include the orientation of the surface areas to each other and the small number of parameters these potentials base their score on.
The investigation of the most contributing parameters for these scores showed that interfaces with large areas of nonpolar, aromatic and positive charged amino acid end groups received a good score. From the regard of the contributions for the BSA potential in the coarse grained representation of Attract, it was further found out that phenylalanine, tyrosine, tryptophan, leucine, isoleucine and methionine are dominant on the interface. Furthermore, arginine might play a key role for only some complexes whereas lysine and negative charged amino acids are disfavoured. This observation explained the preferred contacts which were seen in the atomistic description of the step potentials, but also pointed out possible problems from the classification of atom types in the GAA model.

The creation of differentiable potentials in the form of a Lennard-Jones potential or a saddle point potential, showed some difficulties which may be referred to the usage of exact distances between the atoms for their scores. To estimate the parameters of the differentiable potentials by Monte Carlo Annealing, their range had to be constrained to avoid overfitting. However, the coarse grained saddle point potentials which separate the Lennard-Jones interactions into attractive and repulsive terms, showed performance differences between the training and the test set which could be seen as a slight overfitting. The atomistic potentials in the grouped-all-atom description lagged behind the performance of the coarse grained potentials in the training set but were performing slightly better in the test set. Only the coarse grained Lennard-Jones potential without a repulsive description showed an increase of the performance in both sets, for the fraction of near-natives and the probability to predict a near-native solution for the complexes compared to Attract. Due to the fair scoring performance of the differentiable potentials on decoys from unbound sampling, their ability for sampling and refinement might be of interest for further investigations.

Furthermore, a simple geometric scoring by differently calculated BSAs, an atomistic Coulomb interaction and a coarse grained Coulomb interaction were evaluated on their scoring ability. The BSAs were able to find a sufficient fraction of near-natives

based on the size of their interfaces. Nevertheless, its scoring suffered from the fact that not each protein complex possesses a large interface area. The BSA which was calculated with the rolling probe algorithm using a probe size of 0.8 Å instead of 1.4 Å, found more near-natives due to its ability to account for interruptions and gaps from mismatches on the interfaces of the incorrect decoys. Both Coulomb scores showed insufficient scoring results on their own. They were only able to predict a near-native structure for 50 % of the complexes at rank 1000 and place only 10-20 % of the near-natives in the top 5 % of the decyos in the training set.

Additionally, the sets of near-native and incorrect structures in the top scored 10 % of all decoys were compared by regarding their union, their symmetric difference and their complement. It could be seen from the complements of the sets of near-native structures that the step potentials predicted nearly all near-natives which were contained in the sets of the other functions. Furthermore,from the symmetric difference and the union of the false positive structures could be shown that the predicted sets from the same type of potentials do resemble each other. Consequential, it can be assumed that it is the potential shape which mostly defines which structures can receive a good score.
From the combinations of the different normalized scores by linear regression and the observations from the symmetric differences of the sets, it can be stated that the step potentials, the BSA-potentials and the differentiable potentials account for similar features of the decoys interfaces. Although the scoring of the Coulomb potential was insufficient on its own, in all combinations it received the second largest weight and thus contributed the most to an increase in the scoring performance through combination. For the combinations, an improvement in scoring showed up mostly for the training set and on the probability to predict a near-native or a native structure on rank 1.
The results showed that not even combinations of scoring functions could account better for the diversity of near-native solutions from unbound docking than the long range step potentials. In addition, the long range step potentials are even more selective than the other forms of scoring functions since they score near-native structures for 40-50 % of the complexes in the top 10. Due to the fact that they use contacts in a range of 10 Å for their predictions, they seem to possess the the ability to account for general interface composition but account also for the orientation of the interfaces to each other.

The step potential from Tobi et al. showed impressive selectivity for the native structures, scoring them for 96 % of the complexes on rank 1. This outstanding performance might be a result from its definition of contacts in a range of 6 Å and the set of training decoys which were taken from bound docking solutions. Hence, the scoring function by Tobi accounts just for interactions between atoms by very

close contacts. The quality of the interfaces of decoys from unbound docking is likely to be insufficient and thus the most important contacts for the scoring cannot be enriched on the short range as it would be necessary for a satisfactory scoring by Tobi.
That the generated long range step potentials and the short range step potential from Tobi et al. show completely different scoring characteristics may be explained by their definition of contacts and by the decoys which were used for their generation. The short range contacts, used in Tobi, allow to account for nearly exact positioning of two atoms to each other whereas the long range contacts better account for the overall interface composition by the number of contacts. Furthermore, the training on bound structures may has generated a scoring function which might base their scoring on some specific contacts which are only enriched in the close interfaces of native structures. The long range potential on the other hand, was trained on diverse near-native structures from different complexes and might account first for the enrichment of interface residues and secondary for contacts between them.

Knowledge-based scoring functions highly depend on the training structures on which they were determined. Thus, their scoring is always dependent on the sampling or the refinement algorithm respectively. Based on its knowledge about the structures which have to be scored, it might be useful to adjust the type and form of the scoring potential. Short range step potentials or differentiable potentials may be able to better account for the placement of two atoms to each other, whereas long range step potentials or atomistic BSA-potentials forgive inexact placements on the interface.
For a sufficient development of scoring functions the quality of the sampled structures has to be considered as well as the goal of the prospective scoring function. A scoring function which has to account for the diversity of interfaces of many near-native structures from unbound docking might not be able to detect the native structure with a sufficient probability in a set of well resolved decoys after refinement. Thus, the selectivity of scoring functions can be seen in contest with its ability to account for the diversity of near-native structures for each complex. The better the sampling algorithm would be able to generate very close near-native structures the more would these two qualities of scoring functions correlate. However, due to the fact that sampling algorithms do not sufficiently respect flexibilities which would be necessary to generate the native structures out of its unbound constituents, a perfect scoring function derived from the total physical binding energy cannot be developed. On these grounds, it might not be useful to develop scoring functions on bound structures to solve real docking problems. Furthermore, the scoring problem may rather be seen as a computational prediction problem than a simulation of a physical phenomenon.

An auspicious docking run would therefore consist of a rigid-body sampling followed by a refinement of the best scored 5-10 % docking decoys. To have as many near-native structures as possible in a subset for refinement, rescoring with one of the generated scoring functions based on a 10 Å step potential should be performed. After the refinement, a more selective scoring function might be consulted to tackle the small scoring problem I and predict a near-native structure on top.
A subset of the best 5-10 % of the decoys can still contain a few thousand structures and refinement of them can take several days depending on the complex size. In order to further decrease the total number of decoys by a simultaneous constant number of near-native structures, clustering of ligand positions might lead to good results. Due to fact that the step potentials account well for the diversity of near-native structures the average score of the near-native cluster might be expected to be better than for incorrect clusters. Thus, a refinement of the best clusters or only a refinement for a few chosen structures from the clusters would reduce the set for refinement dramatically and makes refinement feasible in a few hours.
Furthermore, the generated step potentials might also be useful for a sampling by a Monte Carlo search. For that reason, the possibility of clashes has to be avoided by the definition of a minimal distance between two atoms. This Monte Carlo search might also be able to overcome local minima more often than gradient dependent methods and may therefore generate fewer incorrect solutions due to the fact that the algorithm ends up in more global minima. Due to the long range step potential, this might also be seen as further smoothing of the energy landscape on the complex surface. In addition the algorithm might also be able to perform movements and rotations of the ligand along the surface to end up not at the first position where the two constituents come together.

To increase the probability to predict a near-native structure on top, the Lego bricks for the successful development of knowledge-based scoring functions have been presented in this work. Based on the scoring results from the Tobi potential, short range step potentials or differentiable scoring functions which are trained on refined structures are suggested to become a solution for the big scoring problem (or small scoring problem I). Furthermore, the combination of scoring functions can be used to make scoring functions more selective to the native structure. Therefore, not only linear combination but also nonlinear classification algorithms like neuronal networks or support vector machines with nonlinear kernels could be useful because nonlinear classification is able to separate clusters by an ideal boundary.
The investigation of the interfaces by the step potentials and the BSA-potentials showed that interface composition plays a large role for protein-protein interactions. Thus, also the use of interface predictors like WHISCY [67] might be useful to constrain the possible interface for sampling or to incorporate their score for the clustering of structures from unbound docking for a refinement.

Based on the knowledge of hot spots, which are areas with a large portion of hydrophobic residues, also a scoring function which accounts not only for contacts between the protein partners but also for the surrounding interface areas of that contact might lead to an improvement in selectivity due to fact that it might improve scoring for perfectly aligned interfaces.

To sum it up, a perfect scoring function like the total free binding energy in nature, cannot be estimated for the docking with rigid proteins. Furthermore, an investigation of the scoring functions to gain further general insights for improvement is quite difficult since the results have to be averaged over the different protein complexes and their near-native structures from the sampling algorithm. The only valuable evaluation of scoring functions seems to be the average of their scoring performance. Thus, a methodology which is able to create problem-adapted scoring functions in a sufficient amount of time is necessary to try out the nearly infinite possibilities for scoring functions.
This work was able to present the tools for a fast development and evaluation of knowledge-based scoring functions for protein-protein docking. Nevertheless, its methodology and programs can also be used for the variety of biological scoring problems which just differ by the set of decoys which are used for the training of the scoring function.

Appendix A

Tables

A.1 Grouped-All-Atom Model

Table A.1: 27 defined atom types for the grouped all atomistic model.

#	atom type
1	$N^{backbone}$
2	$H^{backbone}$
3	C_{α}
4	$C^{backbone}$
5	$O^{backbone}$
6	$CH_3^{nonpolar/polar}$
7	$CH_2^{nonpolar}$
8	$CH^{nonpolar}$
9	$S^{nonpolar}$
10	$CH_{(2)}^{polar}$
11	$O_H^{polar/aromatic}$
12	$H_O^{polar/aromatic}$
13	S^{polar}
14	$O_{=}^{polar}$
15	N_H^{polar}
16	H_N^{polar}
17	$C_{=}^{polar}$
18	$CH_{(2)}^{aromatic}$

19	$C_{Ring}^{aromatic}$
20	$C(H)^{positive}$
21	$CH_2^{positive}$
22	$N^{positive}$
23	$H^{positive}$
24	$O^{negative}$
25	$CH_2^{negative}$
26	$C_{acid}^{negative}$
27	$C_{\alpha}^{Glycine}$

Table A.2: Description of the residues by the 27 defined atom types for the grouped all atomistic model.

residue	atom/group	#	partial charge
ALA	N	1	-0.570
	HN	2	0.370
	CA	3	0.200
	CB	6	0.000
	C	4	0.500
	O	5	-0.500
ARG	N	1	-0.570
	HN	2	0.370
	CA	3	0.200
	CB	21	0.000
	CG	21	0.070
	CD	21	0.310
	NE	22	-0.700
	HE	23	0.440
	CZ	20	0.640
	NH1	22	-0.800
	HH11	23	0.460
	HH12	23	0.460
	NH2	22	-0.800
	HH21	23	0.460
	HH22	23	0.460
	C	4	0.500
	O	5	-0.500

ASN	N	1	-0.570
	HN	2	0.370
	CA	3	0.200
	CB	10	-0.000
	CG	17	0.500
	OD1	14	-0.500
	ND2	15	-0.850
	HD21	16	0.425
	HD22	16	0.425
	C	4	0.500
	O	5	-0.500
ASP	N	1	-0.570
	HN	2	0.370
	CA	3	0.200
	CB	25	-0.100
	CG	26	0.700
	OD1	24	-0.800
	OD2	24	-0.800
	C	4	0.500
	O	5	-0.500
CYS	N	1	-0.570
	HN	2	0.370
	CA	3	0.200
	CB	10	0.180
	SG	13	-0.450
	HG	12	0.270
	C	4	0.500
	O	5	-0.500
GLN	N	1	-0.570
	HN	2	0.370
	CA	3	0.200
	CB	10	0.000
	CG	10	0.000
	CD	17	0.500
	OE1	14	-0.500
	NE2	15	-0.850
	HE21	16	0.425
	HE22	16	0.425
	C	4	0.500
	O	5	-0.500

GLU	N	1	-0.570
	HN	2	0.370
	CA	3	0.200
	CB	25	0.000
	CG	25	-0.100
	CD	26	0.700
	OE1	24	-0.800
	OE2	24	-0.800
	C	4	0.500
	O	5	-0.500
HIS	N	1	-0.570
	HN	2	0.370
	CA	3	0.200
	CB	21	0.000
	CG	20	0.130
	ND1	22	-0.570
	HD1	23	0.420
	CD2	20	0.100
	CE1	20	0.410
	NE2	22	-0.570
	HE2	23	0.420
	C	4	0.500
	O	5	-0.500
ILE	N	1	-0.570
	HN	2	0.370
	CA	3	0.200
	CB	8	0.000
	CG1	7	0.000
	CG2	6	0.000
	CD1	6	0.000
	C	4	0.500
	O	5	-0.500
LEU	N	1	-0.570
	HN	2	0.370
	CA	3	0.200
	CB	7	0.000
	CG	8	0.000
	CD1	6	0.000
	CD2C	6	0.000
	O	4	0.500
		5	-0.500

LYS	N	1	-0.570
	HN	2	0.370
	CA	3	0.200
	CB	21	0.000
	CG	21	0.000
	CD	21	0.000
	CE	21	0.310
	NZ	22	-0.300
	HZ1	23	0.330
	HZ2	23	0.330
	HZ3	23	0.330
	C	4	0.500
	O	5	-0.500
MET	N	1	-0.570
	HN	2	0.370
	CA	3	0.200
	CB	7	-0.000
	CG	7	0.235
	SD	9	-0.470
	CE	6	0.235
	C	4	0.500
	O	5	-0.500
PHE	N	1	-0.570
	HN	2	0.370
	CA	3	0.200
	CB	18	0.000
	CG	19	0.000
	CD1	19	0.000
	CD2	19	0.000
	CE1	19	0.000
	CE2	19	0.000
	CZ	19	0.000
	C	4	0.500
	O	5	-0.500
PRO	N	1	-0.570
	CA	3	0.285
	CB	7	0.000
	CG	7	0.000
	CD	7	0.285
	C	4	0.500
	O	5	-0.500

SER	N	1	-0.570
	HN	2	0.370
	CA	3	0.200
	CB	10	0.265
	OG	11	-0.700
	HG	12	0.435
	C	4	0.500
	O	5	-0.500
THR	N	1	-0.570
	HN	2	0.370
	CA	3	0.200
	CB	10	0.265
	OG1	11	-0.700
	HG1	12	0.435
	CG2	6	-0.000
	C	4	0.500
	O	5	-0.500
TRP	N	1	-0.570
	HN	2	0.370
	CA	3	0.200
	CB	18	0.000
	CG	19	-0.055
	CD1	19	0.130
	CD2	19	-0.055
	NE1	15	-0.570
	HE1	16	0.420
	CE2	19	0.130
	CE3	19	0.000
	CZ2	19	0.000
	CZ3	19	0.000
	CH2	19	0.000
	C	4	0.500
	O	5	-0.500

TYR	N	1	-0.570
	HN	2	0.370
	CA	3	0.200
	CB	18	0.000
	CG	19	0.000
	CD1	19	0.000
	CD2	19	0.000
	CE1	19	0.000
	CE2	19	0.000
	CZ	19	0.265
	OH	11	-0.700
	HH	12	0.435
	C	4	0.500
	O	5	-0.500
VAL	N	1	-0.570
	HN	2	0.370
	CA	3	0.200
	CB	8	0.000
	CG1	6	0.000
	CG2	6	-0.000
	C	4	0.500
	O	5	-0.500

A.2 The docking Benchmark

Table A.3: The ATTRACT benchmark consisting of 164 complexes from the protein docking benchmark 4.0.

PDB code	class	high	medium	acceptable	total
2I25	rigid	0	0	12	14444
1HIA	rigid	0	0	19	13636
1FQJ	rigid	0	4	53	49022
1E6J	rigid	0	5	16	19641
1B6C	rigid	0	7	14	20230
3D5S	rigid	0	2	72	14885
2OOB	rigid	0	6	36	5985
1US7	rigid	0	3	47	21381
1MAH	rigid	0	3	11	16536

1JTG	rigid	0	1	9	19643
1E6E	rigid	0	5	34	23494
1AZS	rigid	0	8	20	37619
2O8V	rigid	0	4	43	14125
2G77	rigid	0	0	16	23043
1UDI	rigid	0	0	27	13750
1JWH	rigid	0	2	17	39432
1PVH	rigid	0	2	25	21161
1HE1	rigid	0	0	35	16760
1FLE	rigid	0	0	21	11581
1AY7	rigid	0	3	28	9552
3BP8	rigid	0	13	148	49545
2FJU	rigid	0	8	18	36371
1XD3	rigid	0	1	54	10966
1TMQ	rigid	0	4	9	22833
1PPE	rigid	1	2	42	8092
1HCF	rigid	0	5	49	21800
1FFW	rigid	0	4	63	8360
1DQJ	rigid	0	2	27	24757
1AVX	rigid	0	3	13	19506
2MTA	rigid	2	3	29	20894
2FD6	rigid	0	3	11	33913
1JPS	rigid	3	1	17	40687
1ML0	rigid	0	5	57	25798
1QA9	rigid	0	2	20	12292
1VFB	rigid	0	2	27	18235
1NCA	rigid	2	1	14	35611
1K74	rigid	0	2	12	23921
1GL1	rigid	0	3	30	10217
1EFN	rigid	0	7	52	7678
1BUH	rigid	0	5	29	15630
2AJF	rigid	0	2	11	36886
1N8O	rigid	0	1	10	22580
1I9R	rigid	0	8	16	35820
1GHQ	rigid	1	5	31	21530
7CEI	rigid	1	6	38	11844
2OUL	rigid	2	3	17	18255
2HQS	rigid	0	1	102	21777

2ABZ	rigid	0	1	27	12500
1WEJ	rigid	1	4	24	19562
1R0R	rigid	1	1	28	10653
1K4C	rigid	0	4	19	31822
1I4D	rigid	0	1	45	29969
2A5T	rigid	0	0	15	30593
2HLE	rigid	0	2	19	18343
2OOR	rigid	0	0	30	32795
3SGQ	rigid	1	3	39	10274
1E96	rigid	0	4	25	20249
1FSK	rigid	0	4	20	27183
1ZHH	rigid	0	2	38	28707
1MLC	rigid	0	2	16	24565
1WDW	rigid	0	0	18	34348
2A9K	rigid	0	2	15	18417
4CPA	rigid	0	2	41	12664
1BJ1	rigid	0	6	10	35637
1EAW	rigid	0	1	32	12784
1GCQ	rigid	2	3	26	6261
1QFW	rigid	1	3	27	30958
1T6B	rigid	1	0	15	34399
1ZHI	rigid	0	13	53	38281
1KXQ	rigid	0	4	14	26054
1F51	rigid	0	4	58	20009
1CLV	rigid	0	2	47	12196
1AK4	rigid	0	0	24	17916
2UUY	rigid	0	0	31	12258
2J0T	rigid	0	0	9	14803
2BTF	rigid	0	1	24	24050
1YVB	rigid	0	4	17	19060
1S1Q	rigid	0	5	35	10365
1OYV	rigid	0	2	28	20894
1KLU	rigid	0	3	26	49705
1IQD	rigid	0	0	11	27280
1GPW	rigid	0	1	20	22474
1F34	rigid	0	0	31	24099
1CGI	rigid	0	1	20	13937
1AHW	rigid	3	3	29	42976

2SNI	rigid	1	2	21	14497
2B4J	rigid	0	0	22	14635
2PCC	rigid	1	9	57	16164
1A2K	rigid	0	7	23	21489
1BVK	rigid	0	3	35	18015
1EWY	rigid	0	5	52	15439
1GLA	rigid	0	1	19	24339
1KAC	rigid	0	0	23	15523
1J2J	rigid	0	7	43	9519
1NSN	rigid	0	5	17	24019
2B42	rigid	0	0	3	22297
2SIC	rigid	0	1	7	17978
1BVN	rigid	0	2	16	18047
1EZU	rigid	0	0	7	23225
1RV6	rigid	0	7	41	15428
1XU1	rigid	0	1	81	17074
1RLB	rigid	0	9	79	27751
1KTZ	rigid	4	7	33	14175
1OC0	rigid	0	4	34	10557
2AYO	rigid	0	1	42	16969
1Z5Y	rigid	0	1	20	15873
1OPH	rigid	0	2	11	27086
1KXP	rigid	0	0	27	40242
2JEL	rigid	1	2	29	20294
2VIS	rigid	0	1	19	40341
1GXD	rigid	0	0	8	39590
1FC2	rigid	0	1	22	11802
1AKJ	rigid	0	4	45	30158
2VDB	rigid	1	2	35	23398
1D6R	rigid	0	2	72	13409
1OFU	rigid	0	0	30	25124
1H9D	rigid	0	0	28	14292
1Z0K	rigid	0	8	33	10506
1SBB	rigid	0	3	13	31710
1DFJ	rigid	0	2	12	24628
1FCC	rigid	0	2	48	19454
1GP2	medium	0	0	17	32577
1IB1	medium	0	0	3	28923

1MQ8	medium	0	4	7	18100
2CFH	medium	0	3	21	19764
1ACB	medium	0	3	20	12119
1R6Q	medium	0	1	35	18918
1WQ1	medium	0	0	16	23673
2Z0E	medium	0	0	8	19904
1IJK	medium	0	10	24	21162
1K5D	medium	0	4	18	35251
2OZA	medium	0	0	21	31751
2HRK	medium	0	2	28	15872
1LFD	medium	0	2	49	12550
1XQS	medium	0	0	33	28380
1NW9	medium	0	6	26	14894
1SYX	medium	0	2	28	9066
1HE8	medium	0	5	30	33641
1ZM4	medium	0	2	26	52612
2H7V	medium	0	1	11	27194
1GRN	medium	0	1	38	21176
3CPH	medium	0	0	17	34914
2J7P	medium	0	0	10	30689
1KKL	medium	0	4	55	21136
1BGX	medium	0	0	0	56116
1M10	medium	0	0	13	24207
1I2M	medium	0	0	35	23597
1JIW	medium	0	1	34	22222
1BKD	hard	0	0	3	28415
1R8S	hard	0	0	0	17557
1JK9	hard	0	2	20	20022
2OT3	hard	0	0	1	21423
2HMI	hard	0	0	5	60626
2O3B	hard	0	0	19	20870
2IDO	hard	0	0	21	12396
1FQ1	hard	0	0	19	22107
2C0L	hard	0	0	8	16748
1IBR	hard	0	0	1	28841
1ZLI	hard	0	0	17	17312
1ATN	hard	0	0	10	32886
2I9B	hard	0	0	2	25874

1JMO	hard	0	0	2	27480
1E4K	hard	0	0	7	35576
1EER	hard	0	0	17	32978
1JZD	hard	0	2	65	27320
1PXV	hard	0	0	7	18498

A.3 Performance Tables

Table A.4: Small scoring Problem I in the training set: The probability in [%] to predict a near-native structure at the given positions.

potential	top 1	top 10	top 100	top 200	top 500	top 1000	top 2000
Attract	7.14	28.57	66.43	77.14	92.86	100.0	100.0
Vdw_Attract	7.86	23.57	60.71	74.29	85.0	94.29	97.86
Elec_Attract	0.0	5.0	15.0	22.86	37.86	51.43	69.29
Elec_gaa	2.86	12.86	21.43	25.71	42.14	53.57	62.86
Tobi	4.29	23.57	57.86	69.29	84.29	92.86	96.43
MC_gaa_10	17.14	35.0	77.14	86.43	93.57	96.43	99.29
MC_gaa_10*	12.86	35.0	72.86	83.57	91.43	95.71	98.57
MC_gaa_6_10	17.86	38.57	79.29	88.57	92.14	96.43	97.86
MC_attract_step10	15.71	39.29	74.29	83.57	93.57	96.43	99.29
LinReg_gaa_10*	19.29	52.14	86.43	90.0	93.57	97.14	99.29
LinReg_gaa_6_10*	16.43	49.29	81.43	87.14	94.29	95.71	98.57
bsA_wrad_1.4	0.71	12.14	36.43	47.14	66.43	79.29	90.0
bsA_wrad_0.8	1.43	11.43	47.86	61.43	77.86	91.43	96.43
MC_gaa_bsa	6.43	25.0	62.14	75.71	85.0	90.71	95.71
MC_attract_bsa	7.14	27.86	64.29	75.71	84.29	92.86	97.86
LinReg_gaa_bsa*	5.0	22.86	57.86	72.14	83.57	91.43	96.43
MC_attract_saddle+	17.86	48.57	84.29	90.71	93.57	96.43	100.0
MC_attract_vdw	9.29	33.57	79.29	90.71	97.86	99.29	100.0
MC_attract_saddle	7.14	39.29	85.0	95.0	97.14	98.57	98.57
MC_gaa_vdw	7.14	25.71	62.86	76.43	93.57	97.14	98.57
MC_gaa_saddle	6.43	23.57	66.43	81.43	93.57	95.0	99.29
Combination_full	18.57	40.0	81.43	90.0	93.57	96.43	97.86
Combination_no_step	10.71	40.71	72.86	83.57	88.57	96.43	98.57
Combination_elec_vdw	10.71	35.71	72.86	83.57	91.43	95.0	99.29

Table A.5: Small scoring Problem I in the test set: The probability in [%] to predict a near-native structure at the given positions.

potential	top 1	top 10	top 100	top 200	top 500	top 1000	top 2000
Attract	12.5	20.83	37.5	41.67	45.83	54.17	70.83
Vdw_Attract	8.33	16.67	33.33	37.5	45.83	66.67	66.67
Elec_Attract	4.17	4.17	8.33	20.83	20.83	33.33	50.0
Elec_gaa	4.17	8.33	12.5	12.5	12.5	29.17	41.67
Tobi	12.5	20.83	41.67	54.17	58.33	58.33	70.83
MC_gaa_10	12.5	29.17	58.33	70.83	79.17	79.17	83.33
MC_gaa_10*	12.5	20.83	50.0	58.33	70.83	83.33	87.5
MC_gaa_6_10	20.83	25.0	50.0	70.83	79.17	83.33	91.67
MC_attract_step10	8.33	20.83	54.17	62.5	75.0	79.17	87.5
LinReg_gaa_10*	8.33	25.0	54.17	62.5	75.0	79.17	87.5
LinReg_gaa_6_10*	8.33	20.83	50.0	58.33	75.0	83.33	91.67
bsA_wrad_1.4	0.0	16.67	29.17	37.5	45.83	62.5	79.17
bsA_wrad_0.8	0.0	16.67	29.17	41.67	58.33	75.0	79.17
MC_gaa_bsa	8.33	20.83	50.0	62.5	70.83	70.83	91.67
MC_attract_bsa	8.33	20.83	50.0	62.5	66.67	70.83	91.67
LinReg_gaa_bsa*	8.33	29.17	37.5	54.17	62.5	75.0	91.67
MC_attract_saddle+	4.17	4.17	29.17	45.83	54.17	66.67	75.0
MC_attract_vdw	0.0	20.83	50.0	62.5	66.67	79.17	91.67
MC_attract_saddle	4.17	8.33	25.0	37.5	50.0	70.83	79.17
MC_gaa_vdw	4.17	29.17	54.17	58.33	70.83	75.0	91.67
MC_gaa_saddle	0.0	8.33	45.83	54.17	75.0	83.33	87.5
Combination_full	20.83	25.0	50.0	70.83	79.17	83.33	91.67
Combination_no_step	16.67	25.0	50.0	62.5	66.67	75.0	87.5
Combination_elec_vdw	4.17	33.33	41.67	54.17	62.5	70.83	83.33

Table A.6: Small scoring problem II in the training set: The average fraction of near-native structures in [%] in the given fractions of all decoys.

potential	top 0.1 %	top 1 %	top 2 %	top 5 %	top 10 %	top 20 %
Attract	4.05	15.21	23.73	37.08	49.22	64.64
Vdw_Attract	3.41	13.41	21.04	33.58	45.27	60.81
Elec_Attract	0.38	2.95	4.94	9.37	14.86	26.27
Elec_gaa	1.82	9.36	13.51	20.49	30.33	44.89
Tobi	1.63	7.6	11.08	18.78	28.17	41.7
MC_gaa_10	8.87	32.62	44.06	61.63	75.71	88.04
MC_gaa_10*	8.2	31.72	43.19	60.88	76.44	87.88
MC_gaa_6_10	9.08	32.35	44.37	61.09	74.62	88.7
MC_attract_step10	7.38	30.22	42.38	58.4	71.8	84.63
LinReg_gaa_10*	10.8	35.06	46.95	64.18	76.35	87.66
LinReg_gaa_6_10*	9.03	31.25	42.42	59.78	72.35	84.12
bsA_wrad_1.4	1.33	6.92	11.44	22.35	35.31	54.71
bsA_wrad_0.8	1.42	7.32	11.8	23.44	36.32	54.65
MC_gaa_bsa	4.3	20.68	30.89	47.83	63.39	78.15
MC_attract_bsa	4.34	20.94	30.93	47.95	63.15	78.33
LinReg_gaa_bsa*	3.84	18.49	28.19	46.34	61.57	77.03
MC_attract_saddle+	9.38	32.54	42.62	56.0	67.33	78.28
MC_attract_vdw	5.01	24.5	35.53	52.65	66.33	79.57
MC_attract_saddle	6.62	27.46	38.46	53.77	66.23	78.48
MC_gaa_vdw	3.77	17.42	26.88	42.49	59.09	73.15
MC_gaa_saddle	3.46	17.42	26.27	42.04	55.84	71.34
Combination_full	9.84	34.05	45.98	63.12	76.55	88.79
Combination_no_step	6.82	27.37	37.89	56.43	69.4	83.07
Combination_elec_vdw	5.29	22.62	31.79	47.94	62.58	76.52

Table A.7: Small scoring problem II in the test set: The average fraction of near-native structures in [%] in the given fractions of all decoys.

potential	top 0.1 %	top 1 %	top 2 %	top 5 %	top 10 %	top 20 %
Attract	1.88	5.53	7.12	13.25	23.86	35.33
Vdw_Attract	1.82	4.87	6.88	12.28	20.87	36.76
Elec_Attract	0.32	3.1	4.21	6.35	11.1	28.67
Elec_gaa	0.32	3.94	6.67	13.09	25.02	34.23
Tobi	2.94	7.65	9.9	15.08	21.02	40.83
MC_gaa_10	8.76	24.71	34.29	49.56	63.26	80.08
MC_gaa_10*	9.55	23.8	32.65	49.14	63.76	79.53
MC_gaa_6_10	10.27	25.58	33.6	51.79	68.33	79.74
MC_attract_step10	7.95	20.28	25.54	41.18	64.3	79.9
LinReg_gaa_10*	8.5	18.92	24.72	42.09	57.3	73.44
LinReg_gaa_6_10*	4.27	18.21	25.14	38.15	52.41	69.72
bsA_wrad_1.4	0.77	8.58	11.58	20.21	37.41	50.17
bsA_wrad_0.8	4.08	7.43	12.77	19.53	31.19	55.07
MC_gaa_bsa	8.33	16.23	20.87	35.46	59.94	76.13
MC_attract_bsa	6.83	15.19	22.23	37.48	55.95	71.29
LinReg_gaa_bsa*	7.11	15.96	21.25	35.21	54.66	74.64
MC_attract_saddle+	0.35	4.3	9.7	21.2	31.88	50.82
MC_attract_vdw	2.0	14.35	17.61	31.2	45.75	63.85
MC_attract_saddle	0.15	2.81	6.24	19.82	29.4	48.83
MC_gaa_vdw	4.0	14.12	20.75	31.7	47.6	63.9
MC_gaa_saddle	3.73	11.13	18.26	34.75	45.29	60.08
Combination_full	9.75	24.93	32.94	49.58	65.26	79.39
Combination_no_step	5.43	16.89	22.28	37.45	54.76	71.66
Combination_elec_vdw	3.26	13.2	18.25	33.72	41.06	55.78

Table A.8: The big scoring problem in the training set: The probability in [%] to predict the native structure at the given positions.

potential	top 1	top 10	top 100	top 200	top 500	top 1000	top 2000
Attract	23.57	42.14	61.43	68.57	80.0	87.14	89.29
Vdw_Attract	19.29	37.14	57.14	63.57	70.0	76.43	82.14
Elec_Attract	10.71	15.0	21.43	23.57	30.0	33.57	41.43
Elec_gaa	30.0	35.0	38.57	43.57	48.57	55.71	60.71
Tobi	96.43	97.14	97.14	97.86	98.57	98.57	98.57
MC_gaa_10	46.43	65.0	77.14	82.86	86.43	92.14	94.29
MC_gaa_10*	45.0	62.14	75.71	80.0	85.71	90.71	92.14
MC_gaa_6_10	45.0	63.57	75.0	80.71	85.71	91.43	94.29
MC_attract_step10	43.57	60.0	75.0	80.71	85.0	90.0	93.57
LinReg_gaa_10*	42.86	65.0	80.71	82.86	90.0	93.57	95.0
LinReg_gaa_6_10*	40.71	56.43	68.57	72.86	78.57	85.0	90.71
bsA_wrad_1.4	22.86	39.29	55.71	60.71	67.14	73.57	79.29
bsA_wrad_0.8	31.43	47.14	63.57	69.29	72.86	79.29	86.43
MC_gaa_bsa	38.57	55.71	68.57	74.29	78.57	82.86	87.86
MC_attract_bsa	37.86	56.43	70.0	77.14	80.71	88.57	90.0
LinReg_gaa_bsa*	38.57	52.86	67.14	73.57	77.86	83.57	87.86
MC_attract_saddle+	38.57	57.86	72.14	76.43	82.86	87.86	90.0
MC_attract_vdw	27.86	41.43	65.71	72.86	75.71	82.86	87.86
MC_attract_saddle	33.57	52.14	69.29	75.0	79.29	85.0	88.57
MC_gaa_vdw	16.43	25.0	40.71	46.43	57.14	66.43	73.57
MC_gaa_saddle	20.0	32.86	51.43	62.86	73.57	79.29	86.43
Combination_full	50.71	69.29	82.14	86.43	90.0	92.86	94.29
Combination_no_step	47.14	63.57	79.29	85.71	88.57	93.57	97.14
Combination_elec_vdw	25.71	40.0	57.86	64.29	72.86	81.43	85.71

Table A.9: The big scoring problem in the test set: The probability in [%] to predict the native structure at the given positions.

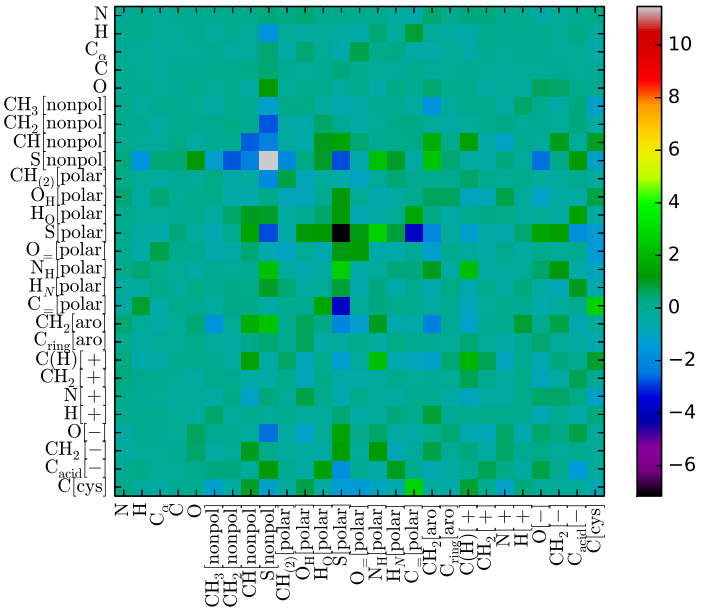
potential	top 1	top 10	top 100	top 200	top 500	top 1000	top 2000
Attract	26.92	42.31	65.38	65.38	69.23	73.08	76.92
Vdw_Attract	19.23	26.92	57.69	57.69	61.54	65.38	73.08
Elec_Attract	3.85	3.85	11.54	15.38	15.38	19.23	26.92
Elec_gaa	23.08	30.77	38.46	38.46	42.31	42.31	61.54
Tobi	88.46	100.0	100.0	100.0	100.0	100.0	100.0
MC_gaa_10	50.0	76.92	84.62	92.31	92.31	92.31	92.31
MC_gaa_10*	46.15	73.08	84.62	88.46	88.46	92.31	92.31
MC_gaa_6_10	50.0	73.08	80.77	84.62	88.46	88.46	88.46
MC_attract_step10	46.15	69.23	76.92	84.62	88.46	88.46	92.31
LinReg_gaa_10*	50.0	61.54	65.38	76.92	84.62	84.62	88.46
LinReg_gaa_6_10*	42.31	57.69	73.08	73.08	80.77	80.77	80.77
bsA_wrad_1.4	53.85	53.85	53.85	53.85	53.85	57.69	69.23
bsA_wrad_0.8	50.0	61.54	61.54	61.54	65.38	69.23	76.92
MC_gaa_bsa	50.0	61.54	69.23	76.92	88.46	88.46	88.46
MC_attract_bsa	53.85	65.38	73.08	80.77	92.31	92.31	92.31
LinReg_gaa_bsa*	53.85	61.54	69.23	73.08	80.77	88.46	88.46
MC_attract_saddle+	38.46	50.0	61.54	65.38	73.08	84.62	88.46
MC_attract_vdw	23.08	46.15	65.38	69.23	80.77	84.62	92.31
MC_attract_saddle	30.77	42.31	57.69	61.54	61.54	88.46	88.46
MC_gaa_vdw	23.08	46.15	57.69	65.38	73.08	73.08	73.08
MC_gaa_saddle	34.62	50.0	61.54	61.54	73.08	73.08	73.08
Combination_full	50.0	76.92	80.77	88.46	92.31	92.31	92.31
Combination_no_step	46.15	69.23	69.23	73.08	84.62	92.31	96.15
Combination_elec_vdw	38.46	50.0	65.38	65.38	73.08	76.92	80.77

Appendix B

Extra figures

B.1 Step Potentials

Figure B.1: Parameter matrix 'LinReg_gaa_10*'.



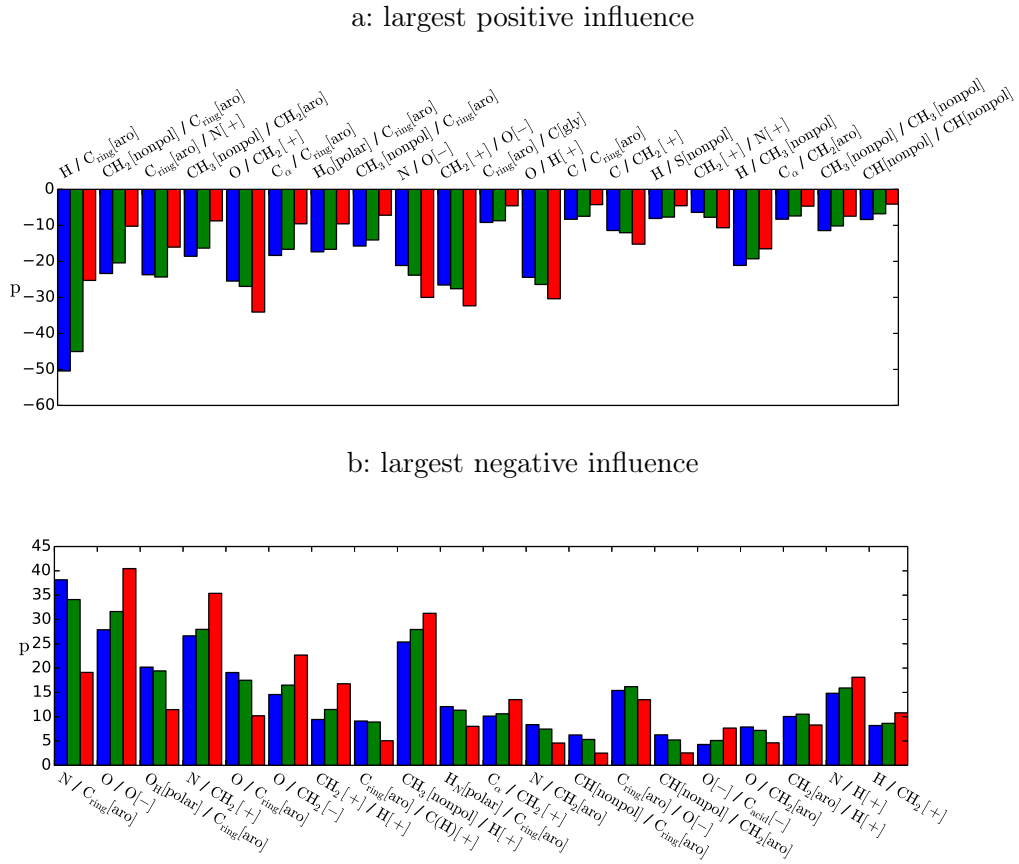


Figure B.2: Contributions 'LinReg_gaa_10*': The parameters are multiplied with the average number of their contacts for the native (blue), the near-native (green) and the incorrect structures (red) to regard the most negative (bottom) and most positive (top) influencing contact types on the score.

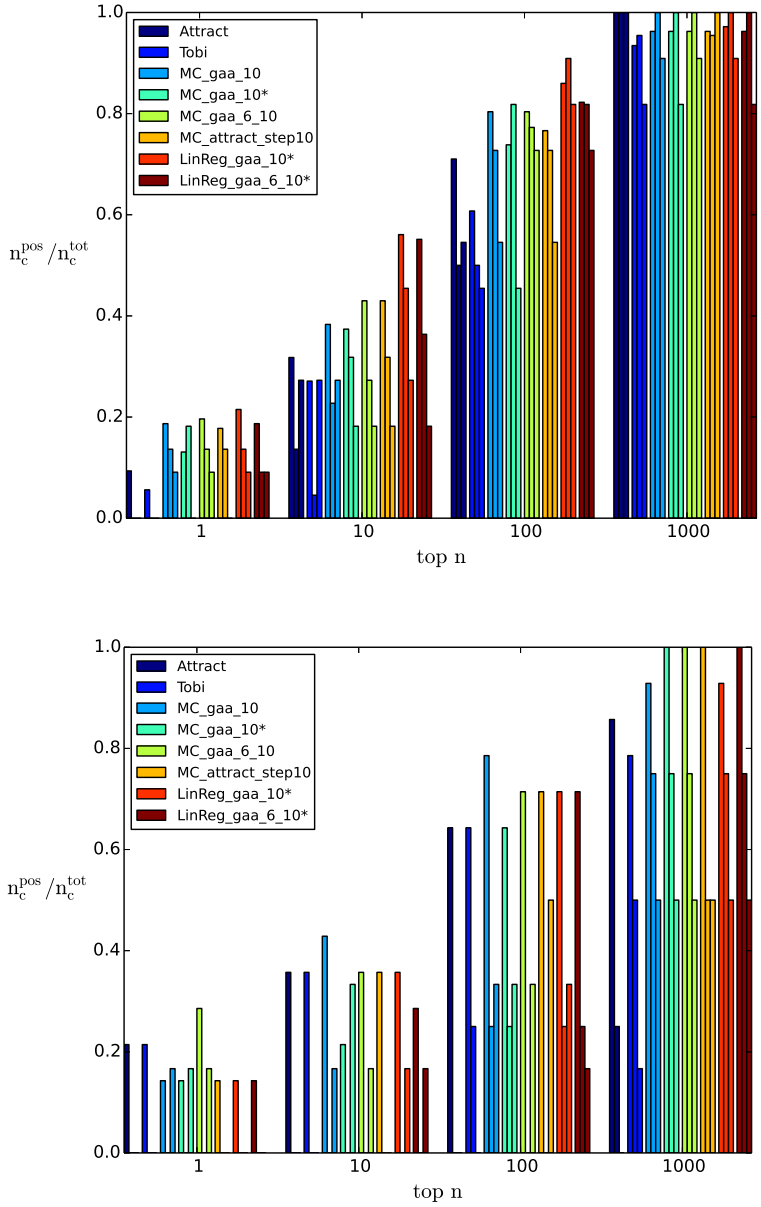


Figure B.3: Small scoring Problem I: The probabilities to predict a near-native structure for each class of difficulty (rigid, medium, hard) in the training (top) and the test (bottom) set at the given ranks is shown for the generated potentials and compared to the scoring by Attract and Tobi. On the ordinate the fraction of complexes for which a near-native structure can be observed is plotted for their position in the decoy set.

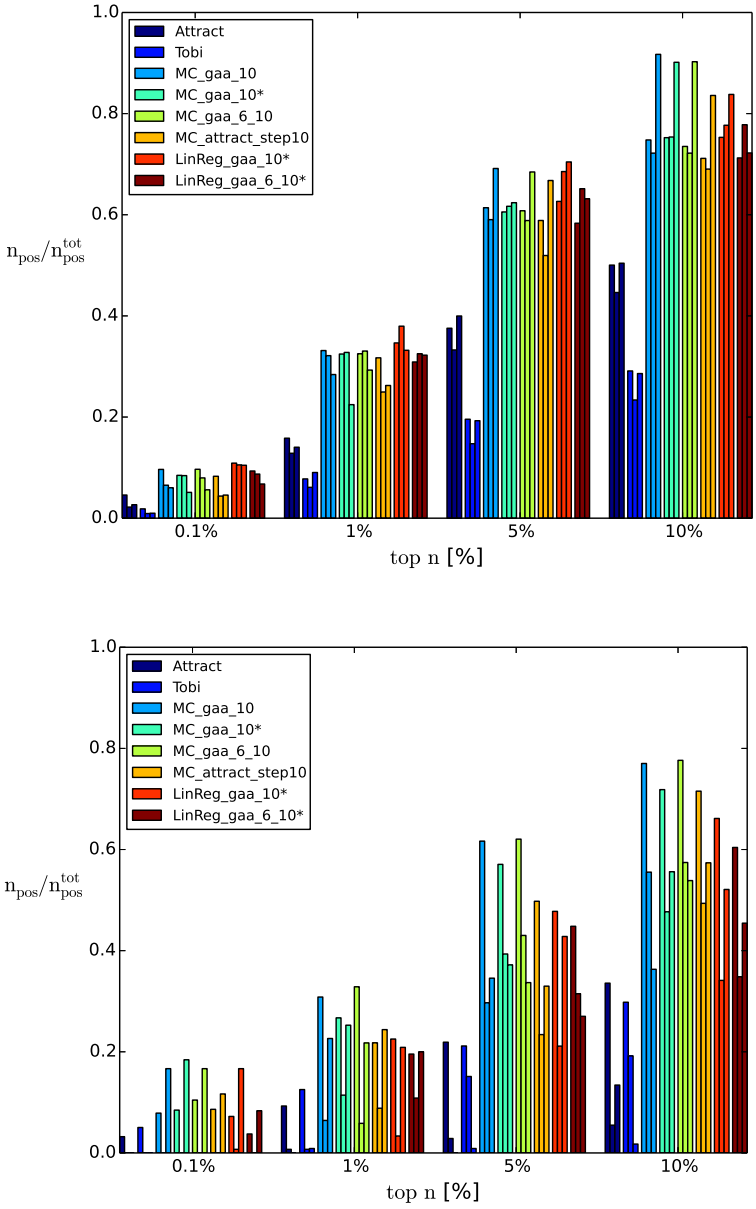


Figure B.4: Small scoring Problem II: The average fraction of near-native structures for each class of difficulty (rigid, medium, hard) in a subset is shown for the generated potentials in the training (top) and the test (bottom) set and compared to the established scoring functions of Attract and Tobi. On the ordinate the average fraction of near-native structures in the decoy sets is plotted for the fraction of all decoys in the set.

B.2 BSA-potential

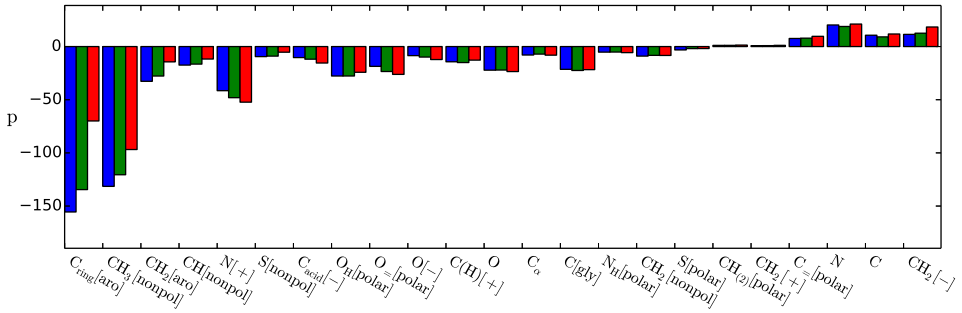


Figure B.5: Contributions to 'LinReg_gaa_bsa*': The parameters are multiplied with the average size of the BSAs for each type of pseudo atom from Attract for native (blue), near-native(green) and incorrect structures.

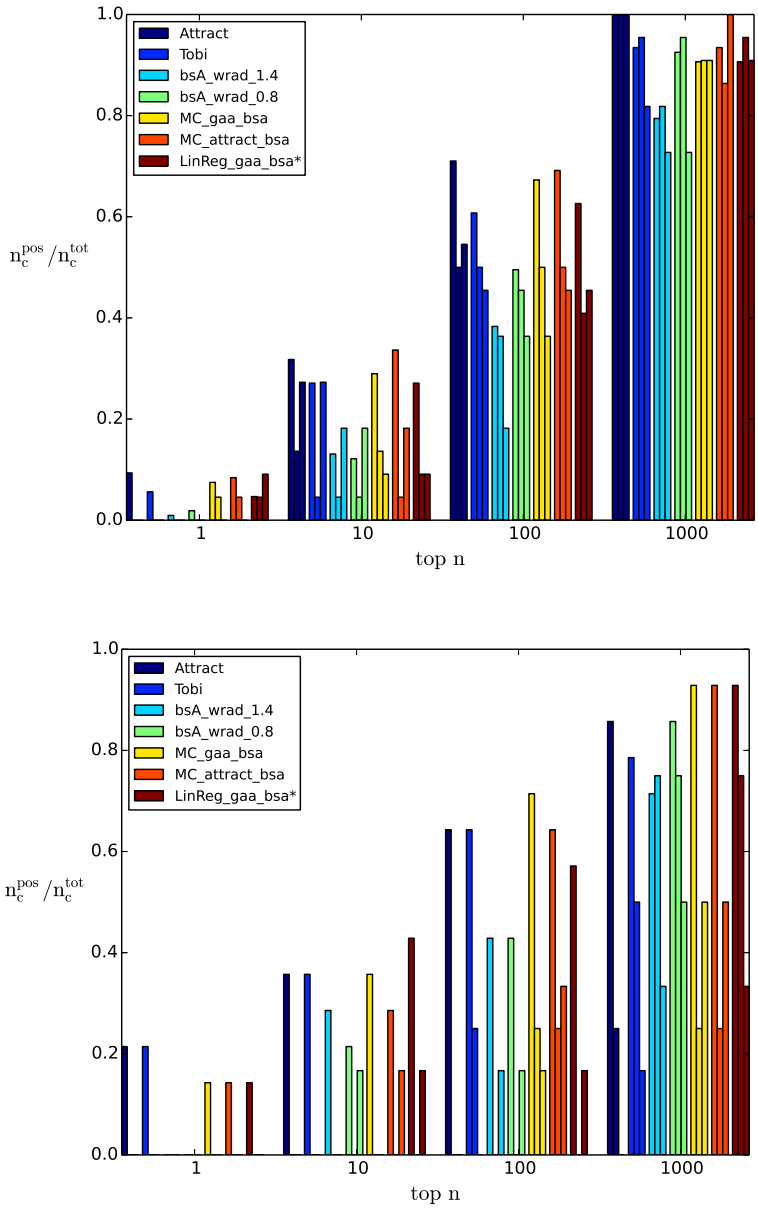


Figure B.6: Small scoring Problem I: The probabilities to predict a near-native structure for each class of difficulty (rigid, medium, hard) in the training (top) and the test (bottom) set at the given ranks is shown for the generated potentials and compared to the scoring by Attract and Tobi. On the ordinate the fraction of complexes for which a near-native structure can be observed is plotted for their position in the decoy set.

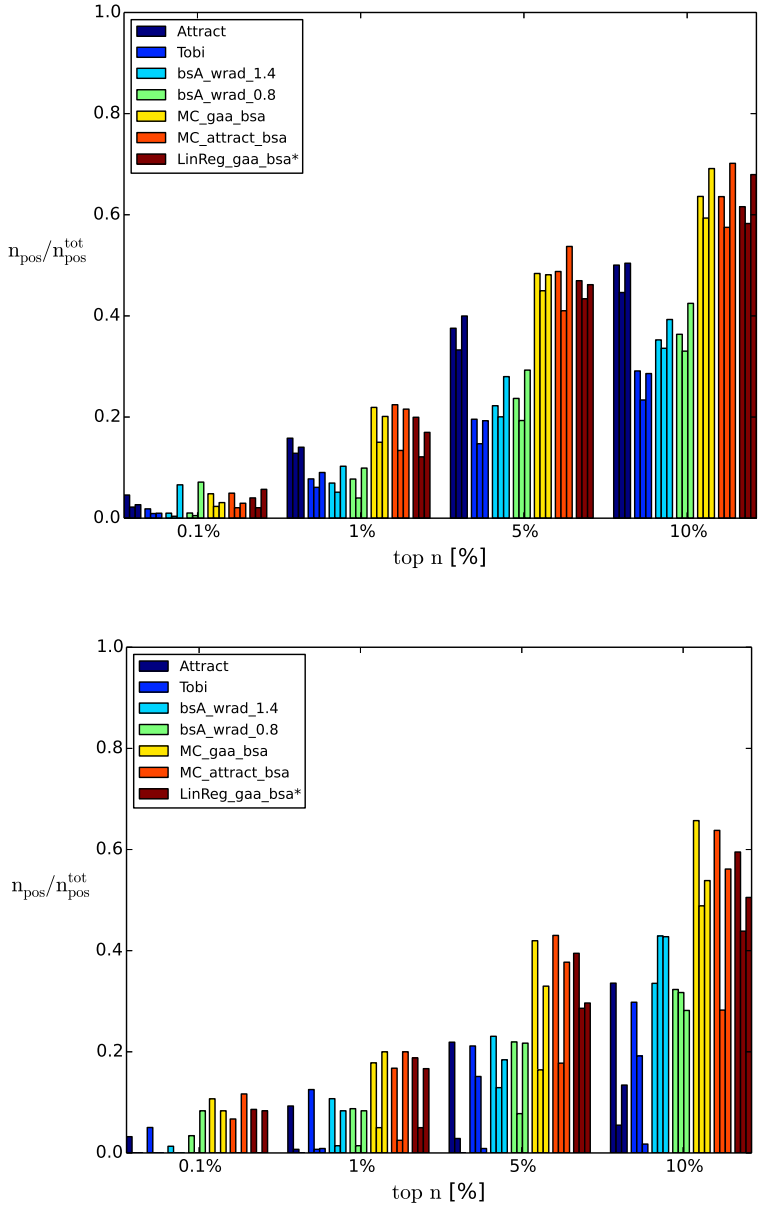


Figure B.7: Small scoring Problem II: The average fraction of near-native structures for each class of difficulty (rigid, medium, hard) in a subset is shown for the generated potentials in the training (top) and the test (bottom) set and compared to the established scoring functions of Attract and Tobi. On the ordinate the average fraction of near-native structures in the decoy sets is plotted for the fraction of all decoys in the set.

B.3 Differentiable Potentials

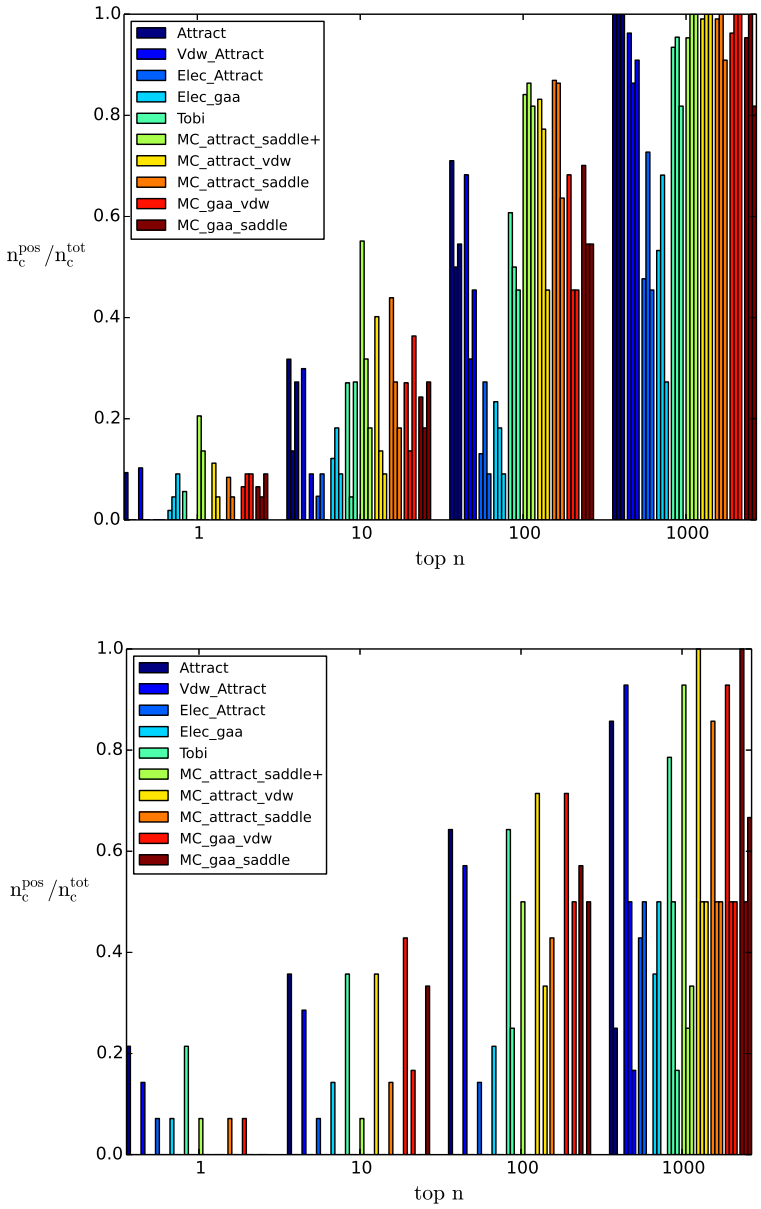


Figure B.8: Small scoring Problem I: The probabilities to predict a near-native structure for each class of difficulty (rigid, medium, hard) in the training (top) and the test (bottom) set at the given ranks is shown for the generated potentials and compared to the scoring by Attract and Tobi. On the ordinate the fraction of complexes for which a near-native structure can be observed is plotted for their position in the decoy set.

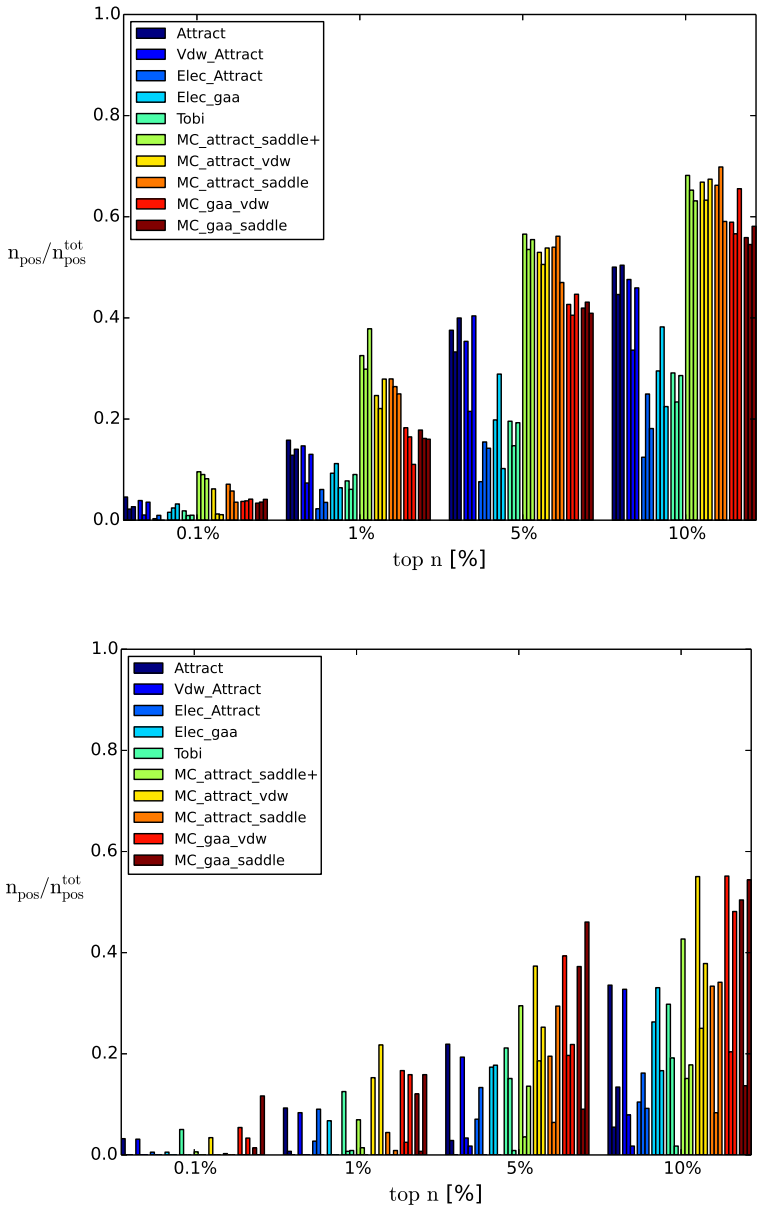
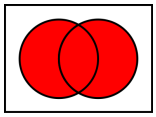
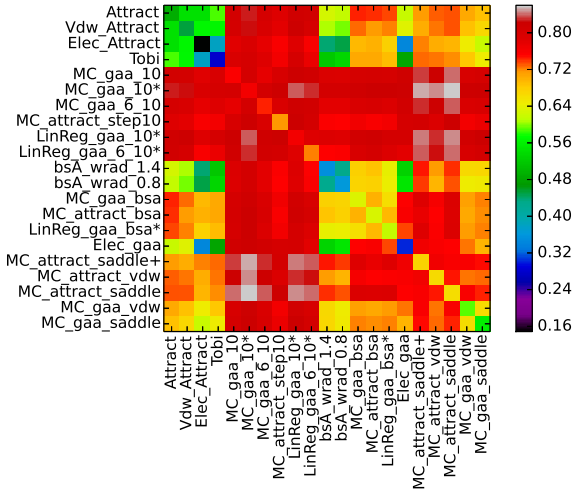


Figure B.9: Small scoring Problem II: The average fraction of near-native structures for each class of difficulty (rigid, medium, hard) in a subset is shown for the generated potentials in the training (top) and the test (bottom) set and compared to the established scoring functions of Attract and Tobi. On the ordinate the average fraction of near-native structures in the decoy sets is plotted for the fraction of all decoys in the set.

B.4 Correlation Analysis



a: correct structures



b: incorrect structures

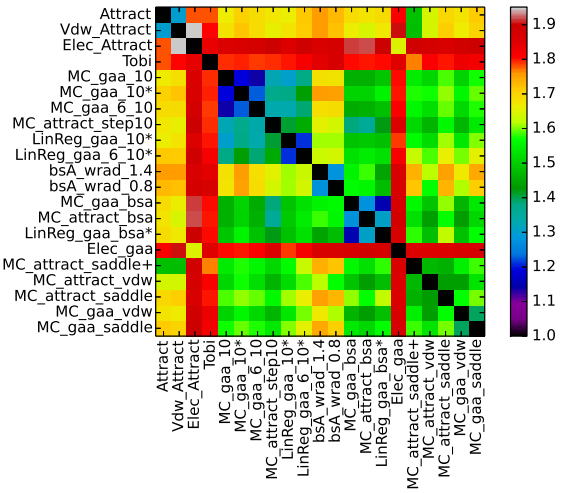
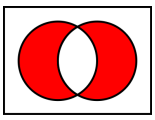
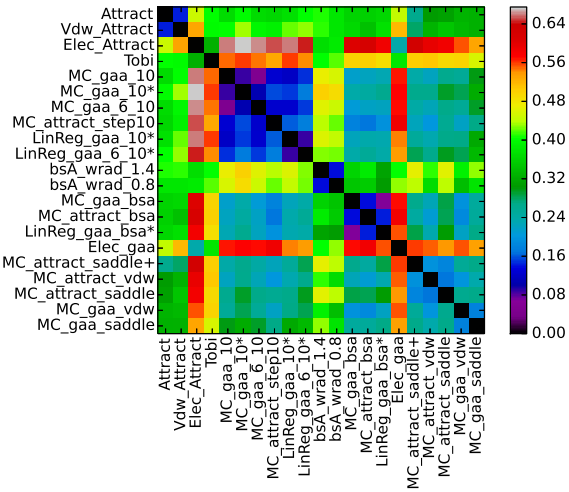


Figure B.10: Training set: The union of the top 10 % atleast acceptable a) and incorrect b) structures of all different scoring functions.



a: correct structures



b: incorrect structures

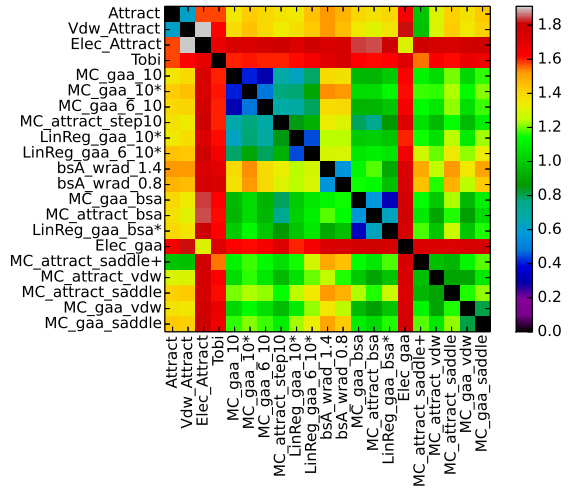
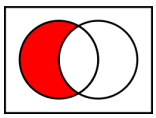
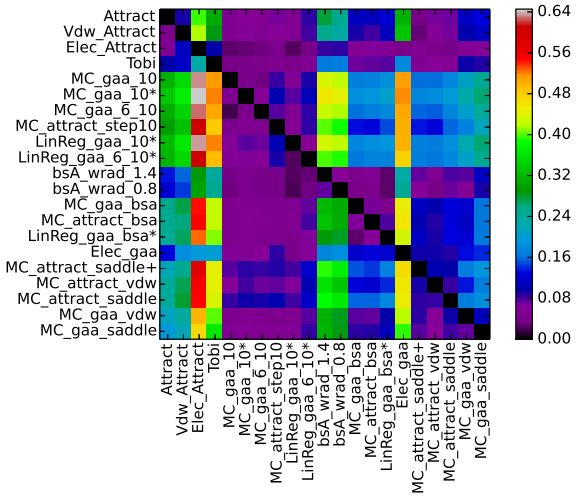


Figure B.11: Training set: The symmetric difference of the top 10 % atleast acceptable a) and incorrect b) structures of all different scoring functions.



a: correct structures



b: incorrect structures

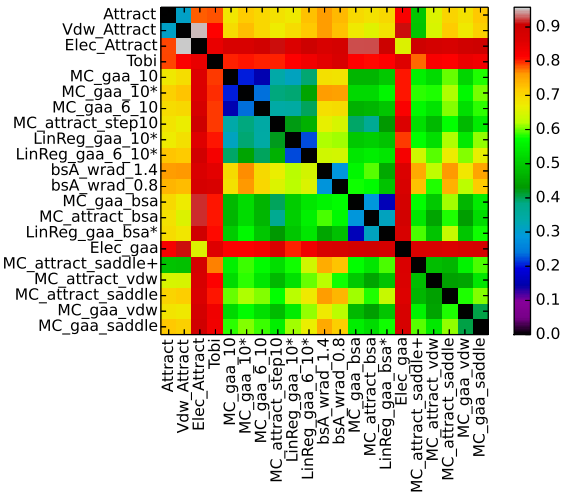


Figure B.12: Training set: The complement of the top 10 % atleast acceptable a) and incorrect b) structures of all different scoring functions.

Appendix C

Manuel for the Development of Knowledge-Based Scoring Functions

C.1 General preparations

To train a scoring function on a benchmark, it is necessary to create folders for each complex in the benchmark containing at least the unbound structures of the constituents as pdb's and an output file from Attract which contains the degrees of freedom. Furthermore, files which determine the quality of the conformations have to present for an optimization. That can be the fnat, Irmsd, Lrmsd or the Capri stars.

C.1.1 Defining complexes for the training set

The benchmark must be divided into a training and a test set. For that reason, a list of all complexes which are supposed to be in the training set has to be prepared.

C.1.2 cross-distribution.py

`cross-distribution.py` is a simple program to permute the list of training complexes to perform crossvalidation. It takes the list of the training complexes and permutes them n times. The last n'th part of the list serves as validation set in a training run.
To use cross-distribution.py the list of the complexes has to be given by the argument `--complexes`, the number of permutations by `--numtestset` and the name of the output file by `--output`. If `--complexes` is not given, it will take by default the names of all the folders in executive directory to prepare the crossvalidation files. As output it generates files with the output name attached by '-cv' and numbered from 1 to n.

```
$python cross-distribution.py --complexes trainset.txt
     --numtestset n
     --output outname
```

```
--exceptions 2 1ACB 1EWY
```

By the use of `--exceptions` the number and names of complexes which should be excluded, can be given.

C.2 Generation of feature vectors

C.2.1 collect-function.py

To train the parameters of the scoring functions and to rescore all the decoy sets quickly, it is necessary to precalculate feature vectors based on the form of the scoring function. The program `collect-function.py` is a python program using a Fortran written library which is implemented into python with **f2py**. Furthermore, it uses `collectlibpy.py` in combination with `collectlib.so` from the Attract program to generate the coordinates of the decoy poses on the run. To use collect-function.py the library `collectgridlib.f` has to be compiled with **f2py** and deposited in the same directory as the program collect-function.py. The library is compiled with f2py by:

```
$f2py -c -m collectgridlib collectgridlib.f
```

In addition, a soft link to `collectlibpy.py` and `collectlib.so` from the `/bin` folder in Attract has to be established.
The program itself can be executed on the terminal by giving it the `coordinates.dat` file from ATTRACT with the degrees of freedom for each decoy pose. Furthermore, with the argument `--proteinmodel` structures of the ligand and the receptor are inserted as pdb's. By default the program generates binary grids for the feature vectors in single precision. It is executed by:

```
$python collect-funtion.py coordinates.dat
```

`--proteinmodel`

1. `[attract, opls, gaa, undefined]`
2. `receptor.pdb`
3. `ligand.pdb`

(4.) `n_atypes`

`--gridtype [spline, none, distances]`

The `--proteinmodel` can whether be `tobi, opls, attract, gaa or undefined`. This has to be defined to reduce the grid size maximal due to the fact that Attract, opls and tobi do not use rising numbers from 1 to n without interruptions for their

atom types. The model `undefined` can be used for any representation if the atom types have no interruptions but the maximum number of atom types `n_atypes` must be inserted after the two pdb's.
The output grid contains all feature vectors and will be in the shape `(bins, numstructures, numparameter)`. The number of parameters `numparameter` arises for contact grids from `numparameter=atomtypes*(atomtypes-1)/2+atomtypes`.
Grids with double precision can be generated by the use of `--double`. By inserting `--maxstruc n_maxstruc(default 100.000)` the number of maximum decoy poses can be defined. The `--gridtype` can be `spline` for grids which use interpolation, `none` for grids which count the number of contacts in given ranges or `distances` which sum up all distances to a given power for all atoms of each contact type.
The argument `--polationparams` defines more precisely the content of the grid.

`--polationparams`

`if spline:`

1. `k_degreee(default=4)`
2. `stepsize(default=0.35)`
3. `start(default=1.4)`
4. `end(default=7.)`

`if none or distances:`

1. `stepsize(default=1.)`
2. `start(default=0;2)`
3. `end(default=10.)`

For the step potentials (`none`), `--polationparams` defines grids which use several equal steps which can be summed up to generate various ranges from that afterwards. The `stepsize`, the `start` and the `end` characterize the ranges of the generated regular steps in the vectors.
However, for the step potentials it might be more convenient to define the steps directly and not to precalculate a grid which can serve for the generation of a final step grid. For this purpose, the argument `--bins` can be used to define the step lengths of the grids by giving first the number of steps and then the distances for its steps.

`--bins n_bins rcut0 ··· rcutn`

For the saddle point potentials (`spline`) the parameters `stepsize`, `start` and `end` define the nodes between which will be interpolated in the training. The argument

`k_degree` determines the number of nodes and the degree of the interpolation for which 2 stands for a linear interpolation, 3 for a quadratic and so on. To find an appropriate degree for its interpolation, the parameter of Attract can be inserted by `--attractpar` to print out the interpolated values for each structure and compare them to the original values.
For the `distances` grids, which are used for the training of LJ-potentials, one can divide the feature vectors into different steps size `stepsize`, stating at `start` and ending at `end` in order to sum them up later in the training program for different cutoffs. Furthermore, the distances grids safe the sum of the distances between two atom types for each power separately. Hence, the number `n_power` and the powers to which the sums are stored has to be given by the argument `--powers`. The final distances grid possess a shape of the form `(bins, n_power, numstructures, numparameter)`:

```
--powers n_powers p0 ··· pn
```

C.2.2 asa.py

The program `asa.py` computes solvent accessible surface areas, the total buried surface area and buried surface areas for defined atom types by the rolling probe algorithm. The total buried surface area can be seen as a score, whereas the bsA for the atom types is used as features for further training or rescoring. As well as collect-function.py the script is dependent on two libraries which have to be present in the same directory as the main program. Softlinks for the files `colleclibpy.py` and `colleclib.so` have to be set from the Attract program. The Fortran library `asalib.f` has to be compiled with **f2py** for its use in the python program. The program is simply executed in the terminal by giving it 4 arguments:

```
$python asa.py
```

1. `receptor.pdb`
2. `ligand.pdb`
3. `coordinates.dat`
4. `[buriedsa, asa, atomsurface, opls, attract, tobi, gaa]`

The fourth parameter defines the data which will be generated. The `buriedsa` and `asa` print out the buried surface area and the solvent accessible area of the complex respectively. The parameter `atomsurface` generates a feature vectors of the bsA for the chemical atom types H, O, C, N, S on the interface. The remaining parameter `opls, attract, tobi, gaa` create grids of atomistic bsA's for the chosen atomistic representation of the pdb's with a grid shape in the form `(N_structures, n_atomtypes)`.

For the usage of pseudo atoms in the `attract` description, the size for the van der Waals radii of the pseudo atoms can be defined with the argument `--coarse_grained`. The coarse grained van der Waals radii were defined by Zacharias 2003 [69].
Furthermore it is possible to change the size of the rolling probe in the algorithm with the argument `--watersize r_probe(default=1.4)`.

C.2.3 decoysetmixer.py

As mentioned in the main text, it might be necessary for a good result to enrich the training set with near-native structures which were scored bad by another score or to supplement structures from another sampling method. This can whether be done by resorting and supplementing the ATTRACT output file with the degrees of freedom or to by using `decoysetmixer.py` to resort and supplement grids of the same form.

C.3 Training Parameter

C.3.1 capristars.py

The Capri stars evaluate the quality of the generated decoys after table 2.1 and thus they may be used as outcomes for regression methods or as quality weights in the Monte Carlo Annealing algorithm. The program `capristars.py` generates a file which contains the assigned stars from the files of the lrmsd, irmsd and the fnat. The output-file possess the name of the file from the ligand rmsd and ends on `.capstars`.

```
$python capristars.py file.lrmsd file.irmsd file.fnat
```

C.3.2 training-MC.py

The program `training-MC.py` is able to use Monte Carlo Annealing to predict a parameter set for scoring functions which are based on various potential forms. Therefore, it uses precalculated grids which contain features for each parameter and structure of each complex. To optimize the parameters, it is possible to define a parameter set as starting position or to start the algorithm from a random distribution. The program uses the subroutine `fgenlib.py` for the computation of function values at the nodes which are defined for the interpolation algorithm. Furthermore, it uses `duplicatelib.py` to duplicate structures in the decoy sets after a given probability (f.e. to be a *-structure after refinement), so that 10 % of the training structures for each complex have a probability except from 0. Both libraries have to be in the same directory as the main program for its execution by:

```
$python training-MC.py
```

```
--complexes [ ]
--grid [ ]
--mcparams [ ]
--evaluate [ ]
--preparameter [ ]
--output [ ]
```

Due to the variety of the potential types which can be handled by `training-MC.py`, the number and type of parameters which have to be given for each argument can vary. Below it is described which parameter have to be given for the arguments above.

`--complexes`

- `list_of_trainset.txt`
- `n_crossvalidation`

For `--complexes` a data-file containing the list of all the protein complexes for training purposes is given. The folders in this list must contain the precalculated grids and the qualitative evaluations (irmsd, fnat, ...) which are used by `training-MC.py`. Secondly, the number `n_crossvalidation` for leave one out crossvalidation has to be defined. The program trains the parameter on the fraction of the first $(n-1)/n$ complexes of the given training benchmark, and validates the results on the last $1/n$ complexes in the list.

`--grid`

- `precalc-grid.npy`
- `n_structures`
- `['interpolate', 'distances', 'step']`

if `'interpolate'`

  - `start_nodes`
  - `end_nodes`
  - `stepsize_nodes`

if `'distances'`

  - `r_cut_off`
  - `number_bins_power`
  - `list_of_bins[...]`

By the argument `--grid`, first the name of the precalculated training grids and secondly the number of structures which will be taken from it, is given. Thirdly, the grid type has to be defined, which can be `'interpolate'`, `'distances'` or `'step'`. For `'step'` potentials no further parameters have to be given. The grid type `'interpolate'` is for grids which contain the sum of Lagrange functions on the nodes for an interpolation of saddle point functions. For this type, one has to define the position of the first node, the position of the last and the stepsize between them. The parameter `'distances'` is used for grids which contain the sum over $1/r^z$ for different powers of z. Due to the fact that these sums can be calculated in different ranges, `r_cut_off` defines the number of range-bins which will be summed up for a total distance cutoff. Thereby, any integer cut off can be generated by summing up the contents for example in the ranges between 0 to 7 Å. Secondly, the number of bins with sums to different powers is given. That is usually 2 for a normal LJ-potential but also potentials which use further terms can be created by MC annealing. For a LJ-potential form, a grid which contains the sum over $1/r^12$ in the fist bin and over $1/r^6$ in the fourth, the argument looks as follows.

`--grid ···  r_cut 2 1 4`

`--mcparams`

- `Δp_type('normal', 'adaptive')`
- `Δp_size`
- `MC_steps`
- `targetfunction('rank', 'refine', 'positiontop', 'simple', 'positionlinear', 'positionquadratic', 'refine-positionlinear')`

if `'rank', 'refine', 'positiontop', 'refine-positionlinear'`

  - `number_subset`

- `Annealing_function('exponential', 'log', 'linear', 'ziczac')`
- `start_temperature`
- `mc_type('interpolate', 'normal', 'keepsign')`

if `'interpolate'`

  - `functiontype`
  - `power1`
  - `power2`

if `'normal', 'keepsign'`

  - `parameter_change('normal', 'saddle')`

if `'saddle'`

- `power1`
- `power2`

The parameters for `--mcparams` determine the way the Monte Carlo search is executed. The type of parameter change `Δp_type` can be `'normal'` or `'adaptive'`. `'normal'` changes a scoring parameter at each step by a given size `Δp_size` whereas the `'adaptive'` scheme adapts the stepsize `Δp_size` to the temperature by the factor t/T_0. `MC_steps` determines the number of Monte Carlo steps, that is equivalent to the number of parameter changes and rescorings.
The `targetfunction` determines the criteria for the optimization of the scoring functions parameters. The `'rank'` and the `'refine'` function builds the sum over `number_subset` structures on top for each complex but the `'rank'` function gives these an extra linear weight which decreases from rank 1 to `number_subset`.
The `'simple'` function takes the rank of the the best scoring near-native structure as its value for the targetfunction.
For `'positoinlinear'` and `'positionquadratic'` each position receives a weight ω_{pos} which decreases linearly and quadratic respectively. The near-native structures possess a weight $\omega_{quality}$ dependent on their quality whereas this weight is 0 for the incorrect structures. The sum over the product between these weights is taken for each complex as target function which has to be maximized.

$$\tau = \sum_{m}^{M_{complexes}} \sum_{n}^{N_{structures}} \omega_{pos}(rk(E_n)) \cdot \omega_{quality}^{(n)}$$

The `'positiontop'` function assigns also increasing linear weights to each position but just until the structure reaches the `number_subset` position. From that rank to the top, the same weight is assigned.
The target function `refine-positionlinear` multiplies the value from the target function `refine` with the value from `'positoinlinear'`.
The `Annealing_function` defines the way the temperature is cooled down from the starting temperature `start_temperature`. The annealing functions can be whether `'exponential'`, `'linear'` or `'log'` which defines a logarithmic decrease. The `'ziczac'` annealing uses a sinus^2 between two linearly decreasing functions to heat up the temperature several times to enable the search algorithm to overcome barriers more easily.
The `mc_type` defines how the scoring parameters will be changed. This is dependent on the given grid. For `'interpolate'` grids only the `'interpolate'` method works. In the library `functionlib.py`, three different `functiontype`'s are predefined. Functiontype `'0'` is the normal ATTRACT potential with attractive and repulsive

terms depending on the given third parameter i_{vor}. Funtiontypes `'1'` and `'2'` are variations of that ATTRACT potential, increasing the range of the attractive potentials at the minimum by a constant value and putting a repulsive term at the end of the potential respectively. For all three potential shapes, it is important to define `power1` for the first term of the LJ-potential and `power2` for the second. For `mc_type = 'keepsign'`, the scoring parameters are constrained to be larger than zero. This is important to use for the generation of a LJ-potential by the distance grids while using the parameters α and β directly. For `'keepsign'` and `'normal'`, it has to be defined in which parameter space the changes are performed. The `parameter_change = 'normal'` changes the parameter α_{AB} and β_{AB} but by choosing `'saddle'`, the parameter ϵ_{AB} and σ_{AB} got changed. If `'saddle'` is chosen as parameter space, `power1` and `power2` need to be defined for the computation of α_{AB} and β_{AB} out of ϵ_{AB} and σ_{AB}.

`--evaluate`

- `evaluation_values.type`
- `type('fnat', 'irmsd', 'lrmsd', 'capstars' 'probabilities', 'duplication')`

if `'duplication', 'probabilities'`

- `list_of_probabilities.txt`
- `n_column`

With the argument `--evaluate` the list of values for the qualitative weighting of the structures $\omega_{quality}$ is given. The second argument defines whether these values are `fnat`'s, `irmsd`'s, `lrmsd`'s, `capstars`'s or `probabilities` based on the quality file. If `'probabilities'` or `'duplication'` is chosen, the name of the `list_of_probabilities.txt` must be given. In that file, the first column consists of evaluation values for which a probability is assigned from the `n_column` column. The parameter `'duplication'` uses the probabilities as weights for the structures, too. Furthermore, it duplicates the structures on their probability to generate at least 10 % of structures for each complex with a probability except from 0.

`--preparameter`

- `startparameter('random', 'saddlepoint', 'ABC')`
- `bins`
- `number_of_parameter`

if `'saddlepoint', 'ABC'`

- `array_of_parameter.par`

By the argument `--preparameter` the format of the starting parameter `startparameter` is defined. Parameter `'random'` generates a random set of scoring parameters as a starting point for the optimization in the form `(bins, number_of_parameter)`. `bins` determines the number of different parameters, that means for example 3 for ϵ, σ and i_{vor}. The `number_of_parameter` defines the number of scoring parameters for the interaction types between λ atoms A and B: $n_{AB} = \frac{\lambda(\lambda-1)}{2} + \lambda$. Whereas for a grid with the atomistic buried surface areas, the number parameters would be $n_{AB} = \lambda$.
By the parameters `'saddlepoint'` and `'ABC'`, it is possible to read in a set of scoring parameters from a file. `'saddlepoint'` defines the parameter file to be form of ϵ_{AB}, σ_{AB} and i_vor_{AB} and `'ABC'` is used for parameters in the form α and β.

`--output`

- `output_name`
- `form_of_outputfile('matrix', 'linear')`

For the output file it is necessary to define the arrangement of the parameter. `'matrix'` leads to a matrix of the form (`bins` $\times \lambda \times \lambda$) and `'linear'` to a form (`bins` $\times$ `number_of_parameter`).

In addition to these main arguments, other arguments can be used to improve its results. By `--maxweight` and `--cutoffweight` the weights $\omega_{quality}$ can be cut off at low values to train only on 'very' good structures and for high weights respectively if overtraining might be coming from them.
Furthermore, `--insertnatives` gives the possibility to insert a grid of one structure for each complex which is supposed to be the native one.
By `--eraseatomtype` it is possible to erase all the entries of the grids for the given atom type due to the possibility of dummy atoms which should not receive parameters.
By the argument `--converge c_steps` it is possible exit the Monte Carlo algorithm before the total number of steps is executed. Therefore, `c_steps` defines after how many steps without a change of the target function, the search is stopped.
For the `spline` and the `distances` grids it might be necessary to constrain the range of the scoring parameters to avoid overfitting. For that reason, the argument `--searchrange` can be used. As parameters, the lower and the upper boundary for each bin has to be given in their order.
Finally, the program generates files containing the parameters in the linear order with the name `MC-Parameter-'outname'.par`. For the parameters in `matrix` form, the output file is named `MC-Paramatrix-'outname'.par`. For the output in form of

the parameter ϵ, σ and i_{vor}, the file is called 'MC-Paramatrix-'outname'_parm.par. Furthermore, a file containing the development of the targetfunction is created with the name Annealing-'outname'.txt.

C.3.3 training-glm.py

The program training-glm.py uses generalized linear models to estimate parameters for a scoring function from linear regression or linear classifications. Therefore, the feature vectors are fitted to a given set of output values. The outcomes have to be characteristics of the quality of the structures. Thus, fnats, irmsds, lrmsds, capstars of fnon-nats can be used as outcome values for the regression or to classify near-natives for a classification.
To enrich the training set with near-native decoys duplicatelib.py is implemented in the program. Thus, duplicatelib.py has to be present in the same folder as the main program. training-glm.py can be executed on the terminal by:

```
$python training-glm.py
    --complexes [ ]
    --grid [ ]
    --regressiontype [ ]
    --evaluate [ ]
    --functionshape [ ]
    --output [ ]
```

The parameters which have to be defined for the arguments --complexes and --evaluate are the same as already described for training-MC.py.
Due to the fact that linear regression can only deal with functions which are linear dependent on its parameters, the import of the grids varies a little from training-MC.py.

--grid

- precalc-grid.npy
- n_structures
- gridtype('distances', 'step')

if 'distances'

  - sign(keepsign, normal)
  - r_cut_off
  - number_bins_power

- `list_of_bins[...]`

The first parameter for the argument `--grid`, represents the name of the precalculated grids for each complex, followed by their number of decoys. Thirdly, the type of the grid is given, which can be whether `distances` or `step`. Grids of the form `distances` contain the sum over the distances for each atom type to a defined power and `step` grids the number of contacts in each defined step.
For the grids `distances`, it must be defined which sign is assigned for each bin. The parameter `keepsign` assigns a negative value to the second sum in the potential. To generate a Lennard-Jones potential, the `--regressiontype nonneglsq` must be used to generate exclusively positive values for α_{AB} and β_{AB}.
The parameter `r_cut_off` defines the number of bins which are sumed up to generate different cutoffs. `number_bins_power` represents the number of bins which contain sums over the distances to different powers. Following to that number, these location of these bins in the grid has to be determined.

The argument `--regressiontype` defines the linear regression or classification algorithm which is used for the parameter determination. The various regression methods perform their regression on different cost functions based on different underlying models. The characteristics of the listed models will roughly be explained in the following.
Furthermore, by the argument `--regressionparams` further parameters can be chosen for each regression method to optimize their results. An explanation for the argument `--regressionparams` must be taken from the source code.

`--regressiontype`

(a) `Robustlinearmodels`
(b) `svr-robust`
(c) `ols`
(d) `nonneglsq`
(e) `Ridge`
(f) `Lasso`
(g) `elasticNet`
(h) `RANSAC`
(i) `Bayesianridge`
(j) `logistic`
(k) `SGDClass`

The `Robustlinearmodel` represents a robust regression based on the statsmodel library which can be found under http://statsmodels.sourceforge.net. Instead of the usual assumption of Gaussian noise, the statsmodel library offers the possibility to use different probability models which can be chosen by:

```
--regressionparams regmodel(Huber, Andrew, Hampel, Ramsay, TrimmedMean,
    Tukey)
```

The regression with `svr-robust` represents a support vector regression. Based on the parameters given by `--regressionparams`, its robustness can be adjusted to be stronger or less strong.
In general, the results of some linear regression algorithms may be affected towards unfavoured solutions if outliers exist in the training set. Robust regression methods try to be more robust against outliers, due to their adapted cost-functions.
The `ols` regression represents ordinary least squares regression which assumes a Gaussian distribution of the noise.
Also `nonneglsq` uses an ordinary least squares fit to generate only positive parameters. Both methods are based on the library from SciPy which can be found under http://www.scipy.org.
The regression and classifier algorithms `Ridge`, `Lasso`, `elasticNet`, `RANSAC`, `Bayesianridge`, `logistic`, `SGDClass` are based on the scikit-learn module which were taken from http://scikit-learn.org.
`Ridge` and `Bayesianridge` regression penalize the size of the scoring parameters to avoid overfitting by including their average value into its cost function.
`Lasso` regression prefers solutions with fewer parameter values and thus may be able to recover the exact set of non-zero weights.
Elastic Networks combine the properties of Lasso and Ridge regression in their cost function and can be used for regression by choosing `elastic`.
`RANSAC` is a robust parameter estimator which uses random subsets of inliers for its predictions. Its result is highly dependent on the number of iterations which can be defined explicitly with `--regressionsparams`.
The regression types `logistic` and `SGDClass` represent a logistic classifier and stochastic gradient descent classifier respectively. The input for the y vectors have to consist of 0's and 1's for each class respectively. For the separation into classes the maximum or minimum value for each class must be given on the second position behind the `--regressiontype`.

With the argument `--functionshape`, the `bins` and the number of atom types `n_atomtypes` for the scoring parameters have to be defined. For grids which use the atomistic buried surface area the number of bins must be defined as 0.

```
--functionshape
```

- `bins`
- `n_atomtypes`

To define the name and the form of the `--output` first the name and then the chosen form has to be given.

```
--output outname outform(normal, saddle)
```

The form of the output can whether be `normal` or `saddle`. `saddle` means that the fitted parameters α and β will be transformed into ϵ and σ (look at equation 2.11). For the the `saddle` form, the power of the first and the second part of the LJ-potential have to be given for the transformation with `--powers p1 p2`.
Due to the different size and thus a diverse average numbers of contacts or buried surface areas for each complex, it might be useful to normalize the data. With `--preprocessing` a method to normalize the input data for each complex can be defined. `Standard` makes the distribution of contacts Gaussian for each complex. `MinMax` distributes the contacts for each structure between 0 and 1. Finally, `meancomplex` divides the contacts through the mean total number of contacts for each complex.

In addition, some more arguments can help to improve the results of regression. `--insertnatives` provides the possibility to insert a grid of one structure for each complex which is supposed to be the native structure. With `--eraseatomtype` it is possible to erase all the entries of the grid for the given atom type due to the possibility of dummy atoms.
The output of the program is a file named `LinReg-Parameter-'outname'.par` which contains the determined scoring parameters. Furthermore, it is possible to check the regression results by `--prediction`. This creates two files which contain the real values and the predicted values for the training and the test set with the names `LinReg-Predictions_'set'_'outname'.txt`.

C.3.4 training-nonlinear-classfier.py

Classifiers possess the advantage that they can separate classes nonlinearly on its features. Therefore, the structures must be divided into a class of near-native and incorrect decoys based on a qualitative assessment by fnat, irmsd, lrmsd or capri stars. The program `training-nonlinear-classfier.py` provides this possibility. It uses the same arguments as `training-glm.py` to read in the grids and to prepare the classification. The program is executed by:

```
$python training-nonlinear-classfier.py
     --complexes [ ]
```

```
--grid [ ]
--classifier [logistic, SVM, SGDclass, gaussianNB]
--evaluate quality.file qualitycut qualitytype
--functionshape [ ]
--output [ ]
```

Just for the argument `--evaluate` an extra parameter has to be defined which divides the structures into a near-native and an incorrect class on the values in the `quality.file`.
The nonlinear classifiers which can be chosen are `logistic`, `SVM`, `SGDclass` and `gaussianNB`. For all four of them, different `--fitparams` can be defined to obtain an optimal result. The exact adjustments for the `--fitparams` must be taken from the source code. The meaning of the adjustments can be looked up on http://scikit-learn.org which contains a description of the classifiers.
The output for these methods is a classifier which is stored in a binary file by the **cPickle** library named `Classpredictor_'classifier-outname.pkl`. This file can be reloaded and used in `grid-reevaluation.py` for the rescoring of decoy sets.

C.3.5 average-params.py

After the generation of n different parameter sets for each set of complexes for cross-validation, it may be a sufficient approach to use the average of these values to prevent overfitting on any set of structures. Therefore, `average-params.py` uses the argument `--scores` followed by the number of parameter sets and their names.

```
$python average-params.py --scores n_scores score1 ··· scoren
```

By using the argument `--scaleparams`, the parameters are normalized by the division through their standard deviation. Thereby, the parameter sets with larger scales do not dominate the total average. The standard deviation is calculated for every type of parameter separately and thus the number of bins has to be provided with `--bins`. If bins is set to 0 the parameters are not given in a matrix form but as a linear vector, for example for the buried surface area potentials or the weights of scoring combinations. To declare the name of the output the argument `--output` can be used.

C.3.6 combine_score.py

the program `combine_score.py` combines scores from different scoring functions linearly and nonlinearly. To determine the weights for the linear combination, it is possible to use Monte Carlo Annealing but also linear regression or linear support

vector machines. Furthermore, the program includes the possibility to combine scores nonlinearly by support vector machines which use nonlinear kernels and by using naive bayesian estimators.
The program can be executed on the terminal:

```
$python combine_score.py
    --complexes [ ]
    --scores [ ]
    --method [bayes, svm, mc regression]
    --yvalue
    --output [ ]
```

For the prediction of weights, a training set of complexes and the number of sets for crossvalidation has to be defined with:

```
--complexes trainingset.txt n_crossval
```

Furthermore, the number and the names of the different scores is given by:

```
--scores n_scores score_0 ··· score_n
```

The files containing the scores must be named `.rescore` if they just contain a list of the pure score or `.dat` when they are in the Attract output format, containing also the degrees of freedom etc.
The `--method` for the estimation of the weights can be whether `bayes`, `svm`, `mc` or `regression`.
The method `bayes` uses the given `--yvalues` from a file to divide the given structures into two classes 'near-native' and 'incorrect' on the given `cutoff`.

```
--yvalues quality.file cutoff
```

As output the program generates a file named `Combine_parameter_'outname'.pkl` from which the classifier can be reloaded with the library `cPickle` in the program `combine_rescore.py`. Furthermore, it stores the deviation σ and the mean μ of the Gaussian probability distribution for each class in a file named `Combine_parameter_'outname'.par`. From these parameters for each class and feature the probability to be in a certain class can be computed.

For the method `svm`, the `--yvalues` and the `cutoff` has to be provided to divide the structures into classes for classification. With the `--fitparams` the support vector classifier can further be defined:

```
--fitparams
```

- kernel[linear, rbf, polynomial](default=linear)
- cachesize[in MB](default=4000)
- Ci(default=1.)
- calc_prob[bool](default=False)
- tolerance(default=0.001)

The cachesize defines how much cache space will be used for the minimization. The parameter Ci represents the weight to which strength classification errors are contribute to the cost function of the svm. The parameter calc_prob∈True,False defines whether a probability distribution is generated for the classes. The probability is derived from the decision function of the support vector machine and serves just as an additional output form. The tolerance adjusts the convergence criteria of the minimization algorithm. The linear kernel produces a list of parameters for the decision function in the file Combine_parameter_'outname'.par. The nonlinear kernels rbf and polynomial generate a file named Combine_parameter_'outname'.pkl which can be reloaded in combine_rescore.py to compute the values of their nonlinear decision function.

For the method mc the --yvalues have to be given and the type of these values must be defined. This type can be fnat, lrmsd, irmsd or capstars. The parameters for the annealing algorithm have to be defined behind the argument --fitparams:

--fitparams

- targetfunction('rank', 'refine', 'positiontop', 'simple', 'positionlinear', 'positionquadratic', 'refine-positionlinear')
- Annealing_function(linear, exponential, ziczac, logarithmic)
- start_temperature
- Δp_type(normal, adaptive)
- Δp_size
- MC_steps

The meaning of the parameter for the Monte Carlo annealing algorithm can be looked up in the manual for training-MC.py.
With the argument --meanscale the different scores can be divided by their mean to adjust the stepsize of the weights for each type of score. Thus, the change of a weight will have the same influence on the combined score for each score. Hence, the parameter space might be sampled more successfully. This division will be respected

in the final output of the weights. As output, a parameter file is generated with the name `Combine_parameter_'outname'.par` which can be used directly as weights for the combination of scores. The development of the target function during the annealing is stored in `Combine_Annealing_'outname'.txt`.

For the method `regression`, only name of the file containing qualitative evaluations has to be provided behind the argument `--yvalues`.
In addition, the `--regressiontype` has to be defined, which can whether be `BayesianRidge`, `ols`, `Ridge` or `svr-robust`. A short description for these types of regression can be found in the manual of `training-glm.py` or in more detail on http://scikit-learn.org.
As mentioned above, the different sizes and chemical compositions of the proteins in all complexes can make a fitting difficult due to the fact that the mean scores will deviate. Therefore, the scores can be preprocessed by the argument `--preprocessing [MinMax, complexmean, Standard]`. The type of the preprocessing can be `Standard` which distributes each score for each complex Gaussian with a mean 0 and a standard deviation of 1. `MinMax` distributes the scores between 0 and 1 and `complexmean` divides the scores for each complex by their mean value.

C.4 Rescoring

C.4.1 grid-reevaluation.py

The program `grid-reevalutation.py` can be used for fast rescoring on the precalculated grids. Due to the fact that only simple multiplications have to executed, the rescoring on grids can be performed much faster than by `@rank.py`. Nevertheless, time delays can occur due to the size of the grids which have to be uploaded into the memory, especially when the program is executed many times in at the same time on a cluster.
The program is executed by:

```
$python grid-reevalutation.py pregrid.npy parameter.par
```

`--gridtype`

- `gridtype(interpolate, distances, step, nonlinear)`
- `bins`
- `atomtypes`

if `'distances'`

- `r_cut_off`
- `number_bins_power`

- list_of_bins[...]

if 'interpolate'

- start
- end
- stepsize
- functiontype
- power1
- power2

The grid type `nonlinear` refers to the possibility of using a nonlinear classifier like svm or naive bayes on a 1-step grid. The parameter file for these grids consists of binary file which contains the classifier from the program `training-nonlinear-classifier.py`.
If `distances` grids are used, the number of the bins which are summed up for the cutoff have to be given first. Secondly, the number of the bins which contain the sums over distances to different powers and their location have to be defined.
For the `interpolate` grids, the `start`, the `end` and the `stepsize` for the placement of the nodes has to be given. Furthermore, the `functiontype` and the power of the first and the second part of the potential have to be defined. The functiontypes were explained in `training-MC.py`, 0 stands for a normal Attract shaped saddle point potential.
For grids which use the atomistic buried surface area, the bins must be defined as 0. Finally the program prints out the scores. For further use of these scores for combination, the files names must end on **.rescore**.

C.4.2 @rank.py

For the rescoring of decoys without precalculated grids, the program `@rank.py` is able to score structures by a LJ-potential (saddlepoint), a step potential, a potential based on the atomistic surface area, a coulomb interaction, the buried surface area and even on a combination of these scores. The combination can be linear by given weights or nonlinear by the use of a classifier. The program is executed by:

```
$python @rank.py
    --input coordinate.dat
    --proteinmodel model receptor.pdb ligand.pdb
    --vdwpotential vdwparameter.par
    --steppotential stepparamter.par --bins n_bins range_0 ··· range_n
```

```
--solvation solparameter.par
--electrostatics
--buriedsa
--combination method combinationweights.par
```

The degrees of freedom for each decoy are given by the ATTRACT output file `coordinate.dat` which is inserted by `--input`. As in ATTRACT, normal modes ensembles and refined structures can be inserted by `--modes`, `--ens` and --name. Each scoring type can be chosen separately or in combination with other functions. If no combination method is given, the scores are just summed up.
The model for the given pdb's of the protein constituents can be `opls`, `attract`, `tobi`, `gaa` or `any`. The models `opls`, `attract`, `tobi` and `gaa` reduce the atom types to the total number of atom types (for opls f.e. 13). The model `any` uses as its maximum number of atom types the input behind the ligand.pdb. To use `any`, the parameter file must contain as many columns and rows as the maximum atom type.

For the scoring by models for van der Waals interactions the parameter file has the be given by the argument `--vdwpotential`.
Furthermore, a `--shift` can be defined which sets every distance between two atoms at least to that value. By shifting the distances to a certain value, clashes between atoms will be avoided. Clashes might result from the change of a coarse grained to an atomistic representation after sampling. In addition, a cutoff can be defined with `--vdwcutoff` which is by default set to 100.
For the van der Waals potentials, the `--functiontype` must be chosen `attract`, `opls` or `free`. `attract` uses a saddlepoint potential with the powers 8 and 6 whereas `opls` uses 12 and 6. Any other potential with other powers can be given by `free power1 power2`.
The parameter file for the step potentials is provided by the argument `--steppotential`. The bins of the step potential have to be defined by `--bins` with the number of bins and their ranges.
The `--solvation` potential uses its parameter file for the computation of a score by the atomistic bsA's. Futhermore, the radius of the probe can be changed with `--watersize`. For the `attract` model, coarse grained van der Waals radii may be used with `--coarse_grained`.
The `--electrostatics` term uses the charges in the pdb files to calculate a Coulomb energy between the proteins.
The `--buriedsa` calculates the negative buried surface area. As for the solvation term, the coarse grained modi can be used and the radius of the probe can be changed by `--watersize` and `--coarse_grained`.

The weights for linear combinations or the classifier to combine the different scores are given by `--combination`. The `method` must be defined first. This can be `linear` to combine the scores linearly by the determined weights, `probability` to estimate the probability for each structure to be in a class, or `decisionfunction` to use the distance from the boundary between classes from svm as a score.
For all nonlinear combinations but also for some linear combinations from linear regression, the normalization of the scores is necessary before combination. Therefore, `--preprocessing` can be used. The `method` can be `Standard` for a Gaussian distribution or `MinMax` for a distribution between 0 and 1 for each score.
The argument `--printout` serves to print out the final combined scores directly. `--rescore` gives out a file ending on `-rescore.dat` which contains each score and the combined score in the Attract file format in the orignal order of the Attract dat-file. `--rerank` creates such a file ending on `-resorted.dat` in which the structures are resorted after their new score.
A new name for the output can be set by `--output`.

C.4.3 combine_rescore.py

The program `combine_rescore.py` combines the given scores from files with a given set of weights linearly or nonlinearly by trained classifiers. The program is executed on the terminal by:

```
$python combine_rescore.py method weights.par
    --scores n_scores s1 ··· sn
```

To receive the right result, the scores must be in the same order as they were when the weights were generated. Usually the order of the scores can be taken from the header of the file which contains the weights. The method to combine the scores is defined as `linear` if weights are used or `nonlinear` if a binary file with a classifier is loaded. For many methods it is necessary to normalize the scores before combination, therefore three methods can be chosen with `--preprocessing`. `Standard` makes a Gaussian distribution, `MinMax` scales the scores between 0 and 1 and `meancomplex` divides them by their mean. The name of the output file can be defined with the argument `--output`.

C.5 Assessment of Performance and Characteristics of Scoring Functions

C.5.1 comparescores.py

To check out whether a new scoring approach was successful, fast rescoring and a performance evaluation is necessary. The program `comparescores.py` takes the

scores and a file which defines the quality of the decoys to evaluate the performance on the whole benchmark. The program can be executed on the terminal by:

```
$python comparescores.py
    --scores n_scores s1 ··· sn
    --evaluate quality.file qualitycut
    --benchmark benchmark.file n_cross
    --classification classtype classification.txt
    --atleast
    --averages
    --ROCcurve
    --plotstructures
    --ROCstructures
```

The files containing the scores can whether be files ending on `.rescore` if they contain only the pure score or files in the Attract output format `.dat`. The names of the scores are given by the argument `--scores` followed by the number of scores `n_scores` and their file names.
The qualitative evaluations of the structures are given by the use of the argument `--evaluate`. `qualitycut` defines the border to divide the structures into near-native and incorrect solutions based on the values of the `quality.file`.
For the evaluation, the structures of each complex are sorted after each score separately. Finally, a binary file is created for each score which contains a step function showing the number of near-native structures in the set for each rank. By this procedure, the evaluation must only be performed once for each score and the time consuming import for each can be avoided. (remark: files could be made smaller and readable by just storing the rank where a near-native structure can be found.)

By providing a `--benchmark`, the output windows will be divided into figures for the training and the test set. The number `n_cross` defines how the benchmark is divided. If `n_cross=1` the whole list of complexes will be taken as the training set and all other folders (complexes) which are not listed are taken as the testset. For any other value, the last $1/n\_cross$ fraction of the complexes in the list and the remaining directories in the executive folder are taken as the test set.
With `--classification` the file `classification.txt` can be used to sort the complexes after their `difficulty` or their `proteintype` which has to be defined as the classtype. If `--benchmark` and `--classification` are chosen, the program sorts the complexes after their sets but divides the bars in the `--barchart` representation

into classes.

For the visualization of the performances, various modes can be chosen. With `--Plotstructures` the step curves for each complex can be regarded. Switching between the complexes can be done by a slider. The modi `--atleast` plots the fraction of complexes for which a near-native complex can be found against its ranks. `--averages` plots the average number of near-natives in the decoy set against their rank. Furthermore, it is possible to use `--ROCcurve` to plot the fraction of near-natives against the fraction of incorrect structures. The same can also be done for each complex by `--ROCstructures`.
The argument `--%` causes that the average fraction of near-natives is plotted instead of the average number for the figure from `--averages`. By using `--double%` the rank on the abscissa in `--atleast` and `--averages` will be changed into the fraction of decoys in the set.
The argument `--barchart` creates a bar-chart for the figures of `--atleast` and `--averages`. The bar-chart shows the values for four ranks and fractions of decoys respectively. Usually, the names for the scores in the legend are taken from their files. Changing these names can be done by `--names name_0 ··· name_n`.

C.5.2 compare-nativescores.py

The program `compare-nativescores.py` plots the fraction of complexes for which a native structure was found against its rank in the decoy set. Therefore, the filenames for the scores of the decoys are given with `--scores`. For the evaluation, a file containing only the native score must be created and named after the file which contains the scores of the decoy set ending on `-native.rescore`.

```
$python compare-nativescores.py

     --scores n_scores s1 ··· sn

     --benchmark benchmark.file n_cross

     --classification classtype classification.txt
```

As in `comparescores.py` the plots can be divided into a test and a training set by the usage of the argument `--benchmark` or can be divided into classes by the argument `--classification`. Just as in `comparescores.py` the values on the abscissa can be changed from the absolute rank into the fraction of all decoys by `--double%`. As well, it is possible to use `--barchart` to plot the figure as a bar-chart for four positions on the abscissa instead as a step function.

C.5.3 parameter-comparison.py

The program `parameter-comparison.py` characterizes the scoring function by comparing them on the structures which they score well in the decoy set. Therefore, the names for the files with the scores of the structures have to be provided by `--scores`. Also a file containing a qualitative evaluation to define near-native structures has to be defined with the argument `--evaluate`.

```
$python parameter-comparison.py
    --scores n_scores s1 ··· sn
    --evaluate quality.file qualitycut
    --benchmark benchmark.file n_cross
    --classification classtype classification.txt
    --Rankerror
    --Rankcorr
    --topcorr topfraction toptype
    --symdifference
    --union
    --complement
```

The structures are divided by the `quality.file` into near-natives and incorrect solutions and sorted by their scores. The value for `qualitycut` defines the border between near-native and incorrect solutions. Using `--Rankcorr` the correlation of the ranks of the structures between the scoring functions is generated. Furthermore, `--Rankerror` generates a matrix of the mean error between the ranks of the scoring functions. Both matrices show the correlations between the ranking of the structures by the scoring functions.
To analyse the well scored structures for each scoring function, subsets of the best scored structures can be compared by `--top`. For that reason, the fraction of decoys which will be regarded has to be defined (`topfraction`$\in [0., 1]$). Furthermore, it has to be determined which structures will be regarded in the subsets, the `good`, the `bad` or `all`. For the chosen set of structures, a matrix of the `--union`, the `--symdifference` and the `--complement` between the sets from all scores is created.

Bibliography

[1] Andrusier, N ; Nussinov, R ; Wolfson, HJ: FireDock: fast interaction refinement in molecular docking. In: *Proteins* 69 (2007), S. 139–159. http://dx.doi.org/10.1002/prot.21495. – DOI 10.1002/prot.21495

[2] Aze, J ; Bourquard, T ; Hamel, S ; Poupon, A ; Ritchie, DW: Using Kendall-tau Meta-Bagging to Improve Protein-Protein Docking Predictions. In: *Lecture Notes in Computer Science, Volume 7036: Pattern Recognition in Bioinformatics* (2011), S. 284–295

[3] Bahadur, R.P. ; Zacharias, M.: The interface of protein-protein complexes: Analysis of contacts and prediction of interactions. In: *Cellular and Molecular Life Sciences* 65 (2008), Nr. 7-8, 1059-1072. http://dx.doi.org/10.1007/s00018-007-7451-x. – DOI 10.1007/s00018–007–7451–x. – ISSN 1420–682X

[4] Berman, Helen M. ; Westbrook, John ; Feng, Zukang ; Gilliland, Gary ; Bhat, T. N. ; Weissig, Helge ; Shindyalov, Ilya N. ; Bourne, Philip E.: The Protein Data Bank. In: *Nucleic Acids Research* 28 (2000), Nr. 1, 235-242. http://dx.doi.org/10.1093/nar/28.1.235. – DOI 10.1093/nar/28.1.235

[5] Bernauer, J ; Aze, J ; Janin, J ; Poupon, A: A new protein-protein docking scoring function based on interface residue properties. In: *Bioinformatics* 23 (2007), Nr. 5, S. 555–562. http://dx.doi.org/10.1093/bioinformatics/btl654. – DOI 10.1093/bioinformatics/btl654

[6] Bishop, Christopher M. u. a.: *Pattern recognition and machine learning*. springer New York, 2006

[7] Bordner, AJ ; Gorin, AA: Protein docking using surface matching and supervised machine learning. In: *Proteins* 68 (2007), Nr. 2, S. 488–502. http://dx.doi.org/10.1002/prot.21406. – DOI 10.1002/prot.21406

[8] Bourquard, T ; Bernauer, J ; Aze, J ; Poupon, A: A collaborative filtering approach for protein-protein docking scoring functions. In: *PLoS ONE* 6 (2011), Nr. 4, S. e18541. http://dx.doi.org/10.1371/journal.pone.0018541. – DOI 10.1371/journal.pone.0018541

[9] Chae, MH ; Krull, F ; Lorenzen, S ; Knapp, EW: Predicting protein complex geometries with a neural network. In: *Proteins* 78 (2010), Nr. 4, S. 1026–1039. http://dx.doi.org/10.1002/prot.22626. – DOI 10.1002/prot.22626

[10] Chakrabarti, Pinak ; Janin, Joël: Dissecting protein–protein recognition sites. In: *Proteins: Structure, Function, and Bioinformatics* 47 (2002), Nr. 3, 334–343. http://dx.doi.org/10.1002/prot.10085. – DOI 10.1002/prot.10085. – ISSN 1097–0134

[11] Chen, Rong ; Weng, Zhiping: Docking unbound proteins using shape complementarity, desolvation, and electrostatics. In: *Proteins: Structure, Function, and Bioinformatics* 47 (2002), Nr. 3, 281–294. http://dx.doi.org/10.1002/prot.10092. – DOI 10.1002/prot.10092. – ISSN 1097–0134

[12] Cheng, TM ; Blundell, TL ; Fernandez-Recio, J: pyDock: electrostatics and desolvation for effective scoring of rigid-body protein-protein docking. In: *Proteins* 68 (2007), Nr. 2, S. 503–515. http://dx.doi.org/10.1002/prot.21419. – DOI 10.1002/prot.21419

[13] Chuang, GY ; Kozakov, D ; Brenke, R ; Comeau, SR ; Vajda, S: DARS (Decoys As the Reference State) potentials for protein-protein docking. In: *Biophys J* 95 (2008), Nr. 9, S. 4217–4227. http://dx.doi.org/10.1529/biophysj.108.135814. – DOI 10.1529/biophysj.108.135814

[14] Comeau, Stephen R. ; Gatchell, David W. ; Vajda, Sandor ; Camacho, Carlos J.: ClusPro: an automated docking and discrimination method for the prediction of protein complexes. In: *Bioinformatics* 20 (2004), Nr. 1, 45-50. http://dx.doi.org/10.1093/bioinformatics/btg371. – DOI 10.1093/bioinformatics/btg371

[15] De Vries, Sjoerd J. ; Dijk, Aalt D. ; Krzeminski, Mickaël ; Dijk, Mark van ; Thureau, Aurelien ; Hsu, Victor ; Wassenaar, Tsjerk ; Bonvin, Alexandre M.: HADDOCK versus HADDOCK: new features and performance of HADDOCK2. 0 on the CAPRI targets. In: *Proteins: structure, function, and bioinformatics* 69 (2007), Nr. 4, S. 726–733

[16] De Vries, Sjoerd J. ; Dijk, Marc van ; Bonvin, Alexandre M.: The HADDOCK web server for data-driven biomolecular docking. In: *Nature protocols* 5 (2010), Nr. 5, S. 883–897

[17] Dominguez, Cyril ; Boelens, Rolf ; Bonvin, Alexandre M. J. J.: HADDOCK: A Protein−Protein Docking Approach Based on Biochemical or Biophysical Information. In: *Journal of the American Chemical Society* 125

(2003), Nr. 7, 1731-1737. http://dx.doi.org/10.1021/ja026939x. – DOI 10.1021/ja026939x. – PMID: 12580598

[18] Duhovny, D ; Nussinov, R ; Wolfson, H: Efficient Unbound Docking of Rigid Molecules. In: *Lecture Notes in Computer Science, Volume 2452: Algorithms in Bioinformatics* (2002), S. 185–200

[19] Eisenberg, David ; Marcotte, Edward M. ; Xenarios, Ioannis ; Yeates, Todd O.: Protein function in the post-genomic era. In: *Nature* 405 (2000), Nr. 6788, S. 823–826

[20] Fernández-Recio, Juan ; Totrov, Maxim ; Abagyan, Ruben: ICM-DISCO docking by global energy optimization with fully flexible side-chains. In: *Proteins: Structure, Function, and Bioinformatics* 52 (2003), Nr. 1, S. 113–117

[21] Fernández-Recio, Juan ; Totrov, Maxim ; Abagyan, Ruben: Identification of Protein–Protein Interaction Sites from Docking Energy Landscapes. In: *Journal of Molecular Biology* 335 (2004), Nr. 3, 843 - 865. http://dx.doi.org/http://dx.doi.org/10.1016/j.jmb.2003.10.069. – DOI http://dx.doi.org/10.1016/j.jmb.2003.10.069. – ISSN 0022–2836

[22] Fink, F ; Hochrein, J ; Wolowski, V ; Merkl, R ; Gronwald, W: PROCOS: computational analysis of protein-protein complexes. In: *J Comput Chem* 32 (2011), Nr. 12, S. 2575–2586. http://dx.doi.org/10.1002/jcc.21837. – DOI 10.1002/jcc.21837

[23] Fiorucci, Sébastien ; Zacharias, Martin: Binding site prediction and improved scoring during flexible protein–protein docking with ATTRACT. In: *Proteins: Structure, Function, and Bioinformatics* 78 (2010), Nr. 15, S. 3131–3139

[24] Geppert, T ; Proschak, E ; Schneider, G: Protein-protein docking by shape-complementarity and property matching. In: *J Comput Chem* 31 (2010), Nr. 9, S. 1919–1928

[25] Gonzalez-Ruiz, D ; Gohlke, H: Targeting protein-protein interactions with small molecules: challenges and perspectives for computational binding epitope detection and ligand finding. In: *Curr Med Chem* 13 (2006), Nr. 22, S. 2607–2625. http://dx.doi.org/10.2174/092986706778201530. – DOI 10.2174/092986706778201530

[26] Gray, Jeffrey J. ; Moughon, Stewart ; Wang, Chu ; Schueler-Furman, Ora ; Kuhlman, Brian ; Rohl, Carol A. ; Baker, David: Protein–protein

docking with simultaneous optimization of rigid-body displacement and side-chain conformations. In: *Journal of molecular biology* 331 (2003), Nr. 1, S. 281–299

[27] Hawkins, Gregory D. ; Cramer, Christopher J. ; Truhlar, Donald G.: Parametrized models of aqueous free energies of solvation based on pairwise descreening of solute atomic charges from a dielectric medium. In: *The Journal of Physical Chemistry* 100 (1996), Nr. 51, S. 19824–19839

[28] Hermann, Robert B.: Theory of hydrophobic bonding. II. Correlation of hydrocarbon solubility in water with solvent cavity surface area. In: *The Journal of Physical Chemistry* 76 (1972), Nr. 19, 2754-2759. http://dx.doi.org/10.1021/j100663a023. – DOI 10.1021/j100663a023

[29] Hou, Tingjun ; Wang, Junmei ; Chen, Lirong ; Xu, Xiaojie: Automated docking of peptides and proteins by using a genetic algorithm combined with a tabu search. In: *Protein Engineering* 12 (1999), Nr. 8, 639-648. http://dx.doi.org/10.1093/protein/12.8.639. – DOI 10.1093/protein/12.8.639

[30] Huang, Sheng-You: Exploring the potential of global protein–protein docking: an overview and critical assessment of current programs for automatic ab initio docking. In: *Drug Discovery Today* (2015), Nr. 0, -. http://dx.doi.org/http://dx.doi.org/10.1016/j.drudis.2015.03.007. – DOI http://dx.doi.org/10.1016/j.drudis.2015.03.007. – ISSN 1359–6446

[31] Huang, SY ; Zou, X: An iterative knowledge-based scoring function for protein-protein recognition. In: *Proteins* 72 (2008), Nr. 2, S. 557–579. http://dx.doi.org/10.1002/prot.21949. – DOI 10.1002/prot.21949

[32] Hwang, H ; Vreven, T ; Janin, J ; Weng, Z: Protein-protein docking benchmark version 4.0. In: *Proteins* 78 (2010), Nr. 15, S. 3111–3114. http://dx.doi.org/10.1002/prot.22830. – DOI 10.1002/prot.22830

[33] *Kapitel* 3. In: Jelesarov, Ilian: *Energetics of Protein–Protein Interactions*, 46-88

[34] Jorgensen, William L. ; Tirado-Rives, Julian.: The OPLS [optimized potentials for liquid simulations] potential functions for proteins, energy minimizations for crystals of cyclic peptides and crambin. In: *Journal of the American Chemical Society* 110 (1988), Nr. 6, 1657-1666. http://dx.doi.org/10.1021/ja00214a001. – DOI 10.1021/ja00214a001

[35] Kortemme, Tanja ; Baker, David: A simple physical model for binding energy hot spots in protein–protein complexes. In: *Proceedings of the National Academy of Sciences* 99 (2002), Nr. 22, S. 14116–14121

[36] Lee, Byungkook ; Richards, Frederic M.: The interpretation of protein structures: estimation of static accessibility. In: *Journal of molecular biology* 55 (1971), Nr. 3, S. 379–IN4

[37] Lensink, MF ; Mendez, R ; Wodak, SJ: Docking and scoring protein complexes: CAPRI 3rd Edition. In: *Proteins* 69 (2007), Nr. 4, S. 704–718. http://dx.doi.org/10.1002/prot.21804. – DOI 10.1002/prot.21804

[38] Lichtarge, Olivier ; Bourne, Henry R. ; Cohen, Fred E.: An evolutionary trace method defines binding surfaces common to protein families. In: *Journal of molecular biology* 257 (1996), Nr. 2, S. 342–358

[39] Liu, S ; Vakser, IA: DECK: Distance and environment-dependent, coarse-grained, knowledge-based potentials for protein-protein docking. In: *BMC Bioinformatics* 12 (2011), S. 280. http://dx.doi.org/10.1186/1471-2105-12-280. – DOI 10.1186/1471–2105–12–280

[40] Lyskov, S ; Gray, JJ: The RosettaDock server for local protein-protein docking. In: *Nucleic Acids Res* 36 (2008), Nr. Web Server issue, S. W233–238

[41] Mandell, Jeffrey G. ; Roberts, Victoria A. ; Pique, Michael E. ; Kotlovyi, Vladimir ; Mitchell, Julie C. ; Nelson, Erik ; Tsigelny, Igor ; Ten Eyck, Lynn F.: Protein docking using continuum electrostatics and geometric fit. In: *Protein Engineering* 14 (2001), Nr. 2, 105-113. http://dx.doi.org/10.1093/protein/14.2.105. – DOI 10.1093/protein/14.2.105

[42] Mashiach, E ; Nussinov, R ; Wolfson, HJ: FiberDock: Flexible induced-fit backbone refinement in molecular docking. In: *Proteins* 78 (2010), Nr. 6, S. 1503–1519

[43] May, Andreas ; Zacharias, Martin: Protein–protein docking in CAPRI using ATTRACT to account for global and local flexibility. In: *Proteins: Structure, Function, and Bioinformatics* 69 (2007), Nr. 4, S. 774–780

[44] May, Andreas ; Zacharias, Martin: Energy minimization in low-frequency normal modes to efficiently allow for global flexibility during systematic protein–protein docking. In: *Proteins: Structure, Function, and Bioinformatics* 70 (2008), Nr. 3, 794–809. http://dx.doi.org/10.1002/prot.21579. – DOI 10.1002/prot.21579. – ISSN 1097–0134

[45] Metz, A ; Ciglia, E ; Gohlke, H: Modulating protein-protein interactions: from structural determinants of binding to druggability prediction to application. In: *Curr Pharm Des* 18 (2012), Nr. 30, S. 4630–4647. http://dx.doi.org/10.2174/138161212802651553. – DOI 10.2174/138161212802651553

[46] Miyazawa, S ; Jernigan, RL: Estimation of effective interresidue contact energies from protein crystal structures: quasi-chemical approximation. In: *Macromolecules* 18 (1985), Nr. 3, S. 534–552. http://dx.doi.org/10.1021/ma00145a039. – DOI 10.1021/ma00145a039

[47] Moal, Iain ; Torchala, Mieczyslaw ; Bates, Paul ; Fernandez-Recio, Juan: The scoring of poses in protein-protein docking: current capabilities and future directions. In: *BMC Bioinformatics* 14 (2013), Nr. 1, 286. http://dx.doi.org/10.1186/1471-2105-14-286. – DOI 10.1186/1471–2105–14–286. – ISSN 1471–2105

[48] Ng, Andrew: *CS 229 Machine learning course materials, Lecture notes 1: supervised learning*. http://cs229.stanford.edu/materials.html : Stanford University,

[49] Norel, Raquel ; Lin, Shuo L. ; Wolfson, Haim J. ; Nussinov, Ruth: Shape complementarity at protein–protein interfaces. In: *Biopolymers* 34 (1994), Nr. 7, 933–940. http://dx.doi.org/10.1002/bip.360340711. – DOI 10.1002/bip.360340711. – ISSN 1097–0282

[50] Palma, PN ; Krippahl, L ; Wampler, JE ; Moura, JJ: BiGGER: a new (soft) docking algorithm for predicting protein interactions. In: *Proteins* 39 (2000), Nr. 4, S. 372–384. http://dx.doi.org/10.1002/(SICI)1097-0134(20000601)39:4<372::AID-PROT100>3.0.CO;2-Q. – DOI 10.1002/(SICI)1097–0134(20000601)39:4<372::AID–PROT100>3.0.CO;2–Q

[51] Pierce, B ; Weng, Z: ZRANK: reranking protein docking predictions with an optimized energy function. In: *Proteins* 67 (2007), Nr. 4, S. 1078–1086. http://dx.doi.org/10.1002/prot.21373. – DOI 10.1002/prot.21373

[52] Pierce, B ; Weng, Z: A combination of rescoring and refinement significantly improves protein docking performance. In: *Proteins* 72 (2008), S. 270–279. http://dx.doi.org/10.1002/prot.21920. – DOI 10.1002/prot.21920

[53] Pons, C ; Talavera, D ; Cruz, X de l. ; Orozco, M ; Fernandez-Recio, J: Scoring by intermolecular pairwise propensities of exposed residues (SIPPER): a new efficient potential for protein-protein docking. In: *J Chem Inf Model* 51 (2011), Nr. 2, S. 370–377. http://dx.doi.org/10.1021/ci100353e. – DOI 10.1021/ci100353e

[54] Qiu, Di ; Shenkin, Peter S. ; Hollinger, Frank P. ; Still, W C.: The GB/SA continuum model for solvation. A fast analytical method for the calculation of approximate Born radii. In: *The Journal of Physical Chemistry A* 101 (1997), Nr. 16, S. 3005–3014

[55] Russell, Robert B. ; Alber, Frank ; Aloy, Patrick ; Davis, Fred P. ; Korkin, Dmitry ; Pichaud, Matthieu ; Topf, Maya ; Sali, Andrej: A structural perspective on protein–protein interactions. In: *Current Opinion in Structural Biology* 14 (2004), Nr. 3, 313 - 324. http://dx.doi.org/http://dx.doi.org/10.1016/j.sbi.2004.04.006. – DOI http://dx.doi.org/10.1016/j.sbi.2004.04.006. – ISSN 0959–440X

[56] Schindler, Christina E. M. ; Vries, Sjoerd J. ; Zacharias, Martin: iATTRACT: Simultaneous global and local interface optimization for protein–protein docking refinement. In: *Proteins: Structure, Function, and Bioinformatics* 83 (2015), Nr. 2, 248–258. http://dx.doi.org/10.1002/prot.24728. – DOI 10.1002/prot.24728. – ISSN 1097–0134

[57] Schueler-Furman, Ora ; Wang, Chu ; Bradley, Phil ; Misura, Kira ; Baker, David: Progress in Modeling of Protein Structures and Interactions. In: *Science* 310 (2005), Nr. 5748, 638-642. http://dx.doi.org/10.1126/science.1112160. – DOI 10.1126/science.1112160

[58] Shrake, A. ; Rupley, J.A.: Environment and exposure to solvent of protein atoms. Lysozyme and insulin. In: *Journal of Molecular Biology* 79 (1973), Nr. 2, 351 - 371. http://dx.doi.org/http://dx.doi.org/10.1016/0022-2836(73)90011-9. – DOI http://dx.doi.org/10.1016/0022–2836(73)90011–9. – ISSN 0022–2836

[59] Still, W C. ; Tempczyk, Anna ; Hawley, Ronald C. ; Hendrickson, Thomas: Semianalytical treatment of solvation for molecular mechanics and dynamics. In: *Journal of the American Chemical Society* 112 (1990), Nr. 16, S. 6127–6129

[60] Stumpf, Michael P. H. ; Thorne, Thomas ; Silva, Eric de ; Stewart, Ronald ; An, Hyeong J. ; Lappe, Michael ; Wiuf, Carsten: Estimating the size of the human interactome. In: *Proceedings of the National Academy of Sciences* 105 (2008), Nr. 19, 6959-6964. http://dx.doi.org/10.1073/pnas.0708078105. – DOI 10.1073/pnas.0708078105

[61] Tobi, D: Designing coarse grained-and atom based-potentials for protein-protein docking. In: *BMC Struct Biol* 10 (2010), S. 40. http://dx.doi.org/10.1186/1472-6807-10-40. – DOI 10.1186/1472–6807–10–40

[62] Tobi, D ; Bahar, I: Optimal design of protein docking potentials: efficiency and limitations. In: *Proteins* 62 (2006), Nr. 4, S. 970–981

[63] Tsai, Chung-Jung ; Kumar, Sandeep ; Ma, Buyong ; Nussinov, Ruth: Folding funnels, binding funnels, and protein function. In: *Protein Science* 8 (1999), Nr. 06, S. 1181–1190

[64] Valdar, William S. ; Thornton, Janet M.: Protein–protein interfaces: analysis of amino acid conservation in homodimers. In: *Proteins: Structure, Function, and Bioinformatics* 42 (2001), Nr. 1, S. 108–124

[65] Van Dijk, Aalt D. ; Bonvin, Alexandre M.: Solvated docking: introducing water into the modelling of biomolecular complexes. In: *Bioinformatics* 22 (2006), Nr. 19, S. 2340–2347

[66] Vries, Sjoerd de ; Zacharias, Martin: Flexible docking and refinement with a coarse-grained protein model using ATTRACT. In: *Proteins: Structure, Function, and Bioinformatics* 81 (2013), Nr. 12, 2167–2174. `http://dx.doi.org/10.1002/prot.24400`. – DOI 10.1002/prot.24400. – ISSN 1097–0134

[67] Vries, Sjoerd J. ; Dijk, Aalt D. ; Bonvin, Alexandre M.: WHISCY: What information does surface conservation yield? Application to data-driven docking. In: *Proteins: Structure, Function, and Bioinformatics* 63 (2006), Nr. 3, S. 479–489

[68] *Kapitel* 9. In: Zacharias, Martin: *Scoring and Refinement of predicted Protein–Protein Complexes*, 236-271

[69] Zacharias, Martin: Protein–protein docking with a reduced protein model accounting for side-chain flexibility. In: *Protein Science* 12 (2003), Nr. 6, 1271–1282. `http://dx.doi.org/10.1110/ps.0239303`. – DOI 10.1110/ps.0239303. – ISSN 1469–896X

[70] Zacharias, Martin: ATTRACT: Protein–protein docking in CAPRI using a reduced protein model. In: *Proteins: Structure, Function, and Bioinformatics* 60 (2005), Nr. 2, 252–256. `http://dx.doi.org/10.1002/prot.20566`. – DOI 10.1002/prot.20566. – ISSN 1097–0134

[71] Zhang, C ; Vasmatzis, G ; Cornette, JL ; DeLisi, C: Determination of atomic desolvation energies from the structures of crystallized proteins. In: *J Mol Biol* 267 (1997), Nr. 3, S. 707–726. `http://dx.doi.org/10.1006/jmbi.1996.0859`. – DOI 10.1006/jmbi.1996.0859