Universität Bielefeld
Angewandte Informatik

Enhancing Protein–Protein Docking by new approaches to Protein Flexibility and Scoring of Docking Hypotheses

Dissertation zur Erlangung des Grades eines Doktors der Ingenieurwissenschaften (Dr.-Ing.)

der Technischen Fakultät der Universität Bielefeld

vorgelegt von

Frank G. Zöllner

Juli 2004

Betreuer: Prof. Dr.-Ing. Gerhard Sagerer
Prof. Dr.-Ing. Franz Kummert

Dipl.-Inform. Frank Gerrit Zöllner
AG Angewandte Informatik
Technische Fakultät
Universität Bielefeld
email: fzoellne@techfak.uni-bielefeld.de

Abdruck der genehmigten Dissertation zur Erlangung
des akademischen Grades Doktor-Ingenieur (Dr.-Ing.).
Der Technischen Fakultät der Universität Bielefeld
am 18. Juli 2004 vorgelegt von Frank Gerrit Zöllner,
am 12. November 2004 verteidigt und genehmigt.

Gutachter:
Prof. Dr. Gerhard Sagerer, Universität Bielefeld
Prof. Dr. Oliver Kohlbacher, Universität Tübingen

Prüfungsausschuß:
Prof. Dr. Jens Stoye, Universität Bielefeld
Prof. Dr. Gerhard Sagerer, Universität Bielefeld
Prof. Dr. Oliver Kohlbacher, Universität Tübingen
Dr. Sven Wachsmuth, Universität Bielefeld

Gedruckt auf alterungsbeständigem Papier nach ISO 9706

Acknowledgements

The work of this thesis was carried out during the years 2001–2004 at the Bielefeld University within the Applied Informatics Group, Technical Faculty under the supervision of Prof. Dr. Gerhard Sagerer and Prof. Dr. Franz Kummert. I would like to thank my supervisors for continuously supporting me, encouraging me and trusting in me to find my own approach to the subjects treated in this work.

Also, I am grateful to Oliver Kohlbacher for the very good support with the BALL library, used for parts of the software developed as well as for reviewing this thesis.

Furthermore, I would like to thank my colleagues of our "Bioinformatics Interest Group" - Kerstin, Steffen, Mathias, Petra, Michaela, Markus and Thomas - for the know how, the discussions, and the feedback on this interdisciplinary field of bioinformatics. Especially, I wish to express my sincere thanks to Kerstin and Steffen with whom I worked on protein–protein docking for nearly three years. Thomas inspired me to use the wavelet approach for deriving features. Also, I would like to thank him for the intense discussion on this topic. Thomas Käster and Michael Pfeiffer kindly introduced me to the field of QbC-Systems and relevance feedback; Jannik kindly proof–read this thesis.

Michaela and I shared the office for nearly the whole time being a member of the group. Thanks for the nice time and the many cups of tee we shared. Recently, Marko joined the group. He gave me good hints while finishing the thesis. I look forward working with him.

Special thanks to my wife Amélie for the support and the patience throughout this work. Proof–reading serveral versions of this thesis was a great job.

Finally, I would like to thank the people of the Applied Informatics Group and graduate program Bioinformatics for the nice time. This work was fincanced by a scholarship within the graduate program Bioinformatics provided by DFG (German Research Foundation).

Abstract

Protein docking is important for understanding the biological functions of proteins. Simulating the interaction between proteins can give insights to the mechanisms behind these functions. In many docking systems proteins are modelled as rigid bodies but in nature proteins behave differently. Especially during, docking proteins change their conformation to fit together optimally. In order to enhance docking results the flexibility of amino acid side chains has to be incorporated.
Within the scope of this thesis, a classification approach to discriminate flexible and rigid side chains is described. In order to model the flexibility, features are calculated and a support vector machine is trained. A classification of side chains can be done at high accuracy. The gained flexibility information is evaluated using the docking system ElMaR. Using the flexibility information shows improvements for most of the used test cases compared to docking them without using any information about the flexibility of the structures.

Another problem in the field of protein docking is the discrimination of true and false docking predictions. In this work, the improvement of scoring docking hypotheses is addressed. Here, a relevance feedback approach is proposed to enhance the scoring of the ElMaR docking system. For different test cases the weighting scheme could be improved so that true and false docking predictions could be discriminated at higher accuracy. An adaptation of these weights to a larger set of test cases belonging to the same enzyme class shows improvements, too.

Zusammenfassung

Für das Verständnis von biologischen Funktionen können Proteindockingverfahren angewandt werden. Die Simulation der Interaktion von Proteinen ermöglicht einen Einblick in die Mechanismen dieser Funktionen. Viele Dockingansätze modellieren Proteine als feste Körper. Proteine sind jedoch flexibel. Besonders während des Dockens verändert sich ihre Konformation um eine höhere Passgenauigkeit zu erzielen. Um die Ergebnisse von Dockingvorhersagen zu verbessern, muss diese Flexibilität modelliert werden.
In dieser Dissertation wird ein Klassifikationsansatz beschrieben, um flexible und starre Seitenketten von Aminosäuren zu unterscheiden. Merkmale werden berechnet, um die Flexibilität zu modellieren. Als Klassifikator wird eine Support Vector Machine eingesetzt. Es lassen sich gute Klassifikationsergebnisse erzielen. Die Klassifikationsergebnisse wurden zudem im Dockingsystem ElMaR evaluiert. Im Vergleich zum Docking ohne Flexibilitätsinformationen werden für fast alle Testfälle Verbesserungen erzielt.

Ein anderes Problem im Bereich Proteindocking ist die Unterscheidung von richtigen und falschen Vorhersagen. In dieser Arbeit soll die Bewertung von Dockinghypothesen des ElMaR Systems verbessert werden. Der hier vorgestellte Ansatz beruht auf Relevance Feedback. Für verschiedene Testfälle kann das Gewichtungsschema verbessert werden, so dass eine bessere Bewertung möglich ist. Eine Adaptierung der modifizierten Gewichte auf Testfälle der selben Enzymklasse zeigt ebenfalls Verbesserungen in der Bewertung.

Contents

Chapter 1
Introduction

Motivation

In the beginning of the 21st century molecular biology has become an emerging field in science. Increasing economic impact on molecular genetics, biochemistry, medicine, and pharmaceutics have driven the research in these fields fast forward. New methods from bioinformatics provide powerful tools so that sequencing whole genomes can be done in an industrial size and manner (Venter *et al.*, 2001) nowadays. The number of sequenced genomes rises fast resulting in a huge amount of data to be analysed.

Thus, the post genomic area becomes more and more emerging. In order to understand or even to simulate whole cells the interaction between the genome, the proteome, and the metabolome has to be analysed (Thornton, 2003; Lengauer *et al.*, 1999).

Therefore, Proteins play an important role as they are involved in many biological systems, e.g. cell stability, immune defence, catalysis, signal transduction, or DNA transcription. The function and the mechanism of proteins are the main keys to describe the metabolic network(s) of a cell at least. The function of proteins can be determined by the analysis of gene expressions (Martínez-Cruz *et al.*, 2003; Greenbaum *et al.*, 2003) or sequence comparison (Ward, 2001), whereas the mechanism behind a protein's function can only be solved by analysing the protein's structure.

The information gained from this analysis can be applied to different fields of life sciences, e.g. drug targeting or design. Knowing the mechanism and the structure of a protein, specific drugs can be built which are more competitive to the ligand in nature and bind optimally to the protein.

Another question in drug design is to built a molecule that precisely binds to the chosen target so that cross reactions will be minimised. Protein docking can help solving this problem. Simulating the binding of two molecules can give information about the docking process. Screening a large library of structures in a 1:N docking scenario enables drug targeting and shows up potentially side reactions of the examined molecule. In order to receive good results the modelling of the docking algorithm is important. First algorithms in the field describe a protein as a rigid body. But the rigid body assumption does not hold as proteins change their conformation, especially during docking. Therefore, the flexibility of proteins has to be taken into account to improve the results.

Protein Docking

Protein docking describes the binding of a molecule to a protein. There are two types of protein docking: protein–ligand docking and protein–protein docking. In protein–ligand docking, the ligand is usually a small organic molecule or a short peptide. An example for a protein ligand docking system is FlexX (Rarey, 1996). Protein–protein docking is the binding of two proteins. In this thesis only protein–protein docking is considered.

Protein–protein docking can be divided into two groups of applications: the bound and the unbound docking. In the case of bound docking a known protein complex is taken and then split into its parts. These components are then re-docked using a docking algorithm. This kind of docking is favourable for testing purposes (e.g. see Ackermann *et al.*, 1998). The more challenging task is unbound docking in which two proteins with native conformations are docked.

In the beginning, proteins have been modelled as rigid bodies. The main assumption was the key–lock principle (Fischer, 1894). Fischer stated that the enzyme specificity is based on geometric complementarity of the enzyme's binding site and the ligand. So that they fit like a key and lock. First algorithms strictly used this assumption (c.F. Connolly, 1983*b*).

In 1958, Koshland (Koshland, 1958) discovered that proteins do not behave like a key and a lock during docking but perform small conformational changes, called "induced fit".

Besides six degrees of freedom (3 through translation and 3 through rotation) of the rigid molecule a vast number of additional variabilities of a protein structure arises. In addition to side chain changes, also movements of larger parts (domains) of a protein have been reported (Gerstein *et al.*, 1994). Searching the whole conformational space is infeasible and therefore new search strategies have been deployed (Ewing *et al.*, 2001; Ackermann *et al.*, 1998; Walls & Sternberg, 1992; Lenhof, 1997). Scoring functions using additional physico–chemical features have been developed to rank the solutions provided by the algorithms. But most of these algorithms neglect the flexibility of proteins and so fail to predict good docking constellations. In order to enhance these approaches it is necessary to incorporate flexibility information.

Therefore, the goal of this thesis is to analyse the flexibility of side chains and to model this flexibility in order to improve rigid body docking algorithms.

Flexibility Approach

The approach described in this work models the flexibility of amino acid side chains in order to incorporate this information into rigid body docking algorithms. Amino acids are classified as "flexible" or "non–flexible" based on energy criteria. Besides these, also other features like the solvent accessible surface area are used. The flexibility is calculated on unbound proteins and is independent of the ligand. It is described by numbers, representing the two classes: 0 for "non–flexible" and 1 for "flexible" residues. A docking algorithm can use this information to flexibilise it's scoring scheme. The flexibility information is calculated

independently from the docking algorithm and therefore does not influence the running time of the algorithm. In order to classify the residues, a threshold based classifier is used, initially. Furthermore, a support vector machine is trained to incorporate more features specific to residue flexibility.

The results of the flexibility predictions are evaluated on a test set of protein complexes and their unbound partners which are automatically derived from the Brookhaven Protein Database (PDB) (Bhat *et al.*, 2001). With respect to the threshold based approach Receiver Operating Characteristic (ROC) analysis is used for evaluation. The support vector machine is evaluated by a 10–fold cross validation.

In a second evaluation procedure the flexibility information is incorporated in the docking system ELMAR (Neumann *et al.*, 2002). ELMAR is an extension to the algorithm proposed by Ackermann (Ackermann *et al.*, 1998). Docking experiments are conducted to estimate the impact of the flexibility information on the results of the docking algorithm.

Scoring Docking Hypotheses

The results produced by a docking algorithm (called docking hypotheses) have to be scored in order to evaluate the accuracy of the algorithm and to discriminate good from bad predictions of a protein complex.

While the problem of searching the high dimensional conformational space to create docking hypotheses has been solved by various methods, the scoring of them is still not satisfying, especially the discrimination between false positive and true hypotheses as stated by Halperin and coworkers (Halperin *et al.*, 2002):

> *"Although some algorithms are able to rank correct solutions within the top hundred or even within the top ten places for some predictive docking cases, for most complexes the highest ranked structures are still false positives, i.e., solutions with a high RMSD from the complex, a high score, and a low rank."*[1]

A second goal of this thesis is to show a different approach to overcome the scoring problem. In order to discriminate false positive from true docking hypotheses an approach is presented that uses expert knowledge without modelling it explicitly in the scoring of the docking algorithm. Therefore relevance feedback techniques are adapted from Query–by–Content retrieval systems (QbC).

Structure of the Thesis

Following this introduction, chapter 2 introduces the biochemistry and structure of proteins. Furthermore, amino acids, the building blocks of a protein are described, including

[1] The RMSD is the root mean square deviation. Here, it is calculated between a hypothesis and a grounded truth, a known complex (see section 7.3.2).

a description of calculating the side chain torsion angles. In chapter 3 different kinds of flexibility within proteins are outlined and recent approaches to side chain flexibility are described. This chapter closes with a discussion of the presented approaches. In the next chapter the ElMaR docking system is introduced. The results of the flexibility classification are evaluated within this docking system. Therefore, the principles of the docking algorithm and the interface for incorporating flexibility information are described. After that, energy based approaches to side chain flexibility are presented. Besides a threshold based classifier, the utilisation of a support vector machine is outlined. In order to train the classifier, features describing the residues have to be extracted. These features are outlined as well.

A second goal of this thesis is to enhance the scoring of ElMaR. In chapter 6 an approach using relevance feedback to estimate better parameters for the scoring function of ElMaR is shown. Subsequently, in chapter 7 the results of the different approaches are presented. The thesis closes with a conclusion and an outlook to further work is given.

Chapter 2
Biochemistry and Structure of Proteins - A Short Introduction

In this chapter a brief introduction to the structure and the biochemistry of proteins is given. First the structure of the smallest parts of a protein, the amino acids, is described (see section 2.1). Then, the structure of a protein is outlined. The last section of this chapter describes the forces within and between different proteins which are responsible for flexibility and the interaction of proteins during docking.

2.1 Amino Acids

The structure of an amino acid can be divided into two components: the *backbone* and the *side chain*. The first part is similar in all amino acids containing two functional groups: the amino group (NH_2) and the carboxyl group (COOH). They are connected via a carbon atom (C_α). The second part, the side chain (R), defines the specificity of the amino acid (see Fig. 2.1).

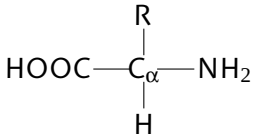


Figure 2.1: *Structure of an amino acid, R denotes the side chain.*

In nature, there exist twenty different types[1] of amino acids (see Appendix C). According to the structure of the side chain, the amino acid can be grouped into apolar, polar, uncharged, or charged side chains (see Stryer, 1996, p. 46).

The geometry of the side chain is determined by the torsion angles of the bonds mediating the atoms of the side chains. The number of torsion angles of a side chain ranges from zero

[1] Besides these, two other amino acids have been found, selenocystein and pyrrolysine. Both are based on standard amino acids (serine and lysine) which are enzymatically modified while attached to a tRNA (Atkins & Gesteland, 2002). Since these two residues are very special, they are not considered in this work.

for Glycine (GLY) to four in case of Arginine (ARG) and Lysine (LYS). The torsion angles are calculated using the coordinates of four surrounding carbon atoms (of the side chain) to set up two planes (see Fig. 2.2). The planes are set up using the vectors $\vec{v_1}, \vec{v_2}$ and $\vec{v_2}, \vec{v_3}$. The angle between the intersecting planes defines the torsion angle (χ). It is equivalent to the angle of the intersecting normals $\vec{a}$ and $\vec{b}$ of the planes:

$$\chi = \arccos \frac{\langle \vec{a}, \vec{b} \rangle}{\|\vec{a}\| \|\vec{b}\|} \tag{2.1}$$

Because the arccos function is only defined on the interval $[-1, 1]$ and takes values from 0 to π, the sign has to be calculated to decide whether a torsion angle lies within the range $[-\pi, 0]$ or $[0, \pi]$. The sign (s) of the torsion angle can be determined by:

$$s = \operatorname{sgn}(\cos x) = \frac{\langle \vec{v_2}, (\vec{a} \times \vec{b}) \rangle}{\|\vec{v_2}\| \|(\vec{a} \times \vec{b})\|} \tag{2.2}$$

Here, $\cos x$ describes the orientation of the normals to each other. A positive value states a parallel orientation of the normals and the sign of χ is positive. A negative value thus determines a negative sign for χ.

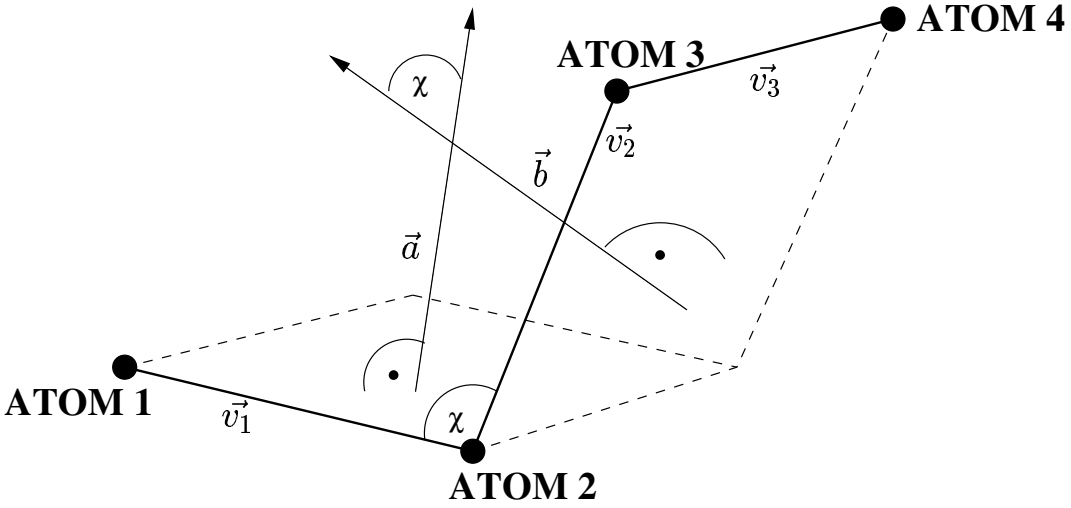


Figure 2.2: *Calculation of the torsion angles. The thick, solid line represents the bonds, atom 1 to 4 the carbon atoms of the side chain. The dashed lines are drawn to visualise the planes.*

Their subscripts are enumerated according to their position in the side chain, e.g. the first torsion angle χ_1 describes the rotation of the bond between the C_α and the first side chain carbon atom, called C_β (see Fig. 2.3).

In the case of Glycine (see Fig. C.8) the side chain is comprised of only one hydrogen atom and therefore a calculation of a torsion angle cannot be performed. Alanine (ALA) has no torsion angles because its side chain consists only of a methyl group (CH_3). Again like for Glycine the number of carbon atoms for calculating a torsion angle is not sufficient (see Fig. C.1). The side chain of Proline (PRO) is special because it is bound to the backbone

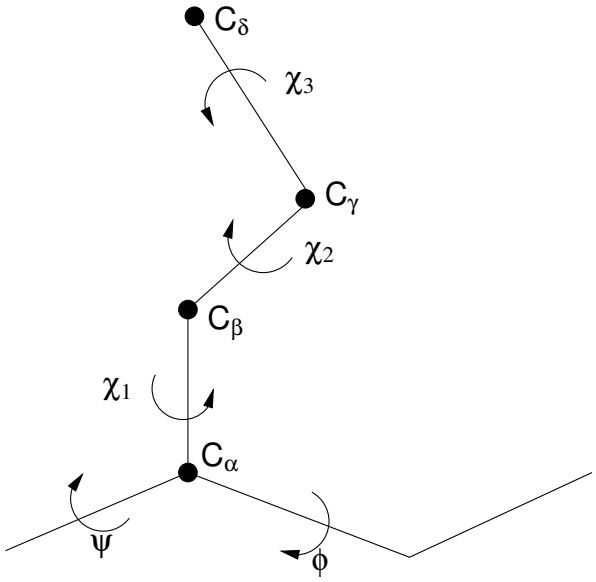


Figure 2.3: *Naming of the torsion angles. Here, also the backbone torsion angles ϕ and ψ are shown.*

forming a loop (see Fig. C.15). Due to these special properties, these three amino acids are not considered in the flexibility predictions presented here.

If an amino acid is solvated, it becomes a *zwitterion* which means that the carboxyl group looses a hydrogen atom to the solvent whereas the amino group receives an extra hydrogen. This results in a doubly charged molecule, carrying a positive and a negative charge at the same time. In this state, two amino acids can perform a reaction emitting a water molecule to the solvent, forming a dipeptide. The bond between the two amino acids is called *peptide bond*. It is planar and inflexible and therefore influences the three-dimensional structure of a protein.

If other amino acids successively bind to the dipeptide, a polypeptide chain is created. If this chain is longer than 35 amino acids it is called a *protein*. Shorter chains are referred to as *peptides*.

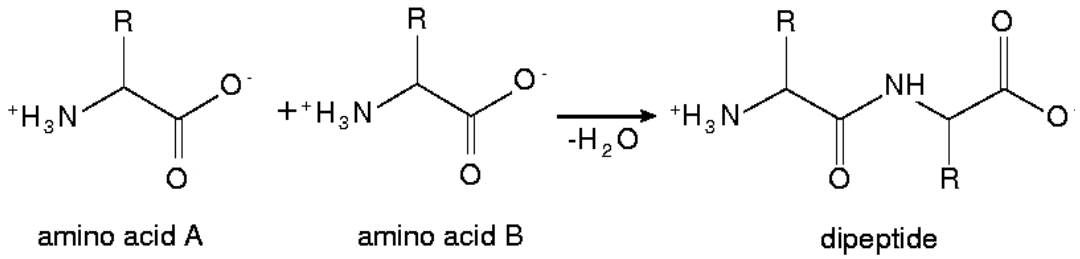


Figure 2.4: *Reaction of two amino acids forming a dipeptide.*

2.2 Proteins

Proteins are built up from amino acids which are also referred to, in this context, as *residues*. Like amino acids, a protein chain has an amino group at one end, the so called *N-terminus* and a carboxylic group on the other, the *C-terminus*. The *backbone* of a protein is defined by the repeated sequence of the atoms N (of the amino group), C_α, and the C of the carboxylic group. In figure 2.5 the backbone is highlighted by the coloured ball and stick model. The tube in shiny blue illustrates the peptide chain.

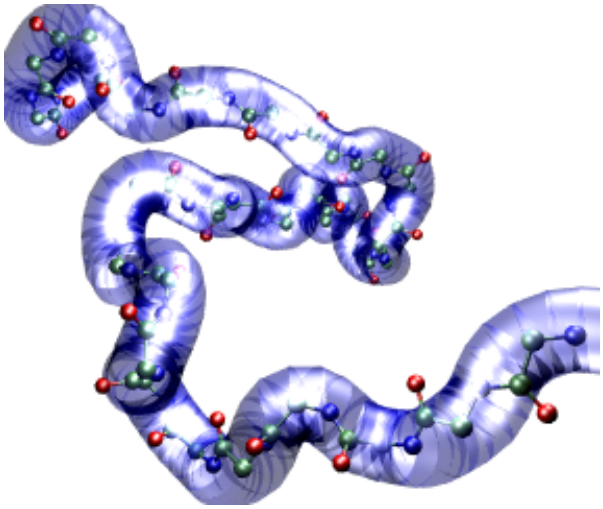


Figure 2.5: *Backbone of a protein. The backbone carbon atoms are coloured in green and the nitrogen is coloured in blue. The red balls represent oxygen atoms.*

The main characteristic of a protein is its well-defined three-dimensional structure which specifies the function. The backbone, C-terminus, and N-terminus form the *primary structure* of a protein. Besides the primary structure, there exist other important structural elements of a protein. The *secondary structure* consists of folded parts of the primary structure. There are two main types: the α*–helix* (see Fig. 2.6) and the β *pleated sheet* (see Fig. 2.7). They can occur on different sections of the primary structure depending on the sequence of the amino acids.

An α–helix has a regular and tight rod like structure. The inner part of the rod is formed by the backbone atoms of the polypeptide chain. The side chains of the residues extend to the outside, away from the backbone. An α–helix is stabilised by hydrogen bonds between the amino (NH) and the carbonyl group (CO) of the backbone atoms (see Fig. 2.6). The CO group of each residue is connected to the NH group of the fourth successor of the amino acid sequence. The connection between two residues of an α–helix is defined by a rise of 1.5Å and a rotation of 100°. Therefore a turn of an α–helix consists of 3.6 residues. Its rotational direction is clockwise (right–handed) for most proteins. Besides this regular α–helix, there exist other special types like 3_{10}–helix or π–helix.

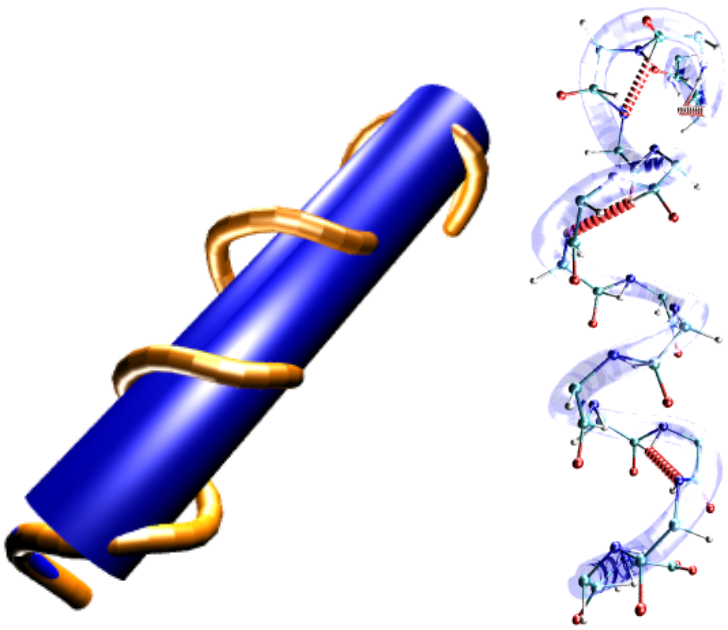


Figure 2.6: *Scheme of an α–helix. On the left side a schematic figure, on the right the residues are added for convenience. The thick red lines indicate the stabilising hydrogen bonds.*

In contrast to an α–helix, a β (pleated) sheet (also called β–*strand*) is long and planar, and the polypeptide chain is almost fully extended. The distance between adjacent residues is about 3.5Å and a β–sheet is stabilised by hydrogen bonds between NH and CO groups in different strands (see Fig. 2.7). Adjacent chains of a β–sheet can run in parallel and anti-parallel direction.

A *super secondary structure* or *motif* is a certain arrangement of two or more adjacent secondary structure elements. Examples are the helix–turn–helix, helix–loop–helix, or the hairpin β motif. Brandon and Tooze (Brandon & Tooze, 1999) give a detailed description of all common motifs found in proteins.

Several motifs can be combined forming a *domain*. Domains are compact globular structures of a protein that usually carry a certain function. Proteins can have more than one domain, each with a different function. These proteins are also called *multi–domain proteins*. Protein structures can be classified according to their domain and motif structures. There are three main groups: α domains, β domains, and α/β domains. Databases like SCOP (Murzin *et al.*, 1995) or Cath (Orengo *et al.*, 1997) classify proteins into families according to this nomenclature.

The totally folded three-dimensional structure, including all secondary structure elements and domains is called *tertiary structure* (see Fig. 2.8(a)). The term tertiary structure is therefore a "container" describing the whole three-dimensional fold of the polypeptide chain.

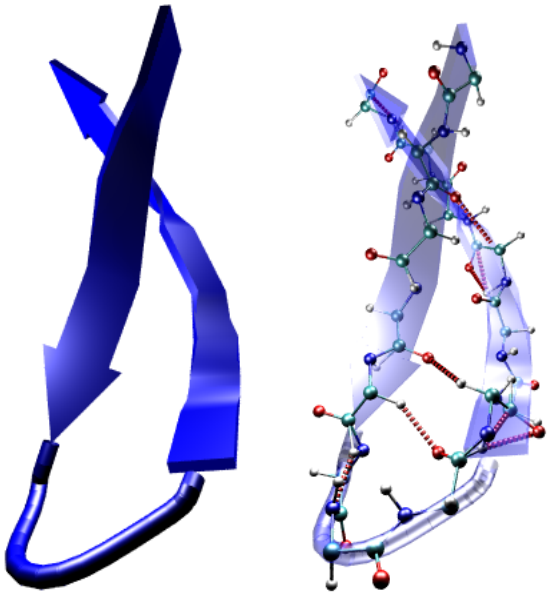
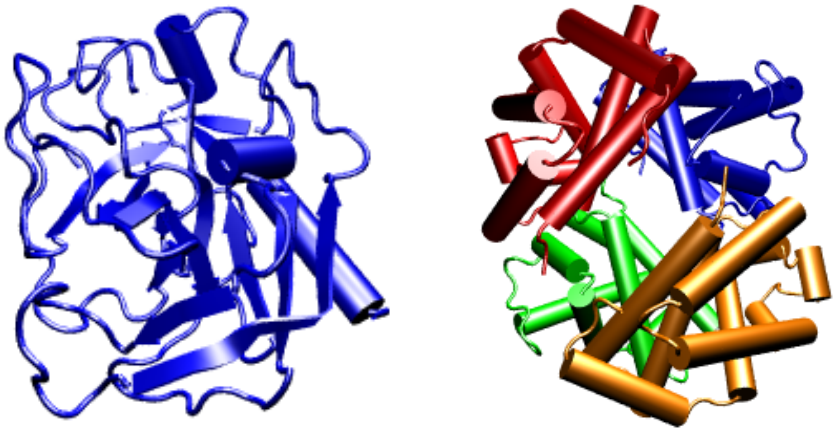


Figure 2.7: *Antiparallel β–sheet. On the left side a schematic figure, on the right the residues are added for convenience. The thick red lines indicate the stabilising hydrogen bonds.*



(a) Tertiary structure of a protein. Here, Trypsin (taken from PDB code 1TAB) is shown.

(b) Quaternary structure of a protein. Here, Deoxyhemoglobin A (PDB code 1A00), involved in oxygen transport is visualised. This protein is built up of four chains.

Figure 2.8: *Tertiary (left) and quaternary structure (right) of proteins.*

Sometimes, a protein does not only consist of one polypeptide chain but is build up from many subunits (polypeptide chains). Together, the different parts enable a certain biological function. The three-dimensional formation of these chains is called *quaternary structure* (see Fig. 2.8(b)).

2.3 Inter- and Intramolecular Forces

Besides the structure of a protein, the forces within and between proteins are important. Intramolecular forces determine the stability of a structure whereas intermolecular forces determine the interactions between structures. Both types of forces influence the flexibility of a structure. In this chapter these two types of forces are described. Here, only an overview is given. In detail discussion of the different forces and their modelling is described by Goodman (Goodman, 1998) or Leach (Leach, 1996).

2.3.1 Bonded Interactions

Bonded interactions are of intra-atomic type. They are determined by the mediating bonds of two atoms. There are three different types, the bond stretching, angle bending, and torsional variations (rotation of a bond). Figure 2.9 illustrates the three types. These intra-

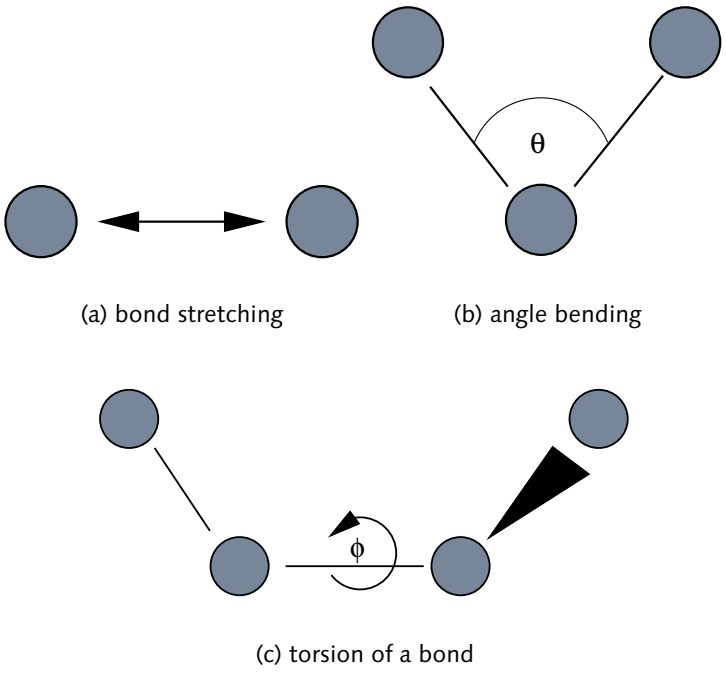


Figure 2.9: *Three types of bonded interactions in a molecule.*

atomic forces do not occur alone but in combination, e.g. stretch–torsion or stretch–bend as well as stretch–stretch in case of three connected atoms. The bond stretching energy can be simply described by the *Hook's law*:

$$E_{stretch} = K_{ij}^{stretch}(r_{ij} - r_{ij}^{eq})^2 \tag{2.3}$$

where i, j are the corresponding atoms of the bond. $K_{ij}^{stretch}$ is the bond stretching constant specific for the bond i, j, and r_{ij} is the current distance between i and j or the bond length, r_{ij}^{eq} is the equilibrium bond length. Similar to this, the angle bending can be described as a *harmonic potential*, here using the angle θ:

$$E_{bending} = K^{bending}(\theta - \theta^{eq})^2 \tag{2.4}$$

The torsion energy depends on the angle ϕ. This energy can be modelled by a cosine function

$$E_{torsion} = \frac{V_n}{2}[1 + \cos(n\phi - \gamma)] \tag{2.5}$$

where V_n gives a qualitative indication of relative barriers to the rotation. Here, n is the multiplicity, giving the number of minimum points in the function as the bond is rotated through 360° and γ is the phase factor determining where the torsion angle passes its minimum value. For details about the parameters refer to Leach (Leach, 1996) or Goodman (Goodman, 1998).

2.3.2 Non–bonded Interactions

Besides the intra–atomic forces, there are also non–bonded interactions: electrostatic and van der Waals. As the term non-bonded indicates, these interactions are not bound to bonds but to the interactions between atoms or molecules.

Electrostatic interactions occur between charges. Atoms consist of two charged elementary particles, the protons and the electrons. Protons are positively charged whereas electrons carry a negative charge. Atoms with a different number of protons and electrons are called *ions* bearing a positive or negative net charge. But also atoms with an equal amount of protons and electrons may have a charge distribution that lead to regions of positive or negative charges, so called *partial charges*. The interaction of these charges can be calculated by the *Coulomb's law*:

$$E_{ES} = \frac{1}{4\pi\varepsilon_0}\frac{q_1 q_2}{r} \tag{2.6}$$

E_{ES} is the energy resulting of the charges q_1, q_2 (e.g. of two atoms) having a distance of r. The constant factor ε_0 describes the permittive vacuum. An electrostatic interaction can be attractive (+-) or repulsive (- -), according to the signs of q_1 and q_2. The force of this energy is given by the gradient of the energy:

$$\begin{aligned} \vec{F_{ES}} &= -\nabla E_{ES} \\ &= -\frac{1}{4\pi\varepsilon_0}\frac{q_1 q_2}{r^3}\vec{r} \end{aligned} \tag{2.7}$$

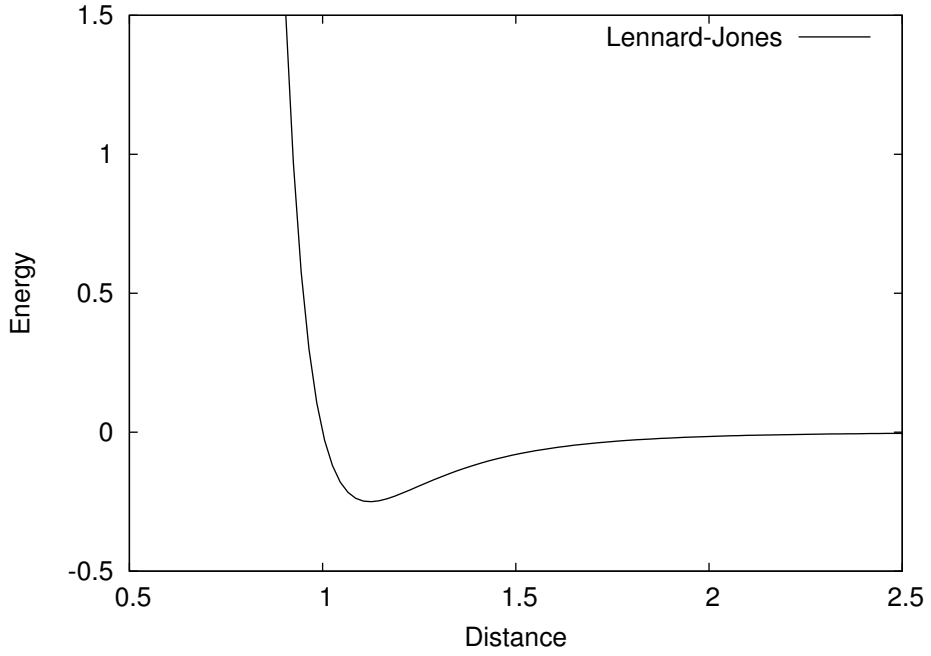


Figure 2.10: *Lennard–Jones potential, the van der Waals energy is distance dependant.*

The van der Waals interaction describes non–bonded interactions consisting of an attractive and a repulsive part. On bases of induced dipole – dipole effects charges fluctuate to neighbouring atoms leading to an attractive electrostatic interaction. Simultaneously, a repulsive force occurs resulting from the *pauli exclusion principle* due to unfavourable energies of overlapping or inter-penetrating electron clouds of the two approaching molecules. The interplay of these two forces leads to an intermolecular potential function, called *Lennard–Jones potential*.

As shown in figure 2.10 the van der Waals energy is nearly zero for great distances of two atoms or molecules. At intermediate distance the energy is negative resulting in an attractive force whereas for short distances the energy is exponentially high resulting in a strong repulsion. The most common description of this potential is given by

$$E_{vdW} = \frac{A}{r^{12}} - \frac{B}{r^6} \tag{2.8}$$

where A and B depend on the atoms involved and r is the distance between them.

Chapter 3
Flexibility within Proteins

In the following chapter, the flexibility within proteins is described. Already in 1958, Koshland (Koshland, 1958) analysed the specificity of enzymes. On different examples he showed that the "key and lock principle" does not explain all enzyme reactions. Thus, he proposed conformational changes occurring during the enzyme reaction, enabling an interaction with the substrate. He also stated that this change is induced by the substrate. Further analyses of protein structures revealed two types of flexibility: domain movements and side chain flexibility.

Although the main focus in this thesis is side chain flexibility, a brief introduction to domain movements is given in section 3.1. Side chain flexibility is described in section 3.2. Here, an overview of recent research work on side chain flexibility is included, additionally. Knowledge about side chain flexibility is worthwhile because it can be used to enhance rigid body docking algorithm, resulting in more precise predictions of complex structures. An overview about docking systems modelling flexibility is outlined in section 3.3. A discussion of the different approaches closes this chapter.

3.1 Domain Movements

Domain flexibility is the movement of larger parts of a protein, e.g. motifs or even domains (see section 2.2). In contrast to side chain flexibility these movements include a conformational change not only within single residues, the backbone is influenced as well. Domain movements typically occur at a hinge point, allowing the structure on the left and on the right of this point to move (see Fig. 3.1). Exemplarily, in case of T4 lysozyme (Faber & Matthews, 1990), in the catabolite gene activator protein (Weber & Steitz, 1987) as well as on binding flexible ligands (Urzhumtsev *et al.*, 1997) and antigen–antibody binding (Rini *et al.*, 1990) domain movements have been reported. Gerstein and colleagues (Gerstein *et al.*, 1994) have analysed and classified domain movements into two groups, shear and hinge bending. Recently, Echols and coworkers (Echols *et al.*, 2003) set up a database called "Molecular Movements Database" collecting domain movements. These are classified according to their type of motion (shear or hinge) and according to their domain type using CATH (Orengo *et al.*, 1999). Docking algorithms modelling domain movements are proposed by different researchers (Sandak *et al.*, 1998; McCammon *et al.*, 1976; Mao & McCammon,

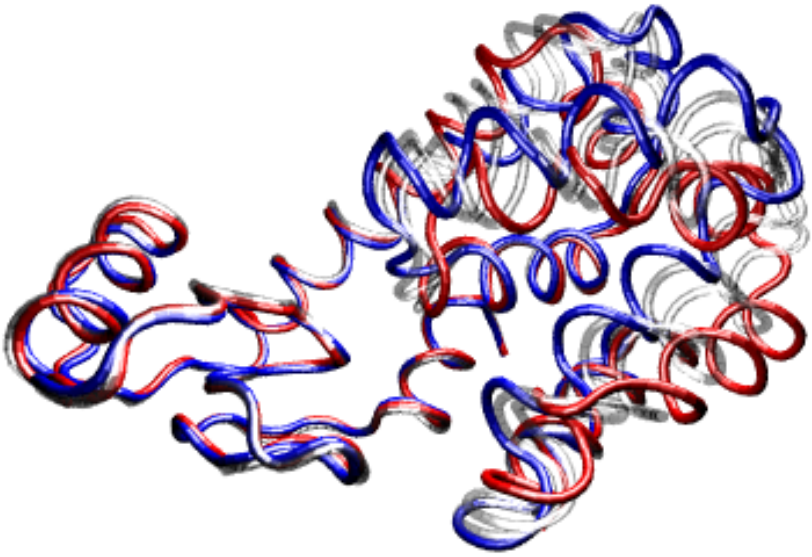


Figure 3.1: *Domain movement of T4 lysozyme mutants at a hinge point (taken from Molecular Movements Database (Echols* et al.*, 2003)). In this figure,several steps from an animation of the hinge move are superimposed. The starting conformation is given in blue, the final conformation is coloured red, intermediate steps are shadowed in grey. The left part remains rigid whereas the right part changes differently.*

1984; Colonna-Cesari *et al.*, 1986). Sandak for instance, uses a General Hough Transformation to simulate the domain movements. In a preprocessing step so called *hinge points* have to be defined as reference points. At each hinge point full three-dimensional rotation of the parts attached to the hinge is allowed. Docking hypotheses are then scored by a voting scheme.

3.2 Side Chain Flexibility

Side chain flexibility in contrast to domain flexibility is bound to local changes within the conformation of the residues. It usually occurs on the surface and around the active site of the protein.

Conformational changes within residues can only occur at the torsion angles of the side chain and at the backbone angles ϕ and ψ (see Fig. 2.3, page 7). Since a rotation of a torsion angle around 360° is (theoretically) possible, the angle space is discretised into so called *rotamers*. According to IUPAC (IUPAC-IUB Commission on Biochemical Nomenclature (CBN), 1967), the rotamers are defined by the ranges as given in table 3.1.

The first row of table 3.1 shows the angle ranges based on the hybridisation of the carbon atom connected to the rotated bond. Here, it has a sp^3 hybridisation which means that atoms connect via bonds to this carbon atom are placed at the corners of a tetrahedron. This

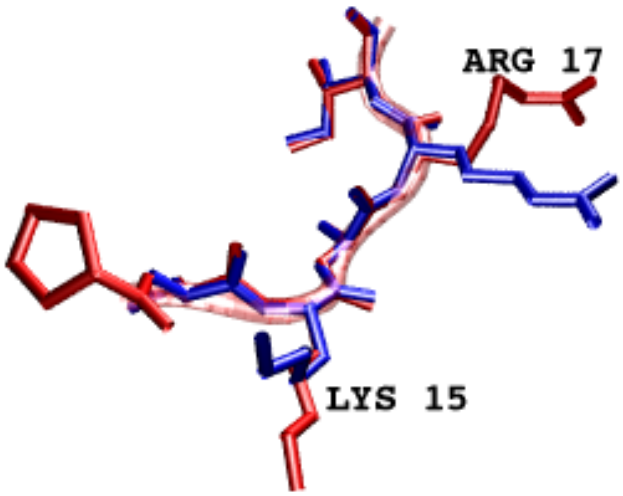


Figure 3.2: *Side chain flexibility on the example of a bovine trypsin inhibitor (1BPI). The original structure (blue) is superimposed onto the corresponding part (red) bound to a beta-trypsin (2PTC). Differences in the side chain conformation can be observed in front (LYS 15) and on the right (ARG 17).*

hybridisation occurs for the χ_1 torsion angle of all residues, the χ_2 of Arginine, Glutamine, Glutamic acid, Isoleucine, Leucine, Lysine and Methionine as well as on the χ_3 of Methionine and the χ_3 and χ_4 of Arginine and Lysine. The second row of table 3.1 defines rotamers based on a sp^2 hybridisation of the torsion angles which corresponds to planar structures of adjacent bonds of the carbon atom. This hybridisation can be found in branched side chains like Asparagine, Aspartic acid, or Glutamine and Glutamic acid. The last row of table 3.1 describes rotamers of side chains with ring systems (e.g. PHE). Here, a planar structure of the bonds and atoms connected to the C_β carbon atom is also present. Due to these steric features of the side chains' ring system, only two rotamers can be observed (see Koch, 2003).

First research work analysing residue conformation was carried out by Janin and Wodak (Janin & Wodak, 1978). They compared the distribution of torsion angles of a small set of 19 protein structures from PDB to energy landscapes received by simple energy calculations on a

hybridisation	g^-	t	g^+
sp^3	0–120°	120–240°	240–360°
sp^2	30–90°	330–360°,0–30°	270–330°
sp^2	30–150°	330–360°,0–30°	

Table 3.1: *Definition of the rotamer ranges. In the first row, rotamers according to IUPAC nomenclature are shown, the second and third row show additional rotamer definitions according to Dunbrack and Karplus (Dunbrack & Karplus, 1993).*

tri-peptide ALA-X-ALA where X is the residue in question. The used energy function consisted of two terms, a torsion angle potential and the van der Waals potential.

The results from the energy calculation on the residue conformation correlate with the distributions of the torsion angles, e.g. a high frequent torsion angle value corresponds to a low energy value, whereas rotamer boundaries correlate to high energy bounds. The distribution of the χ_1 torsion angle is tri modal for all residues, favouring the g^+ rotamer, whereas the distribution for the χ_2 angle showed different characteristics for the different side chains (e.g. branched, aromatic, etc).

From the distributions of torsion angles probabilities for a certain conformation can be derived. This information is compiled into so called *rotamer libraries*. Several different rotamer libraries have been set up (Bower *et al.*, 1997; Lovell *et al.*, 2000; Ponder & Richards, 1987; Tuffery *et al.*, 1997). The libraries differ in the amount of used data (usually unbound structures), the method used to calculate the probabilities (e.g. Dunbrack and Bower uses Bayesian statistics and some hyper distributions to fit the probabilities, whereas Tuffery et al. describe their rotamers from cluster analysis) and whether the backbone torsion angles are included or not. In the first case the libraries are called *backbone dependent*, in the latter *backbone independent*. They are mainly applied in folding task or used for conformational sampling (Althaus *et al.*, 2002).

Rotamer libraries have been extended by Schrauber and coworkers (Schrauber *et al.*, 1993). They analysed the rotamericity of side chains to improve the rotamer library of Ponder and Richards (Ponder & Richards, 1987). A torsion angle is considered as rotameric if it does not differ more than 20° from the mean of the rotamer.

Koch (Koch, 2003) instead compiled a rotamer library especially for the protein–protein docking. In contrast to the other rotamer libraries the protein structure data is divided into complexes and unbound structures. The probabilities for the side chain conformations are calculated using a so called *language model*, a statistical approach used within the field of speech recognition, enabling precise estimates of rotamer probabilities for higher torsion angles (χ_3 and χ_4). On the basis of these distributions flexibility information is derived by comparing bound and unbound structures.

Most approaches to side chain flexibility are based on comparison of bound and unbound protein structures. Hubbard and Thornton (Hubbard *et al.*, 1991) analysed the conformational changes of proteolytic sites and compared them to serine proteinase inhibitors in bound state. They used a least–squares algorithm to superimpose the structures. Parameters like main-chain torsion angles, accessibility, mobility, and protrusion indices have been calculated. Hubbard et al. stated that for cleavage of these structures by the serine proteinase the proteolytic sites have to alter their conformation radically. Betts and Sternberg (Betts & Sternberg, 1999) compared complex and unbound structures also by super-imposition.

Zhao and colleagues (Zhao *et al.*, 2001) analysed side chain flexibility within unbound protein structures. Therefore they paired homologous proteins and compared their torsion angles. Side chain flexibility was evaluated by plotting the distribution of torsion angles as histogram and plotting the torsion angles of each pair and residues. The histogram shows the already known distributions described by Bower and his colleague Dunbrack (Bower

et al., 1997) or Janin (Janin & Wodak, 1978). The plotting of the paired residues shows differences of the residues torsion angles (points off the diagonal) within the paired proteins. Significant levels are set up for each amino acid type to reflect its environments and structure.

Najmanovich and coworkers (Najmanovich *et al.*, 2000) analysed changes on receptor proteins upon ligand binding. On a test set of bound and unbound protein structures they investigated the flexibility of side chains of residues in the active site. Najmanovich stated that only few residues within the binding pocket change their conformation upon binding but within these large and polar residues (e.g. LYS, ARG) tend to be more flexible than other amino acid types.

Beside methods comparing protein structures (either unbound or bound structures) the influence of the environment to side chain flexibility is analysed. McGregor and coworkers (Mc Gregor *et al.*, 1987) examined the influence of the secondary structure to side chain conformations. They stated that within the fixed and well ordered structures of helices or β–sheets the distribution of side chain torsion angles changes significantly towards one rotamer in favour. The first torsion angle (χ_1) is influenced most due to its short distance to the backbone but also higher torsion angles (χ_2, χ_3, χ_4) are influenced.

Koch (Koch, 2003) analysed within her Phd thesis also the influence of the secondary structure to side chain flexibility. In contrast to the work of McGregor, she also included amino acids at the end of a helix or sheet. On comparing three cases of residue environments Koch stated that the more restricted the environment is, the less flexible the residues are.

Statistics on amino acid conformations are also required in the field of structure prediction. In the field of homology modelling preferences of rotamer combinations are helpful to build valid models. Ogata and Umeyana (Ogata & Umeyana, 1998) analysed the influence of environmental residues to torsion angles within homologous proteins. Side chain conformations are modelled using principle components calculated on residues atoms.

Wilson and coworkers (Wilson *et al.*, 1993) used an energy based rotamer search to find an optimal rotamer combination while modelling homologous proteins. Beside force field calculations including a solvation term the conformational searching is started from a rotamer library providing side chain conformations. Side chains in different conformations (according to the rotamer library) are placed around a center residue which is chosen at random. Then iteratively for this environment the globally best combination of side chain conformations is searched using the force field as score function.

Leach and Lemon (Leach & Lemon, 1998) proposed an algorithm to search the conformational space of protein side chains using the Dead End Elimination theorem (DEE) and A* search. The DEE is used to identify the global minimum energy conformation (GMEC) of side chain rotamers, eliminating those conformations not contributing to the GMEC. A* search is a method for finding a "least cost" path in a tree or a graph from the root node to a goal node. It has two components, the one calculates the cost getting from the root to the actual node, the other uses heuristics to estimate the cost to reach the goal node from the actual position. The costs of a path are calculated using DEE.

3.3 Protein Docking using Flexibility Information

Protein docking is usually separated into protein–ligand docking and protein–protein docking. The difference between the two directions is determined by the size of the molecule docked to a specific protein. In protein–ligand docking usually small molecules are used whereas protein–protein docking deals with the docking of two proteins.

In both cases flexibility can not be neglected. Small ligands can change their conformations as well, especially if they are peptides.

3.3.1 Flexibility Information used in Protein–Ligand Docking

In protein–ligand docking flexibility is often only allowed for the ligand and the receptor is kept rigid. In FlexX (Rarey, 1996) ligand flexibility is handled e.g. by a fragment based method. Here, the fragments of the ligand are fitted incrementally into the receptor site. The fitting is done by pose clustering (Rarey *et al.*, 1996).

Claussen and colleagues (Claussen *et al.*, 2001) have proposed an approach also modelling receptor flexibility, called FlexE. FlexE docks flexible ligands into an ensemble of receptor structures which represents the flexibility of the receptor. All structures of an ensemble are superimposed. Then, side chain conformations and backbone parts are clustered to create a "united protein description". After that, an incompatibility graph is applied to exclude parts that can not occur simultaneously.

Within the DOCK system Ewing and coworkers (Ewing *et al.*, 2001) provide an approach called "anchor and grow". Here, similar to FlexX the ligand is divided into segments based on rotatable bonds (the anchors) and rigid segments. At first the anchors are docked and good hypotheses are searched. Then the conformations are extended by adding additional segments. A pruning step avoids the exponential growth of the search step.

AutoDock (Morris *et al.*, 1996) is also a protein–ligand docking program using conformational searching with a grid based energy evaluation on bases of the AMBER force field (Cornell *et al.*, 1995; Weiner *et al.*, 1984).

The GOLD program (Jones *et al.*, 1997) uses a genetic algorithm approach for docking flexible ligands into a rigid active site of a protein. The flexibility information of the ligand and the protein is coded into a binary string to simulate genetic mutations. Here each rotatable bond is used. Its variability is allowed from -180° to 180° with a step-size of 1.4°. The algorithm performs quite well but there are some limitations. For each docking run the size and the position of the active site have to be determined. As genetic algorithms produce solutions on random "mutations" the results may vary from one experiment to the other. Therefore several experiments have to be done in order to verify the results. This is rather time consuming.

A completely different approach is proposed by Nagata and coworkers (Nagata *et al.*, 2002). They apply a force feedback mechanism to explore the molecular potential field of proteins

and ligands. The potentials are calculated by GRID potential energies and a force feedback joystick is used to move (dock) a ligand to a given protein. The electrostatic force is returned to the force feedback device to guide the user moving the ligand in real time. The system also prevents collisions so that the molecules do not stick together.

3.3.2 Flexibility Information incorporated in Protein–Protein Docking

First protein–protein docking algorithms (Ackermann *et al.*, 1998; Lenhof, 1997; Walls & Sternberg, 1992) in the field have been based on the rigid body assumption, ie. modelling the proteins as rigid bodies. Ackermann uses a voxel representation to model the proteins. On bases of a surface segmentation according to physico–chemical features (charge, hydrophobicity) into regions a cross correlation on complementary parts (convex/convex or concave/concave) is done to generate docking hypotheses. The original work of Ackermann is extended by a soft volume model (Neumann *et al.*, 2002) to enable flexibility (see also section 4.2).

Lenhof (Lenhof, 1995) represents the protein surface by triangles (set up from surface points). Docking hypotheses are generated by geometric hashing, searching for similar triangles and their transformations. The number of transformations is reduced using a local complementarity criterion. This criterion is extended by additional fitness functions modelling physico–chemical features (Lenhof, 1997).

The algorithm of Lenhof was extended by Althaus and coworkers (Althaus *et al.*, 2002) to semi flexible docking. Flexibility is handled by a combinatorial approach using a multi-greedy and a branch–&–cut algorithm to search a minimum energy conformation among possible side chain conformations. This approach is called "side chain de-mangling". Based on the rotamer library of Dunbrack the residues are decomposed into two distinct sets, one holds residues having rotamers and belonging to the binding site, the other one holds the rest of the residues. The optimal combinations of rotamers yielding the lowest total energy is then obtained by multi greedy or branch–&–cut search. Additionally the search space is reduced using the DEE theorem (cf. Leach & Lemon, 1998). The resulting side chain conformations are then minimised using the AMBER force field. Finally the free energy of binding is determined to evaluate the docking hypothesis.

Kohlbacher (Kohlbacher *et al.*, 2001) proposed an alternative docking approach using nuclear magnetic resonance spectroscopy (NMR) to avoid time consuming calculations of the free energy of binding. In order to score a predicted complex, ^{1}H–NMR spectra of the complex and the hypothesis are compared. The NMR spectrum of a docking hypothesis is received by calculating the chemical shifts for each proton of the protein complex. The spectrum of the reference complex was calculated from PDB structures.

Other approaches apply so called *soft shells*. Here, a region of the protein surface is marked as "soft", allowing steric clashes within this area. Jiang (Jiang *et al.*, 2002) for instance use varying sizes of their voxel representation of the surface and a cut off criterion for volume overlaps. The best parameters are estimated by statistical analysis of docking runs. Fernández–Reccio and coworkers (Fernández-Recio *et al.*, 2002) utilise grid–based potential

functions to make the surface of the unbound proteins soft. The potentials are refined by extensive Monte–Carlo simulation.

Sandak and coworkers (Sandak *et al.*, 1998; Verbitsky *et al.*, 1999) focused on domain movements (see section 3.1) and used a General Hough Transform to simulate the domain movements. In their approach proteins are represented by the 3D coordinates of the backbone carbon atoms (C_α). The algorithm is divided into a preprocessing step and a recognition step. In the preprocessing a hinge point is chosen so that it divides the set of C_α atoms into a pair of ordered sets. The hinge is used to define a reference frame. A so called frame–invariant is defined to describe invariant features of the protein shapes. For each frame-invariant a transformation between its coordinate frame and the reference frame is calculated and stored with the frame–invariant (called R–Table). In the recognition phase the frame invariants of the target protein are matched to the one of the model protein (which is pre-processed) to find candidate transformations of the protein parts. Candidates are scored by votes which are increased if they already exist in the R–Table.

Lorber and colleagues (Lorber *et al.*, 2002) propose an algorithm that uses multiple residue conformations and substitutions to model the flexibility during docking. The basic assumption here is that each side chain conformation is independent of each other and the whole protein conformation is additive. Therefore in a preprocessing step multiple low energy conformations for each flexible residue (selected among all residues of the protein) are calculated. This ensemble of pre-generated conformations is then processed into a hierarchical data structure and an optimisation of this structure is performed to speed up the docking procedure later. Beside the identification of similar conformations of residues within the ensemble, the atoms of a side chain are ordered by their position in the chain to prune immediately steric clashes. During docking (using the program DOCK) first the rigid parts (backbone, buried residues) of the protein are positioned and then the side chain conformations are explored until one meets the docking requirements. After that the remaining side chain conformations are investigated and those clashing are pruned. The whole conformation of the ligand is then set up out of the best side chain conformations.

3.4 Discussion

In the sections 3.2 and 3.3 approaches to model side chain flexibility and the application of side chain flexibility to protein–protein docking have been described. Side chain flexibility is important in all areas where proteins are involved, e.g. protein structure prediction or protein interactions.

There are in principle two major directions for modelling side chain flexibility. On the one hand the flexibility of a side chain is modelled by the distribution of torsion angles (e.g. rotamer libraries) and probabilities of changing a rotamer. On the other hand side chain flexibility is handled as a combinatorial problem of placing the side chain with an optimal conformation (e.g. side chain de-mangling, conformational searching).

Rotamer libraries are an appropriate method for describing favourite conformations of side chains and therefore can be used as basis for further investigations. But the flexibility information itself can not be extracted from these libraries, a structure comparison is needed (cf. Koch, 2003).

Conformational searching and side chain placement aim to predict a structure or docking constellation and therefore often use rotamer libraries to reduce the amount of possible solutions. The flexibility is not modelled explicitly (cf. Althaus *et al.*, 2002) as a placement is only valid within the current situation and not in general.

Besides this, the modelling of the docking algorithm also plays an important role in how the flexibility information has to be calculated. Algorithms based on a voxel representation (like ElMaR) underlie an abstraction from the atomic model. A benefit of this is a gain in speed. But side chain flexibility is calculated on the torsion angles and therefore cannot be efficiently modelled into this representation directly. An application of a "soft shell" tries to handle the flexibility but it is to coarse.

Algorithms operating close to the atomic model instead can use placement techniques for the calculation of docking hypotheses. This results in more precise predictions but the conformational searching is more time intensive than the voxel representation. So there is a tradeoff between accuracy of the results and the computational speed of the algorithm.

Using NMR techniques to score and predict docking hypotheses (Kohlbacher, 2000) one could avoid modelling the flexibility as it is included within the spectra by default. But then, the shift predictions have to be modelled efficiently. Also, a distance or score between the reference spectrum of the known complex and the hypothesis has to be defined. Another unsolved problem of this approach is the lack of publicly available experimental data.

The only way to handle large amounts of data like in a 1:N protein–protein docking scenario is to set up a hierarchy of algorithms to filter the large search space efficiently. Flexibility information has to be added to each level depending on the used algorithms. Besides this the scoring of possible hypotheses is very important, too. Filtering out a large number of false predictions and keeping only the best hypotheses at the beginning of such a cascade will save time. This time can then be spend on the selected hypotheses.

The ElMaR system (Neumann, 2003) is designed as such a part or module of a cascade. It is very fast. In this thesis a classification approach is described to provide flexibility information to the ElMaR system in a way so that the run time is not affected much but the results improve. In contrast to other algorithms (e.g. Hubbard *et al.*, 1991; Betts & Sternberg, 1999) this approach is based on unbound proteins alone. A flexibility prediction is made on the basis of features characterising the residues. The classification has to be done only once as the protein structure information will not change.

Further the scoring of the ElMaR system will be improved by introducing relevance feedback to re-order the list of hypotheses. Using feedback, special requirements to the data can be easily incorporated without modifying or redesigning the scoring function. A comparison of docking hypotheses can be done on basis of the feature geometry, hydrophobicity and charge (which can be visualised to the structure) within an ElMaR result set.

The results of this level of a cascade of docking algorithms can be scheduled to further modules with a more complex modelling of the docking process or more time consuming scoring function (e.g. energy based scoring).

Chapter 4
Protein–Protein Docking using the ELMAR System

The main goal of this thesis is to set up a classifier to discriminate residue side chains according to their flexibility. In order to test the accuracy of this approach the results of the classification are incorporated into the docking system ELMAR.

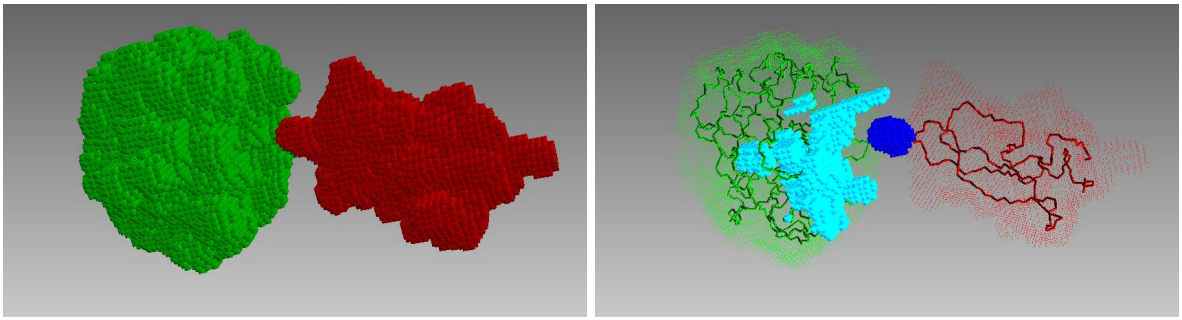
ELMAR is a protein–protein docking system using an algorithm based on a soft volume model to dock proteins. In section 4.1 an outline of the docking system is given. ELMAR can handle local flexibility information to improve its predictions of protein complexes. In section 4.2 the incorporation of the flexibility information is shown.

4.1 Docking System ELMAR

The ELMAR docking system (Neumann, 2003) is a further development of the algorithm proposed by Ackermann and coworkers (Ackermann *et al.*, 1998). In this approach the three-dimensional structure of a protein is discretised into a voxel[1] representation (see Fig. 4.1). From this voxel representation the surface is segmented into a set of concave or convex regions. In order to include physico–chemical properties, the protein's hydrophobicity and charge values of the residues are mapped onto the surface. A match between compatible regions (convex/concave or concave/concave) provides initial docking hypotheses. These are then refined by a cross correlation of the features attached to the surface which score the hypotheses. The algorithm is very fast as this correlation is handled by a fast Fourier transformation. The docking of two proteins can be done in less than 20 minutes[2].

The work of Ackermann focused on bound docking, using a small set of 34 protein complex structures. A docking was performed by breaking the complexes into their parts and then re-docking them again. This approach has been extended by the work of Neumann (Neumann, 2003) introducing a soft volume model and applying unbound protein docking to the algorithm. An interface for flexibility information has also been provided and the algorithm has been enhanced by technical aspects like parallel execution of the docking modules (see

[1] A voxel is a three-dimensional pixel.
[2] Run times have been estimated on a Compaq Alpha 500 Personal Workstation.



(a) Surface, representation as voxel (b) Contact region, representation as voxel

Figure 4.1: *Voxel representation of a beta-trypsin complex (2PTC). The structure is visualised using the visualisation tool* VIWISH *(Klein* et al.*, 1996).*

Figure 4.2) to increase the speed of the algorithm and therefore be able to process a large amount of data (see Neumann, 2003, chapter 5 for details).

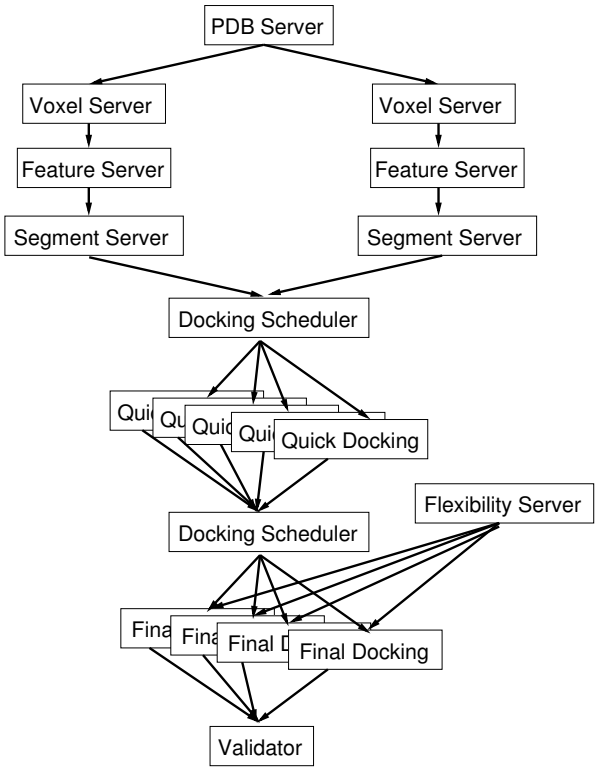


Figure 4.2: *Integration of flexibility information into the* ELMAR *docking system. The flexibility information is included into the "FinalDocking" module. Courtesy of Neumann (Neumann, 2003).*

4.2 Incorporating Flexibility into the Docking System ELMAR

The ELMAR docking system uses flexibility information, especially information on local changes of the protein's conformation to set up a soft volume model. The soft volume model tries to make the rigid modelling of structures more flexible. The flexibility of amino acid side chains can be described e.g. by rotamer statistics (Koch, 2003) or like in this thesis by a classification approach (see chapter 5).

In order to keep the speed of the ELMAR docking system the flexibility information has to be accessed fast. Because of this the calculation of flexibility information is done independently from ELMAR and can be selected as the runtime requirements allow (see Fig. 4.3). If a docking run should be finished in short time, no or flexibility derived from rotamer changes should be included. If more precise results should be calculated, energy based flexibility information can be included. Additionally, the docking algorithm can use several sources of flexibility information simultaneously, like statistically derived data and energy based data (Zöllner *et al.*, 2002; Neumann *et al.*, 2002). Therefore the classification results are stored in a relational database, providing the data on demand to the docking system (see Fig. 4.2). Appropriate index structures on the relations in the database speed up requests.

Within the "FinalDocking" module of ELMAR the classification result of the amino acid side chain is mapped to the corresponding surface regions. It is used to reduce or increase the geometric complementarity factor, a weight to score steric clashes. A steric clash occurs if the matched regions are not complementary, e.g. convex/convex regions are paired. In this case the flexibility information can be used to decide whether this steric clash may occur during docking in a natural environment or not. It will occur if both regions are inflexible which means the corresponding residues are classified as not flexible. This docking hypothesis is then assigned a lower score. If the amino acids of these regions are classified as flexible a steric clash would probably not occur. Thus, this docking hypothesis receives a higher score.

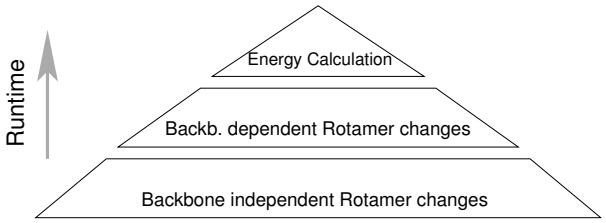


Figure 4.3: *Hierarchy of flexibility information according to run time purposes.* ELMAR *can include different types of flexibility sources like rotamer statistics or energy based flexibility.*

In ELMAR, the external flexibility information p is scaled to form the elasticity weight EL (see Eq. 4.1). EL is designed in such a way that the convolution of flexible voxel results in a lower score whereas the convolution of rigid marked voxel contribute a higher penalty score to the geometry term of the scoring function.

$$EL = \left(1 - \frac{\omega}{2}\right) + \omega\left(1 - \frac{p - \min(p)}{\max(p) - \min(p)}\right) \tag{4.1}$$

The flexibility p is scaled by $1 \pm \frac{\omega}{2}$ so that the distribution of the flexibility has an average of one. ω denotes the scaling factor and can be chosen arbitrary. Exemplarily, a value of 0.5 scales the flexibility values $p \in [0,1]$ between 0.75 and 1.25. Before scaling p is normalised

using the minimum and maximum of all flexibility values of the proteins in question (see Eq. 4.1). If no flexibility information is available, p is set zero.

In equation 4.2, $(P_1 \bullet P_2)(i,j,k)$ represents the convolution of the surface of protein P_1 at grid position (i,j,k) with all surfaces of protein P_2 at positions (i',j',k'). Matching surface points contribute to the score whereas steric overlaps are penalised by the parameter $-q$ [3].

$$(P_1 \bullet P_2)(i,j,k) = \sum_{i',j',k'} P_1(i,j,k) * P_2(i+i', j+j', k+k') * EL(i',j',k') \tag{4.2}$$

with

$$P_1(i,j,k) = \begin{cases} 1 & , (i,j,k) \in \text{Protein 1} \\ 0 & \text{else} \end{cases}$$

$$P_2(i,j,k) = \begin{cases} 1 & ,(i,j,k) \in \text{Surface Protein 2} \\ -q & ,(i,j,k) \in \text{Interior Protein 2} \\ 0 & \text{else} \end{cases}$$

Besides the geometry scoring, also the hydrophobicity (H) and the charge (Q) values of the surfaces are correlated (see Neumann, 2003, section 5.1.4 for details) and the three scores are combined into an overall score:

$$C = (1-\alpha)(1-\beta) * (P_1 \bullet P_2) + \alpha(1-\beta) * (H_1 \bullet H_2) - \beta * (Q_1 \bullet Q_2) \tag{4.3}$$

In order to combine the different features into the scoring function two weights α and β are used. The weights can take values between zero and one. Each combination of the weights influences the impact of the three features to the overall score of a docking hypothesis. This can be visualised by the triangle shown in Figure 4.4. On each edge of the triangle one features is annotated. Each combination of α and β is a point within the triangle. Exemplarily, in case of $\alpha = 0$ and $\beta = 0$ only the geometry component is included into the scoring. For $\alpha = 0$ and $\beta = 1$ only the electrostatics (charge) and for $\alpha = 1$ and $\beta = 1$ only the hydrophobicity is taken into account. Any other combination of $\alpha, \beta \in [0,1]$ results in partial contributions of the three features to the overall score of a hypothesis.

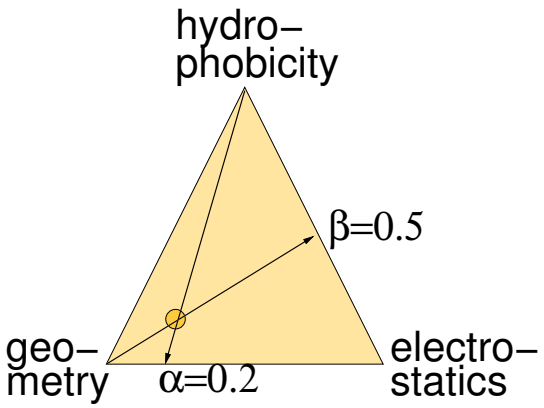


Figure 4.4: *Parameter space of the* α *and* β *weight. Courtesy of Neumann, 2003.*

[3] Value estimated empirically, see Ackermann *et al.*, 1998. In the original work q is denoted as p, here it is renamed to q as p has been used for the flexibility information.

Chapter 5
Predicting Side Chain Flexibility

In this chapter two approaches for predicting side chain flexibility are presented. Both approaches use features derived from scoring side chain conformations by molecular mechanics force fields. At first, an introduction to molecular mechanics force fields is given, focusing on the AMBER force field which has been utilised in this work. Then the two classification approaches are outlined as well as the features used for training them.

5.1 Molecular Mechanics Force Fields

In section 2.3.1 and 2.3.2 different types of interactions within (bonded) or between molecules (non–bonded) have been described. In this section force fields are introduced. Molecular mechanics force fields combine all the different interaction types resulting in a single energy value describing the state of the molecule they are applied to.

Force fields are used in different tasks, e.g. structure prediction (Ulrich *et al.*, 1997; Pillardy *et al.*, 2001), folding of proteins (Lazaridis & Karplus, 2001), or simulating biomolecules within molecular dynamics simulation (Stone *et al.*, 2001; Tapia & Velazquez, 1997; Zuegg & Gready, 2000). They are also applied for scoring conformations, as within docking algorithms (Althaus *et al.*, 2002; Halperin *et al.*, 2002; Kohlbacher, 2000; Lenhof, 1997), or even in sequence analysis to set up amino acid similarity matrices (Dosztanyi & Torda, 2001).

There exist different types of force fields which differ in the number of interactions taken into account and the way how these interactions are modelled. The AMBER (Cornell *et al.*, 1995; Weiner *et al.*, 1986) and the CHARMM (Brooks *et al.*, 1983) force field for example model bending and stretching interactions by a harmonic potential whereas the MM2/MM3 force field (Allinger, 1977; Lii & Allinger, 1991) describes these interactions by the *Morse Potential*, a more accurate but complex function. The GROMOS force field (Scott *et al.*, 1999) uses additional terms called *non physical*. Most of these are for restraining interactions. A short overview of different force fields is given by Norrby and coworkers (Norrby *et al.*, 1996) who compared several of them.

In this thesis the AMBER force field is used. Besides that the BALL library (Kohlbacher, 2000) which has been chosen for handling the protein structures provides already an implementation of this force field, the AMBER force field has been applied in several research works

before (Althaus *et al.*, 2002; Leach & Lemon, 1998; Wilson *et al.*, 1993). It consist of simple energy potentials (see Eq. 5.1) so that an energy score can be computed fast. Another aspect for choosing this force field is that it has well established parameters resulting in a good approximation of the total energy of a given structure.

AMBER **Force Field**

The AMBER force field was developed by Weiner and colleagues (Weiner *et al.*, 1984) in 1984 and refined by Cornell and coworkers (Cornell *et al.*, 1995). It consists of five energy potentials: bending energy, stretching energy, torsion energy, electrostatic and van der Waals energy (see also sections 2.3.1, 2.3.2). The total energy is calculated by summing up all partial contributions. In equation 5.1 all non-bonded interactions are included in the last term like proposed by Cornell et al.:

$$\begin{aligned} E_{total} = & \sum_{ij \in bonds} K_{ij}^{stretch}(r_{ij} - r_{ij}^{eq})^2 \\ & + \sum_{ij \in angles} K_{ij}^{bending}(\theta_{ij} - \theta_{ij}^{eq})^2 \\ & + \sum_{dihedrals} \frac{V_n}{2}[1 + \cos(n\phi - \gamma)] \\ & + \sum_{i<j} \left[\frac{A_{ij}}{R_{ij}^{12}} - \frac{B_{ij}}{R_{ij}^{6}} + \frac{q_i q_j}{\varepsilon R_{ij}} \right] \end{aligned} \tag{5.1}$$

Besides the possibility of fast calculations, the parameters used are the most important parts of a force field. Exemplarily, the stretching energy term requires at least two parameters, the normal bond length (r_{ij}^{eq}) and the bond stretching constant ($K_{ij}^{stretching}$) for each type of bond. The type of a bond depends on the order of binding (single, double or triple) and the appendant atoms. The chemical (e.g. partial charge, chirality, hybridisation) and physical features (e.g. size, mass) of the atoms influence the bond, too. Therefore, Weiner introduced so called *atom types* describing the atom's features. This results in at least 70 parameters[1] for describing the bond stretch.

In addition, to the parameters for the stretching, bending, and torsional variations, the non-bonded potentials need an estimate about the charge distribution within the molecule to calculate the repulsion or attraction forces. Each atom has a radius assigned, called *van der Waals radius*, which describes the equilibrium between the repulsion and attraction force of the atom. There are two models for representing these radii, the *united* and the *all atom model*. The difference between the two is the handling of the non–polar hydrogens. In the united atom model the hydrogens are merged to the adjacent carbon atom resulting in an larger van der Waals radii whereas in the second model all atoms are taken into account. The pros and cons of the two models are described by Kini and Evans (Kini & Evans, 1992) who compared protein structures minimised by both methods. The all atom model yields

[1] This number has been estimated according to the 1996 AMBER parameter set.

better results due to the more precise van der Waals radii and the more detailed modelling of the protein. But a larger set of parameters is needed using this model. The version of the AMBER force field implemented in the BALL library uses the all atom model. In this work, the force field is configured with the standard parameter set that comes along with the BALL Library[2] (Aubertin *et al.*, 2002).

5.2 Classification of the Flexibility of Side Chains

In the following, both classification methods developed for the discrimination of flexible and non flexible residues are described. In the first approach (see section 5.2.3) a threshold based system is outlined. There, an energy based criterion is used to separate the classes of flexible and non–flexible residues. In the second approach (see section 5.2.4), a support vector machine is trained. The set of features derived form energy calculations is extended by other aspects influencing the flexibility of side chains (e.g. solvent accessible surface area).

The flexibility of side chains can be detected by comparing unbound protein structures to sequence identical complexes. If the differences between the torsion angles are that large so that a change of a rotamer of the torsion angle may occur, the side chain might be flexible. Since the docking process cannot be observed by comparing only two structures, in most approaches a statistical analysis is performed in order to support a single observation. Often the used data is also separated into different categories, like buried or exposed (e.g. based on the solvent accessible surface area (SAS) value of the residue) or due to its membership in a secondary structure (see Mc Gregor *et al.*, 1987; Schrauber *et al.*, 1993).

Recently, Koch (Koch, 2003) analysed statistically the flexibility of residues side chains by comparing unbound and complex structures. She reported that flexibility of the torsion angles grows if they are more far away from the backbone. Each residue can be positioned on a flexibility scale where ARG, SER, LYS and GLN are the most flexible residues (according to the probability of rotamer changes) whereas PHE, TYR, TRP and CYS are the most rigid amino acids. Within each residue and torsion angle the direction of change tends to the most favourable rotamer. Exemplarily, in the case of ARG 33% of all side chains change their rotamer. Out of these 66% move to the third rotamer, 37% to the second and only 17% to the first rotamer.

Although these overall tendencies can help e.g. in reducing the conformational search space, they neglect specialities in the local conformations of proteins. Predicting the flexibility of each residue of a protein can be used for modelling the flexibility more precisely. The classification approaches presented here predict the flexibility on the unbound protein structures. There are two main advantages to other methods investigating or calculating flexibility. On

[2] Using non bonded cutoff of 200Å, van der Waals cutoff of 150Å, van der Waals cut-on of 130Å, electrostatic cutoff of 150Å, electrostatic cut-on of 130Å, electrostatic scaling factor for 1-4 interaction of 2.0, Vdw scaling factor for 1-4 interaction of 2.0 and a distance dependent dielectric constant of 1.0 (standard values that come with the BALL library), all calculations are done without solvent molecules.

the one hand structure comparison methods depend on at least two structures that have to be refined, a protein complex and an unbound protein structure that matches one of the chains of the complex. Here, only a single protein structure is used, the unbound one. On the other hand all calculations only have to be done once as the refined structures will not change[3]. Compared to approaches that place side chains during docking the calculated flexibility can be used for all test case scenarios the analysed unbound protein structures are involved in.

5.2.1 Synthetic Conformations

In order to predict the flexibility of the residue's side chain, features have to be calculated that are characteristic for the residue in question and also give information about the residue's flexibility.

A side chain's conformation and the information about its surrounding can be used as a feature to classify its flexibility. Energy based scores like the total energy of a force field describe the conformation of a residue. It implicitly includes neighbourhood informations via the non–bonded interaction potentials (see section 2.3.2 and Eq. 5.1), too.

The conformation of a side chain is defined by the torsion angles χ. Since the torsion angles can take values from 0° to 360° and some residues (ARG, LYS) have up to four χ angles a large number of conformations are possible for a single residue's side chain.

An isolated investigation of side chain conformations has been performed by Lorber and coworkers (Lorber *et al.*, 2002). They assumed that the whole protein conformation is additive, and that single side chain conformations can be handled independently. Torgasin (Torgasin, 2003) analysed in a first approach the influence of changing the side chain conformation of a residue to the surrounding by molecular dynamics simulation. She tested the influence on residues within 4, 6 and 8Å around the modified amino acid. The influence of a changed conformation onto the surrounding gets less the larger the diameter is chosen. The influence depends not only on the changes performed (e.g. change in the torsion angle) but also on the type of the amino acid and the neighbouring residues. Generally, in the experiments run by Torgasin conformational changes are compensated quickly (within the first 5 to 10 ps) by the surrounding. In some cases the residues have been rearranged resulting in low energy values but also not optimal solutions have been carried out. A general pattern describing or modelling rearrangement effects can not be found easily since at first a description of a neighbourhood group and methods for comparing these have to be defined in order to carry out statistical investigations. In case of a flexibility classification influences between neighbouring residues can be neglected because the question is whether a residue changes or not. Residue side chains itself perform concerted movements[4], here one torsion angle is modified at a time and the rest of the side chain and the surrounding conformation

[3] In case an already solved structure is deprecated, the energy calculations have to be repeated for the new structure.

[4] Concerted movements means that more than one torsion angle changed its rotamer when comparing the complex and unbound structure.

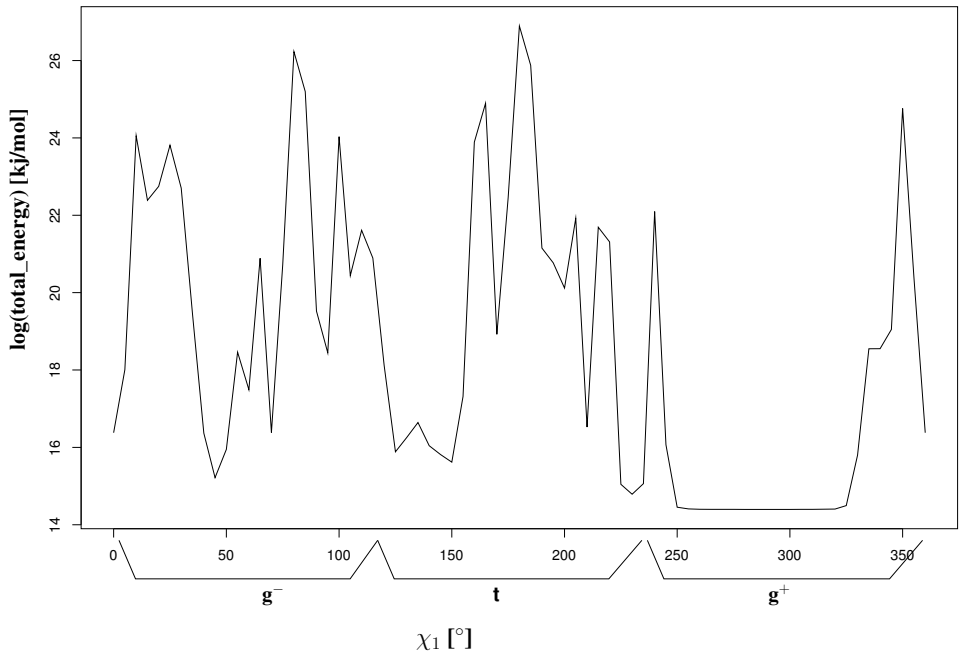


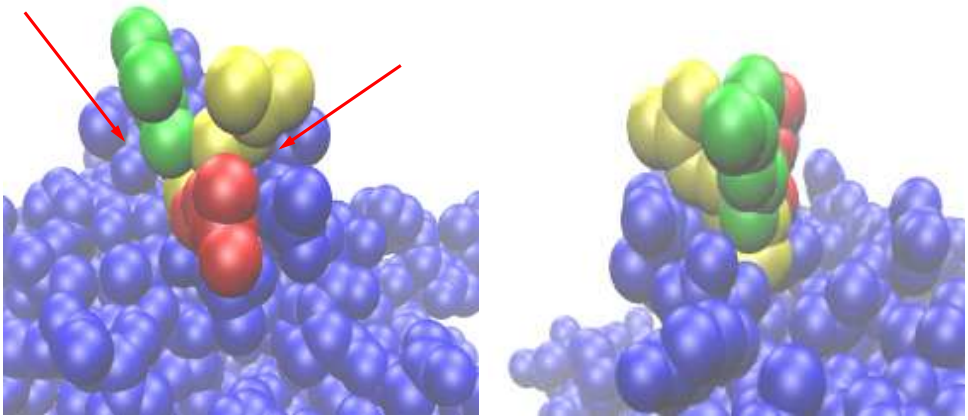
Figure 5.1: *Energy landscape of an ARG residue for χ_1. Here a sampling rate of 5° is used to create synthetic conformations. The surrounding conformation is kept rigid. Below the x-axis the rotamer boundaries are shown.*

is kept rigid because it is unclear whether these movements occur at the same time or after each other. In case of a classification whether a residue's side chain is flexible or not the aspect of interaction of torsion angles within a side chain has no impact on the classification. This can be handled by combining the flexibility predictions for the different torsion angles.

In order to get an energy landscape of different conformations of the side chain the torsion angles are rotated by 360°, sampling in 5° steps. Because these conformations are based on sampling, they are called *synthetic conformations*.

Figure 5.1 reflects the surrounding of the Arginine (ARG) side chain. During the rotation of the χ_1 torsion angle around 360° steric hindrance by neighbouring groups cause high energy values according to the definitions of the non–bonded potentials (see Fig. 2.10). In figure 5.1 it can be observed that within the first (0-120°) or the second (120-240°) rotamer energy minima and maxima alternate often due to steric restrictions. One can assume that within these two rotamers the flexibility of the side chain (here the χ_1 torsion angle) is reduced. Sparse contacts or an optimal arrangement reduce the total energy as shown in the third rotamer of figure 5.1.

Another example is shown in figure 5.2. Here, the 3D structures of an ARG residue with different conformations of the χ_1 torsion angle are superimposed (red: 330°, green: 160°, yellow: 70°). In figure 5.2(a) on the left of the residue no neighbouring groups are located so that the side chain can be placed there without problems. On the other side instead, other groups of the protein are located so that moving the side chain freely is not possible.



(a) Left from the highlighted ARG structures (see red arrow) no surrounding structures causes any steric hindrance whereas on the other side (see the other red arrow) the ARG is near to neighbouring residues.

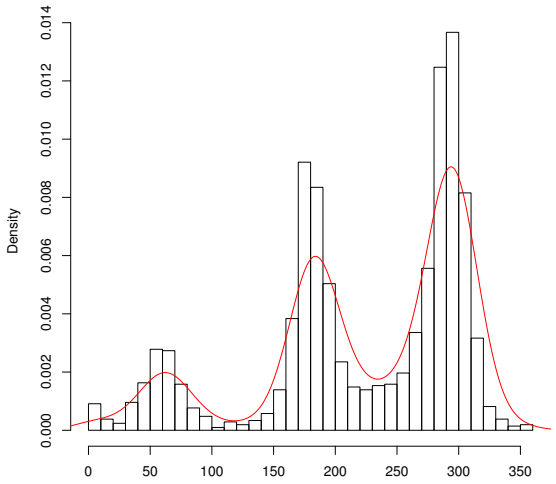
(b) Here, the same conformation is shown as on the left, but the structure is rotated by 180°. On the left of the highlighted residue, one can clearly see groups from the surrounding restricting the flexibility whereas on the right, no other groups are near to the marked residue.

Figure 5.2: *3D Structure of ARG 145, 1ACB chain E in three different conformations. The yellow structure has a χ_1 of 70°, the green structure of 160° and the red structure of 330°. In blue the E chain of 1ACB is given. The size of the balls representing the atoms depends on the atoms' van der Waals radii.*

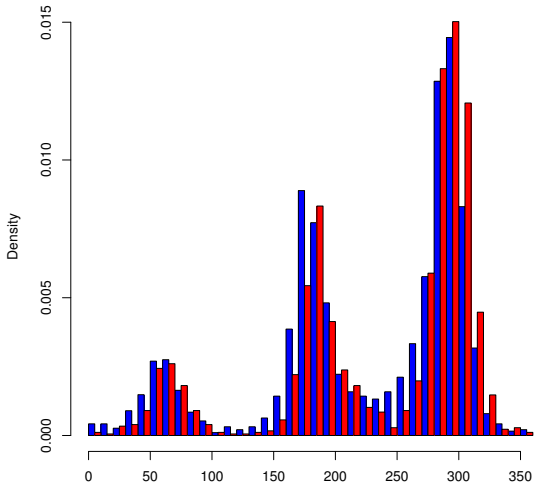
Figure 5.2(b) shows the similar constellation just from another point of view (the protein is rotated by 180°). Here, it is obvious that the yellow coloured residue is in contact with a surrounding residue. Because here the van der Waals radii are used for visualising the atoms, the conclusion can be drawn that there is an increase in the non-bonded energy potential for the given conformation of the residue.

Looking at the distribution of the energy minima (see Fig. 5.3(a)) of all Arginine in the data set, a tri–modal distribution can be observed. Here, a histogram of the synthetic conformations yielding the global energy minimum is calculated for each residue and torsion angle in the data set (cf. Zöllner, 2001; Koch *et al.*, 2001). In figure 5.3(a) a density distribution is fitted on the histogram.

This distribution of the energies correlates with the distributions of statistical approaches (see Koch *et al.*, 2002). In figure 5.3(b) the distributions of the statistical approach of Koch et al. and the energy distribution calculated in this thesis are plotted. The peeks within both distributions lie within the centers of the rotamers whereas at the rotamer boundaries the amount of observed conformations are more sparse.



(a) Distribution of energy minima of χ_1 for all ARG of the test set.



(b) Torsion angle distribution for χ_1 of ARG. In red the distribution from the rotamer library is shown, in blue the distribution of the energy minima per degree (cf. Koch *et al.*, 2002).

Figure 5.3: *Comparison of energetically and statistical rotamer distributions. On the left the distribution of energy minima for χ_1 and ARG is given. On the right, this distribution is superimposed to the statistically derived distribution of the same data.*

Within the distribution of the energy minima, the third rotamer is preferred most whereas the first rotamer is the most unfavourable one. Comparing this observation to the energy landscape of the synthetic conformations in figure 5.1, parallels can be found. As outlined before, the third rotamer shows a broad energy minimum, a region with less steric hindrance for placing a side chain. Within the other rotamers, a placement of the side chain is more unlikely. This is also reflected in the overall distribution and shows that all residues follow the general tendencies given by the rotamer distributions but also, that the energy landscape differs due to the local settings of the residue.

Thus, the conclusion is that the energy landscape, based on the synthetic conformations, holds information about conformations that can possibly be taken by a side chain. Extracting features that represent this information can be used for predicting a side chain's flexibility.

5.2.2 Features for the Flexibility Classification

In this section the features selected for the classification of the flexibility of amino acid side chains are presented. At first energy based features derived from the energy landscape are outlined. Then, additional parameters influencing the flexibility are described.

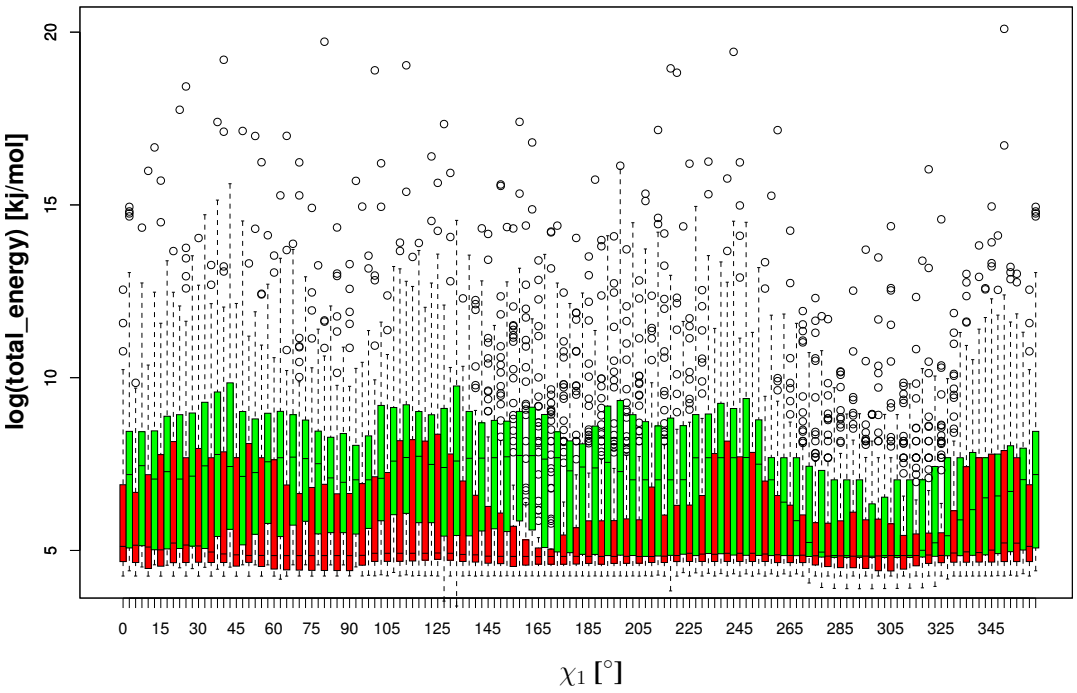


Figure 5.4: *Box-plot of the energy landscape for all ARG,χ_1. The red marked boxes represent the residues labelled flexible, the green boxes represent the non–flexible residues. The energy values on the y–axis are given as logarithms.*

Energy based features

The energy landscape describes the surrounding of the residue in question. In order to get an overview of the data covered by the test set a box-plot is drawn for each amino acid type. From a box-plot one can easily derive the range of energy scores of a given conformation within the data set. In figure 5.4 the total amount of labelled data for ARG is divided into the two classes flexible (red) and non flexible (green). For each conformation (each value of χ_1 during rotation) the energy scores of all examples are represented by a box-plot. The red and green boxes present the range of the energy scores of 50% of the data. The whiskers mark 90% of the data whereas single points above or under the boxes/whiskers represent outliers. Inspecting the energy landscapes (formed by the box-plots) of the χ_1 torsion angle of figure 5.4, there are differences between the two classes, especially within the first (g^-) and the second rotamer (t). Extracting the median curve for both classes (see Fig. 5.6) also shows that within the third rotamer (g^+) the curves align, reflecting that this rotamer is preferred energetically. Looking at the plot of TRP (see Fig. 5.5), the differences between the flexible and non flexible residues are even more obvious. Similar observations can be made for the other residues and torsion angles (cf. App. B.1). From the energy landscape different features can be extracted which are described in the following.

A feature to discriminate flexible from non–flexible residues can be the difference in energy between the original conformation (the one given in the PDB file) and the minimum energy

Figure 5.5: *Box-plot of the energy landscape for all TRP,* χ_1. *The red marked boxes represent the residues labelled flexible, the green boxes represent the non–flexible residues. The energy values on the y–axis are given as logarithms.*

conformation received from the synthetic conformations. In case of protein–protein docking conformational changes are forced by the approaching protein. During the docking process the energy level of the side chains rises in case of steric contacts, enabling a side chain to pass energy boundaries. In this approach, the assumption is made that a conformation is taken which contributes most to an overall lower energy level of the complex than its unbound parts. This assumption is supported by the fact that most changes of the conformation of a side chain directs to the energetically favourable rotamers (see Koch *et al.*, 2002). Thus, this feature has been implemented in the threshold based classifier (see section 5.2.3) as well as it is taken as one component of the feature vector used with the SVM.

Besides the energy difference, the energy landscape itself characterises a residue. Thus, the whole energy landscape could be used as a feature, too. But looking at Fig. 5.7 one can see that the base energy level of each residue differs. One reason for this is the different sizes of the proteins. So features are needed, that are independent from the size of the protein. Also, the feature vector would contain at least 72 components to be trained. A dimension reduction is necessary. The energy landscape of each residue is similar to a signal e.g. from a speech recording. In signal processing, often a linear transformation is taken for a decomposition of the input signal in order to compress and/or to extract features out of it. This is usually done by a Fourier transformation (see Eq. 5.2).

$$F(k) = \sum_{m=0}^{M-1} f(m) e^{\frac{-ik2\pi m}{M}} \tag{5.2}$$

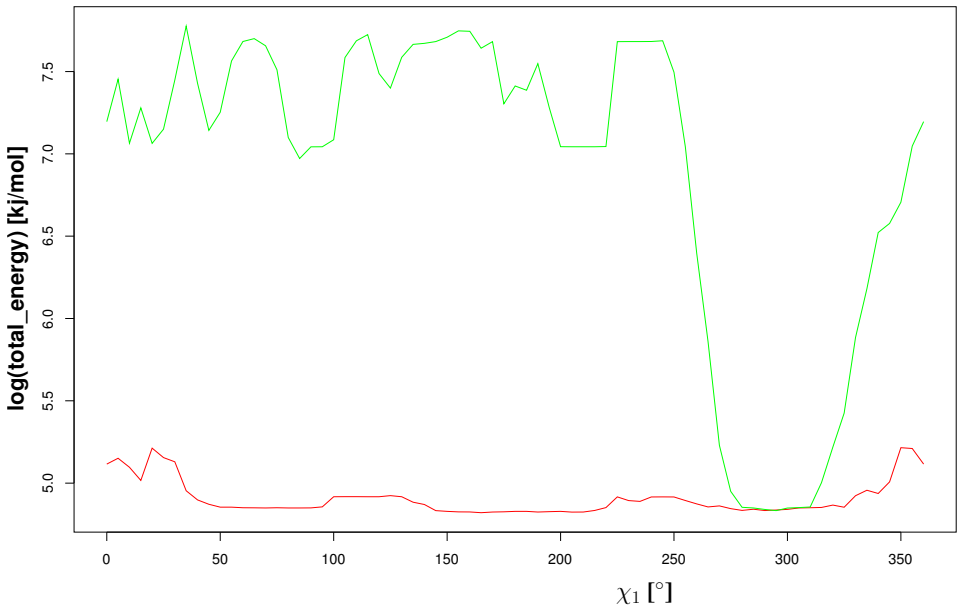


Figure 5.6: *Means extracted from Fig. 5.4. The red curve represents a flexible residue, the green curve represents the non–flexible residue. The energy values on the y–axis are given as logarithms.*

A Fourier transformation decomposes the input signal (here the energy landscape) into a set of sinus and cosine functions and coefficients ($F(k)$). The coefficients describe the contribution of the base (sinus and cosine) functions to the original signal. Applying filters, the signal can be de-noised and important or characteristic parts become visible.

For instance, by cutting off those coefficients representing high frequencies the input signal is smoothed (see Fig. 5.11). A disadvantage of the Fourier transformation is that e.g. filtering high frequencies (by setting some coefficients to zero) effects the whole signal because the corresponding base function spans along the defined ranges (Bäni, 2002). In some cases the base function does not contain any local information and therefore the local information is distributed over several coefficients. So, it is difficult to choose coefficients that can be removed from the signal.

Another problem of the Fourier transformation is the handling of discontinuous parts of the signal. A popular example is the representation of a rectangular signal by the Fourier transformation. The signal itself can be decomposed by sinus and cosine functions but problems occur reconstructing the discontinuous parts, namely occurring at the corners of the rectangular signal. Here, the so called *Gibbs effect* can be observed, the reconstructed signal overshoots at the corner. This is usually a hint that the used base functions do not fit to the signal. A way to avoid this is to choose a different method to decompose the signals, like the wavelet transformation. A wavelet transformation (see Eq. 5.3) is also a linear transformation.

$$f = \sum_k c_k \psi_k \tag{5.3}$$

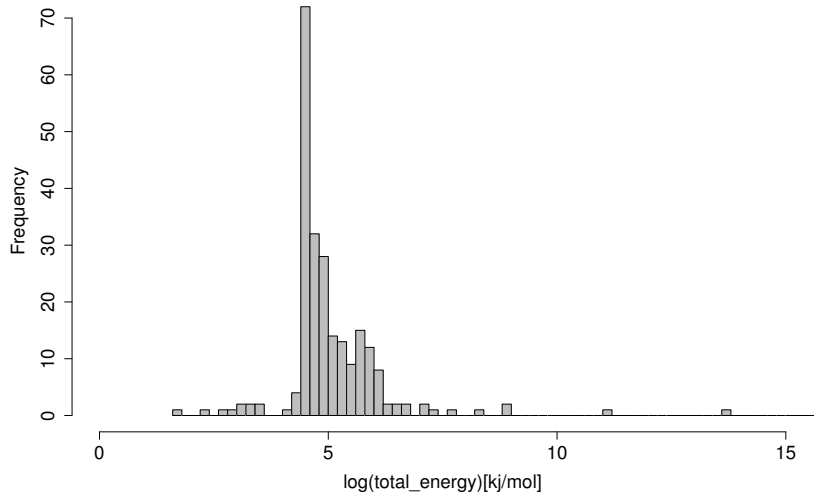


Figure 5.7: *Distribution of the base energies of the proteins in the test set. On the x-axis the total energy is given in logarithmic scale.*

But compared to the Fourier transformation, its base functions have to fulfil several criteria, like a good localisation in time (the basis function only differs from zero on a small range of the signal) as well as in frequency space and they have to form an orthonormal system. The simplest wavelet is the so-called *Haar–Wavelet* (see Fig. 5.8). Other base wavelets, also named *mother–wavelets*, have been calculated e.g. by Daubechies (see Fig.5.9).

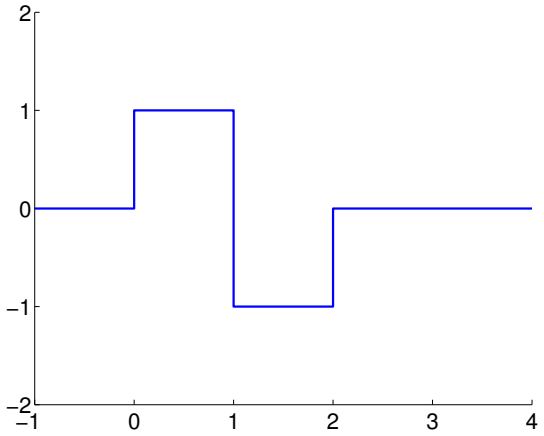


Figure 5.8: *Haar Wavelet. Mother wavelet for* $m = n = 0$.

$$\psi(t) := \begin{cases} 1 & \text{if} \quad 0 \leq t \leq \frac{1}{2} \\ -1 & \text{if} \quad \frac{1}{2} \leq t \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{5.4}$$

As a starting point the function given in equation 5.4 is used. From this the Haar wavelet (see Fig. 5.8) can be constructed using the parameters m and n:

$$\psi_{m,n}(t) = 2^{-\frac{m}{2}} \psi(2^{-m}t - n) \quad m,n \in \mathbb{Z} \tag{5.5}$$

The parameters are used to scale (m) and translate (n) the mother wavelet along the axis. For $m = n = 0$ one obtains the initial wavelet again. The approximation of a function by

the Haar Wavelet looks like a staircase. According to the chosen parameters the steps can be more or less detailed. Exemplarily, one could define a approximation function $T_m f$ with a size of the staircases of 2^m. Changing the value of m, this function can be more or less detailed (e.g. more detailed if m is small). Thus, a more detailed function $T_{m-1} f$ can be defined, too. Because both approximations operate on the same range there exists a relation between the approximations; the difference between the detailed ($T_{m-1} f$) and the less detailed approximation ($T_m f$) is a linear combination of Haar Wavelets:

$$T_{m-1} f - T_m f = \sum_n \nu_{m,n} \psi_{m,n} \tag{5.6}$$

In equation 5.6, $\nu_{m,n}$ denotes a so called *wavelet coefficient*.

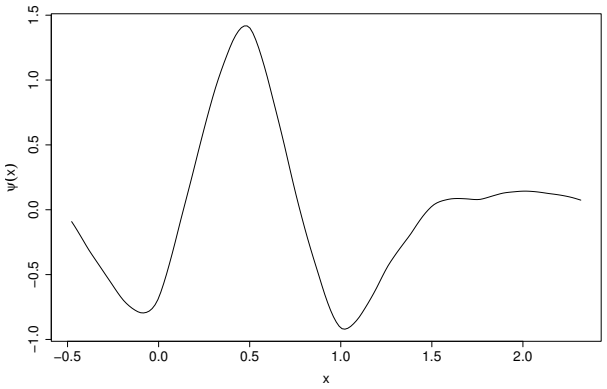


Figure 5.9: *Plot of a Daubechies 6 mother wavelet.*

Furthermore, a detailed approximation ($T_{m_0} f$) can be modelled by the sum of less detailed approximations ($T_{m_1} f$) and a linear combination of wavelets:

$$f = T_{m_0} f = \sum_{m=m_0+1}^{m_1} \sum_{n \in \mathbb{Z}} \nu_{m,n} \psi_{m,n} \tag{5.7}$$

Wavelet transformations have been applied in different fields of science and several approaches have driven the development of the wavelet transformation further. A theory that combines the different developments in wavelet theory is called *multi-resolution analysis* (Mallat, 1989). The multi-resolution analysis (MSA) can be compared to a microscope which can be used to look at a function at a resolution and position of your choice. Thus, a *scaling function* (φ) is used which is like a short impulse. In order to represent a function f within the scale of 2^m, an approximation can be reached by a linear combination of 2^m stretched and by $n2^m$ shifted versions of φ:

$$\begin{aligned} f &\approx \sum_{n \in \mathbb{Z}} u_{m,n} \varphi_{m,n} \\ \text{with} \quad \varphi_{m,n}(t) &= 2^{-\frac{m}{2}} \varphi(2^{-m} t - n) \end{aligned} \tag{5.8}$$

One can show that the calculation of the best coefficients for $u_{m,n}$ can be realised easily if $\varphi_{m,n}$ forms a orthonormal family for each scale (each m) (c.f. Bäni, 2002, chapter 1). For equation 5.8 the parameters can be calculated as follows:

$$u_{m,n} = \langle \varphi_{m,n}, f \rangle = \int_{-\infty}^{\infty} \varphi_{m,n}(t) f(t) dt \tag{5.9}$$

Following this calculation, $u_{m,n}$ can be compared to a weighted mean of f of the environment of the position $n2^m$. The smaller m is, the smaller the environment and therefore the more

detailed are the coefficients $u_{m,n}$ proportional to the sampling points of f. Let $A_m f$ be the best approximation of f within the scale of 2^m then the following equation is true:

$$A_m f = \sum_{n\in\mathbb{Z}} \langle \varphi_{m,n}, f\rangle \varphi_{m,n} \tag{5.10}$$

Also, let V_m be the set of functions f which can be calculated exactly by the scale 2^m then $A_m f$ is the projection of f onto V_m. The function φ is contained within V_0. Further the different approximations should be connected via

$$\cdots \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \subset \ldots \tag{5.11}$$

which means that if f can be represented by 2^m (V_m) it has to be represented also by the more detailed scale 2^p (V_p) if $p < m$. From the multi resolution analysis and its scaling function wavelets can be constructed. As mentioned above wavelets have an orthonormal basis. The scaling function φ itself forms no such basis but it can extended to become an orthonormal basis. Baeni describes an approach to extend the set V_0 to an orthonormal basis in V_{-1}:

$$\varphi = \sum_{k\in\mathbb{Z}} h_{k-2n}\varphi_{-1,k} \tag{5.12}$$

The new system should be constructed by translating one single function ψ:

$$\psi = \sum_{k\in\mathbb{Z}} g_k \varphi_{-1,k} \tag{5.13}$$

The coefficients g_k have be chosen so that they extend V_0 to an orthonormal basis. At the end of this calculation (c.f. Bäni, 2002) the already known equation of the wavelet function results:

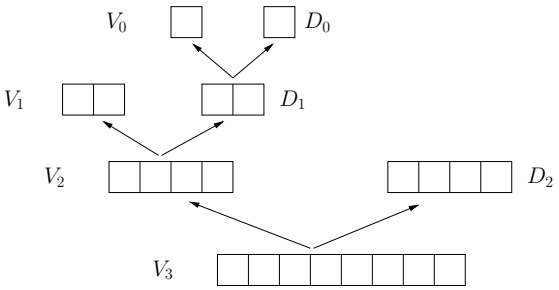
$$f = \sum_{m\in\mathbb{Z}} \sum_{n\in\mathbb{Z}} \nu_{m,n}\psi_{m,n} \tag{5.14}$$

Here, a wavelet function has been constructed from the MSA. Furthermore there is a relation between the approximation function as defined in equation 5.10 and the wavelet of equation 5.14:

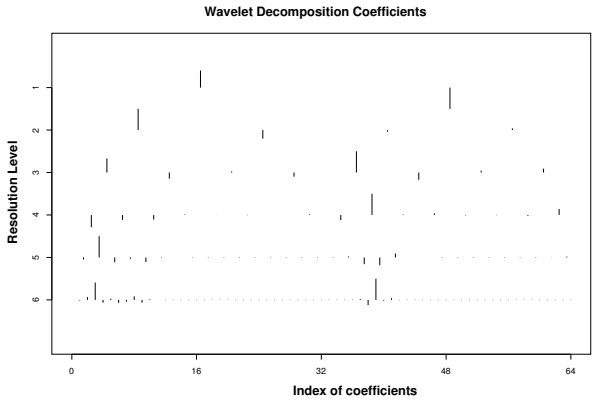
$$\begin{aligned} A_{m-1} f &= A_m f + \sum_{n\in\mathbb{Z}} \nu_{m,n}\psi m, n \\ A_{m_0} f &= A_{m_1} f + \sum_{m=m_0+1}^{m_1} \sum_{n\in\mathbb{Z}} \nu_{m,n}\psi m, n \end{aligned} \tag{5.15}$$

This means that within each scale (resolution) the function f can be represented by the approximation function $A_m f$ and the detail function $D_m f$:

$$\begin{aligned} D_m f &= \sum_{n\in\mathbb{Z}} \nu_{m,n}\psi m, n \\ \text{with} \quad \nu_{m,n} &= \langle \psi_{m,n}, f\rangle \end{aligned} \tag{5.16}$$



(a) Scheme of MSA. Each approximation V_x is decomposed into an approximation and detail function on the next level.

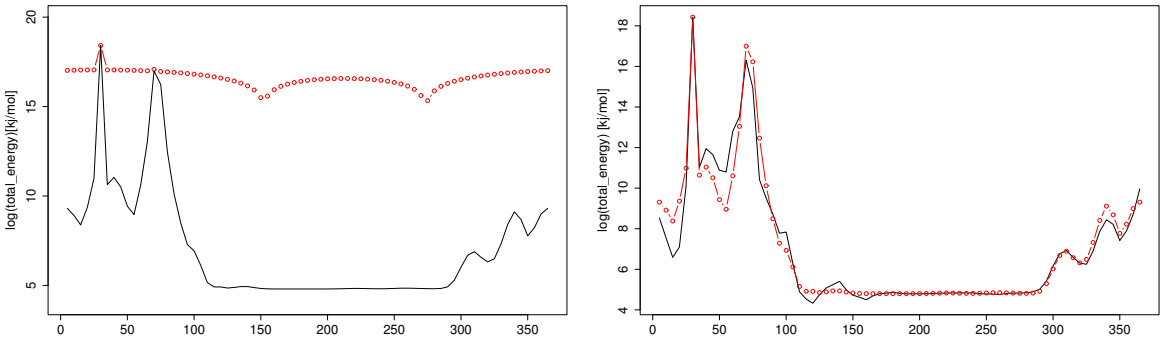


(b) Wavelet decomposition using a Daubechies filter of order 6. Here, the coefficients are plotted for each resolution level. The input signal is padded by zeros to reach a length of the next power of two (128 inputs).

Figure 5.10: *Multi resolution analysis. On the left a scheme of the hierarchies, on the right a plot of the wavelet coefficients after the decomposition of the energy landscape of ARG 20, 1BPI is depicted.*

In figure 5.10 the resolution hierarchy is shown as a scheme (left) and exemplarily for an energy landscape of the data set (here, the coefficients after the wavelet decomposition are shown). An application of the wavelet transformation is the removal of noise within the signal. This is done to enhance further analysis. Noise within a signal is removed by *thresholding* the wavelet coefficients. Thresholding means to remove coefficients by setting their values to zero. Because of the good localisation properties of wavelets a thresholding of the coefficients only has an impact on a small part of the signal. The reconstruction from the thresholded coefficients often results in a good approximation of the original signal. Figure 5.11 shows the reconstruction of an energy landscape after thresholding. In figure 5.11(a) the energy landscape is transformed by a Fourier transformation. Then the last two coefficients are set to zero. Afterwards the signal is reconstructed by applying the inverse Fourier transformation. In case of figure 5.11(b) a wavelet transformation is performed. Here, a Daubechies filter is used. Compared to the Haar Wavelet, these filters are superior because of an improved scaling function. These scaling functions have a better location in time/frequency. Daubechies filters differ in their so-called order (see Bäni, 2002). In case of order 1 one would receive back the Haar MSA. In this work for most residues a Daubechies filter of order 6 is chosen (cf. Fig. 5.9). But in some cases (ARG χ_3/χ_4, GLN χ_1, and TRP χ_1) better results are reached using a filter of order 4.

After the decomposition, a thresholding is performed on the fourth level of the resolution pyramid using the soft threshold method (see Eq. 5.17). Figure 5.12 shows the thresholded coefficients of the wavelet decomposition of the energy landscape of Arginine 20, 1BPI.



(a) FFT, thresholding by setting last 2 of 72 coefficients to zero (red).

(b) Wavelet transformation, soft thresholding on level 4 (red).

Figure 5.11: *Comparison of Fourier and Wavelet transformation. Here, the results of the reconstruction after thresholding are shown.*

Comparing this figure to figure 5.10(b) one can observe that several coefficients have been eliminated.

There are different methods to eliminate coefficients from a signal. Generally, a threshold τ is used to decide whether a coefficient is kept or removed. In this context, *hard thresholding* means that if a coefficient c_k does not exceed the threshold τ its value is set to zero. Otherwise it is kept within the set of coefficients. *Soft thresholding* in contrast also removes the coefficient if the threshold is not exceeded but reduces the value of the coefficients if it is larger than τ (see Eq. 5.17), too.

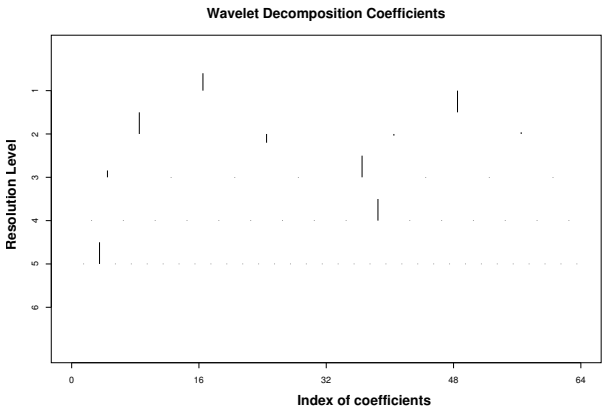


Figure 5.12: *Plot of thresholded wavelet coefficients of ARG 20, 1BPI. Here, a soft thresholding and the "universal method" are applied. All coefficients from level 3 to 6 are thresholded.*

$$\tilde{c_k} := \begin{cases} 0, & \text{if} \quad |c_k| \leq \tau, \\ sign(c_k)(|c_k| - \tau), & \text{else} \end{cases} \tag{5.17}$$

The threshold τ has to be estimated on the set of coefficients. In this work the threshold value is determined by the "universal method" of Donoho (Donoho, 1995):

$$\tau = K\sqrt{2\ln(N)}\sigma \tag{5.18}$$

The threshold τ is estimated on basis of the number of coefficients[5] N and is weighted by the coefficients' standard deviation. K is a constant of order 1 (see Bäni, 2002).

Exposure to Solvent

The *solvent accessible surface area* (SAS) describes the surface area, here of a residue, that is exposed to the solvent (Lee & Richards, 1971). The degree of exposure to the solvent is used to decide whether a residue is *buried* in the core of the protein or *exposed* and lies on the surface of the protein. The SAS can be calculated by moving a "water probe" (a sphere with radius of 1.4Å, see Fig. 5.13) over the protein surface calculating the contact area (Connolly, 1983*a*).

In the literature, amino acids are analysed in context of the SAS (Pacios, 2001; Cyrus, 1976). The degree of exposure to the solvent of a residue correlates with its flexibility. The more exposed the residues are, the more flexible they are (cf. Koch, 2003; Schrauber *et al.*, 1993). Surface residues underlie fewer steric restrictions and tend to be more flexible.

Here, the SAS is calculated for each residue for the conformation given by the original PDB file. In order to be able to compare the SAS values between the different amino acid types, the SAS is normalised using the maximal SAS value of the specific residue type:

$$relSAS(aa) = \frac{SAS(aa)}{\max\limits_{aa}(SAS)} \tag{5.19}$$

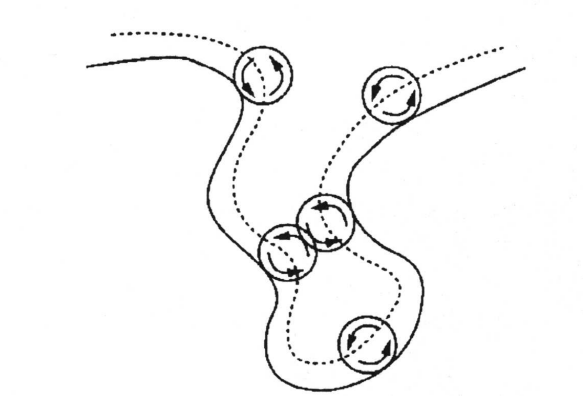


Figure 5.13: *Scheme of calculating the SAS after the algorithm of Connolly (Connolly, 1983*a*). A probe representing the solvent is "rolled" over the surface to measure the contact area.*

Exemplarily, in figure 5.14 the relative SAS is visualised by colouring the residues according to their SAS values. The more the residues are shaded in red, the more exposed they are.

[5]The number of coefficients depends on the resolution level of the MSA.

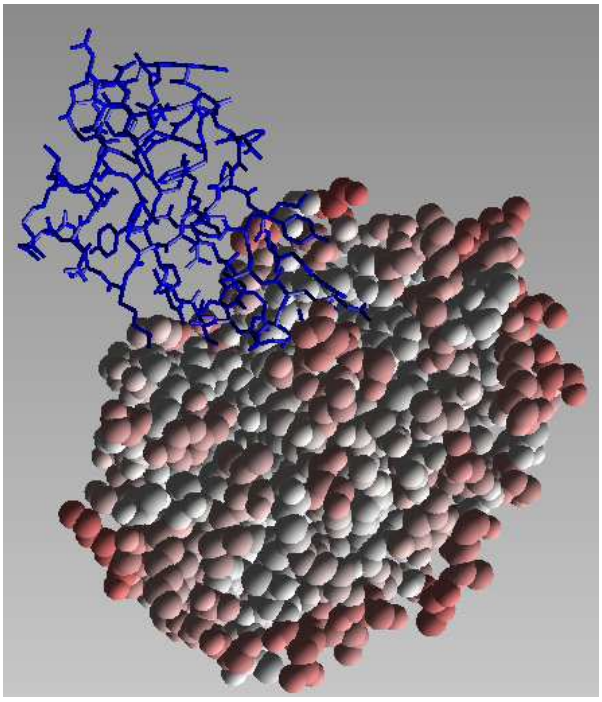


Figure 5.14: *A trypsin (2PTC) coloured by the relative SAS of the residues. The more red the atoms are coloured, the higher the SAS.*

Original Conformation

As an additional feature, the original conformation can be taken. From the distribution of the torsion angles the most favourable rotamer for each residue can be derived. Favourable rotamers fall together with an energy minimum (cf. Koch *et al.*, 2002) whereas non–favourable rotamers tend to have an higher energy level. The direction of change of a rotamer is correlated with the starting rotamer (Koch, 2003) which means that a side chain prefers to change towards the optimal, energetically favourable rotamer. Thus, the original conformation can be used as a feature. Because the classifier is trained for each torsion separately, only the value of the corresponding torsion angle is included. Also, the rotamer distribution is implicitly fed to the classifier since the classifier is trained by a labelled data set of reasonable size.

B-value of Side Chain Atoms

The *B-value* or *temperature factor* is used in crystallography as a measure for the accuracy of the atom positions calculated from the electron density plots. A low B-value indicates a low variance in the atom position. High values instead assume greater fluctuations. On the one hand these fluctuations due to the refinement method (crystallography) but also, on the other hand due to the flexibility of the residue. Karplus and Schulz (Karplus & Schulz, 1985)

used the B-value as a measure of chain flexibility of protein backbones. In this approach the side chain carbon atoms are taken to form a feature to predict flexibility. In order to receive an average B-value (B_{AA}), for each side chain the sum of the carbon atom's temperature factors (B) is normalised by the number of carbon atoms:

$$B_{AA} = \frac{1}{N_{carbon}} \sum_{i=1}^{N_{carbon}} B_i \tag{5.20}$$

Secondary Structure

The secondary structure of a protein (e.g. an α–helix, see section 2.2) influences the side chain conformation of residues (cf. Koch, 2003; Mc Gregor *et al.*, 1987). So, the information whether a certain residue belongs to a secondary structure element can support the classification of the side chain flexibility. In the database used for annotating meta information of the proteins (see section D.1) the output of the program *dssp* (Kabsch & Sander, 1983) is stored. Dssp calculates for a given protein structure (PDB format) all secondary structure elements and labels them by letters (e.g. an α helix is labelled as H). In order to combine this information with the numerical values of the other features the membership of the residue in a secondary structure is denoted by 1, otherwise 0. Here, the different types of secondary structure elements are neglected.

5.2.3 Threshold Based Classification

In order to predict the flexibility of a residue, in a first approach the difference in the total energy of the original (E_{base}) and the minimum synthetic conformation (E_{min}) is used (see Eq. 5.21). The amount of this energy difference expresses how much energy can be gained if the side chain conformation takes the minimum energy conformation within the certain torsion angle space. If the difference in total energy is large it suggests that a rotamer change is possible whereas only small changes in the energies relate to no rotamer changes.

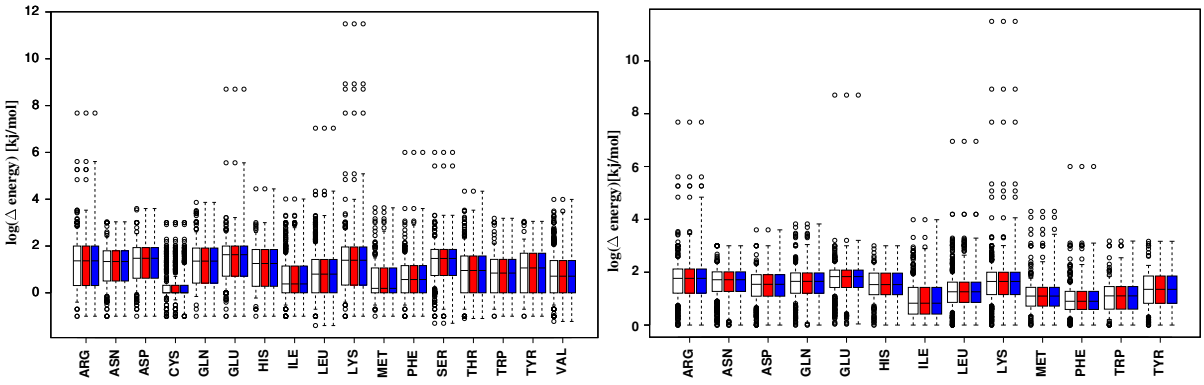
In order to classify the flexibility of residue's side chains a threshold is defined for discriminating between flexible and rigid residues:

$$P_{flex}(aa) = \begin{cases} 1 & \text{if } norm(E_{base} - E_{min}) \geq S, \\ 0 & \text{otherwise} \end{cases} \tag{5.21}$$

In equation 5.21 the residue aa is classified as flexible (1) if its energy difference exceeds a certain threshold S. It is defined as not flexible (0) if the threshold is not reached.

The energy difference is normalised to define threshold values (S) between 0 and 1. Because the energy values differ from protein to protein due to the size of the protein[6], here,

[6]The AMBER force field potentials sum up the energy contributions of each residue pair, see Eq.5.1.



(a) χ_1 torsion angle.

(b) χ_2 torsion angle.

Figure 5.15: *Box-plots of the distribution of energy differences for χ_1 and χ_2 for each residue. The whisker of white marked boxes is calculated by $0.5 * (Q_3 - Q_1)$, the red ones by $1.5 * (Q_3 - Q_1)$ and by $3 * (Q_3 - Q_1)$ in case of the blue marked boxes.*

the normalisation is handled by the distribution of the energy differences using box-plot statistics.

In figure 5.15 the distributions of the energy differences for each residue and χ_1 and χ_2 is shown. The upper border of the boxes (also called upper hinge) represents 75% of all examples in the data set. The upper whiskers extend this amount up to 90%. The value of the upper whisker is taken for normalisation. The whiskers itself are calculated by scaling the distance of the quantiles (Q_1, lower hinge and Q_3, upper hinge) of the box-plot controlling the extend of the whiskers and the amount of data included. In figure 5.15 three different values for the extension of the whiskers are shown to demonstrate the changes of a whisker. For the classification the optimal normalisation factor is determined for each residue and torsion angle using Receiver Operating Characteristic (ROC) analysis plots. ROC analysis will be explained in section 7.3.2.

The optimal threshold is then estimated on the used data set by sampling the interval $[0,1]$. This is done for each residue and torsion angle. The classification results as well as the optimal threshold are then estimated using ROC, too.

5.2.4 Classification of Residues using Support Vector Machines

The results of the threshold based prediction of the flexibility (see section 7.2.1) show that using a simple linear classifier works quite well for the χ_1 torsion angle but for the higher torsion angles the classification performance is low. In order to improve the classification performance on the one hand additional features have to be selected. On the other hand a method superior to linear classification has to be chosen, like a support vector machine (SVM). Before outlining the enhanced classifier, a short introduction to support vector machines is given.

Introduction to Support Vector Machines

In 1992, support vector machines have been introduced by Vapnik and co-workers (Boser *et al.*, 1992). In the field of bioinformatics in several applications SVMs are used, e.g. for classifying monomer and dimer structures of proteins (Neumann, 2003; Zhang *et al.*, 2003) or homology modelling of proteins (see Christianini & Shawe-Taylor, 2000, chapter 8). The following introduction (including the figures) is based on the book of Cristianini and Shawe-Taylor (Christianini & Shawe-Taylor, 2000).

Support vector machines try to learn a separating hyperplane so that it optimally discriminates two classes[7]. The hyperplane is tuned that way during learning so that the SVM's generalisation error is minimised, meaning that this method finds a optimal solution and that it does not end up in a local minima. Besides this, the computational effort is very low so that usually support vector machines can handle large datasets efficiently. SVMs are based on linear classification machines. In linear classification a binary decision is performed by a real valued function $f : X \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$. The input $\vec{x} = (x_1, \ldots, x_n)$ is assigned a positive class if $f(\vec{x}) > 0$ and a negative class otherwise. In this case f is a linear function and for $\vec{x} \in X$ it can be written as:

$$f(\vec{(x)}) = \langle \vec{w}, \vec{x} \rangle + \vec{b} \qquad (5.22)$$

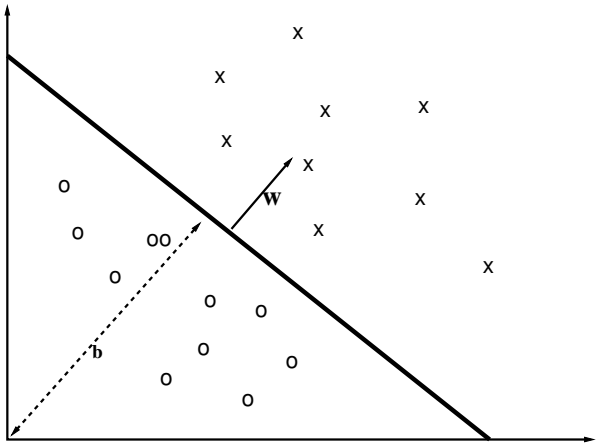


Figure 5.16: *Scheme of linear classification for two dimensions. The separating hyperplane is given by* $(\vec{w}, \vec{b})$*, x and o denote examples from the two classes.*

Thus, $f(\vec{x})$ defines a hyperplane (see Fig. 5.16) with parameters $\vec{w}$ (the direction perpendicular to the hyperplane) and $\vec{b}$ (position vector). Support vector machines are trained by a labelled data set (of size M). The SVM divides the input samples into two classes. The training set can be written as $\{(\vec{x}_i, y_i)\} \quad i = 1, \ldots, M$ with $y_i \in \{-1, 1\}$.

Often the classification performed this way is difficult, since an optimal hyperplane can not be found. The idea within the theory of support vector machines is to transform the original input space into a higher dimensional space (see Fig. 5.17). Usually, a high dimensional space is sparse. Mapping the input space into a higher dimensional space thus makes it easier to find separating hyperplanes. In order to map the data, so–called *kernel* functions are searched so that the hyperplane optimally discriminates two classes. Imagine that ϕ is a non–linear function that maps the input $\vec{x} \in \mathbb{R}^n$ to a higher dimensional space $\mathbb{R}^N$, and there the input is linearly separable. Thus, the function separating the input can be written as:

$$f(\vec{x}) = \vec{w}\phi(\vec{x}) + \vec{b} \qquad (5.23)$$

[7] SVMs can be extended to separate n–classes, as each n–class problem can be defined as (n-1) two class problems. Here only the two class case is referred.

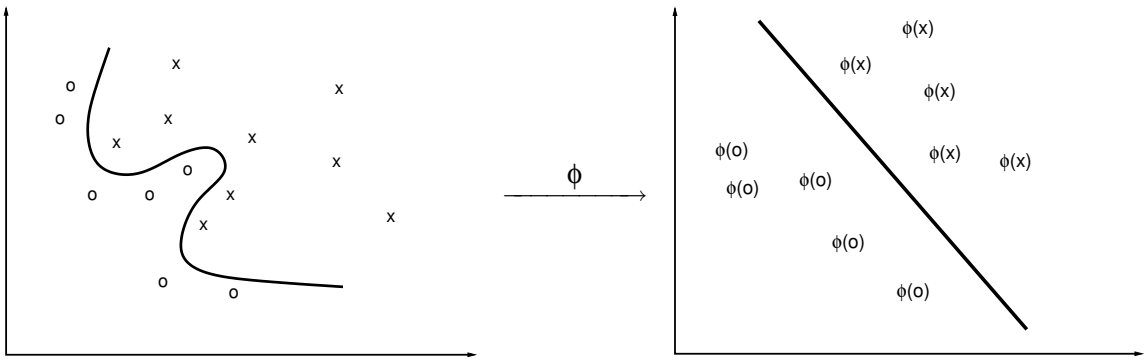


Figure 5.17: *Scheme of mapping input space to feature space using Kernel function ϕ.*

This equation can be reformulated resulting in the Perceptron Learning rule:

$$f(\vec{x}) = \sum_{i=1}^{N} \alpha_i y_i \langle \phi(\vec{x}_i)\phi(\vec{x}) \rangle + \vec{b} \tag{5.24}$$

In equation 5.24, the inner product $\langle \phi(\vec{x})\phi(\vec{x}) \rangle$ is calculated. Here, kernel functions can be applied. Their inner product of the original input ($K(\vec{x},\vec{x})$) is equal to the inner product in feature space. Using this *kernel trick* it is possible to build a classifier with an implicit mapping into feature space:

$$f(\vec{x}) = \sum_{i=1}^{N} \alpha_i y_i K(\vec{x}_i,\vec{x}) + \vec{b} \tag{5.25}$$

Kernel functions have important properties, e.g. a new kernel function can be constructed from other kernels. The properties of kernel functions are discussed in detail in chapter 3 of Cristianini's book. In practice, usually a special transformation function (ϕ) to build a kernel is not searched but already known kernels are taken. A second feature of a SVM is that the classifier can be trained very good, so that the generalisation error is reduced at the same time. This is reached by optimising the distance between the margin of the function separating the classes (functional margin) and the input examples. The functional margin $\tilde{\gamma}_i$ of the training example ($\vec{x}_i, y_i$) is defined as

$$\tilde{\gamma}_i = y_i(\vec{w}^T\vec{x}_i + b) \Rightarrow \tilde{\gamma}_i > 0 \tag{5.26}$$

if $\vec{x}_i$ is classified correctly. The functional margin can be transformed into the geometric margin (γ_i) by using the normalised weight vector $||\vec{w}||$:

$$\gamma_i = \frac{\tilde{\gamma}_i}{||\vec{w}||} \tag{5.27}$$

In case of the hard margin classifier, the margin is then defined as the minimum over all examples I in the training set:

$$\gamma = \min_{1 \le i \le I} \gamma_i \tag{5.28}$$

Figure 5.18 shows the margin for an exemplarily training set. The generalisation error is high if the training set is not compact. But also, the larger the training set is, the smaller the generalisation error. Furthermore, the greater the distance of the input samples to the margin γ is, the smaller is the generalisation error, too.

In order to optimise the SVM a discriminating function is searched that maximises the margin. In case of the hard margin classifier, γ is given by $\gamma = \frac{\tilde{\gamma}}{||\vec{w}||}$ a maximisation of γ is the same as minimising $||\vec{w}||$ for a fixed $\tilde{\gamma}$. Setting $\tilde{\gamma} = 1$ the optimisation problem can be formulated as

$$\begin{aligned} & \vec{w}^T \vec{w} \rightarrow \textit{minimise} \\ \text{with subject to} \quad & y_i(\vec{w}^T \vec{x}_i + b) \geq 1 \quad 1, \ldots, N \end{aligned} \tag{5.29}$$

and can be solved by using Lagrange multipliers and the Karush–Kuhn–Tucker–Theorem (KKT).
Solving the equations above results in:

$$\alpha_i^* [y_i(\vec{w}^* x_i + b^*) - 1] = 0 \quad i = 1, \ldots, N \tag{5.30}$$

Equation 5.30 implies that only those inputs x contribute to the solution for which the functional margin is one. These data points lie closest to the separating hyperplane. Their corresponding α_i^* are non-zero. All other parameters α_j^*, $i \neq j$, $i, j \in N$ are zero and the corresponding input vectors are not involved. Therefore, the contributing inputs are called *support vectors*. The optimal hyperplane for separating the classes is then given by:

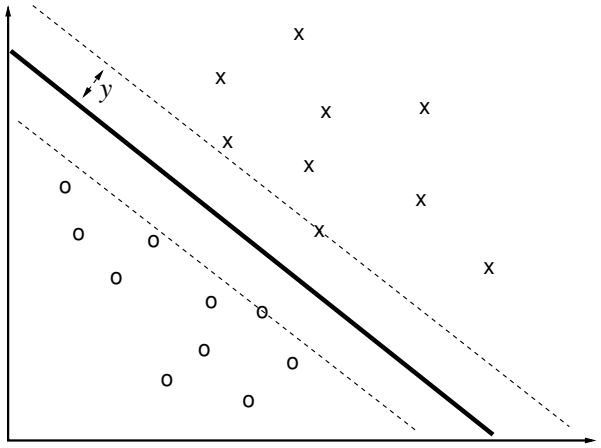


Figure 5.18: *Margin for an example training set.*

$$\begin{aligned} f(x, \alpha^*, b^*) &= \sum_{i=1}^{N} y_i \alpha_i^* (x_i x) + b^* \\ &= \sum_{sv} y_i \alpha_i^* (x_i x) + b^* \end{aligned} \tag{5.31}$$

Here, α^*, b^* are the Lagrange multipliers used for the optimisation and *sv* represents the set of support vectors. In figure 5.19 a maximal margin (bold line) is shown that optimally separates the two classes (x,o). The highlighted examples in the plot mark the support vectors.

Classifying Residue Flexibility using SVMs

After having outlined the principles of support vector machines in this section the application of a SVM as a classifier is described. In this thesis a support vector machine is used

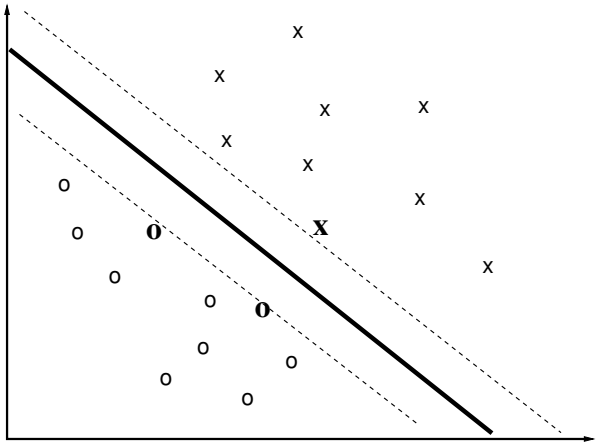


Figure 5.19: *Scheme of a maximal margin. The maximal margin hyperplane (bold) separates the input (x,o) optimally. The bold marked inputs (**x**,**o**) denote the support vectors setting up the margin hyperplane.*

to classify amino acid side chains as flexible or non–flexible. Because every amino acid side chain has specific properties (e.g. charge, polarity, size and number of torsion angles) for each torsion angle and side chain a SVM is trained. In section 5.2.2, several residue specific features have been outlined. From these a feature vector has to be created. In order to choose those features that represent the side chains best a data driven approach is applied. Here, principle component analysis (PCA) is used to select appropriate features and to reduce the dimension of the feature vector at the same time. A reduction of the dimensionality of the feature vector can increase the classification power of the SVM.

In PCA the aim is to produce a set of uncorrelated variables representing the original information. Therefore, the input data ($\{\vec{x}\} \in \mathbb{R}^M$) is rotated to the principle axis using a orthogonal linear transformation. For dimension reduction, then the first n principle components are selected.

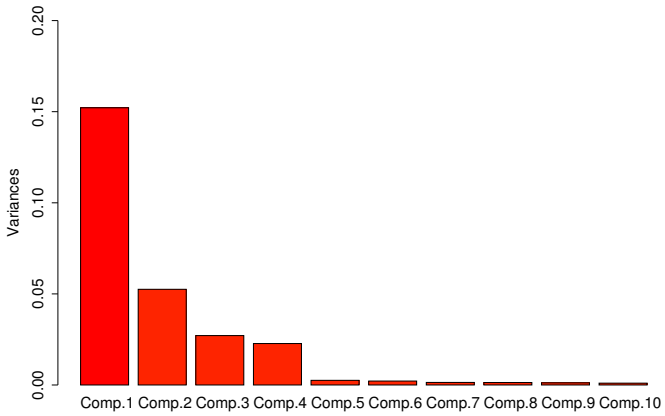


Figure 5.20: *Spectrum of the eigenvalues of all features for ARG and χ_1. Here, only the first 10 (of 21) eigenvalues are shown.*

The selection of principle components can be guided by analysing the spectrum of the eigenvalues (see Fig. 5.20). A hint for cutting off the first n components can be obtained by comparing the differences in variance starting from left to right. In figure 5.20 the variance within the first four components is obviously greater than in the rest of the coefficients. Thus, selecting the first four coefficients is reasonable. Another possibility is to take the first two or only the first component because of the differences in the variances. In case of different possibilities, here the SVM is trained with different numbers of principle com-

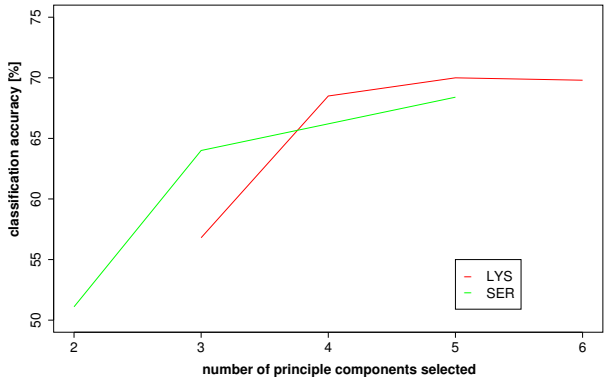
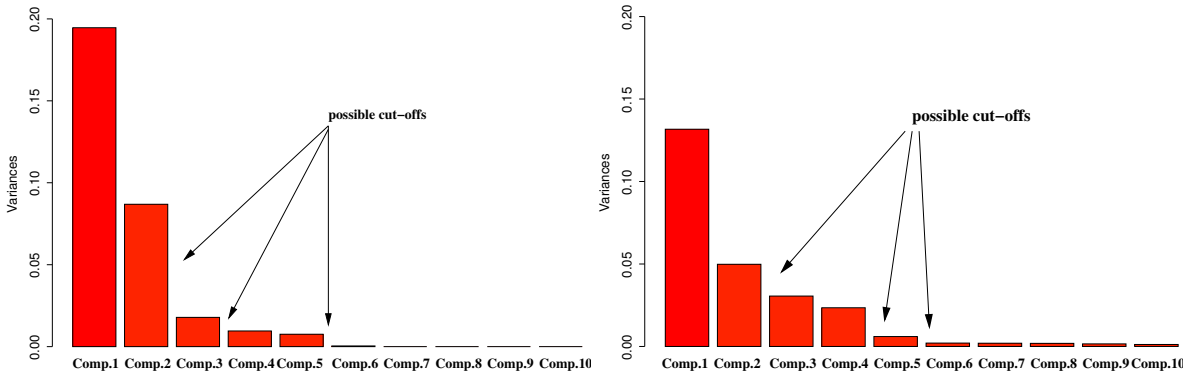


Figure 5.21: *Plot of the total classification accuracy of classifying LYS and SER for different numbers of principle components, used as features.*

ponents. The combination which achieves the highest classification accuracy is then taken. Figure 5.21 shows the classification accuracy for different numbers of principle components of Lysine and Serine. In both cases, the eigenvalue spectrum supports several possible cut–offs (see Fig. 5.22(a) and 5.22(b)).



(a) Eigenvalues of SER

(b) Eigenvalues of LYS

Figure 5.22: *Eigenvalue spectra for LYS and SER of the features. Here, only the first ten eigenvalues are plotted.*

Initially, for all residues the feature vector consists of the following components (cf. section 5.2.2 and Fig. 5.23): a set of wavelet coefficients, the energy difference, the original conformation of the residue, the secondary structure information as well as the solvent accessible surface area value and the temperature factor. The order of the components within the feature vector is not important because the feature vector is processed by the PCA and transformed into lower dimensional space. The features get merged via the transformation into the lower dimensions and a mapping is impossible.

For the first torsion angle (χ_1), the first three to five principle components are chosen according to the analysis of the eigenvalue spectra. The concrete numbers for each residue and torsion angle are given in table 5.1. The eigenvalue spectra for each residue and torsion angle are shown in appendix B.4.

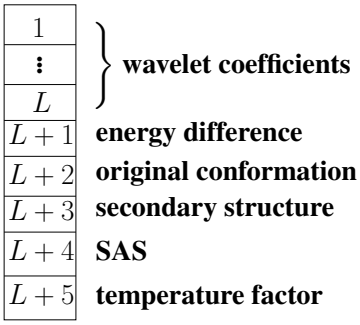


Figure 5.23: *Initial feature vector before the PCA is applied. The number of the wavelet coefficients depends on the level of the MSA they are taken from. The order of the components within the feature vector is arbitrary because the vector is processed further by a PCA.*

The resulting low dimensional feature vectors are then used to classify the residues' side chain as flexible or non–flexible using the support vector machine. Here, the support vector machine implemented in the R–package (Ihaka & Gentleman, 1996; Dimitriadou *et al.*, 2004) is trained. Because the number of examples for training of some residues is small, the SVM is trained using a leave-one out cross test. Within each training iteration the input material is divided randomly in a small test and a larger training set. The SVM is then presented the training set and afterwards it is evaluated by the test set. Here, a 10–fold cross evaluation is chosen.

residue	No. of principle components			
	χ_1	χ_2	χ_3	χ_4
ARG	3	4	5	3
ASN	4	5	–	–
ASP	4	6	–	–
CYS	3	–	–	–
GLN	3	4	4	–
GLU	3	5	4	–
HIS	3	3	–	–
ILE	3	5	–	–
LEU	5	5	–	–
LYS	5	4	3	3
MET	3	6	5	–
PHE	5	5	–	–
SER	5	–	–	–
THR	4	–	–	–
TRP	4	6	–	–
TYR	5	5	–	–
VAL	5	–	–	–

Table 5.1: *Number of the first n principle components selected for each residue and torsion angle. The principle components are taken as features to train the SVM.*

5.2.5 Calculating an Overall Flexibility for Amino Acid Side Chains

In the previous sections two approaches to predict side chain flexibility have been outlined. In both approaches the flexibility has been predicted independently for each single torsion angle. This has been done to avoid infeasible sampling of conformations if all residues torsion angles would have been included at once. In this section a combination of the single torsion angles is described. So, a whole amino acid side chain can be scored according to its flexibility.

Since up to now, the torsion angles were handled independently. A simple method to combine them is to sum up the single contributions:

$$P_{flex}(aa) = \frac{1}{N}\sum_{i}^{N} P_{flex}(aa)_i \tag{5.32}$$

Here, N denotes the total number of torsion angles, $P_{flex}(aa)_i$ the predicted flexibility of the residue aa and torsion angle χ_i. Because $P_{flex}(aa)_i$ is either 0 or 1, the overall flexibility $P_{flex}(aa)$ takes values between 0 and 1.

Another way to combine the single flexibility predictions is to weight each prediction by the distance of the torsion angle from the backbone. The higher torsion angles tend to be more flexible (see Koch, 2003) than the lower ones. Thus, the distance can be incorporated to avoid a bias towards the higher torsion angles:

$$\begin{aligned} P_{flex}(aa) &= \sum_{i}^{N} \frac{1}{d_i} P_{flex}(aa)_i \\ \text{with} \quad d_i &= \sqrt{(C(\chi_i) - C_\alpha)^2} \end{aligned} \tag{5.33}$$

Here, d_i is the distance of the carbon atom (see Tab. 5.2) associated with the torsion angle χ_i from the C_α carbon atom of the backbone.

torsion angle	carbon side chain atom
χ_1	C_β
χ_2	C_γ
χ_3	C_δ
χ_4	C_ε

Table 5.2: *Mapping of torsion angles to the corresponding carbon side chain atoms as used for weighting the flexibility by distances (cf. Fig. 2.3).*

The flexibility $P_{flex}(aa)$ can then be incorporated into the docking system ElMaR (see section 4.2) in order to improve the docking results.

Chapter 6
Enhancement of the ELMAR Scoring Function

In the previous chapters, the enhancement of rigid body docking algorithms by flexibility predictions has been described. In the following a second step to enhance protein docking is outlined. Here, the scoring of docking hypotheses is addressed.

Generally, not only for the ELMAR system, the ranking of docking hypotheses is still not solved today. The main problem is to distinguish between correct and false positive solutions (see Halperin *et al.*, 2002). An optimal set of weights for the individual components can vary between different query proteins. During development those weights can be modified explicitly, but knowledge about the implementation details, especially about the scoring function is needed.

In this chapter a method is proposed to enhance the docking system ELMAR. This approach (**I**ntelligent **P**rotein **H**ypothesis **Ex**plorer, IPHEx) addresses the weighting scheme used in ELMAR, trying to adapt better weights by using relevance feedback techniques.

Following this introduction, the scoring scheme of ELMAR is outlined. Different methods for optimising weights used for scoring are discussed and the aim of the IPHEx system is given. In section 6.2 the principles of Query–by-Content (QbC) systems are presented and their application to protein docking is shown.

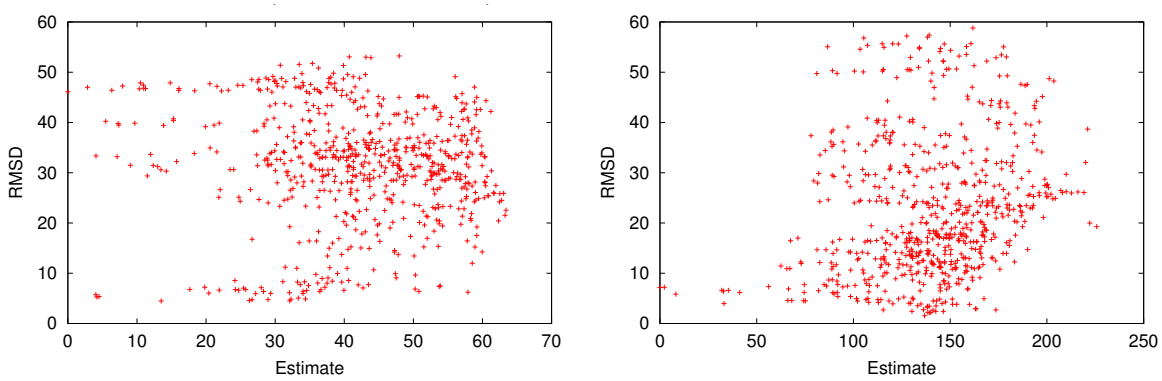
6.1 Ranking Docking Hypotheses using ELMAR

As already outlined in section 4.1 the ELMAR docking system uses the features geometry (P), hydrophobicity (H), and charge (Q) to score the hypotheses proposed by the docking algorithm. In order to combine these features to an overall score two weights (α,β) are used:

$$C = (1-\alpha)(1-\beta) * (P_1 \bullet P_2) + \alpha(1-\beta) * (H_1 \bullet H_2) - \beta * (Q_1 \bullet Q_2) \tag{6.1}$$

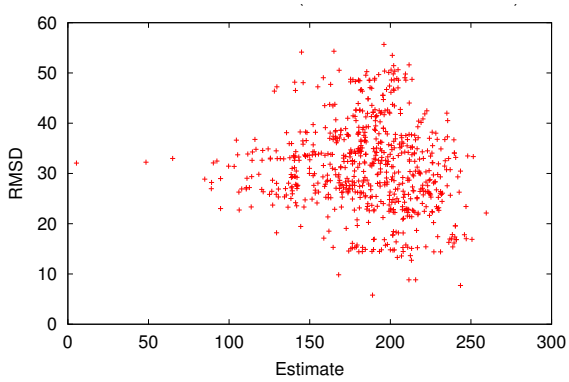
The cost function (C) reflects the ranking of a hypothesis of a complex[1]. For testing the algorithm the root mean square deviation (RMSD) to a known crystallised complex is calculated as standard of truth. Plotting estimate against RMSD gives an overview about the docking algorithms performance (see Fig. 6.1). Ackermann and coworkers (Ackermann *et al.*, 1998)

[1] Hypotheses with large costs are assigned a low rank whereas hypotheses with low costs receive high ranks.

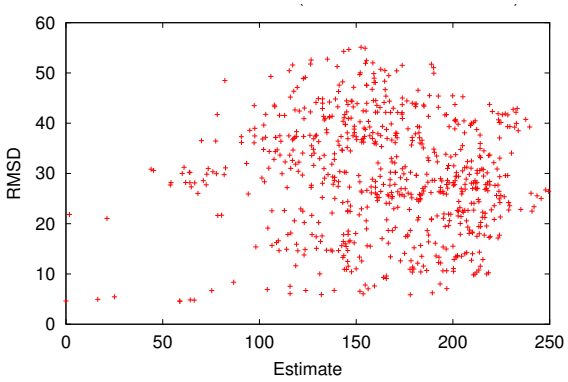


(a) Docking of 1AQ7 and 1BPI.

(b) Docking of 1CHG and 1HPT.



(c) Docking of 1BRA and 4PTI.



(d) Docking of 2PTN and 5PTI.

Figure 6.1: *Results of an* ELMAR *docking run for different test cases. Here, the parameters are chosen as* $\alpha = 0.5$ *and* $\beta = 0.2$*. Each point in the graphics represents one docking hypothesis. On the x-axis the estimate of the costs is plotted against the RMSD on the y-axis.*[2]

estimated the parameters α and β by sampling. For the test set used in their work best results have been achieved for $\alpha = 0.5$ and $\beta = 0.2$. Since the parameters are established for bound docking, they do not necessarily fit for unbound docking. Thus, the weights should be adapted.

Comparing the results of unbound docking shown in figure 6.1, in most cases, besides good hypotheses, also hypotheses with large RMSD values are placed on low ranks by the scoring function (especially in the case of Fig. 6.1(a)). In fact the parameters established on the whole test set do not fit for all test cases. A better approach might be to establish separate sets of parameters for certain subsets of the test set, e.g. protein families or proteins that have similar reaction schemes. Another example of a wrong assignment of ranks is given in figure 6.2. Here, the two docking hypotheses (blue/green) are superimposed to the

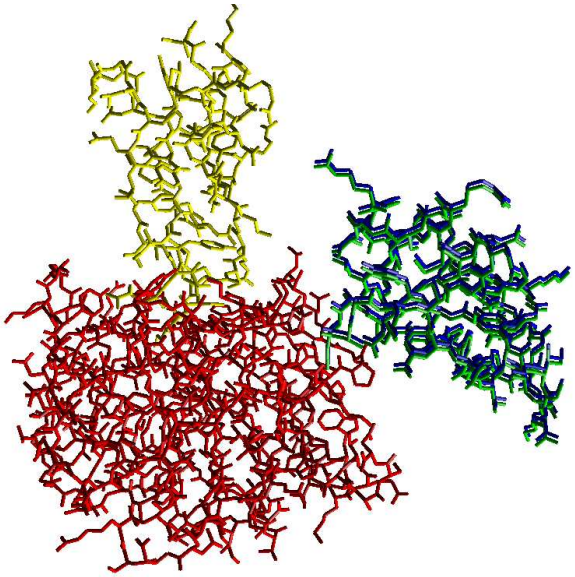


Figure 6.2: *Ranking of hypotheses by the* ELMAR *scoring function. In yellow and red the conformation of the complex 2PTC is shown. In green the best ranked hypothesis is given. It's RMSD is 33Å. In blue a similar hypothesis with the same RMSD (33Å) is shown. This hypothesis has been assigned a rank of 37.*

reference complex 2PTC (yellow/red). The green coloured hypothesis has an RMSD of 33Å. Although it has been assigned the first rank, the blue coloured hypothesis is similar in its transformation and structural error (RMSD of 33Å). In this case ELMAR ranks the hypothesis on position 37.

In the literature, several approaches solve the parameter estimation by formulating an optimisation problem. Rosen and coworkers (Rosen *et al.*, 2000) e.g. propose the ENPOP algorithm that tries to find globally optimal parameters minimising energy landscapes of the problem under investigation. An iterative approach is proposed by Zien and colleagues (Zien *et al.*, 2000), where two steps are performed: first the original application is run (here fold classification of sequences using the program 123D) to produce hypotheses and afterwards a calibration of the data is performed using an external "standard of truth". The calibration method is based on the assumption that good solutions according to the classification score better than bad ones. Two methods called VIM and CIM are formulated that optimise the weights so that an optimal solution to a system of inequalities is found.

Comeau and coworkers (Comeau *et al.*, 2004) proposed a clustering approach applied to a rigid body docking based on a fast Fourier transformation approach like ELMAR. Before clustering the active sites of the docking hypotheses, an energy filter (using electrostatic and desolvation potentials) is applied to cut down the number of possible solutions. Clustering is then performed on bases of pairwise RMSD calculations within the active site. Therefore, the receptor is held fix and for each ligand, all residues within 10Å from its receptor are picked. The RMSD is then calculated between the residue sets of the ligand of each hypothesis. This

approach returns in 31 out of 48 test cases at least one near native structure with an average RMSD of 5Å. The processing of one complex takes up 4 hours on a 1.3GHz 16 CPU IBM pSeries690 server.

In contrast to the approaches mentioned before, the IPHEx system uses relevance feedback techniques adapted from Query-by-Content (QbC) systems (cf. Salton & McGill, 1983; Rui *et al.*, 1998), especially from the INDI (**I**ntelligent **N**avigation in **D**igital **I**mage databases) system (Kämpfe *et al.*, 2002; Bauckhage *et al.*, 2003) which is rooted in information retrieval and was transduced to the field of image databases. Providing an easy to use interface hiding a potentially complex scoring function from the user, a set of hypotheses can be evaluated and scored easily. The 3D visualisation of the docking hypotheses enables the human expert to inspect and compare a hypothesis to other ones or to a known (homologue) complex. The hypotheses are scored from highly relevant to highly non-relevant. The comparison and inspection of a hypothesis is not restricted to the positioning (the geometric complementarity) of the docked proteins. Other features like hydrophobicity and charge can be mapped onto the three–dimensional structures providing additional criteria to score the hypotheses.

After scoring a set of hypotheses, the system modifies the weights within the scoring function according to the feedback. In contrast to optimisation methods this approach also works if no "standard of truth" is available (unknown reference complex). In this case, e.g. one hypothesis out of the predicted ones has to be chosen[3].

Besides this, the approach formulated here can be used to navigate through a large set of docking results searching for hypotheses fulfilling certain criteria (defined by the user). Here, query by content retrieval can also be used. Criteria describing hypotheses to be searched for can be easily defined by picking a hypothesis as a query object. Similarity search can then be applied to find corresponding hypotheses within the set of docking results.

One goal of this work is the improvement of the scoring function and to find good weights (α, β). In a bootstrapping approach, the result of a feedback session is mapped onto all docked test sets in the database that have the same enzyme number assigned. Here, the idea is that proteins possessing the same enzyme number perform the same chemical reaction and thus, also the same biological function. This idea is derived from the definition of the enzyme numbers (c.f. NC-IUBMB, 1992). The enzyme numbers group proteins into classes according to their reaction scheme. Because of this, the docking mechanism of each enzyme class might be similar and the weights can be simply adapted.

6.2 Adapting QbC Techniques for Scoring Docking Hypotheses

In this section an approach to relevance feedback is described that enhances the scoring of the ELMAR docking system. At first a brief overview on Query–by–Content systems and their application to the scoring of docking hypotheses is given. Then the IPHEx system is

[3] Of course additional knowledge about the docking test case is needed, e.g. the location of the active sites of the docking partners.

outlined. Finally the adaptation of the weights which are used for the scoring of docking hypotheses is described.

6.2.1 Query-by-Contents Systems and Protein Docking

In Query-by-Contents systems retrieval is based on similarity search, i.e. the system has to find similar items to a given query item. Relevance feedback is taken to tune the similarity measurement towards the response of the user. The system provides an intelligent interface hiding the individual components of the similarity function. The user gives feedback navigating through the result set one at a time and ranking them from highly relevant to highly non-relevant. Examples for such systems are MARS (Rui *et al.*, 1998) or INDI (Kämpfe *et al.*, 2002). Those systems are applied in the field of image retrieval systems.

IPHEx re-ranks docking hypotheses generated by the ElMaR system based on relevance feedback. In order to describe the method first a docking hypothesis has to be defined formally. A docking hypothesis is a tuple of the transformation $M = (\vec{t}, \vec{r})$, containing a translation $\vec{t} = (t_1, t_2, t_3)^T$ and a rotation vector $\vec{r} = (r_1, r_2, r_3)^T$, and the feature components geometry, charge and hydrophobicity (P, H, Q). Looking at the scoring function of ElMaR (see Eq. 6.1), the features can be interpreted as a similarity measure between docking hypotheses. The weights α, β control the influence of the three features. The user's feedback can be taken to tune the weights of the scoring function.

In QbC–Systems the distance between each feature is calculated and combined into an overall distance measure between two objects. In Figure 6.3, a scheme of the application of QbC techniques to the protein docking is given. On top of the figure, different docking hypotheses calculated using ElMaR are shown. For each hypothesis, a set of the three features (P, H, Q) is determined by the docking system. On bottom of the graphic the reference structure (here a known complex of the data set) is shown. For this complex, the mentioned features are calculated using ElMaR, too. In this case a "faked" hypothesis with a translation of $\vec{t} = \vec{0}$ and a rotation of $\vec{r} = \vec{0}$ is presented to the "Final Docking" module of ElMaR (see section 4.1) to score the complex. The distance d_i between the reference structure and the hypothesis is calculated as follows:

$$\Delta(P_1 \bullet P_2) = (P_1 \bullet P_2)_{complex} - (P_1 \bullet P_2)_{hypothesis_i} \tag{6.2}$$

$$\Delta(H_1 \bullet H_2) = (H_1 \bullet H_2)_{complex} - (H_1 \bullet H_2)_{hypothesis_i} \tag{6.3}$$

$$\Delta(Q_1 \bullet Q_2) = (Q_1 \bullet Q_2)_{complex} - (Q_1 \bullet Q_2)_{hypothesis_i} \tag{6.4}$$

In a QbC-system each of these distances is combined via a corresponding weight to an overall score. The scoring of ElMaR (see 6.1) can be reformulated similar for a single hypothesis i as:

$$C = w_1 * \Delta(P_1 \bullet P_2) + w_2 * \Delta(H_1 \bullet H_2) - w_3 * \Delta(Q_1 \bullet Q_2) \tag{6.5}$$

In equation 6.5 the weights $\vec{w} = (w_1, w_2, w_3)^T$ regulate the contribution of the features to the overall score C. In order to map these weights to the original weights α and β, the following

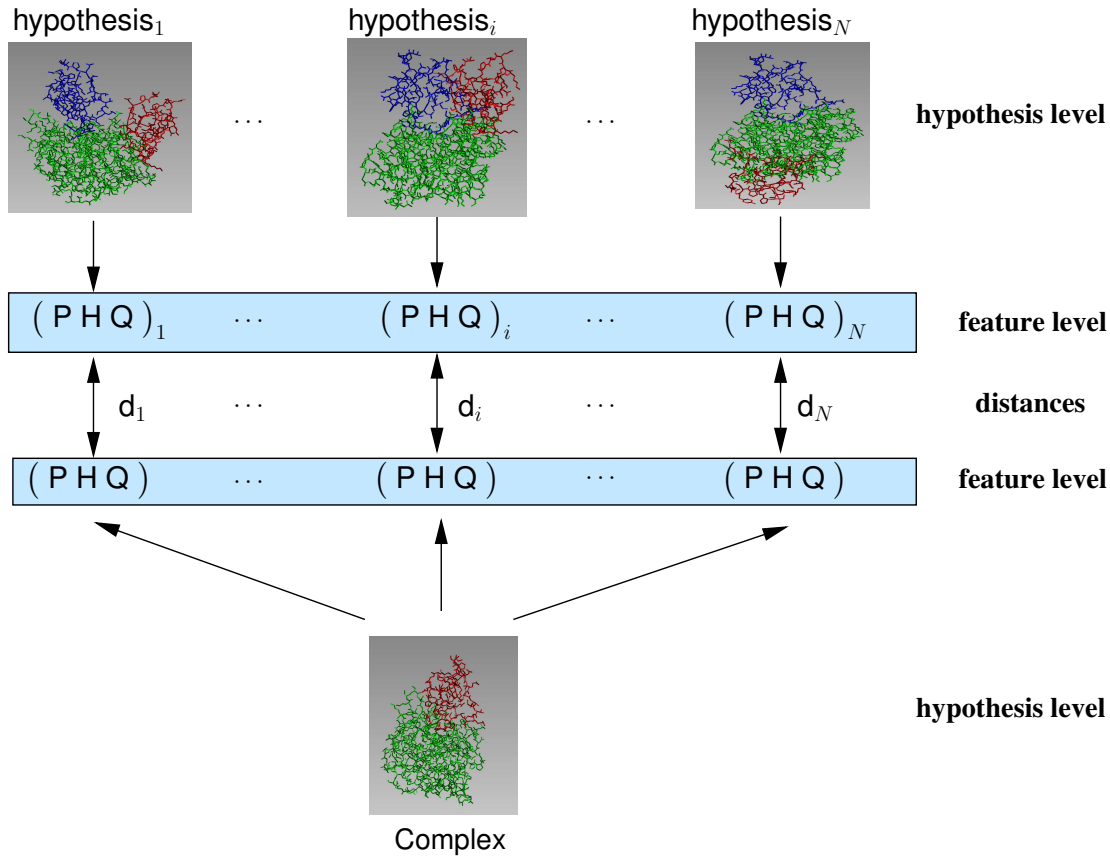


Figure 6.3: *Scheme of using QbC techniques for scoring hypotheses generated by* ELMAR. *The distance d_i between a hypothesis i and the reference complex is calculated from the features (P,H,Q) and is combined via weights to a score (see Eq. 6.5).*

equations are set up:

$$w_1 = (1-\alpha)(1-\beta) \qquad w_2 = \alpha(1-\beta) \qquad w_3 = \beta \tag{6.6}$$

An additional condition has to be introduced to solve these equations :

$$\sum_i w_i = 1 \tag{6.7}$$

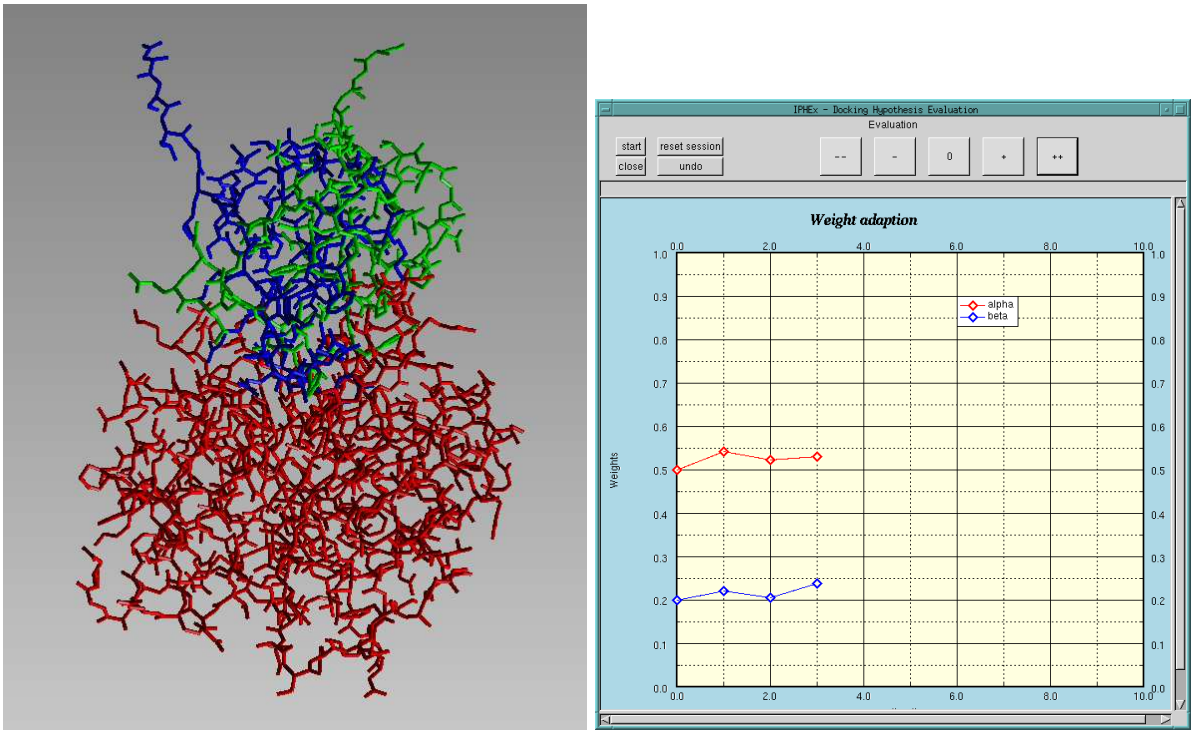
Thus, α and β can then be expressed as follows:

$$\alpha = \frac{w_2}{1-w_3} \qquad \beta = w_3 \tag{6.8}$$

In order to take care of equation 6.7 the weights w_1, w_2 and w_3 are normalised and the weight w_1 is incorporated indirectly.

6.2.2 The IPHEx System

After having outlined the similarity calculations used within IPHEx, the next step is to explain how relevance feedback can be applied to the system, and how the adaption of the weights is realised.



(a) 3D visualisation of the enzyme (1CHG) coloured red, and the inhibitor (1HPT) blue, a potential docking solution is coloured in green. This is an example how docking hypotheses are presented to the user by the IPHEx system.

(b) User interface for the navigation through the set of docking hypotheses and for giving feedback. On top, buttons for navigation (left) and feedback (right) are localised, below the navigation panel the development of adapted weights is shown in a plotting widget.

Figure 6.4: IPHEx *system. On the left, the super-imposition of a true complex and a hypothesis which is presented to the user, is shown. On the right, the navigation and feedback panel is given.*

The IPHEx system consists of two modules, the visualisation component ViWish (Klein *et al.*, 1996) (see Fig. 6.4(a)) for presenting the docking hypotheses to the user and a feedback module (see Fig. 6.4(b)) to navigate within the set of hypotheses and to give feedback. The docking constellation visualised by the ViWish can be coloured according to the biochemical features, aiding the user's interpretation. The navigation panel in the second module lets the user navigate through the set of hypotheses and rank the hypotheses from highly non–

relevant to highly relevant ($--, -, 0, +, ++$). Here, highly relevant means that the given hypothesis is similar to the reference structure, e.g. the known complex. Highly non-relevant hypotheses are those differing from the given reference, respectively. Below the navigation and feedback panel the development of the adapted weights is plotted for each iteration of feedback.

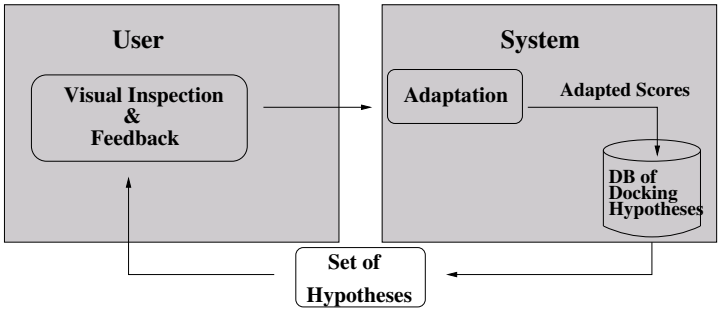


Figure 6.5: IPHEx *system. Left: comparison and feedback by the user. Right: adaptation performed by the system.*

Figure 6.5 gives an overview of the running IPHEx system. The ELMAR docking system generates a set of 700 hypotheses per docking test case. These hypotheses are stored within a relational database. Because a user can not rank the total number of hypotheses initially a subset, here 20 out of the 700 hypotheses, is chosen randomly. This set is ordered according to the distance between the hypotheses and the reference complex (see Eq. 6.5). Afterwards it is presented to the user. The user inspects the set of hypotheses and gives feedback according to the similarity to the reference structure. After having scored the set of hypotheses an adaptation of the weights is performed by the system resulting in an updated set of hypotheses.

6.2.3 Adapting Weights of the Scoring Function using QbC Techniques

The adaptation of the weights $\vec{w} = (w_1, w_2, w_3)^T$ is controlled by the user's relevance feedback. In section 6.2.1 the similarity between a docking hypothesis and the reference structure is defined by the distance of the features hydrophobicity (H), geometry (P) and charge (Q). In order to adapt the weights the user's feedback and the similarity of the hypotheses have to be combined.

During an iteration the user ranks the list of hypotheses HL by providing scores S from highly non-relevant to highly relevant. After each scoring iteration the feature weights w_i are updated using the following equation:

$$w_i' = w_i + \varepsilon \sum_{m \in HL} S(H_m^{HL}) F(R(H_m^{HL}, HL_i)) \tag{6.9}$$

Updating the weight w_i is done by adding up the feedback of all scored hypotheses. In order to reflect the user's feedback first the rank of a hypothesis within the ordered feature list HL_i,

$i \in \{1,2,3\}$ (denoted as $R(H_m^{HL}, HL_i)$) is determined[4]. Each feature list consists of the ordered hypotheses according to their distance to the reference structure (see Eq. 6.2, 6.3, and 6.4). A low rank within a feature list implies that this feature supports the hypothesis, a high rank does not. For combining the rank $R(H_m^{HL}, HL_i)$ with the feedback $S(H_m^{HL}) \in \{-3,-1,0,1,3\}$[5]

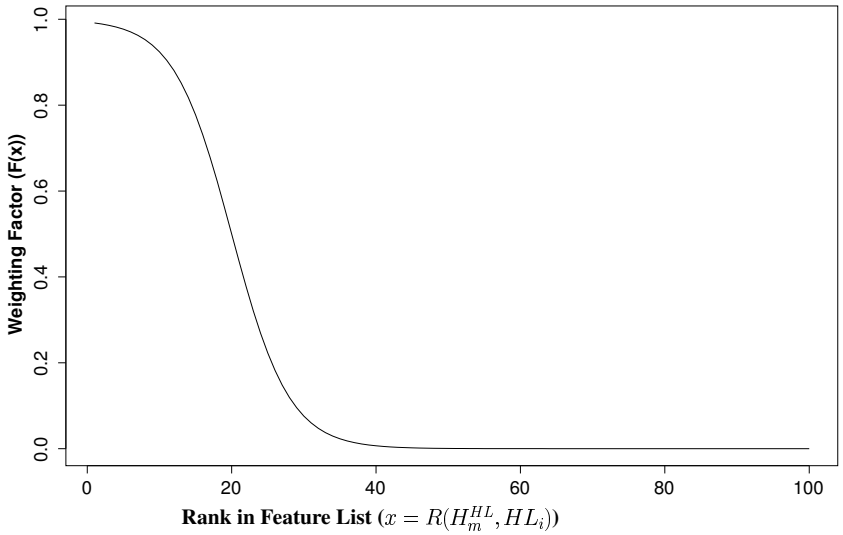


Figure 6.6: *Function F, on the x–axis the rank is plotted, the feedback of a hypothesis with a low rank is propagated stronger than the feedback of a hypothesis with a higher rank.*

of the hypothesis H_m^{HL} the function F is used. It weights the feedback reflecting the support of the feature using the rank of H_m^{HL} within each feature list. Thus, the value of the product of F and S represents the value, the weight w_i is changed for the hypothesis H_m^{HL}.

The functionality of F can be described as follows: If a hypothesis H_m^{HL} has been scored as relevant by the user and its rank within the feature list HL_i is low than the according weight w_i is increased because the function F propagates the score to the weight. An increase of the weight assumes that the corresponding feature is important for the docking process of the test case. Similar, if H_m^{HL} has been assigned a low score the corresponding weight is decreased as its assigned feature should not contribute further to the scoring. In case of a hypothesis with a high rank in the feature list the assumption is that the feature is not relevant. Therefore, its score should not contribute to a change of the corresponding weight and the function F returns a value near to zero. So, the function F is designed monotonic and decreasing (see Fig. 6.6).

Finally, the sum over the feedback scores is weighted by the learning rate parameter ε to lessen the impact of the feedback to the weight. At the end of a session the modified

[4]The index i of the feature list HL_i corresponds to the index of the weight w_i.

[5]Numerical representation of the feedback from highly non–relevant (-3) to highly relevant (3). The values are chosen arbitrary.

weights are stored within a database for re-use or to score other complexes in the same family.

Chapter 7
Results

In this chapter the results of this study are presented. At first the data on which all experiments have been carried out is described. In section 7.2 the performance of the classifiers is evaluated. The threshold based method is tested using Receiver Operation Characteristic (ROC) analysis (see section 7.2.1). The results and the accuracy of the SVM classifier are outlined in section 7.2.3. A second evaluation procedure is described in section 7.3. Here, the gained flexibility information is incorporated and tested in the docking system ElMaR.

A second goal of this thesis is to improve the scoring of the ElMaR system. The results of this approach, using a relevance feedback method (IPHEx), are presented in section 7.4. This chapter summarises with a discussion.

7.1 Data Set

Several approaches have been proposed in this work: flexibility prediction, evaluation of flexibility information within the docking system ElMaR, and improvement of scoring by relevance feedback. Thus, to train and evaluate these approaches three dimensional structures of proteins are required. Publically available protein structures are provided by the Brookhaven Protein Database – Pdb (Bhat *et al.*, 2001). This database consists of nearly 25.000 structures (cf. Research Collaboratory for Structural Bioinformatics (RCSB), 2003) mostly refined by crystallography and fewer by NMR spectroscopy or homology modelling. The number of deposit structures is steadily growing (see Fig. 7.1). Therefore, automatic methods have been developed to fetch new Pdb-releases and to update all information calculated on the protein structures (see section 7.1.1).

Besides this, the data set used in this work has to fulfil several criteria. The classification is done on unbound protein structures. So, at first the Pdb structures have to be classified into complex and unbound proteins. Thus, test cases (consisting of a complex and two unbound proteins) for the protein–protein docking can be derived automatically without additional calculations.

Since most Pdb structures are refined by crystallography, the resolution should be as high as possible to guarantee precise atom coordinates. Furthermore, these protein models lack of hydrogens due to the refinement method. Since the hydrogens are important for the energy

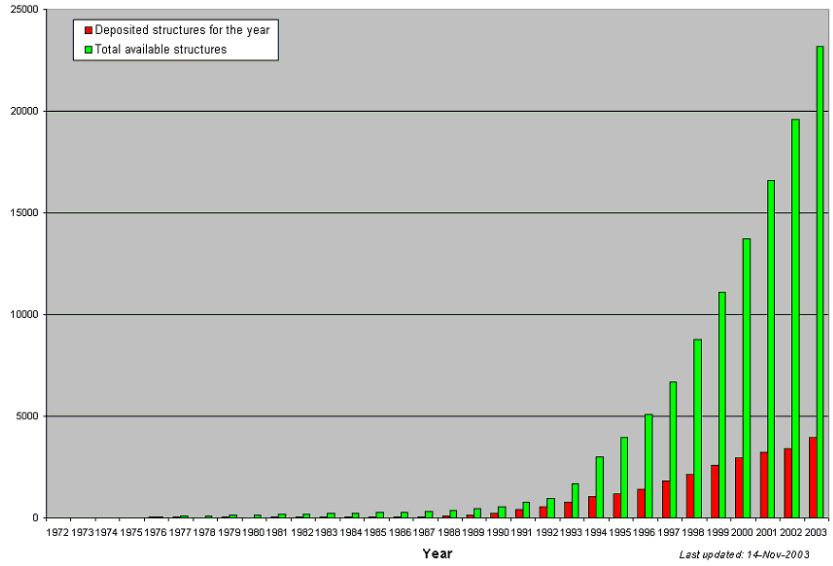


Figure 7.1: *Growth of* PDB *(Research Collaboratory for Structural Bioinformatics (RCSB), 2003). The exponential growth of protein structures is obvious. In green the total number of structures available is depicted. The red bars represent the deposited structures per year.*

evaluation, they have to be added. Additionally, all structures are checked for completeness (no missing atoms) and valid bonds.

In order to evaluate the classification a labelled test set is needed. Using the structure comparison methods of Koch (Koch, 2003), each residue and torsion angle of the data set is labelled. Here, automatically generated test cases are used for comparison of complex and unbound structures.

7.1.1 Automatic Test Set Generation

In order to use PDB structures, a lot of preparation has to be done. Therefore, an automation of this procedure is helpful. Here, a modular and pipelined system has been set up. It can process new structures by applying certain defined criteria to the structures. The system is divided into two parts. In figure 7.2 the first part calculating test cases for protein–protein docking is shown. Some of the modules (cf. blue boxes) have to be run every time an update of the test cases is scheduled, e.g. due to an update of the PDB. The **init_first** module has to be run once while installing and setting up the system for the first time. All other modules can be run incrementally or in batch mode (processing all entries sequentially). In case of an incremental update, new entries can be distributed to other CPUs to increase speed. The system is back ended by a database system (MYSQL 4.1) for storing and serving data to the pipeline. A control framework is attached to the pipelined system. It watches the processing of the modules and checks the dependencies between them. If one criteria fails,

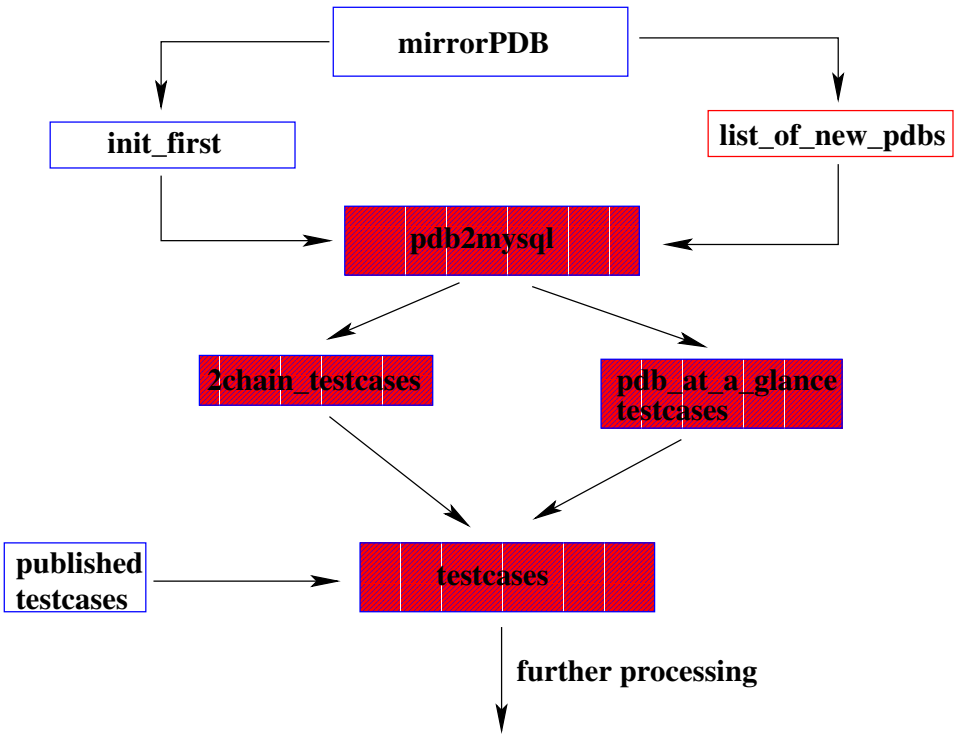


Figure 7.2: *First part of the automatic processing of* PDB *structure for flexibility investigation and docking. The blue and red shaded boxes represent modules working incrementally, whereas the blue outlined boxes have to be run every time. The boxes filled with the pattern can be run in both modes.*

the structure is omitted from the final test set. The control framework is realised using the Java build-tool *ANT* (Loughran, 2002).

The **mirrorPDB** module compares the locally stored version of the PDB to the one hosted by the official server. New entries are copied to the local repository of protein structures. This is done using the perl tool *mirror* (McLoughlin, 2003). Then, a list of new structures is compiled. The module **list_of_new_pdbs** generates a job file for each entry so that the calculations can be distributed to multiple computers.

In a next step (**pdb2mysql**), either in batch mode or incrementally, meta information from the PDB files is extracted and stored within a relational database similar to 3DInsight (An *et al.*, 1998). The extracted information, e.g. the number of chains, is used to search for protein complexes and their unbound sequence identical parts. The search for protein complexes is based on three different heuristics (cf. Neumann, 2003). On the one hand, classified complexes from the PDB_at_a_glance (Pearlstein & FitzGerald, 1996) are taken (cf. PDB_**at_a_glance testcases** in Fig. 7.2). On the other hand, protein structures are defined as complexes if they consist of two chains and if there exist proteins within the PDB that are sequence identical to one part of the complex (cf. **2chain_testcases** module). This pair of single chained proteins have not be sequence identical to each other. As a third method,

certain naming conventions are considered: Entries with the chain identifiers A, B and I are usually an enzyme (consisting of the chains A and B) and its inhibitor (chain I). Some other name conventions can be found, like chain identifiers L and H for antibodies. The search process can be formulated as a SQL query and is executed on the database (cf. Neumann, 2003, chapter 4). The intersection of the result sets of the three methods forms an initial test set.

In order to compare docking results, test sets (Chen & Weng, 2002; Halperin *et al.*, 2002; Norrel *et al.*, 1999; Betts & Sternberg, 1999) also published are incorporated. According to certain quality criteria (e.g. resolution of the structure or chain length) individual subsets can be created for further processing. Since most protein–protein docking test cases are hand picked by each researcher the automatic test set generation is provided as a web service (Zöllner *et al.*, 2004) called *agt-sdp* (**A**utomatic **G**enerated **T**est-**S**et **D**atabase for **P**rotein–Protein Docking) to access and select data sets easily.

Based on these test cases further preparation of the PDB structures may be run. This is also done automatically. Figure 7.3 gives an overview of this second part. At first, the structures are checked for completeness and verified (e.g. correct bond lengths). This is done to ensure that the energy calculations will be correct. Missing atoms or wrong bond length will influence the total energy. Since hydrogen atoms cannot be detected in the refraction pattern during crystallography, these atoms are not included within the original PDB structure file. They have to be added to complete the structure (see section D.1).

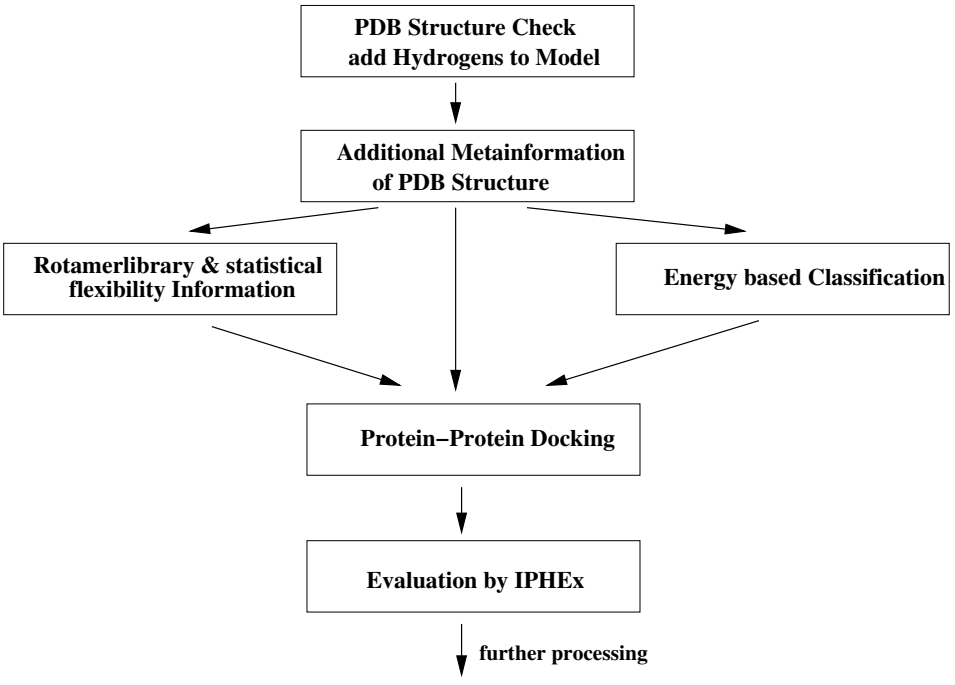


Figure 7.3: *Second part of the automatic processing of* PDB *structure for flexibility investigation and docking. The docking module was implemented by Neumann (Neumann, 2003), whereas the module of the rotamer library was developed by Koch (Koch, 2003).*

For the structure comparison and the evaluation of the flexibility, several additional information, e.g. secondary structure or solvent accessible surface area (SAS) of the structure have to be collected. For details about the different tools which have been summarised within the module **additional Meta-information** see appendix D.1.

In the next step, flexibility information (e.g. using rotamer statistics (Koch, 2003) or the classification approach (cf. chapter 5)) is calculated on the protein structure which is then provided to the docking algorithm. The results of the docking are handed over to further evaluation modules. In a first step docking hypotheses can be re–ranked using the IPHEx module (see section 6.2).

7.1.2 Description of the Data Set

In this section the data set used in this study is described. Initially, 24475 PDB structures (mirror of 2nd February, 2004) have been registered within our database. Out of these a test set of 77 complexes and 88863 test cases has been automatically derived. Table 7.1 shows the initial numbers found by each method and the results of the intersection of these. For

Method	raw test cases		final test cases	
	# Complexes	# Test cases	# Complexes	# Test cases
Complex2Unbound	57	88803	49	87531
Unbound2Complex	487	175036	28	1272
from literature	75	197	75	197

Table 7.1: *Intersection of the individual sets found by each method. Besides the intersection several quality criteria have been applied for the final set.*

the data set the resolution of each structure should be between 0.1 and 2.5Å to pick good structures and to reject structures refined by modelling techniques or NMR. The minimum chain length should be more or equal than 30 residues to avoid peptides. For flexibility classification all protein structures have to pass the PDB **structure check** module.

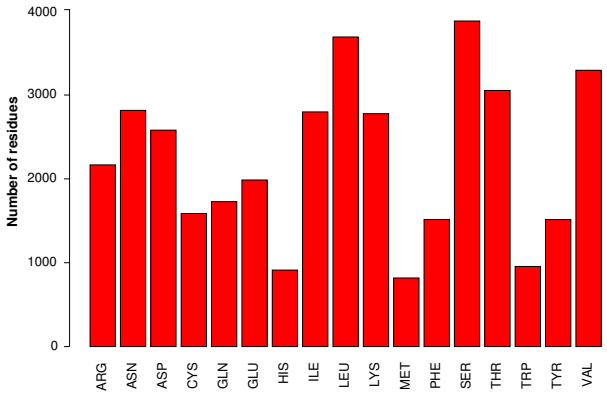
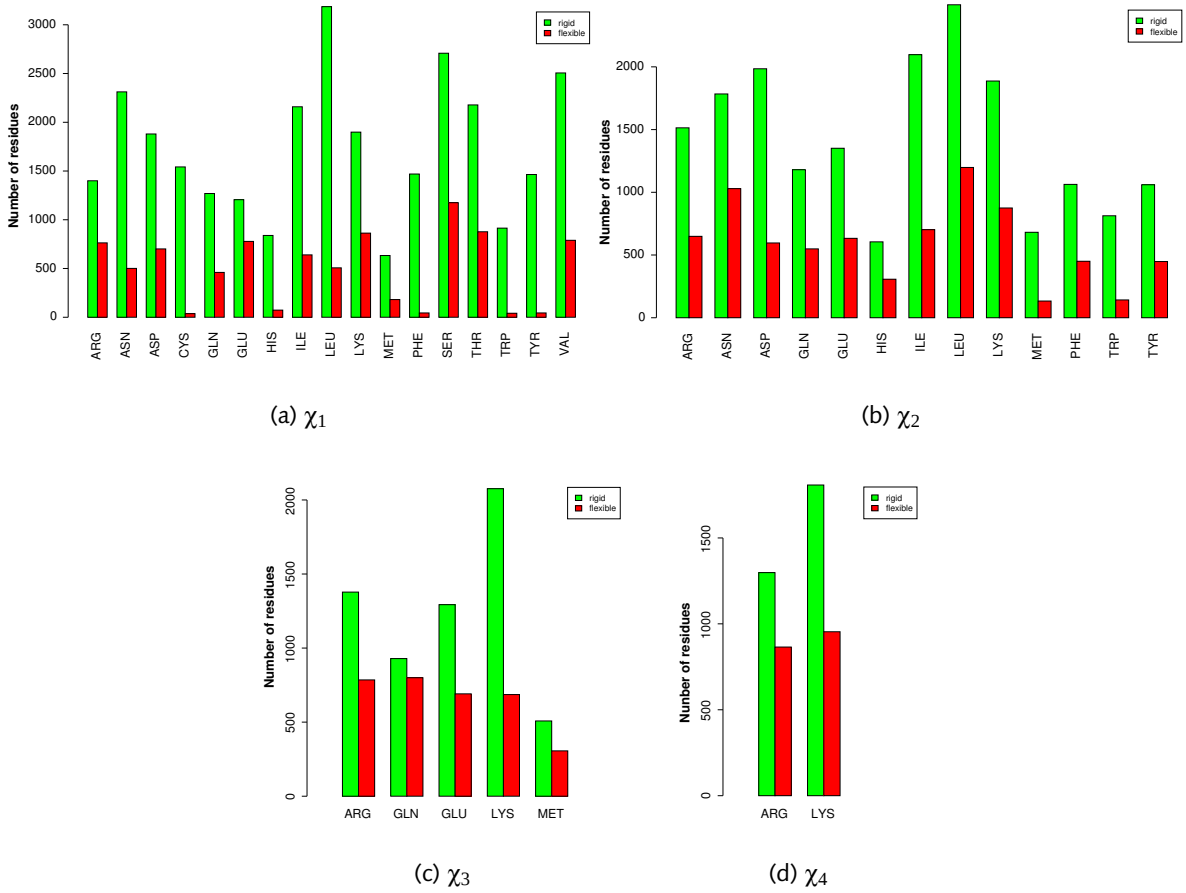


Figure 7.4: *Histogram of residues per amino acid type. The total amount of residues is 44345 from 232 protein structures, not counted ALA, GLY and PRO.*



(a) χ_1 (b) χ_2

(c) χ_3 (d) χ_4

Figure 7.5: *Histogram of labelled data set used for the prediction. The red bars visualise the number of residues changing their rotamer, the green marked bars represent the number of non–flexible residues.*

The final data set used for the classification tasks contains 232 unbound proteins and 44345 residues. Each residue in this data set is labelled whether it changes a rotamer or not (see section D.1). Figure 7.4 shows the number of residues for each amino acid type whereas figure 7.5 gives the histogram of the two classes for each residue type and torsion angle.

The unbound protein structures are from the different enzyme classes (NC-IUBMB, 1992) as shown in table 7.2. In the last line of the table the number of all proteins are summarised which have no EC number assigned[1].

Unequal class sizes can influence the result of a classifier[2]. The number of flexible residues is usually lower than the number of rigid residues due to the packing of the protein. Most amino acids reside within the core of the protein. This can be observed in the data used as

[1] Either these proteins are no enzymes, e.g. `1G7H` is involved in gene regulation or those proteins have no yet an EC number assigned.

[2] The classification can be biased towards the bigger class.

EC number	Description	counts
1.1.1.1	Alcohol dehydrogenase.	2
2.1.2.2	Phosphoribosylglycinamide formyltransferase.	1
2.4.2.9	Uracil phosphoribosyltransferase.	1
2.6.1.1	Aspartate transaminase.	10
3.1.27.5	Pancreatic ribonuclease.	10
3.2.1.17	Lysozyme.	65
3.2.1.91	Cellulose 1,4-beta-cellobiosidase.	1
3.4.21.1	Chymotrypsin.	4
3.4.21.4	Trypsin.	37
3.4.21.62	Subtilisin.	1
3.4.23.16	HIV-1 retropepsin.	19
3.4.24.17	Stromelysin 1.	7
5.2.1.8	Peptidylprolyl isomerase.	16
6.3.2.19	Ubiquitin–protein ligase.	1
6.3.4.4	Adenylosuccinate synthase.	7
–	–	61

Table 7.2: *Data set of unbound proteins used for classification of residue flexibility, grouped by EC number. In the last line, all proteins without no EC number assigned and which are used for the flexibility classification are listed.*

shown in figure 7.5. In order to avoid this the number of examples is reduced so that each class (flexible, non–flexible) has an equal size.

For testing the flexibility information within the docking system ElMaR, test cases have to be compiled. Out of the 88863 test cases (see table 7.1) derived from Pdb and for the 232 unbound proteins used for the flexibility calculations, a set of 17023 examples has been extracted from the database. Although, the ElMaR system is designed for speed, the calculation of all test cases will last nearly a full year[3]. Thus, a subset of 245 test cases matching 18 different complex structures has been chosen finally according to the crystallographic resolution of the structures. A detailed list of the used structures is given in table A.3. Table 7.3 gives the number of test cases per complex.

Complex	1A2W	1A7X	1ADE	1ADI	1AFK	1AFL	1AFU	1AO6	1APN	1ARG	1ASM	1ASN	1B2K	1BM0	1CGI	1LYS	1TPA	2PTC
counts	7	6	1	1	9	6	10	2	13	8	4	4	113	1	2	54	2	3

Table 7.3: *Test Cases grouped by their reference complex.*

A bias towards the complex 1B2K and 1LYS due to the fact that the test cases are chosen according to the unbound proteins used for the flexibility predictions. The largest group of unbound proteins are Lysozymes. The complexes 1B2K and 1LYS and also their unbound parts belong to this class. Thus, these proteins are more frequent within the set of test cases.

[3] Here, a docking run is averaged by 20 minutes, see (Neumann, 2003).

7.2 Classification Results

In the last section, the data used for classifying the flexibility and performing docking experiments have been outlined. In the following, the results of the different approaches are presented.

7.2.1 Evaluating Threshold based Classifier by ROC Statistics

The first classification approach is based on a threshold to discriminate flexible and non–flexible residues (see section 5.2.3). In order to evaluate this approach, here Receiver Operating Characteristics (ROC) analysis is chosen. ROC is a straight forward method to analyse the performance of various kinds of applications. Before evaluating the classification, a brief introduction to ROC is given. Then, the performance of the classification system is evaluated. Here, ROC analysis is also used to search for good thresholds to classify the flexibility.

Introduction to Receiver Operating Characteristics

Receiver Operating Characteristic analysis is a method originating from the field of "Signal Detection Theory" (Egan, 1975) and was developed for the analysis of radar images. The main objective in signal detection theory is to decide whether there is a certain information within a signal (like a spot on a radar image) or just noise. Because of this, ROC analysis can be used as a method to test a decision system's accuracy. Beside the traditional field of signal detection theory ROC analysis has been applied to the field of medicine and health care, especially in radiology to measure the accuracy of diagnostic systems (Swets, 1988; van Erkel & Pattynama, 1998) and has been recently applied in the field of machine learning (Bradley, 1997), data mining (Drummond & Holte, 2000) and bioinformatics (Bilban *et al.*, 2002).

In ROC analysis the basic idea is to calculate the probability that a residue is flexible under the consideration given a certain feature (here energy difference greater a certain threshold). The selection of a threshold influences the performance, e.g. here the number of correctly predicted residue side chains (see Fig. 7.6). Moving the threshold causes an increase in false positive examples or false negative examples according to the direction the threshold is moved.

		Predicted	
		$PRC(aa) = 0$	$PRC(aa) = 1$
True	not flexible	true negative (TN)	false positive (FP)
	flexible	false negative (FN)	true positive (TP)

Table 7.4: *Scheme for confusion matrix applied to flexibility prediction.*

The result of a prediction given a certain threshold can be evaluated by a 2×2 table, also called *confusion matrix* (see Tab. 7.4). Here, the four fields of this table are defined as follows:

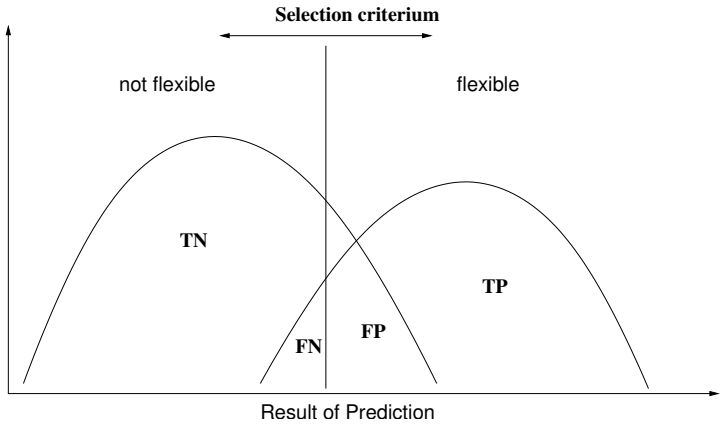


Figure 7.6: *Scheme of a threshold based prediction. Here, the possible distributions of a prediction result given a certain threshold (selection criteria) is shown. The examples left of the threshold (marked by the vertical line) are not flexible examples whereas the examples on the right are flexible. Moving the threshold to the left or right influences the number of correctly and wrong classified examples.*

- TP, number of residues which have been predicted as flexible and which are flexible.
- FN, number of residues which have been predicted as not flexible, but are flexible.
- FP, number of residues which have been predicted as flexible, but are not flexible.
- TN, number of residues which have been predicted as not flexible and are not flexible.

A disadvantage of these tables is that the performance cannot be seen easily from the numbers. Even more, one is interested in the specificity and sensitivity of the method. The specificity here is the proportion of residues which are not flexible and which are not predicted as flexible. The sensitivity is defined as the proportion of residues which are flexible and also have been classified as flexible. These values can be calculated using the following equations:

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP}+\text{FN}} \qquad \text{Specificity} = \frac{\text{TP}}{\text{TP}+\text{TN}} \tag{7.1}$$

In medical statistics the word “specificity" is often used in a different sense, meaning the chance of correctly predicting a negative example (Baldi & Brunak, 2001). It is also called *selectivity* or false alarm rate (see Eq. 7.2).

$$\text{Selectivity} = \frac{\text{TN}}{\text{FP}+\text{TN}} \tag{7.2}$$

The performance of a method depends on the tradeoff between sensitivity and specificity, e.g. how many false positives and false negative can be allowed. In order to visualise the numbers, a so called ROC curve can be drawn, plotting the sensitivity (or hit rate) against false positive rate. An example of such a curve is shown in figure 7.7.

ROC curves give an overview about the performance of the system depending on the threshold. The position and form of the curve is an indicator how well the prediction system works. A diagonal in the plots (see Fig. 7.7) marks the chance line, indicating that the prediction is random. Curves above the diagonal represent good prediction results. The prediction is better the nearer the ROC curve is to the left and upper boundary of the plot. A curve below the diagonal indicates that the predictions failed.

A performance measure independent of the threshold can be derived directly from this curve, using the area (A_{ROC}) under the ROC curve (Bradley, 1997).

$$A_{ROC} = \int_0^1 ROC \tag{7.3}$$

This criterion can be used to decide which method is best on a given problem (like here, different normalisation factors, see Fig. 7.7).

Another question arising from this analysis is which threshold is optimal so that the classifier performs best. In ROC analysis theory, the best threshold corresponds to the point on the ROC curve that is closest to the upper left corner of the plot (see Eq. 7.4). On the one hand one could assign costs to the classification results and then perform a maximum likelihood optimisation (Metz & Pan, 1999; Drummond & Holte, 2000) to estimate this point. But therefore, corresponding costs have to be estimated or defined (Foster & Fawcett, 1997). Here, simply the distance of each point ($\vec{x_i}$) to the upper left corner ($\vec{c}$) is evaluated, since a cost function is difficult to estimate.

$$T = \min_i(\sqrt{(\vec{c} - \vec{x_i})^2} \tag{7.4}$$

7.2.2 Results of the Threshold based Classification

The protein data described in section 7.1.2 has been applied to the threshold based classifier. At first an optimal normalisation factor has been searched by testing six different scaling factors for the whiskers. The optimal normalisation factor (Nf) is picked by comparing the area under the ROC curves. Table 7.5 summarises the results for the χ_1 angle.

For most of the residues best results for χ_1 are reached using a normalisation factor of $0.5 * (Q_3 - Q_1)$. Here, $(Q_3 - Q_1)$ denotes the inter–quantile distance of the box-plots (see section 5.2.3). For CYS and MET residues the optimal normalisation factor is $6 * (Q_3 - Q_1)$. For the other torsion angles (χ_{2-4}) the largest ROC areas are received by a normalisation factor of $0.5 * (Q_3 - Q_1)$ (see also Tab. B.1, B.2, B.3, and B.4).

Figure 7.7 shows the ROC curves of ARG (χ_1), TRP (χ_2), MET (χ_3) and ARG (χ_4). For each plot 1-specificity is plotted against the sensitivity. The curves are coloured according to their normalisation factor used within the classification. The diagonals drawn within the plot represent the "chance line" meaning classifying a residue by random. The different normalisation factors do not affect the prediction results much except for Q_4. The Q_4 normalisation factor extends the whiskers to the outmost point including all energy differences

AS	χ_1		χ_2		χ_3		χ_4	
	A_{ROC}	Nf	A_{ROC}	Nf	A_{ROC}	Nf	A_{ROC}	Nf
ARG	0.74	$0.5 * \Delta Q$	0.63	$0.5 * \Delta Q$	0.64	$0.5 * \Delta Q$	0.61	$0.5 * \Delta Q$
ASN	0.69	$0.5 * \Delta Q$	0.62	$0.5 * \Delta Q$	–		–	
ASP	0.64	$0.5 * \Delta Q$	0.51	$0.5 * \Delta Q$	–		–	
CYS	0.75	$6 * \Delta Q$	–		–		–	
GLN	0.72	$0.5 * \Delta Q$	0.69	$0.5 * \Delta Q$	0.55	$0.5 * \Delta Q$	–	
GLU	0.60	$0.5 * \Delta Q$	0.55	$0.5 * \Delta Q$	0.54	$0.5 * \Delta Q$	–	
HIS	0.74	$0.5 * \Delta Q$	0.75	$0.5 * \Delta Q$	–		–	
ILE	0.62	$0.5 * \Delta Q$	0.65	$0.5 * \Delta Q$	–		–	
LEU	0.65	$0.5 * \Delta Q$	0.64	$0.5 * \Delta Q$	–		–	
LYS	0.66	$0.5 * \Delta Q$	0.55	$0.5 * \Delta Q$	0.51	$0.5 * \Delta Q$	0.49	$0.5 * \Delta Q$
MET	0.63	$6 * \Delta Q$	0.51	$0.5 * \Delta Q$	0.76	$0.5 * \Delta Q$	–	
PHE	0.67	$0.5 * \Delta Q$	0.58	$0.5 * \Delta Q$	–		–	
SER	0.54	$0.5 * \Delta Q$	–		–		–	
THR	0.70	$0.5 * \Delta Q$	–		–		–	
TRP	0.88	$0.5 * \Delta Q$	0.72	$0.5 * \Delta Q$	–		–	
TYR	0.59	$1.5 * \Delta Q$	0.63	$0.5 * \Delta Q$	–		–	
VAL	0.71	$0.5 * \Delta Q$	–		–		–	

Table 7.5: *ROC areas and normalisation factor for all torsion angles, $\Delta Q = Q_3 - Q_1$. Dashes indicate that the corresponding torsion angle does not exist within the residue type.*

and therefore the maximum difference is used for the normalisation. Because this value is extremely huge in comparison to the average energy difference, most of the values are nearly zero resulting in a random prediction.

Based on the normalisation factor the optimal threshold is calculated. Thus, the corresponding classification results are taken and the distance of each point within the ROC curve to the upper left corner is determined (see Eq. 7.4). Because each value in the ROC plots is linked to a certain threshold, the optimal threshold can be estimated easily. Table 7.6 shows the thresholds for all amino acids and torsion angles. Using these thresholds, all residues in the test set are classified. The result is stored in the relational database for serving this flexibility information to the docking system ELMAR.

Figure 7.8 on page 78 shows the three-dimensional structures of the trypsin `1AQ7`. Here, the residues are coloured according to their flexibility prediction based upon the threshold method. The green coloured atoms are correctly predicted whereas the red coloured residues are wrongly predicted. Each sub-figure shows the coloured protein structure for each torsion angle.

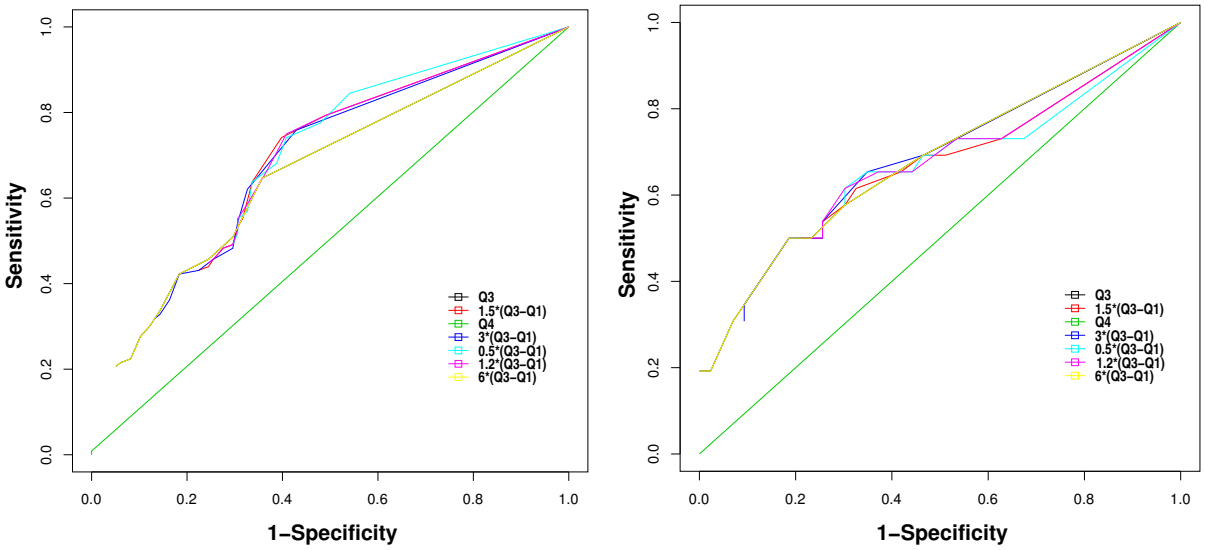
Comparing the four figures, it is obvious that the number of coloured residues drops for the higher torsion angles. The reason for this is that only a few residues have long side chains and possess a χ_3 or χ_4 torsion angle. In case of the first torsion angle (Fig. 7.8(a)) a lot of misclassifications occurred (many red coloured atoms) whereas for the higher torsion angles

amino acid	threshold			
	χ_1	χ_2	χ_3	χ_4
ARG	0.2	0.2	0.2	0.2
ASN	0.3	0.3	-	-
ASP	0.3	0.3	-	-
CYS	0.8	-	-	-
GLN	0.4	0.4	0.4	-
GLU	0.3	0.3	0.3	-
HIS	0.2	0.2	-	-
ILE	0.2	0.2	-	-
LEU	0.2	0.2	-	-
LYS	0.2	0.2	0.2	0.2
MET	0.1	0.1	0.1	-
PHE	0.3	0.3	-	-
SER	0.3	-	-	-
THR	0.2	-	-	-
TRP	0.2	0.2	-	-
TYR	0.1	0.1	-	-
VAL	0.2	-	-	-

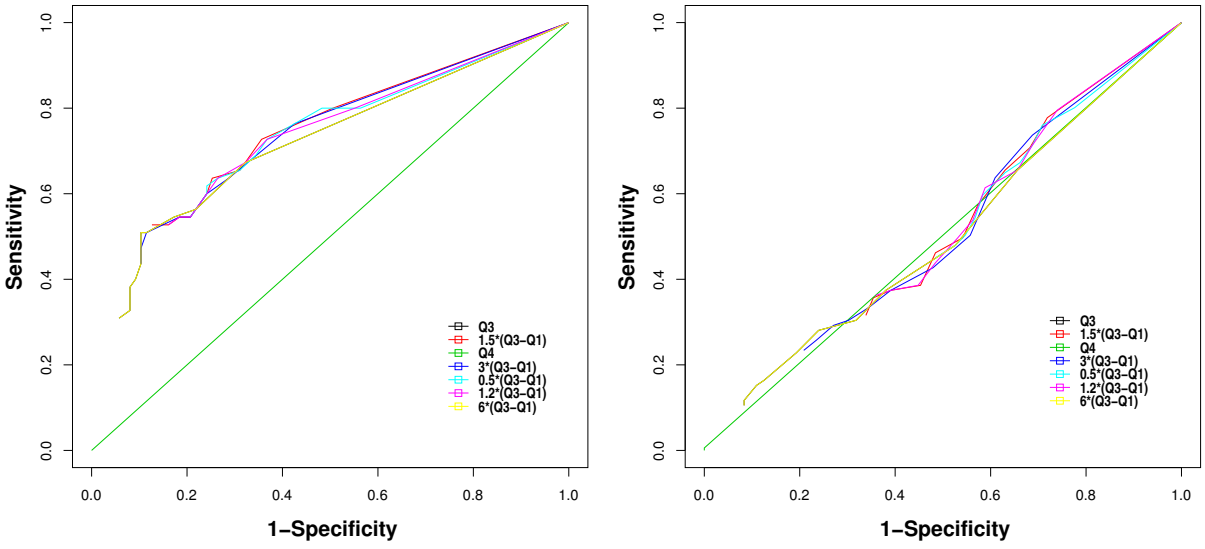
Table 7.6: *Thresholds used for the classification of residue flexibility using the energy difference as feature.*

the amount of wrongly classified residues is nearly equal to the number of correct classified residues (cf. Fig. 7.8(c) or 7.8(d)). According to the visual inspection, best results are reached for the χ_2 torsion angle (see Fig. 7.8(b)).

A further evaluation of the threshold based prediction of amino acid side chain flexibility is carried out by incorporating these results into the docking system ElMaR (see section 4.2). The results of the docking evaluation are given in section 7.3.

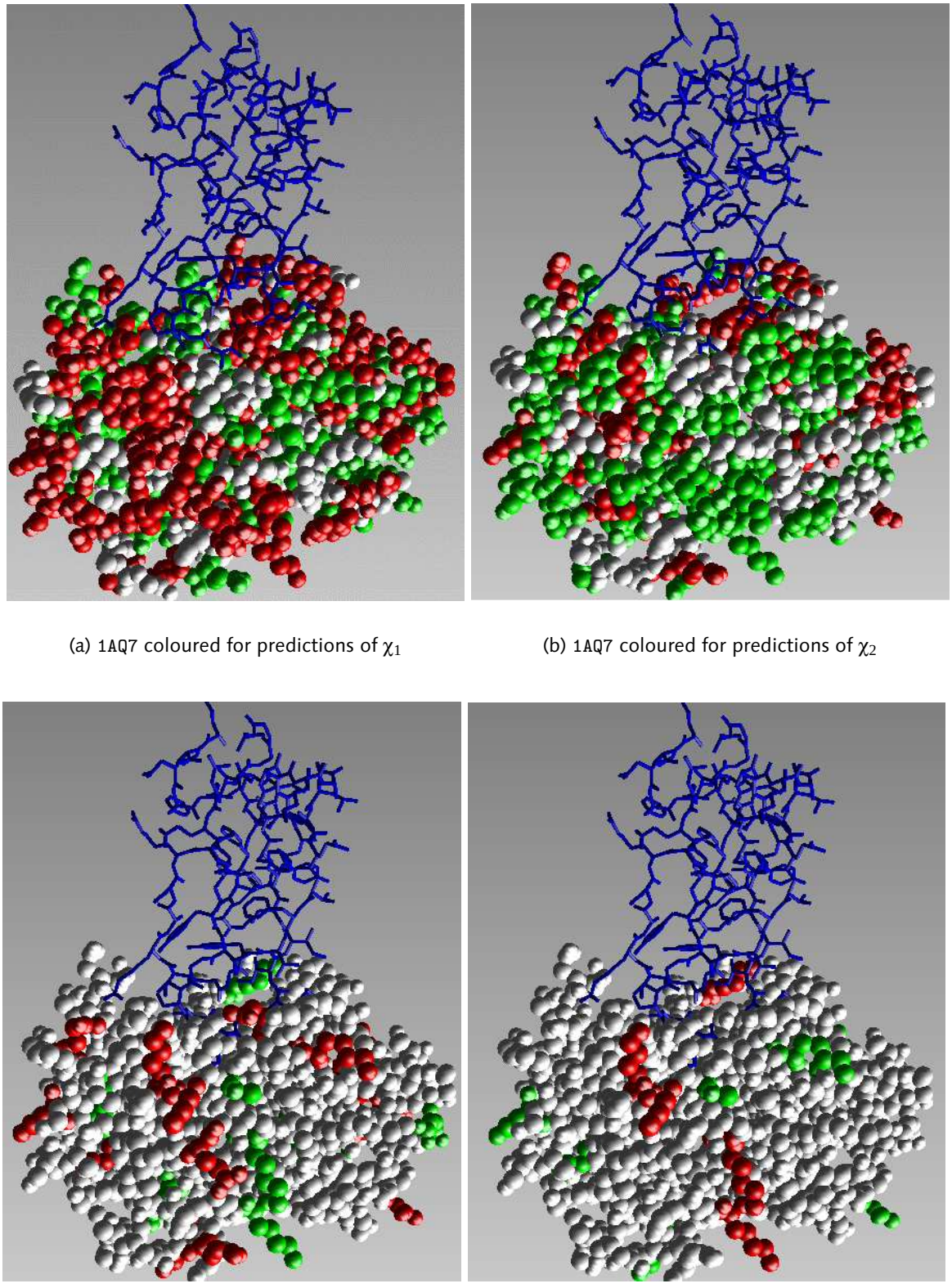


(a) ARG,χ_1 (b) TRP,χ_2



(c) MET,χ_3 (d) ARG,χ_4

Figure 7.7: *ROC curves for different residues and torsion angles. The different colours mark the normalisation factors used. On the x-axis 1-specificity, on the y-axis the sensitivity is plotted. The diagonal line marks the "chance line".*



(a) 1AQ7 coloured for predictions of χ_1

(b) 1AQ7 coloured for predictions of χ_2

(c) 1AQ7 coloured for predictions of χ_3

(d) 1AQ7 coloured for predictions of χ_4

Figure 7.8: *The protein structure of a trypsin (1AQ7) is coloured according to the threshold based flexibility prediction. Here, the structure is coloured in green in case of a true prediction and red in case of a false prediction. Gray coloured atoms are residues not considered like ALA, GLY or PRO and in case of the higher torsion angles (χ_{2-4}) those residues that do not possess these torsion angles. In blue, a docked trypsin inhibitor (1BPI) is shown.*

7.2.3 Classification Results using the Support Vector Machine

In this section the results of the classification of the flexibility of residues' side chains using a support vector machine are presented. Several features were calculated on the energy landscape of each torsion angle (see chapter 5) and combined to single feature vectors for the classification. Here, a support vector machine of the free statistics package R (Ihaka & Gentleman, 1996; Dimitriadou *et al.*, 2004) is used. The R package providing the support vector machine uses the *libsvm* (Chang & Lin, 2001) internally.

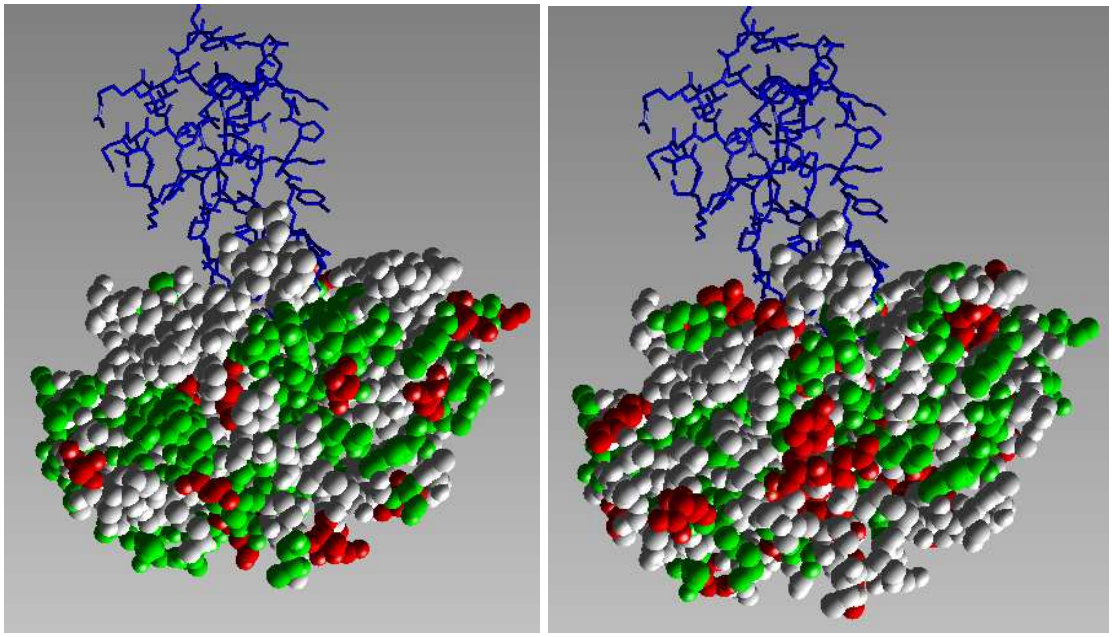
The classes of flexible and non–flexible residues are not balanced (the non–flexible residues outnumber the flexible ones) as shown in figure 7.5 on page 70. These proportions result from the folding of a protein burying most residues within the core and placing only fewer residues on the protein surface. The residues on the surface tend to be more flexible as steric restricts are less than within the core.

The SVM is biased to the larger class. During the generalisation, when training a SVM the classification is tuned towards reducing the number of misclassified examples of the larger class. Thus, the class containing the non flexible residues, neglecting classification faults is optimised. Therefore, equal sized classes of non–flexible and flexible residues have been chosen.

residue	χ_1	χ_2	χ_3	χ_4
ARG	77.1%	70.6%	64.5%	66.6%
ASN	73.3%	63.3%	-	-
ASP	70.8%	70.2%	-	-
CYS	89.2%	-	-	-
GLN	78.8%	74.9%	65.1%	-
GLU	68.4%	70.1%	63.1%	-
HIS	75.3%	69.7%	-	-
ILE	75.0%	69.1%	-	-
LEU	76.8%	63.4%	-	-
LYS	70.0%	75.6%	80.8%	69.2%
MET	82.8%	79.3%	75.3%	-
PHE	83.0%	66.4%	-	-
SER	68.4%	-	-	-
THR	64.2%	-	-	-
TRP	85.7%	75.7%	-	-
TYR	81.0%	62.1%	-	-
VAL	68.1%	-	-	-
average	75.8%	59.8%	69.8%	67.9%

Table 7.7: *Overall accuracy for classification of the different torsion angles as flexible or non–flexible. Here, a radial basis function kernel and a 10–fold cross evaluation are used.*

A 10–fold cross evaluation is performed to measure the classification accuracy and to avoid over-fitting. Besides the total accuracy normally printed out, the *libsvm* source code has been modified to print out the classification of the examples used within the testing of each validation step (see Tab. 7.8) to get a more detailed view of the classification process. The R package provides different kernel functions[4]. Here, radial basis functions are chosen as they perform well in most applications.



(a) 1AQ7 coloured for predictions of χ_1 (b) 1AQ7 coloured for predictions of χ_2

Figure 7.9: *Visualisation of SVM based flexibility predictions for χ_1 and χ_2. Correct predictions are coloured in green, false predictions are shown in red and gray coloured atoms represent residues not considered in the flexibility predictions like ALA, GLY and PRO.*

Table 7.7 summarises the classification results. Here, for each amino acid type and torsion angle the total accuracy is shown. In table 7.8 detailed results for the flexibility classification of χ_1 of arginine are presented. The numbers of true positives, false positives, false negatives and true negatives are the average over the ten validation runs. The results of the other amino acids and torsion angles are given in appendix B.5.

In summary, the flexibility of a residue for the χ_1 torsion angle can be predicted with a classification accuracy of nearly 75%. For the second torsion angle (χ_2) the accuracy of classifying the side chain flexibility is 60% whereas for the other torsion angles average classification rates of 68% and 67% are reached.

Inspecting the results given in table 7.8 one can see that flexible and non–flexible residues can be successfully predicted at same percentages. Only in few cases (e.g. for PHE, see

[4]Besides, radial basis functions, also polynomial, sigmoidal and linear functions are provided.

		Predicted	
		not flexible	flexible
True	not flexible	78.9% (15)	21.1% (4)
	flexible	23.3% (7)	76.7% (23)

Table 7.8: *Classification of the flexibility of χ_1 for ARG. Here, the average classification rate for all examples of the test sets randomly chosen within the 10–fold cross evaluation, is shown. A total accuracy of 77.1% is reached (see Tab. 7.8).*

Tab. B.5(l)) one class can be predicted better than the other one. These observations are also valid for the higher torsion angles.

Similar to the evaluation of the threshold based predictions the classification results are visualised by colouring a protein's structure according to true or false predictions. In figure 7.9 the protein 1AQ7 is used. The sub-figures show the classification results for the χ_1 and $\chi 2$ torsion angle. In figure 7.9(a) the averaged good results classifying the flexibility for the first torsion angles are visible. Most residues have been classified correctly indicated by the green colour whereas only few amino acids are assigned a wrong class. Inspecting the figure 7.9(b), the increase of false predictions is obvious. This observation corresponds to the results given in table 7.7 for the second torsion angle.

7.3 Docking Results using Flexibility Information

In order to test the flexibility approach in a docking scenario, in this section the flexibility information calculated is provided to the docking system ElMaR (see section 4.2) . First, the experimental setup is described, then the methods for evaluating and comparing the docking results are outlined (see section 7.3.2). Finally, the results for incorporating flexibility information into the docking system are shown.

7.3.1 Docking Experiments

In order to investigate how much impact the new flexibility information has onto the resulting docking hypotheses two docking experiments were set up. In the first experiment a docking without flexibility information is performed using the generated test sets (see section 7.1.2). The resulting docking hypotheses are stored in a relational database to compare them later to the results of the second experiment. In the second experiment flexibility information is provided to the ElMaR system. In order to compare the performance of the two flexibility approaches the information gained is presented independently to the system. Then, the docking is carried out on the same data as used in the first experiment. The docking hypotheses are stored in the database for further evaluation. For testing the influence of

the scaling factor, two values are chosen: $\omega = 0.5$ and $\omega = 1.0$. Each of the outlined docking experiments are run with both scaling factors.

7.3.2 Evaluating and Comparing Docking Hypotheses

For testing docking algorithms, the hypotheses predicted by ELMAR are compared to a known complex[5] with identical sequence. A well known measure to compare two structures is the so called *root mean square deviation* (RMSD):

$$RMSD = \frac{1}{N}\sum_{i}^{N}\sqrt{(a_i - b_i)^2} \tag{7.5}$$

The RMSD value gives the euclidian distance (in Å) between two structures a and b. A small value indicates a good similarity whereas a large value shows significant differences between them. In equation 7.5, N is the number of C_α atoms. a_i and b_i denote the atoms of the two structures to be compared. In order to calculate the RMSD the two structures are superimposed.

Besides the RMSD, the ranking of docking hypotheses is of interest, too. Halperin and coworkers (Halperin *et al.*, 2002) defined a set of different measures including rank and RMSD, called *DRUF* (Docking Results Unified Format). Here, the N10, N50 and N100 measures are chosen because these scores reflect the quality of predictions of a complex from two unbound proteins. The measures are defined as follows:

N10: Number of hypotheses within the first 10 ranks with an RMSD $\leq$ 3Å.

N50: Number of hypotheses within the first 50 ranks with an RMSD $\leq$ 4Å.

N100: Number of hypotheses within the first 100 ranks with an RMSD $\leq$ 5Å.

By the definition of these measures, changes within the result set a test case processed in the different experiments is very simple. Since all results are stored in the database, the calculation of the N_X scores can be realised through SQL queries (see Fig. 7.10).

```
select Entry, count(*) from Hypothesis
       where rmsd<[3|4|5] and rank<[10|50|100] group by Entry;
```

Figure 7.10: *SQL query for the N10, N50, N100 measure of the DRUF Protocol.*

Other measures defined within DRUF are:

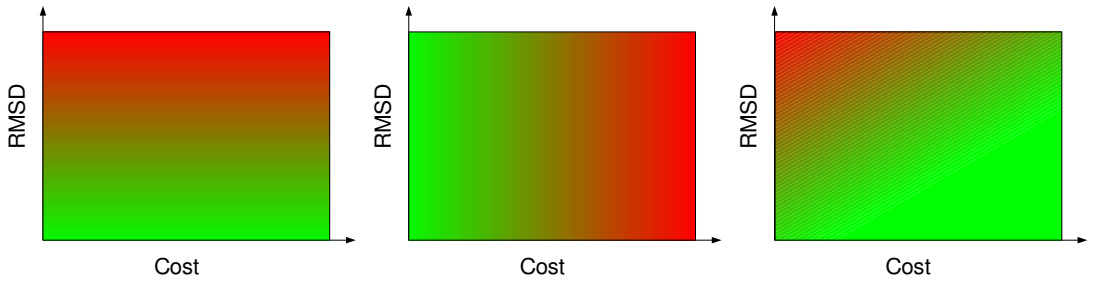
[5] Known complex means a crystallographically refined complex structure, deposited in the PDB.

- RMSD of the hypothesis at Rank #1
- Rank of the first solution with RMSD <5Å
- Rank of best RMSD hypothesis

A disadvantage of these scores is that the flexibility of a side chain is not taken into account. The different side chain conformations of multiple hypotheses cannot be compared by the C_α RMSD. Furthermore, the DRUF protocol only focuses on the top ranked hypotheses. A measure that summarises the performance of a docking run over the whole set of hypotheses would be desirable. Neumann (Neumann, 2003) proposed the *IPI* (Integrated Performance Indicator). The IPI summarises the performance by a weighted sum of the scores of all hypotheses:

$$IPI = \sum_i \underbrace{\frac{\text{Rank}_{max} - \text{Rank}_i}{\text{Rank}_{max}}}_{\text{Rank weighting}} \cdot \underbrace{\frac{\max(10\text{Å} - \text{RMSD}_i, 0)}{10\text{Å}}}_{\text{RMSD weighting}} + \begin{cases} p_a & if \quad \frac{\text{RMSD}_i}{\text{score}_i} > \frac{\text{RMSD}_{max}}{\text{score}_{max}} \\ p_b & else \end{cases} \tag{7.6}$$

The score of a hypothesis i is the product of the normalised rank and the weighted RMSD. Additionally, an error term is added whether the hypothesis has an RMSD above or below the diagonal. The IPI is the sum of all hypotheses of the test case. Figure 7.11 visualises the components of the IPI measure.



(a) RMSD between 0-10 Å normalised to $[0..1]$ (RMSD weighting).

(b) Rank normalised to the interval $[0..1]$ (Rank weighting).

(c) Exclusion of false positives by the error term (last term of equation 7.6).

Figure 7.11: *The components of the integrated performance indicator. Hypotheses that fall into the green area contribute to a good score. Courtesy of Neumann (Neumann, 2003).*

The IPI can give an overall score of the whole set of hypotheses. For a detailed analysis within parts of the result sets other methods have to be used. Besides the IPI, the minimal RMSD within the first 10, 50 and 100 ranks is considered. Here, changes within the best hypotheses can also be observed for test cases that are hard to predict (e.g. `1A2W`).

Besides methods calculating scores that express the accuracy of the docking experiments, visualisation techniques can be used for qualitative analysis of docking experiments. Therefore, the rank or the costs are plotted against the RMSD. In this thesis several docking

experiments for one test case are conducted and have to be compared. Simply, the results of different experiments can be drawn into one plot. Since the number of hypotheses predicted by ELMAR is large (700 hypotheses per test case), these plots become rather complex. In order to avoid this a different method for comparing and visualising the differences between the docking runs is used. The whole plotting area is sampled into rectangles. Here, a rectangle of size of 10×1.5 is used. The size of the rectangle is abutted from the N10 measure. A width of "10 ranks" and a height of "1.5Å" yields good results. On the one hand, the rectangle is not too small, e.g. several hypotheses are covered. On the other hand, the rectangle is not too large, so that a fine sampling is possible, visualising changes in detail. Within each rectangle the number of hypotheses of each experiment placed here are counted. By calculating the difference ($\Delta C_{x,y}$) between these numbers, changes can be easily observed:

$$\Delta C_{x,y} = C^B_{x,y} - C^A_{x,y} \tag{7.7}$$

Here, $C^B_{x,y}$ denotes the number of hypotheses of an experiment using flexibility information within the rectangle at position x,y. $C^A_{x,y}$ represents the number of hypotheses for the same docking experiment and the same rectangle but without using flexibility information. A positive difference means that the number of docking hypotheses placed within the rectangle increased whereas a negative value shows a decrease, respectively.

Furthermore, the differences can be visualised by applying different colours for positive and negative differences and plotting the rectangles. The quantities of the changes can be expressed by the lightness of the colours. Light colours indicate few changes whereas dark shades represent many changes[6].

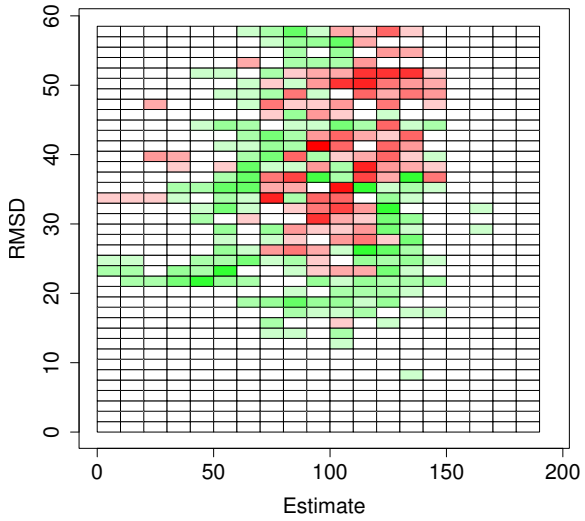


Figure 7.12: *Visualisation of changes between the docking of 1BEL and 1RAT.*

[6] In case of no changes due to equal numbers of hypotheses or if no hypotheses are placed within that regions of the plot, the rectangles are coloured white.

Figure 7.12 visualises the application of this method to the results of the docking of 1BEL and 1RAT. Here, the docking with flexibility information is compared to a docking without flexibility. The differently coloured rectangles show the changes within the set of hypotheses. Green coloured parts show an increase, red coloured boxes a decrease in the number newly placed hypotheses.

7.3.3 Results for the Docking Experiments

In the last section the experiments carried out for testing the flexibility predictions within the docking and the methods for evaluating the docking results have been outlined. In the following the results of the experiments are shown. In all experiments carried out unbound 3D structures of proteins from the PDB are used. In order to evaluate the results of the docking runs test cases have been compiled as described in section 7.1.1. Table A.3 in the appendix shows a detailed list of unbound proteins and complex pairs.

During docking, the ELMAR system for could not process some test cases. Usually the calculation and prediction of hypotheses takes about 20 minutes for a single test case (cf. Neumann, 2003). Here, for some test cases the calculations takes several days which was surprising. Due to this the docking run was aborted, since the calculation did not seem to come to an end. This error is non deterministic, since it has been observed occurring on all levels of the docking system (cf. Fig. 4.2) and with different test cases. Even test cases that e.g. have been successfully calculated for the χ_1 flexibility could not be calculated for the overall flexibility and vice versa. Thus, the number of test cases within the different docking experiments differ.

The evaluation of docking results is complex, since different evaluation methods focuses differently on the results. The analysis of the minimal RMSD, i.e. only returns the RMSD of the best predictions. The amount of good hypotheses that has been predicted is not reflected. Therefore, the N10, N50, and N100 scores of the DRUF protocol can be used. Furthermore, the IPI measure calculates a score including all hypotheses of a docked test case. Thus, an analysis of the results using these measures is reasonable.

The flexibility of a residue's side chain is predicted for each torsion angle separately. It is also combined via a weighted sum (see section 5.2.5) to a score of the whole side chain. At first the influence of the docking is tested using only the flexibility classification for the first torsion angle χ_1. It has been chosen because the first torsion angle is the most restricted one and because all residues have at least this torsion angle. So, the influence of the flexibility is distributed over the whole protein and does not only effect few residues in case a higher torsion angle is chosen (e.g. only ARG and LYS residues consist of a large side chain covering the χ_4 torsion angle). In a second step the overall flexibility score is tested within the docking. In the following, at first the results for the docking incorporating the threshold based flexibility predictions are presented. Then, the results of the experiments using the SVM based flexibility predictions are shown.

Test of the Threshold based Flexibility Predictions

In this work the flexibility of a residue's side chain is predicted by two approaches. Here, the results of incorporating the flexibility information of the threshold based approach are evaluated.

Threshold based Flexibility of χ_1

At first, the results for the χ_1 torsion angle are outlined. The evaluation is performed on 81 test cases matching 14 different reference complexes. The number of test cases per complex is given in table 7.9.

Complex	test cases
1A2W	3
1ADI	1
1AFK	3
1AFL	3
1AFU	5
1AO6	1
1APN	8
1ARG	2
1ASM	3
1ASN	2
1B2K	41
1CGI	2
1TPA	2
2PTC	3

Table 7.9: *Number of test cases per reference complex used for docking experiments. These proteins are taken for evaluating the threshold based flexibility predictions of χ_1.*

The figures 7.13, 7.14 and 7.15 show the distribution of the minimal RMSD reached within the top 100, 50 and 10 ranks, respectively. Each box-plot represents the hypotheses yielding the minimal RMSD for the given reference complex. For half of the complexes an improvement can be observed if flexibility information is applied. Further five examples show improvements in the minimal RMSD for at least one of the two docking experiments carried out. Only in two cases no improvements can be reached. For `1AFK` no better RMSD than for the docking without flexibility can be achieved whereas for `1ADI` the predicted hypotheses within the different docking experiments have the same RMSD. For some single test cases the flexibility predictions fail and no good hypotheses are predicted (e.g. `1APN`, $\omega = 1.0$ or `1ASM`, $\omega = 0.5$) but on average better results are achieved.

The length of the box (also called inter-quantile distance) represents the deviation within the data represented by the box-plot. Since each box-plot summarises several test cases

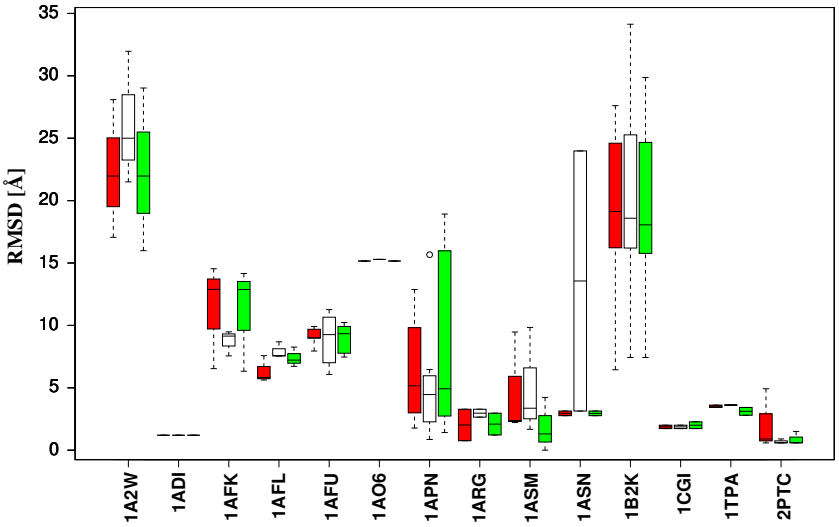


Figure 7.13: *Box-plots of all test cases grouped by the reference complex. Here, the flexibility information is estimated by the threshold based classifier for χ_1. Each box-plot represents the minimal RMSD within the top 100 ranked hypotheses (Rank$\leq$100). The white boxes represent the docking results without flexibility, the red for docking results with flexibility ($\omega = 0.5$) and the green for $\omega = 1.0$.*

for the same reference complex, the deviation gives a hint for the behaviour of the scoring function. Large deviations assume that for similar test cases[7] no similar scores are assigned. These test cases cannot be predicted easily by ELMAR. Instead of this, in case of a small deviation the conclusion can be drawn that the scoring function assigns similar scores for similar test cases. Here, the test cases can be predicted at a higher accuracy.

Comparing the three figures, for some examples the deviation of the minimal RMSD is small (e.g. 1ADI, 1AFL, 1CGI, 1TPA, and 2PTC). For these test cases the deviation within all three docking experiments is similar and good predictions are reached. In case of 1ADI, 1CGI, 1TPA, and 2PTC in 90% of the test cases the best hypothesis has an RMSD less than 5Å and is ranked within the top ten ranks (see Fig. 7.15). In case of 1ASN the deviation within the results of the docking without flexibility is high. But for the experiments using flexibility it is very small. Here, the flexibility information yields very good results for all test cases. In case of 1ASM on average for all test cases the best hypothesis is placed within the top ten ranks and has an RMSD of 5Å or less. But in case of the docking experiments using flexibility information and a scaling factor of $\omega = 0.5$ the deviation of the hypotheses with minimal RMSD is high within the first 10 and 50 ranks (cf. Fig. 7.15, 7.14). Here, the assumption can be drawn that for few test cases the scoring function failed. A hypothesis with a large RMSD value has been assigned a high rank wrongly. This observation can be supported comparing all three figures. In case of the top 100 ranked hypotheses with minimal RMSD the dis-

[7] The similarity of the test cases due to fact that each test case is defined via the sequence identity between the unbound proteins and its corresponding complex parts (see section 7.1.1). Since several unbound proteins match the same complex part, these test cases are similar.

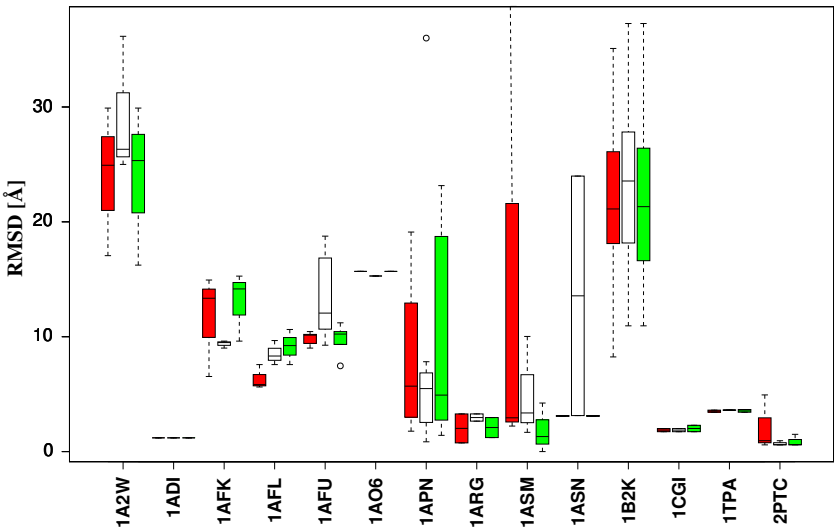


Figure 7.14: *Box-plots of all test cases grouped by the reference complex. Here, the same information is shown as in figure 7.13 but for all hypotheses with Rank* ≤ 50.

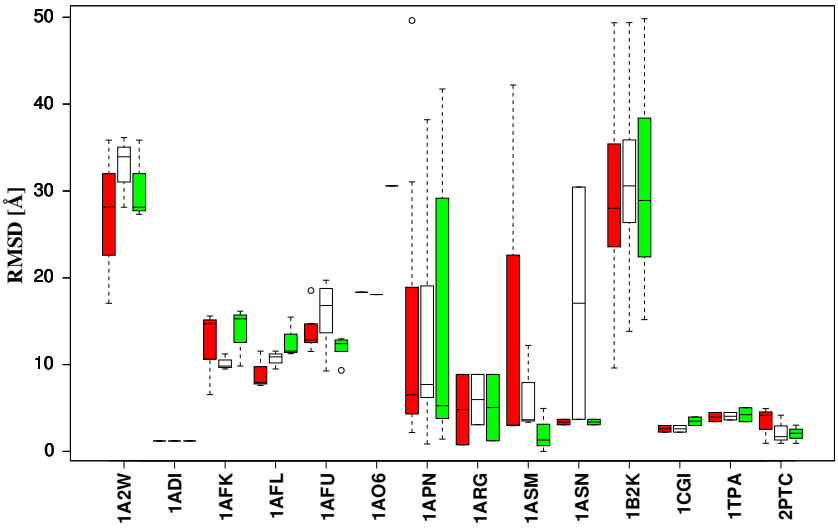


Figure 7.15: *Box-plots of all test cases grouped by the reference complex. Here, the same information is shown as in figure 7.13 but for all hypotheses with Rank* ≤ 10.

tribution of the RMSD is smaller than for the other groupings. Therefore, the hypotheses counted within the top 100 ranks differs from the ones included in the figures 7.14 and 7.15 for the "outlier test cases".

Besides the minimal RMSD also the IPI measure is applied to the data. The IPI measures the overall performance including all hypotheses predicted for a test case. Large values indicate a good performance whereas small values show a low performance in predicting correct docking solutions (see section 7.3.2).

The evaluation of the docking results by the IPI measure (see Fig. 7.16) confirms the observation made before. Inspecting figure 7.16 one can see that for example the overall performance for the test cases of `1A2W` is worse (IPI value near zero) for all three docking experiments. But for the flexible docking using a $\omega = 0.5$ a slightly better performance is measured (due to the better ranked hypotheses). For other test cases the docking perfor-

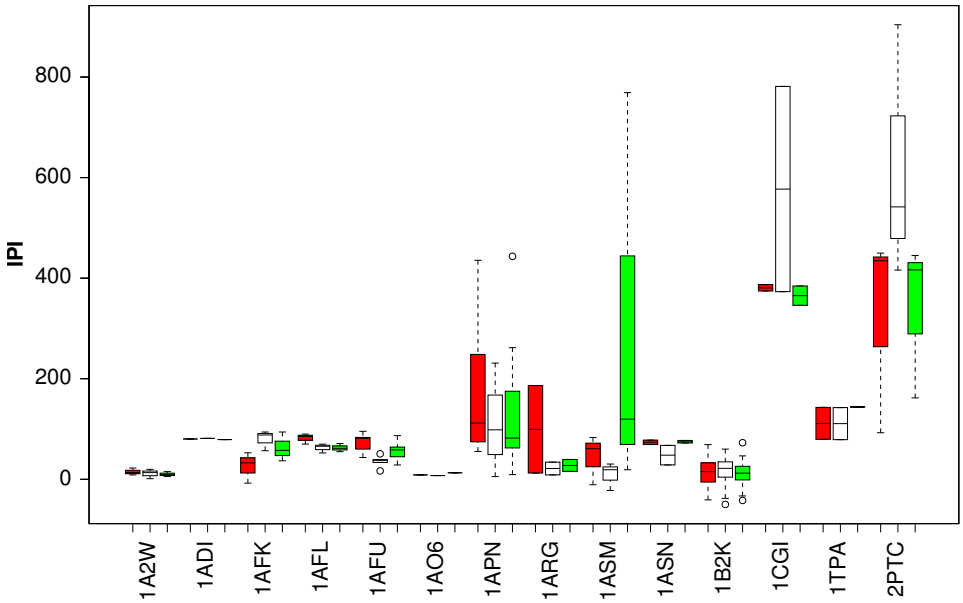


Figure 7.16: *IPI evaluation of the docking using the threshold based flexibility classification for χ_1. In green and red the results of the docking with flexibility and in white the results using no flexibility are depicted.*

mance is more obvious. Exemplarily, in case of 2PTC the minimal RMSD distribution of the test cases is better for the docking without flexibility. This is reflected by the IPI evaluation, too. But also the improvements are measured (cf. Fig. 7.16). In case of 1ASM and 1ASN the docking with flexibility outperforms the results reached using no flexibility.

In table 7.10 the evaluation of the docking results by the N10/50/100 measure of the DRUF protocol is shown. Since for some test cases (e.g. 1A2W) no near native hypotheses have been predicted at all, these are not listed within the table. For all other docking examples the initial numbers and changes calculated according to the three DRUF measures for each docking experiment are given. Dashes denote those test cases where no changes in the number of hypotheses between the experiments occurred.

Generally, for 47% of the test cases no changes can be estimated using the N10 measure. The number of improvements is equal to the number of test cases showing no better results. For the N50 in nearly half of all examples improvements can be observed whereas 35% of the test cases do not show any improvements. Similar observations can be made for the N100 evaluation. Inspecting table 7.10 in detail, for some docking examples the flexibility predictions fail and no improvements can be reached. Most of these changes due to false scored hypotheses place within the top ranks. Good hypotheses are then moved to higher ranks. Exemplarily, for the test case 2PTC(1AUJ/1BPI) 1 hypothesis is lost within the N10, 5 within the N50 and 4 within the N100. Looking at the set of hypotheses in the first case (N10), the hypothesis is shifted from rank 10 to rank 12. Similar changes are observed for the other hypotheses. An explanation of these changes is given in the discussion of the results (cf. section 7.5). In three of four test cases similar to the reference complex 1APN and

Test Case	N10		N50		N100	
	$\omega = 0.5$	$\omega = 1.0$	$\omega = 0.5$	$\omega = 1.0$	$\omega = 0.5$	$\omega = 1.0$
1ADI(1QF4/1QF5)	–	8**-1**	–	–	29**+1**	–
1APN(1C57/1DQ1)	–	–	9**-9**	9**-9**	17**-17**	17**-17**
1APN(1C57/1DQ5)	–	–	–	–	2**-2**	2**-2**
1APN(1CON/1DQ1)	–	0**+5**	3**-3**	3 **+7**	8**-8**	8**+4**
1APN(1CON/1QNY)	6**-6**	6 **-6**	31**-31**	31**-31**	39**-35**	39**-35**
1ARG(1ARS/1ASA)	0**+8**	0**+10**	18**+16**	18**+2**	22**+45**	22**+7**
1ARG(1CQ7/1CQ8)	–	–	–	–	–	10**-1**
1ASM(1AMQ/1CQ8)	0**+6**	0**+10**	8**+8**	8**+23**	13**+12**	13**+38**
1ASM(1ASA/1ASE)	–	0**+10**	1**+27**	1**+49**	2**+40**	2**+98**
1ASN(1ASE/1CQ8)	–	–	–	–	–	–
1CGI(1CHG/1HPT)	1**+1**	1 **-1**	13**+1**	13 **-1**	38**+2**	38**-17**
1CGI(1GCD/1HPT)	–	–	19**+2**	19 **+2**	–	–
1TPA(1AUJ/1BPI)	–	–	2**+1**	2**-1**	8**+1**	8 **+11**
1TPA(1BJU/1BPI)	–	–	4**+2**	4 **+5**	13**+2**	13**+5**
2PTC(1AQ7/1BPI)	–	–	20**-2**	20**-1**	54**-2**	54 **-21**
2PTC(1AUJ/1BPI)	3**-3**	3**-1**	30**-30**	30**-5**	67**-65**	67**-4**
2PTC(1BJU/1BPI)	–	8 **-2**	–	38**-2**	70**-1**	70**-1**

Table 7.10: *Evaluation of docking results by DRUF protocol. The initial numbers for N10, N50 and N100 are compared to the results of the docking using flexibility information. The changes are given in bold numbers, dashes denote no changes. Here, the threshold based flexibility information for χ_1 is incorporated.*

for the test case 2PTC(1AUJ/1BPI) nearly no hypotheses are placed within the top 100 ranks. These changes are not influenced by the docking but due to calculation errors generating the hypotheses (see section 7.5). But for five examples the flexibility information can improve the results very much. An increase up to 98 hypotheses (in case of 1ASM(1ASA/1ASE)) for the N100 is reached. For the test case 1ASM(1AMQ/1CQ8) within the top 10 ranks the number of hypotheses with an RMSD less than 3Å rises from 0 to 6 ($\omega = 0.5$) and 0 to 10 for $\omega = 1.0$, respectively.

For the other test cases the results differ. On the one hand improvements within the N50 and N100 ranges are reached but no improvements for N10 (e.g. 1TPA(1AUJ/1BPI)) are yielded. On the other hand there are improvements e.g. for the docking experiment using a scaling factor of $\omega = 0.5$ but a decrease of hypotheses for the other flexibility docking experiment. Exemplarily, the test case 1CGI(1CHG/1HPT) yields improvements for N100 ($\omega = 0.5$) but a loss of 17 hypotheses for a scaling factor of $\omega = 1.0$.

Summarising the results the threshold based flexibility prediction improves the docking results. Although, for some test cases no improvements are reached (e.g. 1AFK) or top ranked hypotheses are lost (e.g. 2PTC) for most residues better results are yielded incorporating the flexibility information. Besides placing in nearly half of test cases the best hypotheses ($\leq 5\text{Å}$) within the top 10 ranks, also the overall performance of the docking is improved as proved

by the IPI evaluation. The prediction of test cases that are hard to predict for ElMaR (e.g. 1A2W) is improved, too.

Threshold based Flexibility Scores for the whole Side Chain

Since the flexibility is derived independently, for each torsion angle in a second docking run the combination of the single flexibilities is tested. In table 7.11 the test cases used for this experiment are given. Similar to the previous experiment, here in figure 7.17, 7.18 and 7.19

Complex	test cases
1A2W	3
1AFK	2
1AFU	3
1APN	2
1B2K	17
1CGI	3
1LYS	9
1TPA	2
2PTC	2

Table 7.11: *Number of test cases per reference complex used for docking runs incorporating the threshold based overall flexibility prediction.*

the minimal RMSD reached are visualised by box-plots. For this experiment improvements

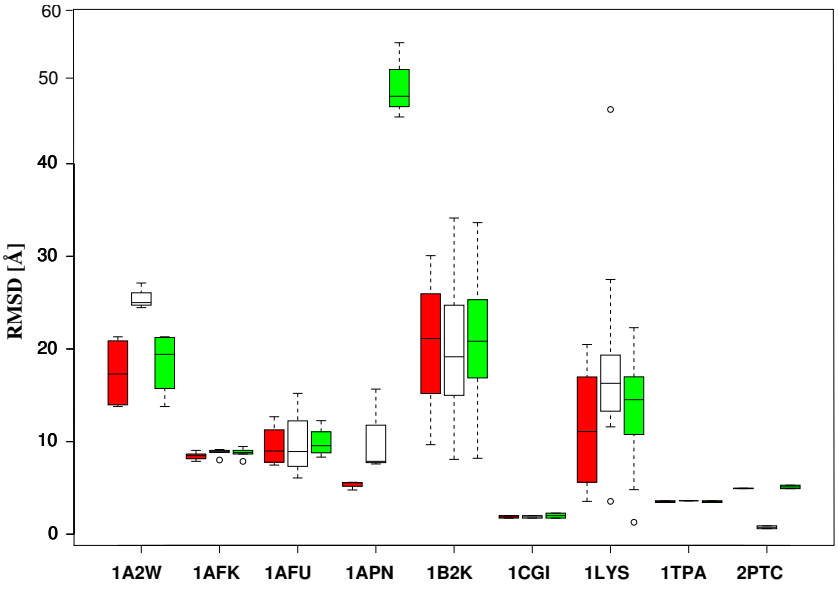


Figure 7.17: *Box-plots of all test cases grouped by the reference complex. Here, the flexibility information is calculated from all torsion angles. The single flexibilities are estimated by the threshold based classifier. Each box-plot (white: without flexibility, red: flexibility ($\omega = 0.5$), green: flexibility ($\omega = 1.0$)) represents the minimal RMSD within the top 100 ranked hypotheses (Rank$\leq$100).*

can be observed, too. In case of 1LYS, 1TPA or 1A2W for each rank criterion better results are achieved using flexibility information. But the number of test cases with no improvements or

even worser results increase for the top ranked hypotheses (Rank $\leq$ 10). In case of 2PTC for example no good results can be achieved at all if the combined flexibility is used. Extreme differences between the results of 1APN can be observed. Here, for a flexibility scaling factor of $\omega = 0.5$ very good results are reached. The predicted hypotheses yield a RMSD below 10Å. In contrast to this result the minimal RMSD reached using an $\omega = 1.0$ is between 50 and 60Å.

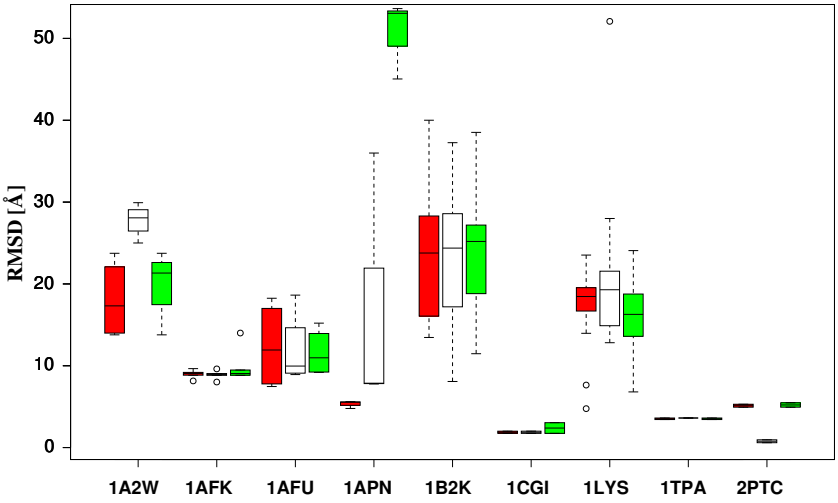


Figure 7.18: *Box-plots of all test cases grouped by the reference complex.Here, the same information is shown as in figure 7.17 but for all hypotheses with Rank* ≤ 50.

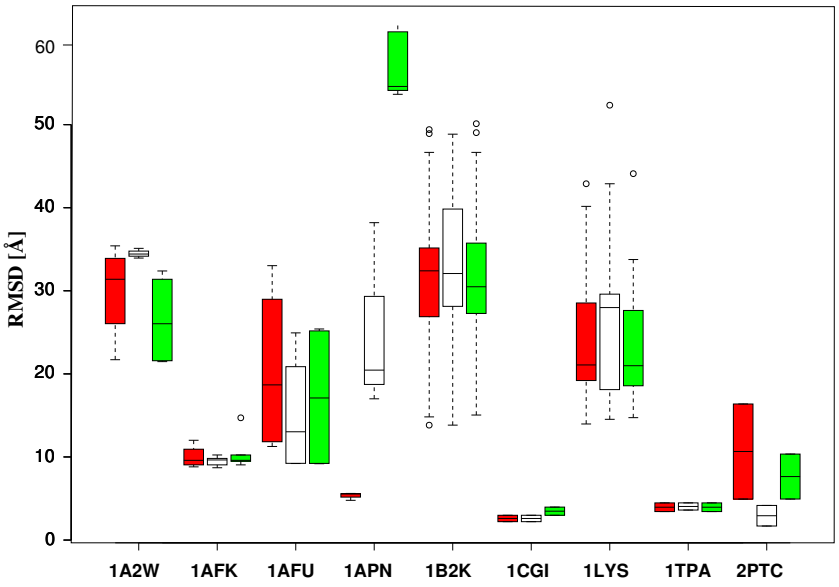


Figure 7.19: *Box-plots of all test cases grouped by the reference complex. Here, the same information is shown as in figure 7.17 but for all hypotheses with Rank* ≤ 10.

For 1LYS and 1B2K the top ten ranked hypotheses have high RMSD ranks. The best hypotheses for these test cases have a rank between 50 and 100. Although the best hypotheses have been assigned to high ranks, the flexibility information on average can improve the results.

As for the results using only the flexibility information of the first torsion angle, the deviation of the RMSD within the test cases for a complex varies. Comparing the three figures, an

Test Case	N10		N50		N100	
	$\omega = 0.5$	$\omega = 1.0$	$\omega = 0.5$	$\omega = 1.0$	$\omega = 0.5$	$\omega = 1.0$
1CGI(1CHG/1HPT)	–	1 **-1**	–	13**-7**	–	38**-22**
1CGI(1GCD/1HPT)	–	–	–	–	–	–
1LYS(1AKI/1LZ9)	–	–	–	–	–	2 **-2**
1TPA(1AUJ/1BPI)	–	–	2**+1**	2**+2**	8**+1**	8**+2**
1TPA(1BJU/1BPI)	–	–	4**+2**	4**+5**	13**+2**	13**+5**
2PTC(1AQ7/1BPI)	–	–	20**-20**	20**-20**	54**-53**	54**-54**
2PTC(1AUJ/1BPI)	3**-3**	3**-3**	30**-30**	30**-30**	67**-65**	67 **-65**

Table 7.12: *Evaluation of docking results by DRUF protocol. The initial numbers for N10, N50 and N100 are compared to the results of the docking using flexibility information. The changes are given in bold numbers, dashes denote no changes. Here, the threshold based flexibility information for the whole side chain is incorporated.*

increase of the deviation in RMSD can be observed for the top ten ranked hypotheses (see Fig. 7.19) underlying that only few good hypotheses have been assigned to such ranks.

This observation is supported by the evaluation of this docking using the N10, N50 and N100 measure (see Tab. 7.12). In only two of nine test cases having hypotheses ranked within the top 100 improvements can be observed (1TPA(1AUJ/1BPI) and 1TPA(1AQ7/1BPI)). For all other test cases similar results to the docking without flexibility are reached focusing on hypotheses within the top 100 ranks and a RMSD below 5Å. Since the results of 2PTC are similar to the ones for the docking using the flexibility predictions of χ_1, the conclusion can be drawn that the ElMaR system failed, too.

The IPI plot (see Fig. 7.20) shows that overall good results are reached for 2PTC and 1CGI compared to the other complexes but the flexibility predictions cannot improve the results. Although, for 2PTC the hypotheses are not comparable due to the same reasons as for the χ_1 predictions, within the IPI plot these results score overall well. The test case 1APN showed great differences between the minimal RMSD values using the different scaling factors. Inspecting the IPI scores, the conclusion can be drawn that in case of $\omega = 1.0$ no good hypotheses can be ranked on low ranks but on average a good performance is reached. Generally, the previous observations are reflected by this plot. Good results using the flexibility information are not only reached for the top ranked hypotheses but also for the whole set (e.g. 1LYS or 1AFU).

In summary the threshold based flexibility prediction of the whole side chain can improve the docking predictions of the ElMaR system. Especially for the test cases of the reference complexes 1ASN, 1ASM, 1ARG, 1TPA, and 1CGI similar or even better results compared to the docking without flexibility is reached. For 1A2W and 1B2K no good hypotheses can be predicted at all but using the flexibility information on average an improvement of 5 to 10Å is yielded.

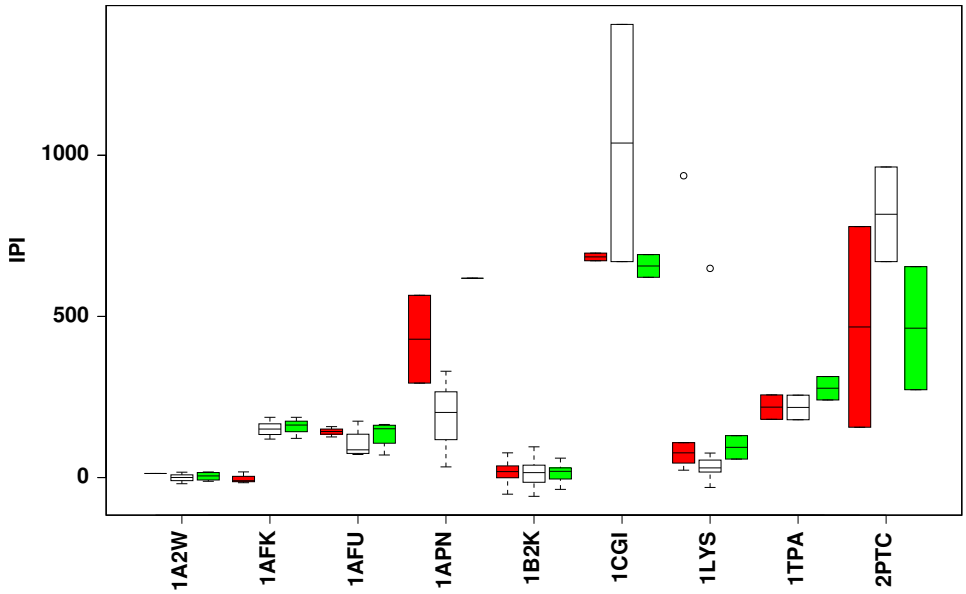


Figure 7.20: *IPI evaluation of flexible docking. Here, the overall flexibility information calculated by the threshold based prediction for the side chains is used. The differently coloured box-plots (white: no flexibility, red:* $\omega = 0.5$*, green:*$\omega = 1.0$*) represent the IPI scores of the test cases grouped by their reference complex.*

Test of the SVM based Flexibility Predictions

In this section, the results of the incorporation of flexibility information predicted by the support vector machine (SVM) into ELMAR are shown. At first the results for the flexibility prediction of the first torsion angle (χ_1) are presented.

Support Vector Machine based Flexibility Predictions for χ_1

As for the threshold based predictions the test cases are grouped by their reference structure. In table 7.13 the number of test cases for each reference complex used for evaluating the flexibility is shown.

In figure 7.21, 7.22 and 7.23 the minimal RMSD of all test cases belonging to the same complex are plotted as box-plots. For each docking run separate box-plots are drawn. Like in the previous section, the white coloured boxes depict the results without incorporating flexibility. The green and red coloured box-plots represent the minimal RMSD of the hypotheses with flexibility information provided to the docking algorithm.

For most docking test cases improvements are reached if flexibility information is used (see `1A2W`, `1AFK`, `1ASN`, `1TPA`, and `2PTC`). Hypotheses with low RMSD values are ranked within the

Complex	test cases
1A2W	1
1A7X	3
1AFK	1
1AFU	3
1APN	1
1ASN	1
1B2K	51
1CGI	3
1LYS	30
1TPA	1
2PTC	3

Table 7.13: *Number of test cases per reference complex taken for evaluating the SVM based flexibility prediction of* χ_1.

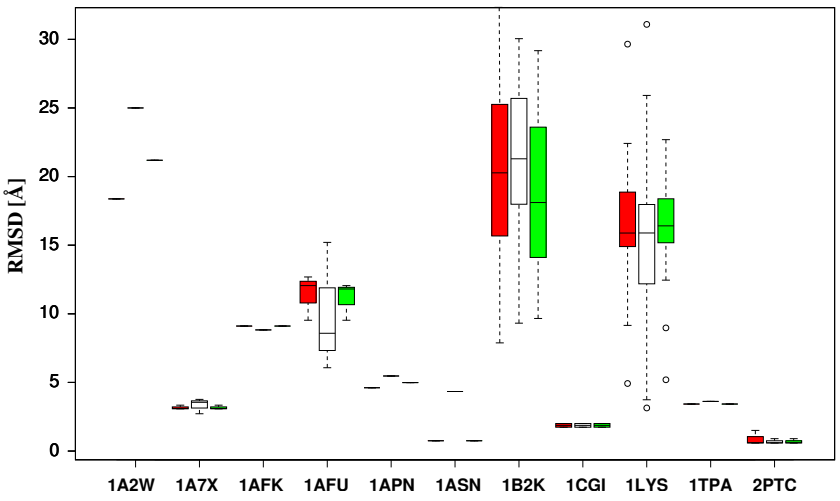


Figure 7.21: *Box-plots of all test cases grouped by the reference complex. Each box-plot represents the minimal RMSD within the top 100 ranked hypotheses (Rank*$\leq$*100). The white boxes represent the docking results without flexibility, the red stand for docking results with flexibility and scaling of* $\omega = 0.5$ *and the green for an* $\omega = 1.0$.

top 100. Only in case of `1A7X` and `1AFU` the flexibility information does not lead to improvements. For the other test cases no predictions of near native hypotheses[8] are reached.

Inspecting figures 7.21, 7.22 and 7.23 one can see that for most of the 11 groups[9] improvements are achieved. In four of these even good hypotheses (RMSD $\leq$ 5Å) lie within the top 10 ranked solutions (see `1ASN`, `1CGI`, `1TPA` and `2PTC`). In case of the test case `1ASN(1ARS/1ART)` the best solution is assigned to a rank within the top 10 solutions. The RMSD for this test case is 0.75Å (for both scaling factors) whereas the RMSD of the best solution of the docking run without flexibility is 4.34Å.

[8] Hypotheses with low RMSD score are called near native, thus they are very similar to the solved structure of the complex.

[9] Here, the test cases belonging to one reference complex are referred as a group.

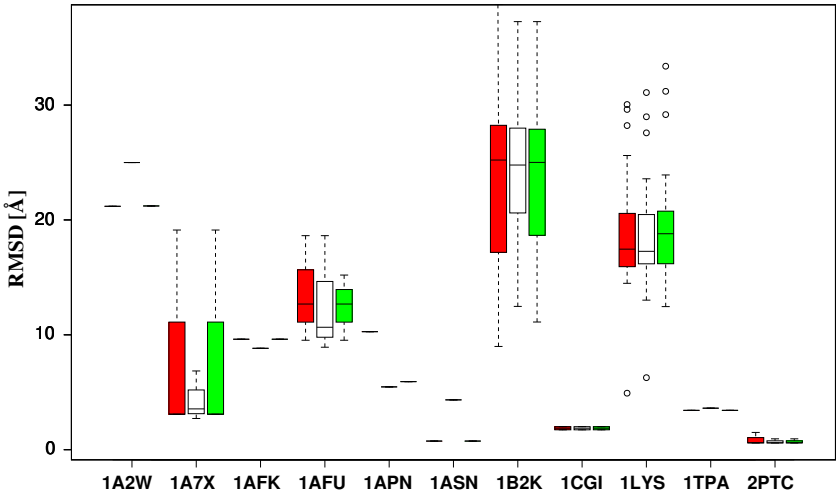


Figure 7.22: *Box-plots of all test cases grouped by the reference complex. Here, the same information is shown as in figure 7.21 but for all hypotheses with Rank* ≤ 50.

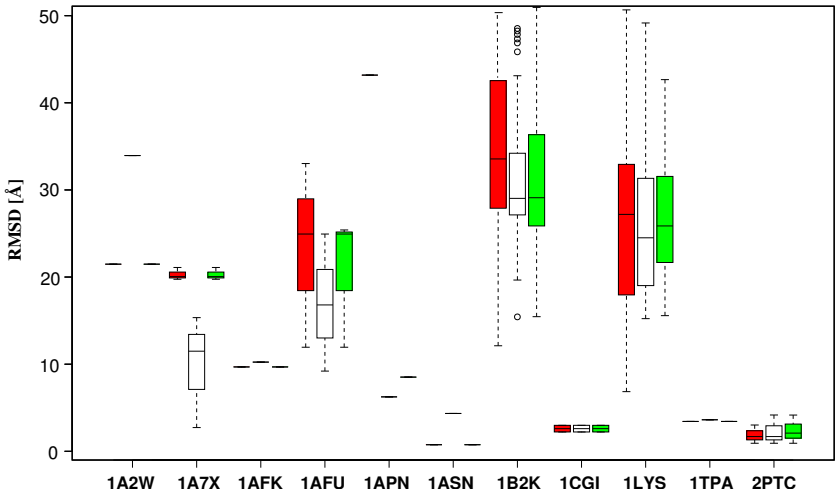


Figure 7.23: *Box-plots of all test cases grouped by the reference complex. Here, the same information is shown as in figure 7.21 but for all hypotheses with Rank* ≤ 10.

Comparing the minimal structural error of the top 50 scored hypotheses with the minimal RMSD of the hypotheses within the top 10 ranks one can see that for some cases (`1B2K`, `1A7X`) improvements are achieved using flexibility information although the solutions are not scored that high. Especially, for the examples of the complex `1A7X` within the docking experiments using flexibility information no good hypotheses are placed into the top 10 ranks but within the top 50. Only in 2 of 11 cases (`1AFU`, `1APN`) no improvements could be achieved, compared to the docking without flexibility.

Besides the minimal RMSD, the docking results have been evaluated by the IPI measure, too. In figure 7.24 a box-plot of the test cases for each reference complex is drawn, showing the IPI scores.

The results of the IPI evaluation correlate with the observations described above. Exemplarily, for 1A2W no good hypotheses were predicted. The IPI value is low. But improvements are obtained for the test cases using the flexibility information (cf. Fig. 7.23, see lower minimal RMSD values), too. These improvements are also reflected by the higher IPI scores

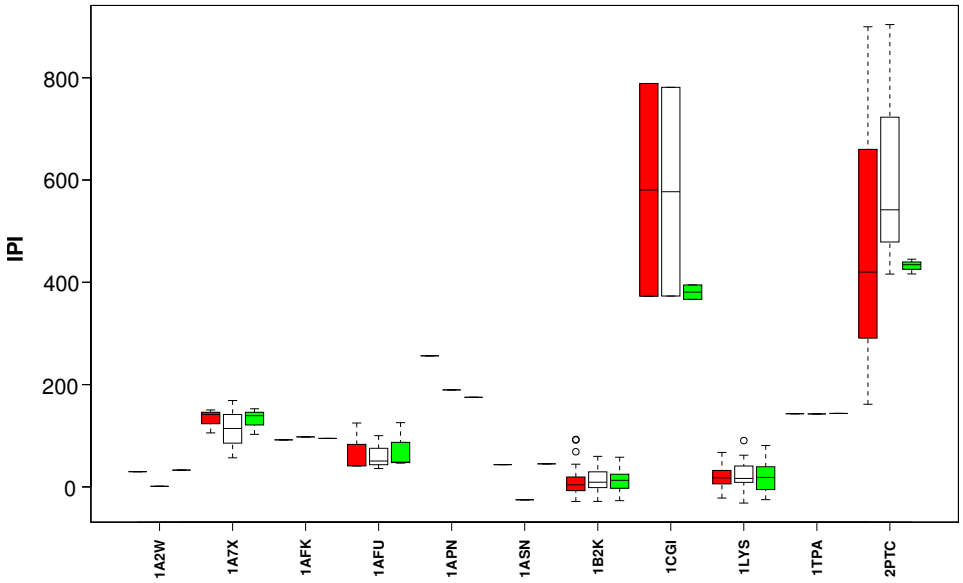


Figure 7.24: *IPI evaluation of docking experiments. In red and green the results of the SVM based flexibility prediction used within the docking are given whereas the white coloured box-plots represent a docking run without flexibility.*

of the docking results using flexibility (see red and green box-plots of 1A2W in Fig. 7.24). Test cases that can be predicted very good received high scores (e.g. 2PTC and 1CGI). Comparing these results to the minimal RMSD evaluation, the observation could be made that hypotheses with a lower or equal RMSD value compared to a docking without flexibility are yielded but in general the flexibility predictions do not reach that high IPI scores. A detailed analysis of the top ranked hypotheses is given by the DRUF evaluation in the following.

In table 7.14, the docking experiments were evaluated by the N10, N50 and N100 measure. For most test cases no changes within the number of the top 10, 50 or 100 ranks can be observed. Only in two cases (1A7X(1FKB/1FKF) and 2PTC(1AQ7/1BPI)), a large number of good hypotheses that have been ranked within the top 100 by ElMaR without flexibility information is lost when using the flexibility prediction and a scaling factor of $\omega = 0.5$. In five other cases only few hypotheses are shifted to higher ranks but remain within the top 100 (e.g. see 1Ax7(1FKF/1FKJ)). Very good results are reached for 1ASN(1ARS/1ART). Using the SVM based flexibility predictions 8 hypotheses with an RMSD of 3Å or less are ranked within the top 10, further 3 (4) hypotheses are placed within the top 50 and for $\omega = 0.5$ another 6 hypotheses have been assigned to ranks within the top 100. Improvements are also yielded for three other test cases (the remaining two test cases for the complex 1A7X and 1TPA(1BJU/1BPI)) with an increase of up to 14 good hypotheses.

Two test cases, 1A2W(1BEL/1RAT) and 1TPA(1BJU/1BPI) have been chosen for a more detailed analysis of the changes within the resulting set of hypotheses. These two test cases are interesting because for 1A2W no near native hypotheses could be predicted but obvious changes have been observed comparing the box-plots of the minimal RMSD scores. In case

Test Case	N10		N50		N100	
	$\omega = 0.5$	$\omega = 1.0$	$\omega = 0.5$	$\omega = 1.0$	$\omega = 0.5$	$\omega = 1.0$
1A7X(1FKB/1FKF)	1**-1**	–	9**-6**	–	28**-9**	–
1A7X(1FKB/1FKJ)	–	–	0**+4**	0**+4**	2**+14**	–
1A7X(1FKF/1FKJ)	–	–	1**-1**	–	2**+6**	–
1ASN(1ARS/1ART)	0**+8**	0**+8**	0**+11**	0**+12**	1**+18**	–
1CGI(1CHG/1HPT)	–	–	–	–	–	–
1CGI(1GCD/1HPT)	–	–	–	–	–	–
1LYS(193L/1LSC)	–	–	–	–	3**-3**	–
1LYS(1AKI/1HSX)	–	–	–	–	1**-1**	–
1TPA(1BJU/1BPI)	–	–	4**+2**	–	13**+2**	–
2PTC(1AQ7/1BPI)	–	–	20**-1**	–	54**-20**	–
2PTC(1AUJ/1BPI)	–	–	30**-2**	–	67**-1**	–
2PTC(1BJU/1BPI)	–	–	–	–	70**-1**	–

Table 7.14: *Evaluation of docking results by DRUF protocol. The initial numbers for N10, N50 and N100 are compared to the results of the docking using flexibility information. The changes are given in bold numbers, dashes denote no changes. Here, the SVM based flexibility information for χ_1 is incorporated.*

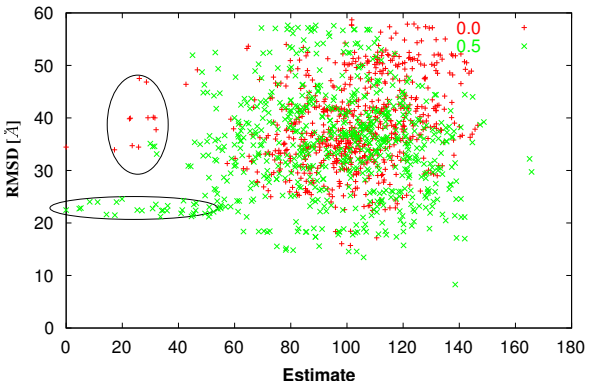
of 1TPA very good results have been predicted but on average the docking results using flexibility information are only slightly better then without flexibility comparing the minimal RMSD scores.

In figure 7.25 and B.10 the results of docking the unbound proteins 1BEL and 1RAT are shown. As reference for calculating the RMSD, the complex 1A2W is taken. The unbound proteins are sequence identical to the chains of the protein. The results of the docking runs incorporating flexibility are compared to the results of the docking run without flexibility information.

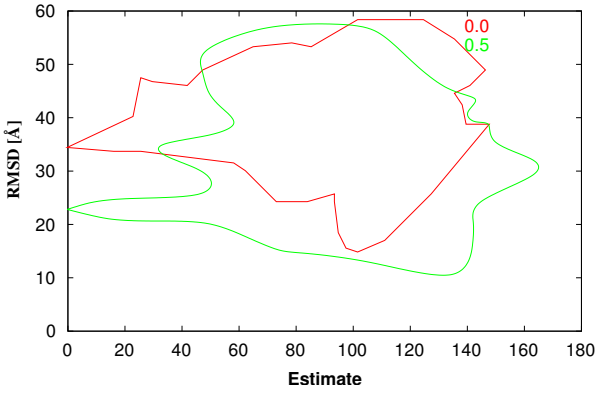
In both figures, three different views onto the information are given. The top left figures (7.25(a) and B.10(a)) show the super-imposition of the results. Each point in the plots represents one hypothesis. The red coloured points depict hypotheses predicted without flexibility whereas the green coloured dots represent hypotheses predicted using flexibility information. Here, the estimated costs[10] are plotted against the RMSD. Beside this plot on the right, the outer hulls calculated from the sets of hypotheses are drawn. Below the two plots the differences between the docking results compared within rectangular areas (cf. section 7.3.2) are visualised. Green coloured rectangles depict an increase in the number of hypotheses located within this region whereas rectangles in red denote a decrease in the number of hypotheses.

For the test case 1A2W(1BEL/1RAT) the best scored hypothesis has a RMSD of 35Å (22Å using flexibility). The hypothesis with the minimal RMSD, a score of 100 (140 using flexi-

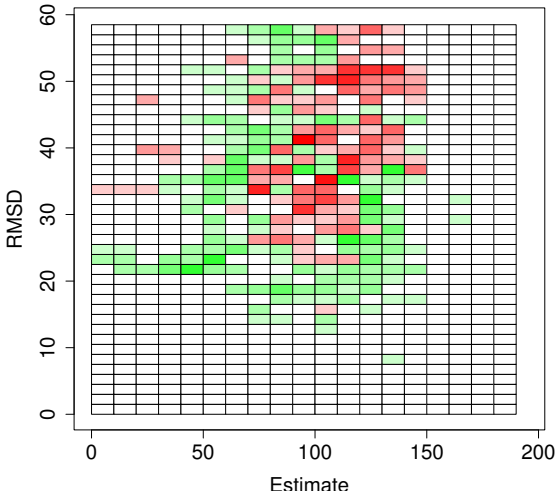
[10] ELMAR assigns high scores for good hypotheses and low scores for bad hypotheses. Here, $\max(costs) - costs$ are plotted so that a low RMSD correlates with low costs, respectively.



(a) Comparison plot of the docking results. Here, the flexibility is scaled by $\omega = 0.5$.



(b) Plot of the outer hulls (red: without flexibility, green: flexibility, $\omega = 0.5$).



(c) Changes between docking with and without flexibility (green: increase in number of hypotheses, red: decrease in number of hypotheses, white: no or equal changes).

Figure 7.25: *Visualisation of the docking results of 1BEL/1RAT. In this experiment only the flexibility information of the first torsion angle χ_1 is used. In figure 7.25(a) each point in the plot represents one docking hypothesis (red: without flexibility, green: with flexibility). Here, the estimated costs are plotted against the RMSD. The outer hulls fitted around the points representing the hypotheses are shown in figure 7.25(b). Below these two figures, the changes within a rectangular area of size 10×1.5 are plotted.*

bility) has been assigned. Although the results of this docking experiment are not optimal with respect to the RMSD, some aspects are worth to be mentioned. In both cases that

incorporated flexibility the best ranked hypothesis has a lower RMSD as the one from the experiment without flexibility. Obviously, false positives (hypotheses located in the upper left area of the plot) are moved towards lower ranks (here higher costs, respectively) whereas hypotheses with a RMSD of about 20Å were scored better (see Fig. 7.25(c)). Inspecting figure 7.25(b) one can see that the whole set of hypotheses is moved towards the x-axis and is slightly rotated. Comparing figure 7.25(c) and 7.25(b) the intersections of the convex hulls correspond to the coloured areas in the rectangle plot. For instance, the intersections of the hulls that lie outside the green coloured polygon are similar to the red shaded areas whereas the intersecting regions that lie within this polygon correspond to the blue shaded parts in figure 7.25(c). The same observations can be made inspecting figure B.10 given in the appendix.

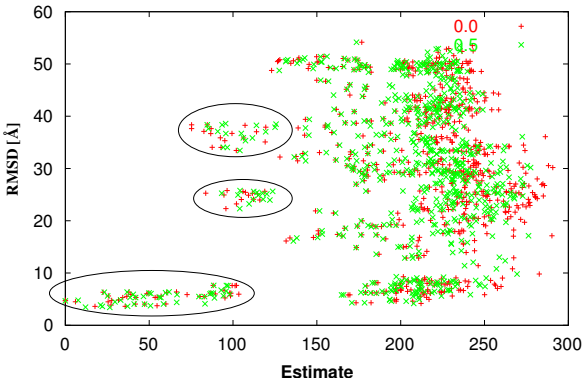
In the second example, the docking of `1BJU` and `1BPI` is analysed in more detail. In all experiments a lot of hypotheses with an RMSD equal or less than 5Å are ranked top. Since, the changes using a flexibility scaling factor of $\omega = 0.5$ do not differ much from those using a $\omega = 1.0$ (see Fig. 7.26 and B.11), in the following the results using $\omega = 0.5$ are outlined (the corresponding figure for $\omega = 1.0$ is given in App. B.6). Similar to the previous example, here, the super-imposition of the docking hypotheses, the outer hulls, and the differences plot are given, too.

From the standard evaluation methods (IPI, DRUF, minimal RMSD) applied before one would expect no significantly changes between the docking experiments using flexible and the reference experiment. In fact, within the top scored hypotheses only small changes can be observed (i.e. see the light colours in Fig. 7.26(c)). But the important differences occur for the false positive hypotheses. In figure 7.26 ane even better in the plot of the outer hull (see Fig. 7.26(b)) hypotheses are shifted towards higher ranks. In nearly all cases these hypotheses are assigned ranks around 100 or above. Thus, in this case a final results list is free of hypotheses with high RMSD scores.

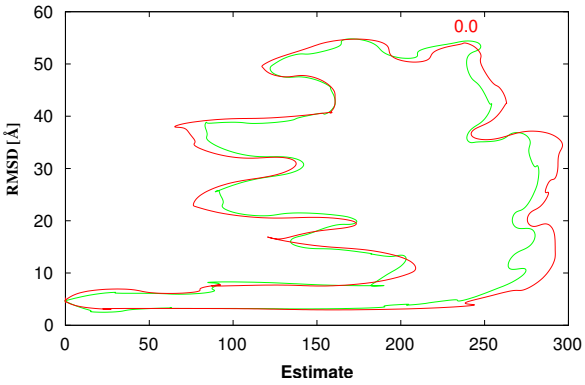
Another interesting aspect is that the result set forms cluster. There exists three different clusters of hypotheses. Their hypotheses are scored by costs of 100 or below. Two cluster hold the false positive hypotheses whereas the other one covers the best predictions obtained by the ElMaR. The rest of hypotheses form a stripe covering hypotheses with RMSD between 5 and 50Å and costs between 150 and 300. Since the results of the two docking experiments are very similar, the conclusion can be drawn that the hypotheses within the clusters are very similar and thus are scored similar.

Comparing the docking without flexibility and the docking with flexibility and a scaling factor of 1.0 of this test case, the same observation can be made. In appendix B.6 the corresponding plots are given. The only difference is that the false positives within the two clusters are shifted more towards higher ranks. Several hypotheses within the stripe are assigned similar scores as reflected by the changes plot (cf. Fig. B.11). Thus, the conclusion can be drawn that the scoring improved since more similar hypotheses are scored equally.

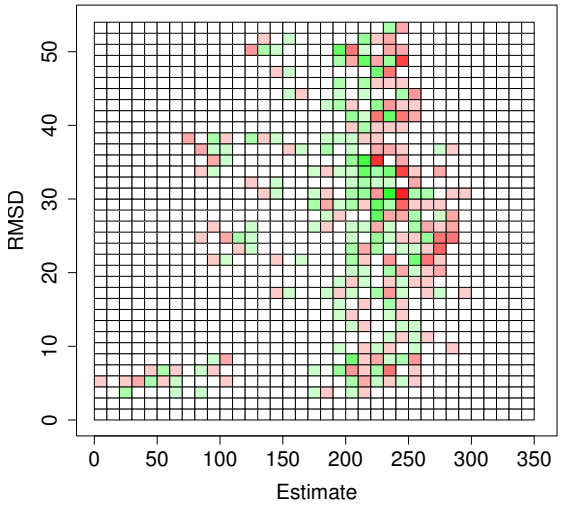
In summary, the SVM based flexibility for the first torsion angles improves the results of the docking. Compared to the docking of the threshold based flexibility fewer errors are propagated and the scoring function is only misled in few cases. The detailed analysis for



(a) Comparison plot of the docking results. Here, the flexibility is scaled by $\omega = 0.5$.



(b) Plot of the outer hulls (red: without flexibility, green: flexibility, $\omega = 0.5$).



(c) Plot of the changes between the two experiments (green: increase in number of hypotheses, red: decrease in number of hypotheses, white: no or equal changes).

Figure 7.26: *Visualisation of the docking results of 1BJU/1BPI. Each point in the plots represents one docking hypothesis. Here, the estimated costs are plotted against the RMSD. In red the results without incorporating flexibility are shown. The green coloured points are hypotheses from docking with flexibility information. In this experiment only the flexibility information of the first torsion angle χ_1 is used.*

1A2W and 1TPA proves that the flexibility information can improve the scoring of ElMaR. False positive hypotheses are assigned to higher ranks or costs, respectively.

Flexibility Scores for the whole Side Chain using the SVM

As for the threshold based predictions, here also docking runs have been scheduled using an overall flexibility prediction of the whole side chain. In the following the results of these experiments are outlined.

Complex	test cases
1A2W	4
1A7X	4
1ADE	1
1ADI	1
1AFK	8
1AFL	6
1AFU	8
1AO6	1
1APN	7
1ARG	7
1ASM	2
1ASN	1
1B2K	24
1BM0	1
1CGI	2
1LYS	9
1TPA	2
2PTC	2

Table 7.15: *Number of test cases per reference complex used for the evaluation of the overall flexibility score of the side chain predicted by the SVM.*

Inspecting the box-plots of the minimal RMSD values reached, for nearly half of the groups (1A2W, 1A7X, 1AFK, 1AFL, 1ARG, 1ASM, 1ASN, 1B2K, and 1TPA) improvements and for most other examples equal results are reached. Only for the test cases of 1LYS no better ranked hypotheses are yielded at all. Comparing the two different scaling factors large variations between the results can be observed (see 1ASN in figure 7.27 or 1BM0 in figure 7.29).

Furthermore, the deviation of the RMSD values of the hypotheses within the first 10, 50 and 100 ranks is larger for the flexibility docking than compared to the docking only using the flexibility predictions of the first torsion angle (see Fig. 7.21, 7.22 and 7.23). Thus, the conclusion can be drawn that the additional flexibility information has a different impact on the scoring of ElMaR then only using the flexibility predictions of χ_1. Here, the scoring function ranks the similar test cases differently. Exemplarily, for 1A2W a small variance is reached using a scaling factor of $\omega = 1.0$ whereas for the other flexibility experiment a larger variance can be observed. In case of 1A7X the difference in variance is just switched.

This observation can be supported by evaluating the results using the N10, N50 and N100 measure of the DRUF protocol. In table 7.16 on page 105 the detailed numbers of the test

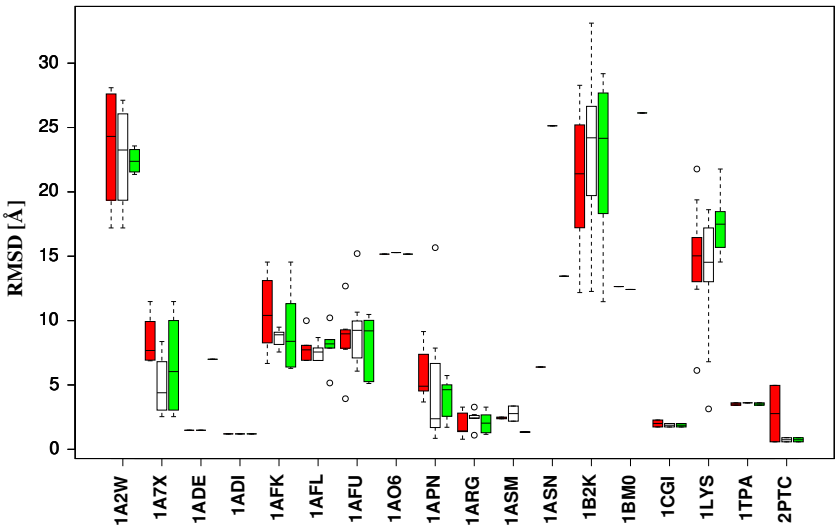


Figure 7.27: *Box-plots of all test cases grouped by the reference complex. Each box-plot represents the minimal RMSD within the top 100 ranked hypotheses (Rank≤100). The white boxes represent the docking results without flexibility, the red boxes stand for docking results with flexibility and scaling of* $\omega = 0.5$*, the green for an* $\omega = 1.0$*. Here, the overall flexibility information gained through the SVM based predictions is used.*

cases having hypotheses assigned within the top 100 ranks are listed. Like in the evaluation described before here, the initial numbers (of the docking without flexibility) and the changes are given (in bold). The dashes denote no changes between docking with and without flexibility. The complex situation described above is reflected within the results of this evaluation method, too. Only for two test cases (`1APN(1CON/1QNY)` and `1APN(DQ1/1QNY)`) no improvements have been reached at all. For those test cases no hypotheses have been ranked within the top 100 ranks using flexibility information.

For most other hypotheses different results between the two scaling factors used with the flexibility are obtained. Exemplarily, in case of `1ADE(1CIB/1QF5)` for $\omega = 0.5$ equal numbers of hypotheses are reached for N10, N50 and N100 whereas for the other scaling factor ($\omega = 1.0$) no hypotheses are ranked within the top 10, 50 or 100, respectively. Another contrary example is the test case `1APN(1CON/1DQ0)`. Here, for the N50 on the one hand a loss of 6 hypotheses is counted whereas on the other hand, for $\omega = 1.0$, the same number of new hypotheses is gained. These differences in the results due to the fact, that the classification accuracy for the higher torsion angles is lower (see section 7.2.3). Thus, more errors are propagated to the docking misleading the scoring function. Additionally, the scaling factors used have an different impact on the flexibility information (cf. section 7.5).

But improvements can be observed, too. In some cases there is an increase of up to 52 good hypotheses (see `1ARG(1ARS/1CQ7)`). Docking the proteins `1ASA` and `1ASE` yields improvements for all three scores. Within the N10 4 (2) new hypotheses are counted. Here, the docking using no flexibility assigned no hypotheses to the top 10 ranks.

Inspecting the IPI plot, one can observe that the differences between the two scaling factors is reflected by several groups (`1BM0`, `1APN`, or `1AFU`. Low prediction performances are

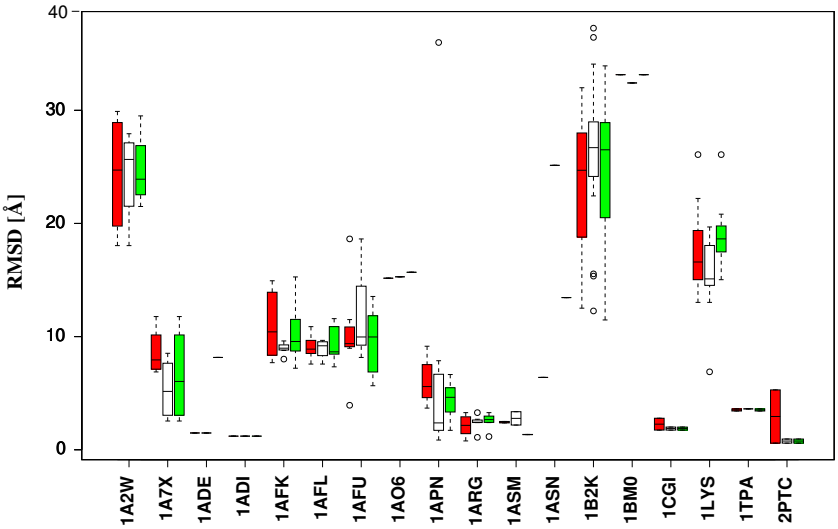


Figure 7.28: *Box-plots of all test cases grouped by the reference complex. Here, the same information is shown as in figure 7.27 but for all hypotheses with Rank* ≤ 50.

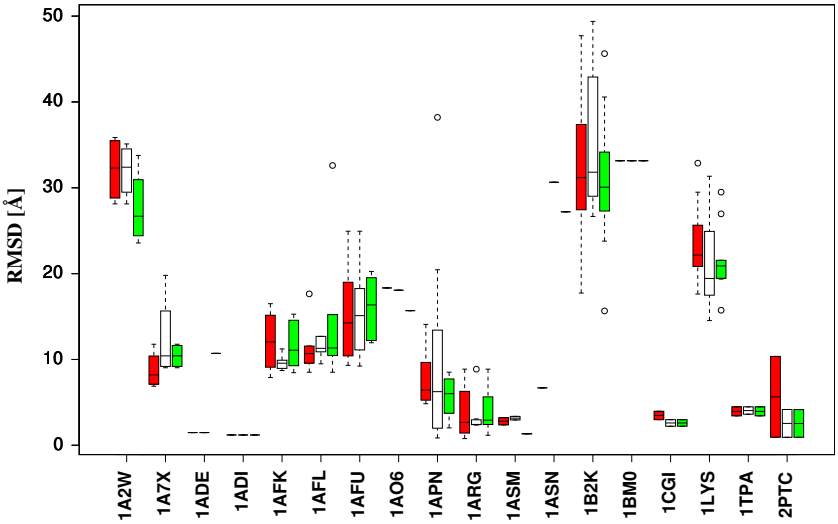


Figure 7.29: *Box-plots of all test cases grouped by the reference complex. Here, the same information is shown as in figure 7.27 but for all hypotheses with Rank* ≤ 10.

depicted by low IPI scores whereas good predictions are scored high. In three cases no improvements are reached (1ADI, 1CGI and 2PTC). Although the performance predicting good hypotheses is lower for 2PTC and 1CGI when applying flexibility information, in both cases these results outperform the other groups. For nearly half of the groups improvements are yielded using the flexibility. In theses cases the two scaling factors have an equal influence to the scoring function. For most other test cases for one of the two scaling factors improvements are reached.

The docking of 1ARG(1AMR/1ASE) shows different results for the two scaling factors (see Tab. 7.16). In figure 7.31 and 7.32 a detailed view on the docking results is given. Here, the same evaluation was performed like for the test cases docked using the SVM based flexibility predictions for χ_1 (cf. Fig. 7.16). In figure 7.31 one can see that the hypotheses are moved towards the lower left corner (see Fig. 7.31(a)). Some hypotheses have been placed on high ranks yielding even better RMSD scores than docking without flexibility

Test Case	N10		N50		N100	
	$\omega = 0.5$	$\omega = 1.0$	$\omega = 0.5$	$\omega = 1.0$	$\omega = 0.5$	$\omega = 1.0$
1A7X(1FKF/1FKJ)	–	–	1**-1**	–	2**-2**	–
1A7X(1FKG/1FKJ)	–	–	1**-1**	–	2**-2**	2**+1**
1ADE(1CIB/1QF5)	–	10**-10**	–	31**-31**	–	36**-36**
1ADI(1QF4/1QF5)	–	–	–	–	29**+1**	–
1APN(1C57/1CON)	–	0**+2**	4**-4**	4**+4**	10**-10**	10**-2**
1APN(1CON/1DQ0)	1**-1**	1**+3**	6**-6**	6**+6**	20**-20**	20**+3**
1APN(1CON/1QNY)	6**-6**	6**-6**	31**-31**	31**-31**	39**-39**	39**-39**
1APN(1DQ1/1QNY)	8**-8**	8**-8**	25**-25**	25**-25**	34**-32**	34**-34**
1ARG(1AMQ/1AMR)	6**-4**	6**+4**	33**-16**	33**-8**	56**-18**	56**-30**
1ARG(1AMR/1ART)	1**-1**	–	–	2**+2**	8**+2**	8**+7**
1ARG(1AMR/1ASE)	0**+9**	0**+1**	14**+10**	14**-9**	24**+8**	24**-12**
1ARG(1AMR/1CQ8)	8**+1**	8**-7**	12**+15**	12**-7**	18**+24**	18**-1**
1ARG(1ARS/1CQ7)	1**+7**	–	14**+21**	14**-8**	18**+52**	18**-8**
1ARG(1ASE/1CQ8)	6**-6**	6**-6**	12**-9**	12**-9**	18**-8**	18**-8**
1ARG(1CQ7/1CQ8)	–	–	–	5**-1**	–	10**-1**
1ASM(1ART/1ASE)	1**-1**	–	27**-18**	27**-12**	42**-15**	42**-18**
1ASM(1ASA/1ASE)	0**+4**	0**+2**	1**+15**	1**+10**	2**+27**	2**+26**
1CGI(1CHG/1HPT)	1**-1**	–	13**-4**	13**-1**	38**-19**	38 **-2**
1CGI(1GCD/1HPT)	–	–	–	19**-1**	–	34**-1**
1LYS(193L/1LSC)	–	–	–	–	3**-3**	3**-3**
1TPA(1AUJ/1BPI)	–	–	2**+1**	2**+2**	8**+1**	8**+2**
1TPA(1BJU/1BPI)	–	–	4**+2**	4**+5**	13**+2**	13**+4**
2PTC(1AQ7/1BPI)	–	–	20**-20**	20**-7**	54**-53**	54**-2**
2PTC(1BJU/1BPI)	–	8**-2**	–	38**-2**	70**-1**	70**-1**

Table 7.16: *Evaluation of docking results by DRUF protocol. The initial numbers for N10, N50 and N100 are compared to the results of the docking using flexibility information. The changes are given in bold numbers, dashes denote no changes. Here, the SVM based flexibility information for the whole side chain is incorporated.*

(see Fig. 7.31(b)). This is also verified inspecting the changes depicted in figure 7.31(c). But also high ranked hypotheses are moved to lower ranks. The most obvious changes due to the flexible docking effect hypotheses ranked on lower ranks (Rank $\geq$ 400). For the flexible docking these hypotheses are assigned to a small range of ranks then the hypotheses predicted without flexibility. False positive hypotheses with an RMSD of around 20Å are assigned to lower ranks but also a lot of hypotheses with higher RMSD score are ranked higher. Since the estimated cost are high for these hypotheses they do not interfere with the top ranked hypotheses. Selecting hypotheses with costs up to 100 from this test case one would obtain hypotheses with RMSD values below 10 or even 5Å.

For the results using a scaling factor of 1.0 (see Fig. 7.32) a different observation could be made. Here, most hypotheses are moved towards the right border of the plot as shown in

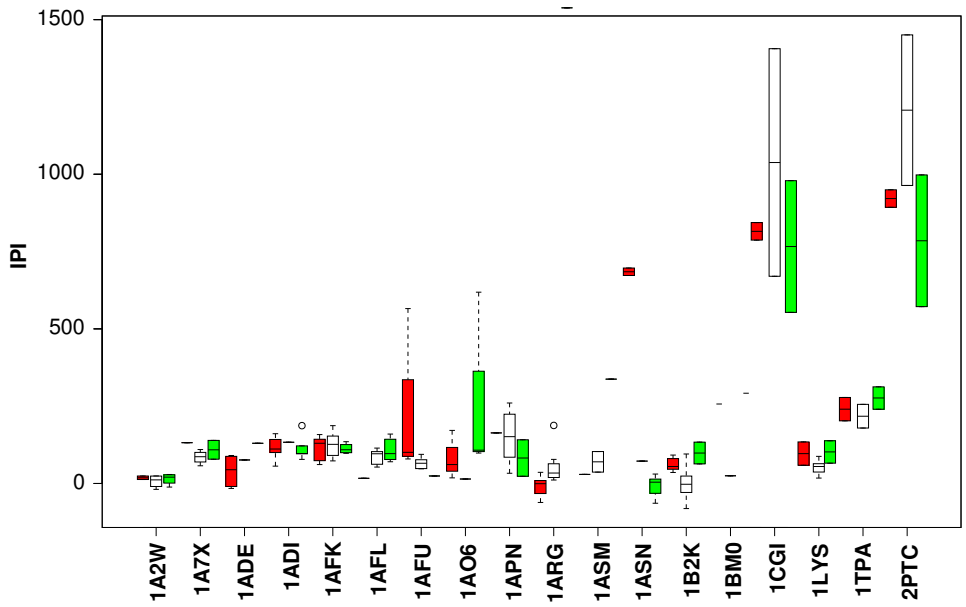
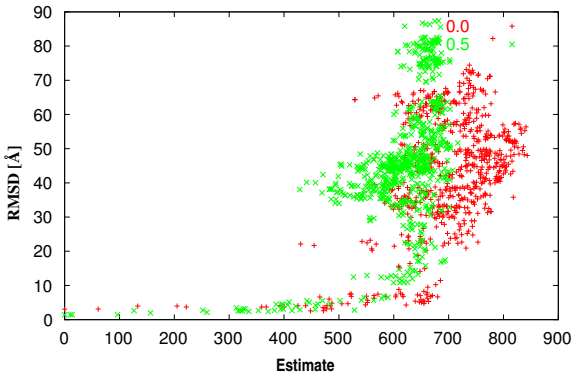


Figure 7.30: *IPI evaluation of flexible docking. Here, the overall flexibility information calculated by the SVM based prediction for the side chains is used. The differently coloured box-plots (white: no flexibility, red:* $\omega = 0.5$*, green:*$\omega = 1.0$*) represent the IPI scores of the test cases grouped by their reference complex.*

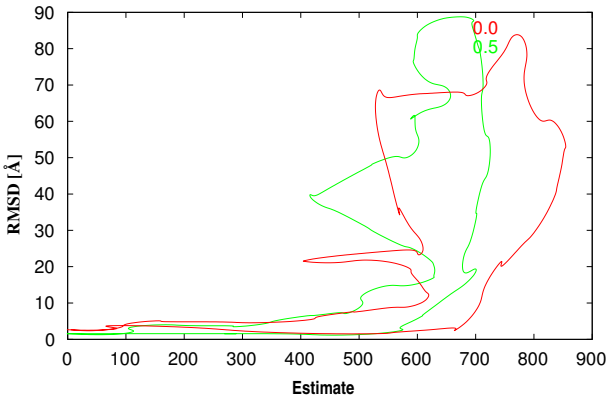
figure 7.32(b). Also, near native hypotheses[11] are yielded. They are assigned to low ranks but compared to the docking without flexibility, the RMSD values are larger (cf. Fig. 7.32(a)). Inspecting the plot of changes (see Fig. 7.32(c)) this becomes obviously. Most of the newly placed hypotheses depicted by the green rectangles are positioned above the red coloured boxes of the hypotheses resulting from the docking without flexibility. The only exception is the best ranked hypothesis. Here, an improvement can be observed. The best ranked prediction yielded docking the test case `1ARG(1AMR/1ASE)` without flexibility information has a RMSD of 2.56Å. For the docking with enabled flexibility a RMSD of 1.4Å and 0.8Å ($\omega = 1.0$) is reached. This is also reflected by the DRUF evaluation (see Tab. 7.16). Within the top 10 ranked hypotheses an increase in the number of hypothesis by 1 is counted. For the other two measures, the number of hypotheses decreased by 9 and 12 respectively.

Comparing the two figures, most differences in the scoring effect hypotheses on low ranks. In figure 7.31 a large number of false positive hypotheses are scored better then for the docking without flexibility. For the scaling factor 1.0 these number is reduced. Only few hypotheses with a RMSD around 50Å remain hypotheses with wrong scores assigned.

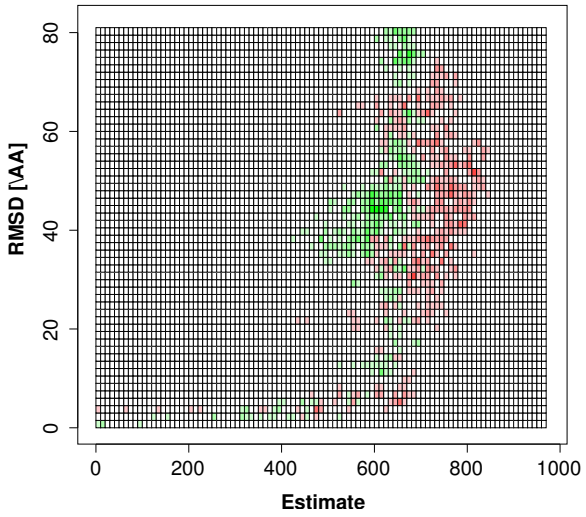
[11] Hypotheses with a low RMSD score.



(a) Comparison plot of the docking results. Here, the flexibility is scaled by $\omega = 0.5$

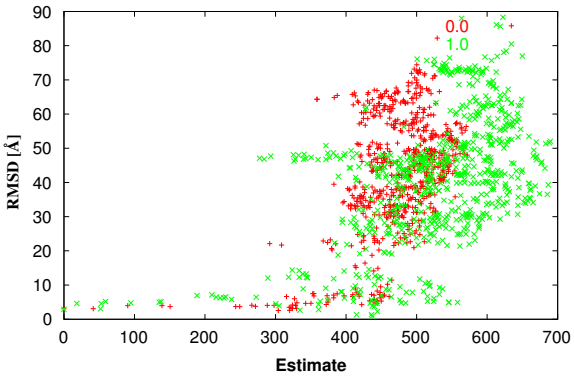


(b) Plot of the outer hulls of the docking results. Here, the flexibility is scaled by $\omega = 0.5$

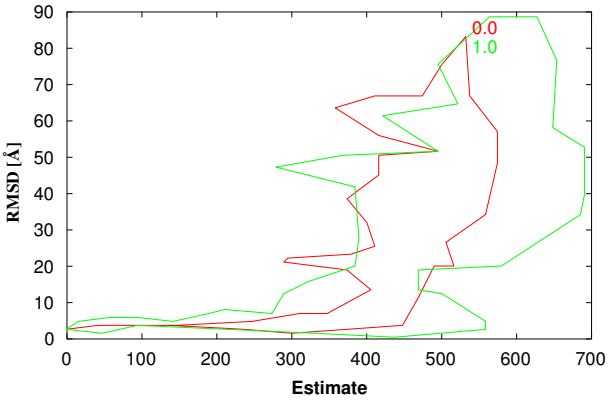


(c) Changes plot of the docking results. Here, the flexibility is scaled by $\omega = 0.5$

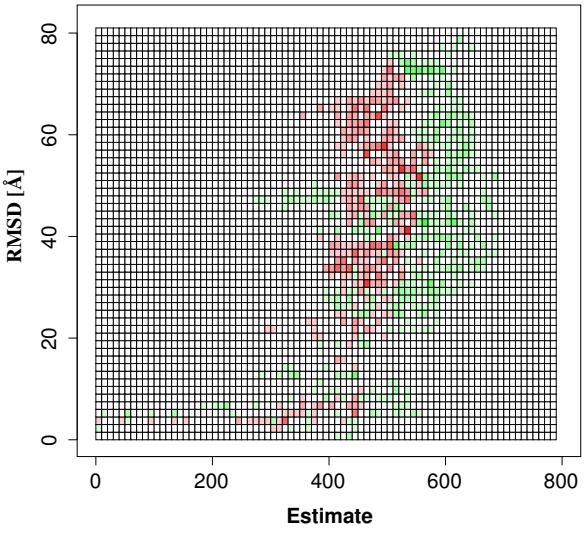
Figure 7.31: *Visualisation of the docking results of 1AMR and 1ASE. In this experiment the combined flexibility predicted by the SVM is used. It is scaled by $\omega = 0.5$. On the top left the costs are plotted against the RMSD. On the right, the outer hulls calculated on the hypotheses are shown. Below the two plots, the changes within rectangular grids is given. In red, docking results without using flexibility information and in green the results using flexibility are coloured. For figure 7.31(c) red and green fields represent the changes (see section 7.3.2)*



(a) Comparison plot of the docking results. Here, the flexibility is scaled by $\omega = 1.0$



(b) Plot of the outer hulls for the docking results. Here, the flexibility is scaled by $\omega = 1.0$



(c) Changes plot of the docking results. Here, the flexibility is scaled by $\omega = 1.0$

Figure 7.32: *Visualisation of the docking results of 1AMR and 1ASE. In this experiment the combined flexibility predicted by the SVM is used. It is scaled by* $\omega = 1.0$*. On the top left the costs are plotted against the RMSD. On the right, the outer hulls calculated on the hypotheses are shown. Below the two plots, the changes within rectangular grids is given. In red, docking results without using flexibility information and in green the results using flexibility are coloured. For figure 7.32(c) red and green fields represent the changes (see section 7.3.2).*

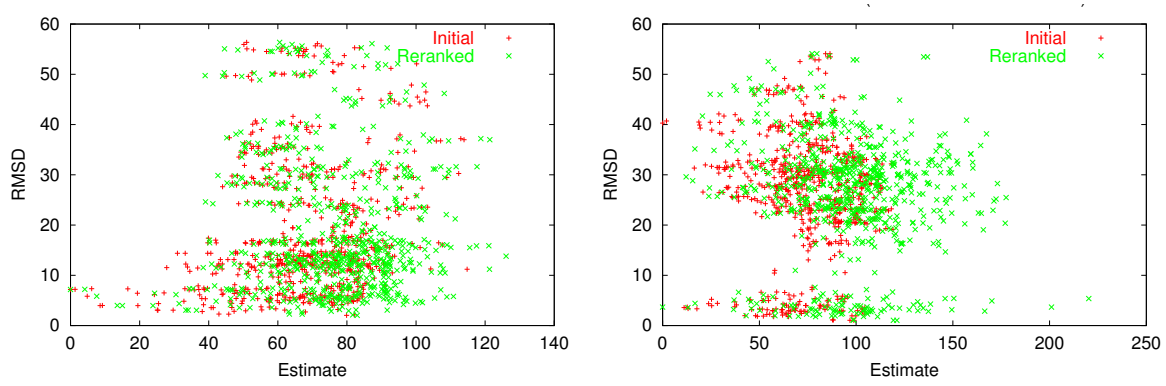
Summarising the results, the overall based flexibility predictions obtained using the SVM improves the docking results. Because the flexibility of higher torsion angles is harder to predict, the docking results show a lower performance as for the χ_1 predictions. But good results are reached, too. The results of the DRUF evaluation as well the example shown in detail prove that in most cases hypotheses with low RMSD could be assigned to high ranks. False positive hypotheses are reduced using this flexibility.

Comparing all four tested flexibility predictions, the SVM based approach yields better results. This is due to the higher prediction accuracies obtained by the SVM. The combination of the flexibility prediction for the different torsion angles do not perform that good as incorporating only the flexibility information of χ_1. In summary, the results prove that the flexibility approach is reasonable and that the ELMAR docking system can be improved.

7.4 Results from Relevance Feedback

In this section the results of re-ranking docking hypotheses using the IPHEx system are presented. In order to test the approach several experiments have been conducted.

Neumann (Neumann, 2003) run several docking experiments on a large set of test cases. The test cases have been automatically derived from PDB (see section 7.1.1). The resulting docking hypotheses are stored within our database of docking results. Because of the large number of already docked test cases, these results are taken for evaluating the IPHEx system. From this set different test cases are chosen randomly to be processed by IPHEx.

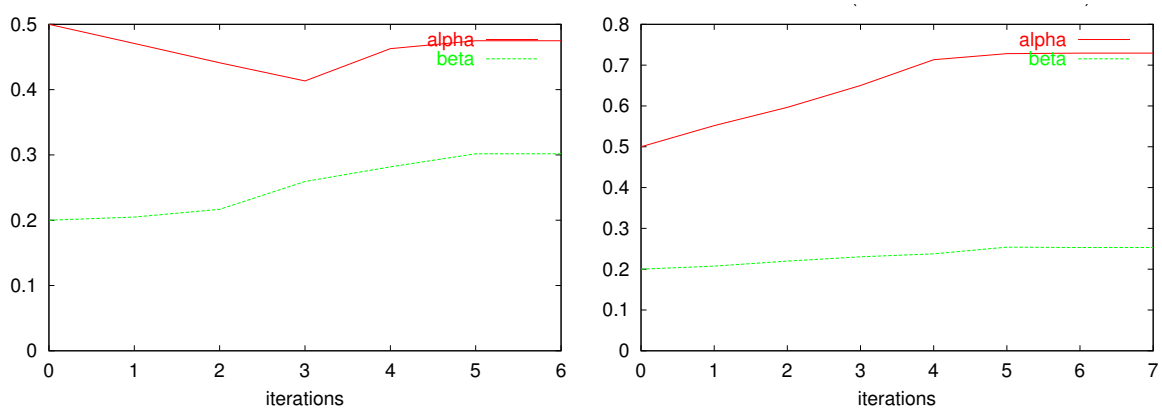


(a) Docking 1CHG (chymotrypsinogen) and 1HPT (inhibitor).

(b) Docking 2PTN (trypsin) and 6PTI (inhibitor).

Figure 7.33: *Evaluation of feedback session (red: original docking hypotheses, green: re-ranked by user) for two examples. Plot of estimated costs against RMSD.*

A typical feedback session can be described as follows: A sequence of 20 docking hypotheses (ordered by similarity measure of the features) from the result set of a docking test case



(a) Weight development for 1CHG/1HPT.

(b) Weight development for 2PTN/6PTI.

Figure 7.34: *Development of weights during feedback session. Here, the adaptation of the weights α and β for two feedback sessions is given.*

is presented to the user. During the evaluation these hypotheses are superimposed to the 3D structure of the known complex (see Fig. 6.4(a) on page 61). The number of hypotheses to be scored is chosen according to usability aspects but can be adjusted by a command-line parameter (see section D.3). Scoring only 20 of 700 hypotheses results in a very short iteration cycle. Therefore, changes within the ranking of the docking hypotheses can be done in a short time (e.g. few iterations). Having finished these 20 hypotheses a weight adaptation step is performed, re-ordering the whole list of hypotheses. These steps can be repeated. Usually after few iteration a steady state is reached which means that there are no more changes within the weights. The feedback session should be terminated since further scoring of the hypotheses will not yield better results. Since the number of iterations differs from test case to test case here the maximum number of feedback iterations is limited to five iterations arbitrarily. The learning rate ε (see Eq.6.9 on page 62) is set to 0.01 for all experiments.

In order to evaluate the results after processing the hypotheses with IPHEx parts of the evaluation methods described in section 7.3.2 are taken. For a quick overview, the cost estimates (see Eq. 6.1) of the original and re-ranked hypotheses are plotted against the RMSD (see Fig. 7.33). In order to evaluate the improvements of the modified weights α and β, parts of the DRUF protocol (Halperin *et al.*, 2002) are used. In order to express changes during docking the differences in the N10, N50 and N100 values (see Tab. 7.17) are calculated. Additionally the results are evaluated using the IPI measure.

First results show that weights modified by relevance feedback improve the cost function and thus the ranking of docking hypotheses. As an example, figure 7.33(a) shows the initial distribution of docking hypotheses (red) and the distribution of the same hypotheses after a feedback session of 5 iterations (green) for the unbound proteins 1CHG and 1HPT. As reference for comparison, the structure of the homologue complex 1CGI is used. It can be seen

that good hypotheses (low RMSD) are re-ranked towards lower ranks whereas bad hypotheses are assigned to higher ranks. Another example is shown in figure 7.33(b). Hypotheses with a lower RMSD are scored better using the modified weights and originally bad hypotheses (RMSD at around 30Å) are negatively scored. Figure 7.34 shows the development of the weights during the feedback. Here, the number of iterations is larger than five, to illustrate the stable state usually reached after five iterations of feedback.

Inspecting figure 7.33(b), the initial best hypothesis has an RMSD of around 40Å, whereas after re-ranking the best hypothesis has an RMSD of 3.6Å. This change is also reflected by the DRUF measurement (see table 7.17, 2PTC(2PTN/6PTI)). In two cases hypotheses are re-ranked wrongly (1ASN(1AMQ/1AMR), 3ENR(1NLS/1SCS)).

Test Case	N10	N50	N100
1ASN(1AMQ/1AMR)	2+**1**	15-**2**	29+**3**
1DYJ(1DRH/7DFR)	–	–	–
1TPA(1BJV/6PTI)	1+**1**	7+**2**	20+**6**
1TPA(1C5R/4PTI)	–	8+**6**	26+**2**
1TPA(1C2D/4PTI)	–	5+**5**	17+**3**
1TPA(1C2J/4PTI)	–	4+**3**	29-**8**
1LZS(1JSF/1REZ)	–	–	–
2PTC(1AQ7/1BPI)	–	17+**9**	30+**16**
2PTC(1QB9/4PTI)	0+**3**	14+**2**	23+**6**
2PTC(2PTN/6PTI)	–	5+**1**	–
2TGP(2TNL/4PTI)	6+**3**	23+**15**	46+**27**
3ENR(1NLS/1SCS)	–	1+**1**	4 -**1**
8RSA(1EOW/1RAT)	0+**1**	3+**2**	10+**1**

Table 7.17: *Feedback results: initial numbers and absolute changes (bold). A dash denotes no change.*

In table 7.17 the results of the experiments are shown. For most test cases an improvement of the ranking is achieved using the relevance feedback approach. In half of the test cases good hypotheses are re-ranked into the top 10 ranks (e.g 2TGP(2TNL/4PTI) or 2PTC(1QB9/4PTI)). In the other cases no hypotheses are placed in the N10, but within the less restrictive groups N50 and N100 (see 1TPA(1C2D/4PTI) or 2PTC(1AQ7/1BPI)).

Comparing the results of the evaluation by the DRUF protocol to the IPI scores (see Tab. 7.18), one can see that for most test cases no improvements are yielded but the same IPI score is reached. Although on average no improvements can be observed using the IPI measure, for four test cases higher scores are reached while re-ranking the hypotheses. In case of 1DYJ(1DRH/7DFR) during re-ranking the number of hypotheses within the N10, N50, and N100 range keeps constant but inspecting the overall set of hypotheses a slight increase of the IPI value is measured. Exemplarily, for 2PTC(1QB9/4PTI) a large number of hypotheses have been ranked additionally within the top 100 by the relevance feedback approach. This increase is reflected within the IPI scores. Here, the IPI value is raised from initially 21.7 to 44.4.

Test Case	IPI	
	initial	re-ranked
1ASN(1AMQ/1AMR)	41.4	40.9
1DYJ(1DRH/7DFR)	9.6	10.2
1TPA(1BJV/6PTI)	77.8	81.5
1TPA(1C5R/4PTI)	55.1	49.7
1TPA(1C2D/4PTI)	53.8	48.0
1TPA(1C2J/4PTI)	63.4	63.4
1LZS(1JSF/1REZ)	26.9	24.2
2PTC(1AQ7/1BPI)	33.0	37.3
2PTC(1QB9/4PTI)	21.7	44.4
2PTC(2PTN/6PTI)	95.7	92.4
2TGP(2TNL/4PTI)	72.7	72.5
3ENR(1NLS/1SCS)	27.5	25.5
8RSA(1EOW/1RAT)	20.9	19.8

Table 7.18: *Evaluation of feedback results by IPI measure.*

One application of the IPHEx system is to find good parameters for each protein class. Therefore, the modified weights are mapped to all proteins having the same EC number as the test case used within IPHEx. Since Neumann performed a large number of docking runs, several enzyme classes can be covered. But in most cases only one reference complex exists for a class. In order to have test cases for different complexes within one enzyme class only those are chosen that have at least three reference complex structures.

EC Class	reference complex	examples	overall improvement	α	β
1.5.1.3	1DYJ	836	–	0.47	0.21
2.6.1.1	1ASN	140	40%	0.53	0.22
3.1.27.5	8RSA	165	15%	0.47	0.21
3.2.1.17	1LZS	34	–	0.41	0.23
3.4.21.4	2PTC	442	12.5%	0.4	0.41

Table 7.19: *Improvements per EC class after re-ranking using* IPHEx*. Here, the percentage of the overall improvement is given. The weights α and β given in the last two columns have been applied to all examples in the corresponding enzyme class. Dashes denote no improvements.*

In table 7.19 the results of mapping the improved weights of the feedback sessions to all test cases of the corresponding enzyme classes, are shown. Here, the overall improvement summarised over the N10, N50 and N100 counts is given. In three of the five tested enzyme classes improvements could be observed. In case of the class 2.6.1.1, 40% of the test cases show improvements within the first 100 ranks. In case of the other two enzyme classes improvements between 10 and 15% can be reached.

7.5 Discussion

In the previous sections the results of the different experiments have been outlined. In the following they will be discussed. At first, the two classifiers for predicting the side chain flexibility are compared. Then, the docking experiments and the results of the IPHEx system are discussed.

7.5.1 Classification of Side Chain Flexibility

The flexibility of a side chain has been classified by two approaches, one based on a threshold criterium and the other using support vector machines. Comparing both approaches they have in common that the classification accuracy decreases in case of the higher torsion angles whereas for the first torsion angle best results are achieved. The threshold based classifier performs worse than the SVM. A reason for this is that only one feature (energy difference) is used. In case of the support vector machine the combination of several features ensures a more robust classification.

Inspecting the different residues one can observe that some residues can be classified easier than others. In case of the threshold based classifier, the worst classification results for the χ_1 are received for SER, TYR, and GLU whereas CYS, TRP, and ARG perform best. Exemplarily, for SER an explanation for the low prediction accuracy can be found when inspecting the energy landscape (see Fig B.1(m)). For SER, the energy landscapes for each class do not differ significantly. Thus, distinguishing flexible and non–flexible Serine residues is difficult.

For the higher torsion angles the classification power drops near the chance line (ROC area of 0.5). Here, best results are achieved for MET (χ_2, χ_3), TRP (χ_2) and HIS (χ_2). All other residues cannot be predicted that good. A reason for this weak classification within the higher torsion angles is due to that the overall flexibility increases (cf. Koch, 2003). For the higher torsion angles steric hindrance is reduced because of a greater distance to the backbone. Here, influences of neighbouring groups are reduced allowing more flexibility. Of course this influences the energy landscape calculated. Since there are fewer steric clashes or interactions with the surrounding, the energy landscape is more flat and the difference between the base energy and the energy minimum is smaller, thus resulting in less discrimination power of the feature.

The same effects can be observed for the SVM based approach. Here, the classification accuracy is also reduced within the higher torsion angles. Generally, the average accuracy for each torsion angle is better than for the threshold based method. In contrast to the threshold based method the average classification rate does not decrease steadily (see Tab. 7.7) but drops for χ_2 and then increases again for χ_3.

Looking at the single residues' results, for χ_2 one can observe that the branched residues ASN and LEU as well as TYR and PHE, both possessing a ring system within their side chain, perform worst. In case of LEU, PHE and TYR the second torsion angle is located at a branch position. In all cases the branch is symmetric. Because of the symmetry a correct labelling

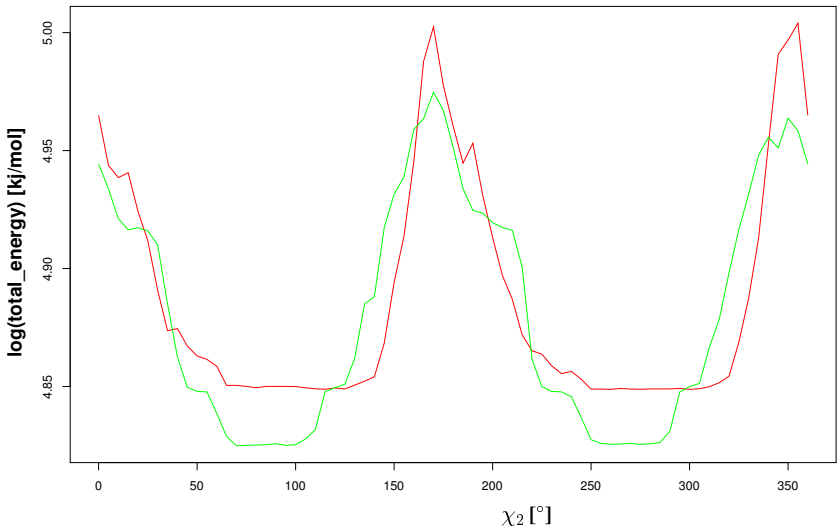


Figure 7.35: *Mean energy landscape of PHE for χ_2. In red the means of the total energy of flexible labelled residues, in green of rigid residues are given.*

during crystallography is hard and sometimes the C_γ (needed for the correct calculation of χ_2, see section 2.1) is assigned wrongly. This can effect the labelling of the data set, if such a side chain of unbound protein and a corresponding complex are resolved differently because of wrong labelling of the carbon atoms. Either, the side chain may be assigned as flexible (if the atom labels differ) and in fact it is not flexible, or the side chain's carbon atoms are labelled similar although this is wrong. In this case a flexible residue is assigned rigid. Thus, the features calculated on these wrongly labelled data may falsify the training of the classifiers.

Another aspect resulting from the symmetry of the branched side chain is that the rotamer distribution is bimodal and thus, the distribution of the energy landscapes, too (cf. Koch *et al.*, 2002; Zöllner *et al.*, 2002). The difference between the energy landscape of a flexible and a rigid side chain (see Fig. 7.35) is less than e.g. for the first torsion angle (cf. Fig.B.1(l)) because large parts of the energy landscape are similar due to the reduced number of rotamers. In case of branched side chains or residues with a ring system only two of three rotamers can be assigned for χ_2.

For the χ_3 and χ_4 torsion angle a reduction in the classification accuracy can be observed for all residues. Here again, a reason for this may be a smaller difference in the energy landscapes between the classes because of less steric restrictions of the environment. For χ_3 the classification accuracy of LYS and MET is obviously higher than for the other residues (ARG, GLN, GLU). The three residues GLU, GLN and MET possess three torsion angles. In case of GLU and GLN at the end positions of the side chains a functional group is located ($CONH_2$ in case of GLN and COOH for GLU, see Fig. C.6 and C.7) whereas for MET a sulfur atom is located at the C_δ position of the side chain. Attached to this side chain a CH_3 group marks the end of the side chain (see Fig. C.13). The sulfur atom compared to a carbon atom

is slightly bigger and has a sp^2 hybridisation. This means that the torsion angle between the C_γ, the sulfur atom and the methyl group usually is planar and about 120°. In case of GLU or GLN the functional groups are attached by an angle of 109°, the standard tetrahedral angle. So, a rotation of the end groups of the side chains is different to that of MET. For MET the assumption can be made that changing the χ_3 torsion angle differs more from flexible to rigid residues than for the other two residues.

The highest torsion angle – χ_4 – is only occupied by two residues: ARG and LYS. The classification results for both residues do not differ much because their side chains are rather similar. Both side chains are four carbon atoms long and carry a charge at the end. The only difference is the end group. LYS has a NH_2 at the end whereas ARG consist of a more complex functional group (cf. Fig. C.2). Since these groups are far away from the backbone, steric restrictions only occur, if direct neighbours also possess large side chains or the residue is buried, respectively.

Although some residues and torsion angles cannot be predicted that good, most residues can be classified at high accuracy, especially utilising the support vector machine classifiers. Here, a classification with up to 90% (CYS, χ_1) accuracy is yielded. On average a classification accuracy of 70% is reached. Comparing the two classification approaches improvements can be observed for the SVM based approach. Providing additional features characterising the specific residue more precise a better discrimination of flexible and non-flexible residues is achieved. Furthermore, the feature vectors enable the support vector machine to find good a hyperplane for separating the two classes. Inspecting the distribution of the examples in the test set used for evaluating the SVM, the observation can be made that the examples are distributed equally between the classes, e.g. the number of false positive and false negative are equally as well as the true positives and the true negatives (cf. App. B.5).

Summarising this section, the flexibility of side chains can be predicted using the methods proposed in this work. The higher torsion angles are harder to predict, since the energy landscapes for these torsion angles have less discrimination power. This can be observed for both classifiers. Best results are reached for the first torsion angle (χ_1). Steric restrictions of the backbone ensure separable energy landscapes. The additional features used within the SVM make the classification more robust. Thus, the SVM based approach should be preferred when classifying the flexibility of residue side chains.

7.5.2 Protein–Protein Docking using Flexibility Information

Before discussing the results of the flexibility evaluation within the docking system ElMaR, some general aspects of ElMaR should be pointed out. The ElMaR docking algorithm is designed for speed. Docking results can be received within minutes. This is reached by an abstraction from the protein model as outlined in section 4.1. Although ElMaR reaches on average good results and docking hypotheses can be scored on high ranks, sometimes no good predictions can be made (e.g. see results of `1A2W`). Besides this, the abstraction from the protein structures has also an influence if flexibility information (especially, if modelled like in this work) is incorporated. Since the protein's 3D structure is sampled into discrete

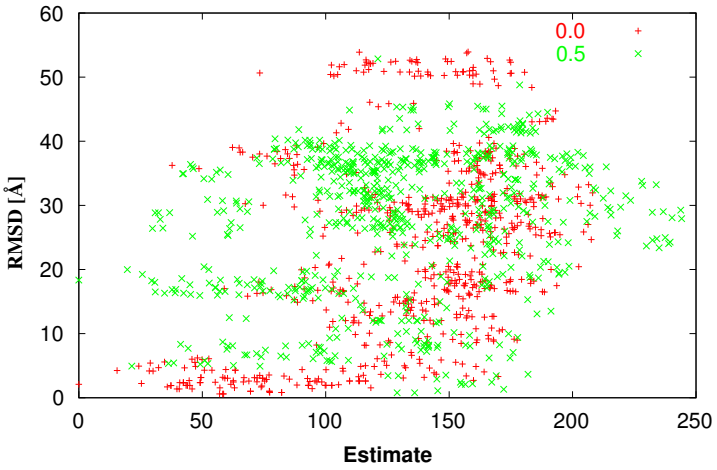


Figure 7.36: *Comparison of test case 2PTC(1AUJ/1BPI). Here, the results of docking the test case with flexibility (green) and without flexibility (red) is shown. For the flexibility information the threshold based predictions for χ_1 are taken.*

voxels, they do not always cover a single residue but several or only a part of a side chain. So, the flexibility information is smoothed over the area of the voxel. On the one hand this can be helpful if the contact side of the proteins in question is large. In this case, the flexibility is distributed across the whole contact side allowing similar scores for steric clashes within this area. But on the other hand, this also allows flexibility where it occurs only at a few points or in a small area (e.g. in case of an enzyme that has a specific selection of its target) in reality. In this case false hypotheses may be predicted.

Furthermore, the ElMaR system is a research system which means that it cannot be used in production environment, yet. Neumann just finished his work when the experiments of this thesis have been started. As already outlined in section 7.3.3, the different modules do not run stable. Thus, the resulting set of processed test cases is reduced and differs between the experiments. Fixing these instabilities would have to be done so that an evaluation of larger sets of test cases will be possible without problems.

Besides this, another error occurred running this system. For some test cases (e.g. 2PTC(1AUJ/1BPI)), no good results are obtained. Inspecting table 7.10, one can see that within the experiments using the scaling factor $\omega = 0.5$ all hypotheses are lost for all three evaluation measures. In order to explain these losses, at first the incorporation of the flexibility is outlined. Docking a test case is done in three steps. A fast compilation of initial hypotheses is done first by matching the surfaces of the unbound proteins. Then, the search space of the hypotheses is explored by rolling the one protein over the other around the initial hypotheses. From these, a subset according to correlation of the features is done. At this point no flexibility is used at all. In the next step, flexibility information is introduced, re-scoring the selected subset from the previous step. Thus, the flexibility information only has an impact on the scoring function, not on the translation or rotation of the proteins to set up a hypotheses.

In order to analyses these negative changes mentioned above, the set of hypotheses of the initial reference experiment (without flexibility) is compared to the set of hypotheses using the flexibility. Both experiments were conducted on the same input data. The comparison is done in order to find out where the hypotheses are moved (e.g. to higher ranks), since the flexibility only has an impact onto the scoring of the hypotheses. Therefore, the translation and rotation vector are compared. Surprisingly, no matching hypotheses are found. Thus, the assumption is that two different sets of hypotheses have been compared by the DRUF protocol. Furthermore, this error is produced by the ElMaR system, since the flexibility information has no impact on the exploration of docking hypotheses. Figure 7.36 shows the super-imposition of the to result sets. The differences between the hypotheses in the lower left corner of plot are obvious.

Same observations could be made for the other test cases loosing hypotheses. Since other docking runs perform well (e.g. the same experiment but using a scaling factor of $\omega = 1.0$) this error may due to some rounding errors during generating the hypotheses. Here, a more detailed analysis of the error would have to be performed. The conclusion that can be drawn is, that these results should be neglected in the evaluation of the flexibility.

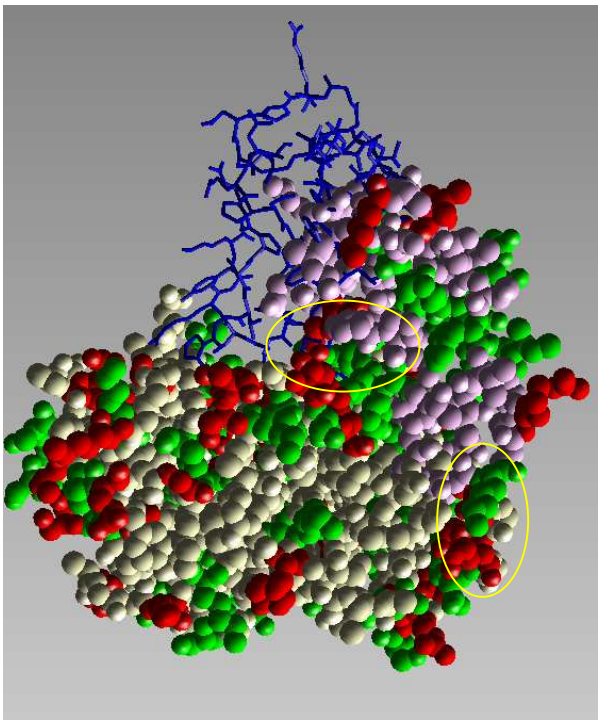


Figure 7.37: *False positive hypothesis from the docking of 2PTC(1AUJ/1BPI). Here, the proteins are coloured according to the correct (green) and false (red) predictions of the flexibility. The residues not involved within the flexibility predictions are coloured in beige for the enzyme (1AUJ) and in pink for the inhibitor (1PBI) in order to visualise the different structures. The marked areas depict regions where errors occurred. The blue structure represents the true docking constellation. The hypotheses has been moved from rank 114 to 10. The geometry score dropped from 76 to 74 using the additional flexibility.*

A different aspect that should be kept in mind is that the flexibility information is based on a prediction. Although the classification accuracy of the side chains is quite good, also some residues are classified wrongly. By including such flexibility information there will be a risk to increase the number of false hypotheses.

Analysing the impact of the flexibility information is rather complex, since the flexibility predictions for each residue differ. The contact sides of the hypotheses also differ because of the translation and rotation of the proteins. Here, two examples are shown to explain a false ranking of hypotheses.

The changes of hypotheses are analysed by comparing the experiments conducted. In most cases it can be observed that good hypotheses are shifted towards higher ranks because false positive hypotheses are moved from higher ranks to lower ones. Exemplarily, in figure 7.37 and 7.38 two docking hypotheses are visualised. They are taken from the test case `2PTC(1AUJ/1BPI)`. The docking was performed using a scaling factor of $\omega = 1.0$ for the flexibility and the threshold based flexibility for χ_1 is used. In both figures the proteins are coloured by the true and false flexibility predictions. In green correct predictions are shown whereas false prediction are coloured in red. In order to distinguish the enzyme (`1AUJ`) from the inhibitor (`1BPI`) the residues not classified are coloured in beige and pink. In blue the correct solution is given.

In figure 7.37 several wrong predicted residues have a contact to the surface of the inhibitor (see yellow marked parts). Since this hypothesis has been shifted towards lower ranks, the wrongly predicted flexibility mislead the scoring function. Here, the geometry score of the hypothesis which includes the flexibility drops from 76 to 74. Thus, the overall score is reduced and the hypothesis is assigned to a lower rank (rank 11).

Inspecting the second example, only small changes within the geometric scoring are observed. Here, false predictions are covered by the correct ones (see Fig. 7.38). The good scoring is not changed much. Thus, most impact on changing the rank of this hypothesis – it is moved from rank 10 to 12 – dues to the newly placed hypothesis shown in figure 7.37.

Although, there exists a risk to incorporate errors made by the classification approach, the obtained flexibility information can be used to enhance the soft volume model of ElMaR as the results proof.

For an overall estimate of the results, two directions to analyse the results can be made: comparing the results between different complexes and evaluating the results within the group of test cases belonging to the same reference complex.

Inspecting the evaluation of the flexibility information incorporated into ElMaR, one can see that in most cases improvements are reached. Besides this, placing the near native solutions[12] within low ranks also hypotheses with high RMSD which have been assigned low ranks within docking without flexibility are assigned higher ranks in case flexibility information is present.

[12] Hypotheses with low RMSD score.

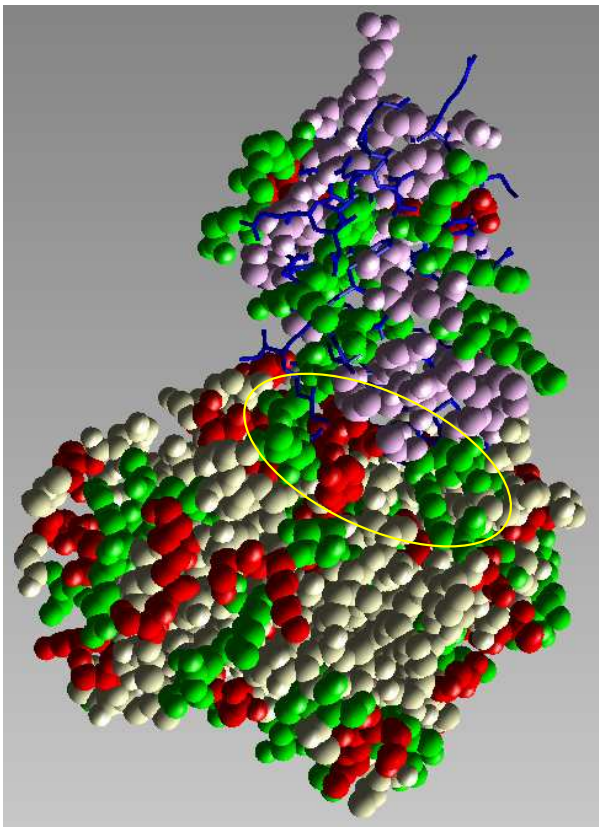


Figure 7.38: *False positive hypothesis from the docking of 2PTC(1AUJ/1BPI). Here, the proteins are coloured according to the correct (green) and false (red) predictions of the flexibility. The residues not involved within the flexibility predictions are coloured in beige for the enzyme (1AUJ) and in pink for the inhibitor in order to visualise the different structures. The marked area depict regions where errors occurred.The blue structure represent the true docking constellation. The hypotheses has been moved from rank 10 to 12. The geometry score differs by 0.002.*

For most reference complexes more than one unbound test case was available so that the influence of the flexibility is not bound to a single example. This gives the opportunity to analyse the behaviour of the scoring function. Since box-plots are used to summarise the results of the different test cases, the size of the box (covering 75% of the data) gives a hint, how the different test cases are scored. In most cases (e.g. 1LYS, 1A2W) the variance within the samples for the flexible docking is smaller then for the docking without flexibility. This observation is also supported inspecting the plots of the detailed analysis for the test cases 1A2W(1RAT/1BEL) and 1TPA(1AUJ/1BPI). Here, the hypotheses predicted using the flexibility information cluster more often then for a docking without flexibility (cf. Fig. 7.25 or 7.26). Thus, the conclusion can be drawn that the flexibility information not only improves the results but also supports the scoring of ElMaR.

Comparing the results using the differently predicted flexibility information better results are reached applying the SVM based approach because of the higher accuracy predicting flexible residues correctly and thus, errors incorporated by false prediction like shown in figure 7.37 are reduced. Furthermore, similar results compared to the docking without flexibility are reached for those test cases, that did not perform that good when using the threshold based predictions. But also for the threshold based predictions good results are reached (cf. docking of 1ASM(1ASA/1ASE)). Differences are found comparing the docking using the χ_1 predictions to the experiment incorporating the overall flexibility. For both flexibility scores one can observe that the docking applying the overall side chain flexibility performs little weaker then the docking incorporating the flexibility predictions for χ_1. A reason is that the lower prediction accuracy of the higher torsion angles. As already outlined in section 7.5.1 a flexibility prediction of the higher torsion angles is more difficult. Therefore, the prediction error can increase when combining all the torsion angles to an overall score.

Since two different scaling factors are tested within the docking, these have to be compared, too. For the docking experiment using the predictions for the χ_1, only minor differences in the results are observed applying the different scaling factors. Only in case of the experiment run with the SVM based overall flexibility score, the different factors influence the results. Here, for a $\omega = 1.0$ a better performance is reached than for $\omega = 0.5$. This is due to the fact that the flexibility information for the whole side chain takes values from 0 to 1 because of the sum over the different parts (see section 5.2.5). In case of the χ_1 flexibility a binary decision is taken (either 0 or 1). Here, the flexibility values lie on the borders of the scaled range. Thus, the distribution of the flexibility scores is similar for all scaling factors, just the its value differs due to the scaling. But in case of the overall flexibility, the different scores are distributed over different ranges and therefore, have a greater impact on the scoring function. In order to analyse the impact of this scaling factor in detail, experiments would have to be performed.

In summary one can state that the flexibility information has an impact on the docking and that it improves the predictive power of the ElMaR system. Using the flexibility predictions of the first torsion angle, best results can be obtained. The SVM based flexibility predictions are more accurate. Thus, incorporating this information should be considered.

7.5.3 Enhancing ElMaR Scoring by Relevance Feedback

The second approach presented in this thesis tries to enhance the scoring function of the ElMaR docking system. Here, a relevance feedback based approach is proposed. The modified weights are adapted to the corresponding enzyme classes to test whether improvements can be reached for the other test cases.

Although the feedback approach has been tested on a limited set of test cases, clear trends can be observed. Within the top 100 ranks for most test cases an increase in good hypotheses is yielded. The relevance feedback approach can be used for improving the weighting scheme of ElMaR.

These changes are only reflected in few cases using the IPI scores (e.g. `1TPA(1BJV/6PTI)`). This is not surprising, since the IPI score summarises over the whole set of hypotheses. But using the relevance feedback, the adapted weights are applied to all these hypotheses in order to re-calculate the list of hypotheses (*HL*) which is presented to the user in the next iteration for giving feedback. This usually results in a change of the costs and rank for all hypotheses (cf. Zöllner *et al.*, 2003). So, the IPI measure verifies that the feedback approach can be used to improve the scoring function. Specific hypotheses, e.g. good predictions are scored more correct whereas the other hypotheses remain unchanged. If differences can be observed comparing the IPI score these are a hint for important changes within the scoring of these hypotheses. Exemplarily, for the test case `2PTC(1QB9/4PTI)` an increase of the IPI scores correlates to an increase in the N10 of 30%. Here, the IPI score increases by nearly 50%.

In case of the adaptation of the modified weights for few enzyme classes no improvements can be reached. But for most classes tested here, improvements in the ranking of hypotheses are yielded. Best results have been reached for the enzyme class 2.6.1.1 (Aspartate amino transferase). No improvements within the adaptation can have different reasons. On the one hand the N10, N50, and N100 scores of the DRUF protocol only focus on the top ranked hypotheses. Changes e.g. in the range above the 5Å are not taken into account. Thus, the IPI score would have to be applied to the results of this adaptation, too. But also the adapted weights may only have an impact on the test case, the feedback has been assigned to. For other test cases within the same enzyme class the changes do not have an effect. For instance, good feedback results have been obtained for the test cases of the complexes `1TPA`, `2PTC` and `2TGP` but an adaptation of the weights onto the all test cases of corresponding enzyme class (3.4.21.4) only shows small improvements.

Up to now the concept of this approach has been proven. But of course, the results of this method depends on the feedback given by the user. Therefore, experiments would have to be scheduled (see section 8.2) to estimate, how the results of IPHEx change with respect to the feedback of different users.

Chapter 8
Conclusions & Outlook

The post genomic era will become more and more important in the next years. The analysis of the underlying mechanisms of interactions, e.g. of proteins within a metabolic system is required to interpret the huge amount of genomic data produced. The correct modelling of the docking mechanisms of proteins will be essential, since proteins are involved at all metabolic levels, from DNA transcription to immune defence or signalling.

This chapter summarises the thesis, and gives an outlook to further research in this field.

8.1 Summary

In this thesis two different approaches to enhance protein–protein docking have been outlined. On the one hand the modelling of flexibility information has been addressed to simulate conformational changes during docking ("induced fit"). On the other hand an another important part of a docking system, the scoring of predictions is addressed. Here, especially the scoring of the ELMAR docking system is improved.

Flexibility Approach

The flexibility of amino acid side chains enables proteins to change their conformations during docking in order to recognise a possible target and to initiate a biological function (e.g. enzymatic reaction). In this work, the flexibility of side chains is modelled by a classification approach. Features from different sources (among others energy calculations) are combined to discriminate residue side chains. A classification is performed on unbound protein structures because the flexibility information is then reusable and not bound to a certain test case.

Energy based features are calculated by scoring synthetic conformations applying the AMBER force field. The synthetic conformations are received rotating the torsion angles of the residue's side chain. Besides the energy difference also the solvent accessible surface area, the original conformation, secondary structure information as well as the temperature factor of the side chain are used. Environmental information is gathered from the energy landscape resulting from the synthetic conformation. This signal is decomposed by a wavelet

transformation in order to receive a set of coefficients that are characteristic for flexible or non–flexible side chains.

In the first approach, the energy difference between the original conformation and the optimal conformation estimated by the rotation of a torsion angle is taken to discriminate flexible from non–flexible residues. A threshold estimated on a training set is used for classification.

Several features are combined to predict the flexibility of side chains in the second approach. For the classification a support vector machine was chosen. The selection of features is guided by a principle component analysis: the different features are combined to a single feature vector and the principle component analysis is applied. Then according to the eigenvalue spectrum a set of principle components are selected (for each residue type) and the support vector machine is trained.

The two methods were trained on a set of 232 unbound proteins. The threshold based approach was evaluated using Receiver Operating Characteristic analysis. It has been also used for estimating the threshold. In case of the SVM, a 10–fold cross evaluation is performed. Furthermore, the results of both approaches are tested within the docking system ElMaR.

Both approaches can be used to classify the flexibility of amino acid side chains. Comparing the two methods the SVM reaches better results than the threshold method. It yields on average an accuracy between 60 and 75% for the different torsion angles and residues. The docking results verify that these predictions can improve the protein–protein docking. In most cases the docking results were improved if flexibility information was presented to the algorithm.

Scoring of Docking Hypotheses

A second goal of this thesis was to improve the scoring of docking hypotheses. Here, an approach using QbC techniques, especially user based relevance feedback is proposed. Humans still have superior capabilities in discriminating patterns then machines. In this approach a subset of hypotheses is presented to the human expert who ranks them by their relevance (here difference) compared to the known complex. The IPHEx system then adapts the weights of the scoring function and re-ranks the list of hypotheses. By repeating this procedure several times, an improvement of the weights is reached resulting in a better ranking of the hypotheses. A benefit of this method is that no docking experiments have to be carried out since the adaptation and re-ranking is only applied to the scoring function. Additionally the docking results have been stored within a database. Therefore re-scoring hypotheses can be run by querying the database.

Up to now, ElMaR uses a fixed set of weights for scoring the hypotheses. One application of IPHEx is to adapt these weights for protein classes, e.g. proteins performing a similar reaction. Here, the weights from a feedback session are applied to proteins possessing the same EC number as the test case that has been re-ranked. The changes within the ranking are then evaluated.

The results of this approach show that improvements can be made when applying relevance feedback. Also the adaptation works for most enzyme classes within the data set.

In summary, the goals of this thesis – predicting the flexibility of amino acid side chains in order to improve the ElMaR protein–protein docking and improving the scoring of hypotheses by relevance feedback – are reached. The results have proven that both approaches are reasonable and that the obtained information improves the docking results.

8.2 Outlook

In this thesis, several techniques have been proposed to enhance the protein–protein docking but there is still room for extensions and further work. Large amounts of genomic data and even more data derived from this information have to be analysed and interpreted in the next years. In the field of protein docking high throughput methods are needed to process these amounts of data. The ElMaR system can be used for docking a large number of test cases fast. It discriminates good from bad predictions to some extend and the search space is reduced. The methods provided here improve the capabilities of ElMaR. In the following, further research topics related to this work are outlined.

Automatic Test Case Generation

Since the number of solved structures grows exponentially, protein docking algorithms can be tested and verified by larger data sets than in the past. But compiling test sets by hand will be infeasible when using large amounts of data. Thus, automatic methods have to be developed in order to generate new test cases. In this work, an approach has been developed to derive test cases automatically (Zöllner *et al.*, 2004). This approach can be extended by including other data sources containing information on possible test cases. Databases like BRENDA (Schomburg, 2003) or KEGG (Kanehisa & Goto, 2000) contain information about reactions and pathway where proteins are involved. Linking these information to the database driven approach, new test cases may be derived.

Flexibility Predictions

The prediction or calculation of flexible regions within the protein structure is essential for predicting near native hypotheses of a complex. In this thesis, the flexibility of residue side chains was modelled using a wavelet decomposition of an energy landscape. Here, Daubechies filters were taken and thresholding was applied to reduce the number of wavelet coefficients.

Recently, Cosic and coworkers (Trad *et al.*, 2002) applied wavelet decomposition for comparing protein sequences. They showed that features characterising a protein class can be

identified within the decomposed input signals. They even assigned certain features to specific detail levels of the decomposition. Thus, further analysis of the energy landscapes and wavelet filters may result in improved features that will lead to a more robust discrimination of flexible and non–flexible side chains.

Another direction of further investigation is to estimate the influence of the torsion side chain angles on each other. Knowledge about the interaction of the torsion angles can help while combining predictions of single torsion angles. Here, molecular dynamics simulation can be applied for simulating e.g. rearrangements of side chains from unfavourable conformations (cf. Torgasin, 2003). By analysing rotamer changes over the time maybe specific patterns can be extracted describing the internal movements of the side chain.

Also the influence of rotamer changes on the neighbouring groups can increase the prediction accuracy of side chain flexibility. If a model or rules can be extracted from interaction patterns of small groups of residues, the derived information can be incorporated (e.g. as feature) into the flexibility classification. First attempts towards this have been made by Torgasin (Torgasin, 2003). For a robust analysis at first a residue neighbourhood and measures for comparing several of them have to be defined.

Scoring of Hypotheses

The improvement of the scoring step of a docking system is very important because an imprecise discrimination of the docking hypotheses may result in pruning good hypotheses from the result set while propagating bad hypotheses to further processing steps. At the end no good hypotheses could be predicted although the system is capable to predict good hypotheses.

Energy based scoring methods are the most precise methods to describe molecules. In the field of protein–protein docking energy scoring function are used by several approaches (see section 3.3) but they are computationally expensive and very slow.

Implementing these scores in post processing modules has several advantages. On the one hand the speed of a docking system (like ElMaR) is not reduced. On the other hand only the best results have to be evaluated by this module reducing time requirements. Running several modules parallel one could apply different energy based scoring functions at once. Besides scoring the hypotheses by the Amber force field, also the free energy on binding could be estimated. It can be easily calculated using atomic contact energies (ACE, cf. Zhang *et al.*, 1997).

Neumann (Neumann, 2003) applied the scoring function of ElMaR to monomer and dimer structures in order to discriminate them. In his approach a monomer is identified as a false prediction whereas a dimer structure is taken as a correct prediction of a complex. In the classification approach hydrophobicity, charge and geometry are taken and a SVM is trained. Good results were reached on a test set provided by Ponstingl (Ponstingl *et al.*, 2000). This approach has been extended by using energy scores based on the Amber force field and ACE energies as additional features reaching similar results. The idea here is to

change the point of view: if monomer and dimer structures can be separated by the scoring function of ElMaR, also the hypotheses should be discriminated by this approach. So, time consuming scoring of docking hypotheses by energy based methods may be replaced by a classification approach. Since, support vector machines are utilised, the classification will be very fast even for large amounts of hypotheses. The time consuming energy calculations will be moved towards the training of the classifier. This can be performed offline.

Intelligent Navigation in Large Data Sets

The IPHEx system on the one hand provides methods for improving the scoring of ElMaR. The adaptation of weights is based on the similarity search between a reference structure and a set of hypotheses. Since the system uses the relevance feedback of human experts, the feedback is dependent of the user. In order to integrate more user independent feedback the IPHEx system can be extended to a multi user system. Therefore, appropriate methods for fusing the (possibly contrary) feedback of different users have to be developed.

Up to now, the similarity search within IPHEx is only used for re-scoring hypotheses. This search capability can be extended in order to search similar docking hypotheses across large data sets. This can be helpful in drug targeting and development. Today drug targets are identified by screening large libraries of structures. Attempts are made to simulate screening tasks (e.g. 1:N docking) by computers. These libraries can be queried easily by the similarity approach. By customising or extending the set of features (e.g. certain composition of the docking site) specific queries can be performed. Results can be investigated and refined similar to the scoring approach presented here. This can help in cutting down expensive wet lab experiments.

Appendix A
Test Sets

In this section the protein data used for the flexibility approaches and the docking experiments is shown. All data is taken from the PDB (Bhat *et al.*, 2001).

A.1 Unbound Protein Data Set

The data set given in the following tables has been used for training and evaluating the flexibility classification approaches. Beside the PDB identification code of the protein also the Swissprot identifier (Bairoch & Apweiler, 2000), the enzyme class[1], and a short description are given.

Protein	Swissprot Name	EC number	Description
132L	LYC_CHICK	3.2.1.17	HYDROLASE(O-GLYCOSYL)
193L	LYC_CHICK	3.2.1.17	HYDROLASE (O-GLYCOSYL)
194L	LYC_CHICK	3.2.1.17	HYDROLASE (O-GLYCOSYL)
1A3C	PYRR_BACSU	2.4.2.9	TRANSCRIPTION REGULATION
1AFU	RNP_BOVIN	3.1.27.5	HYDROLASE
1AJX	POL_HV1B1	3.4.23.16	ASPARTYL PROTEASE
1AKC	AATM_CHICK	2.6.1.1	TRANSFERASE(AMINOTRANSFERASE)
1AKI	LYC_CHICK	3.2.1.17	HYDROLASE
1AMQ	AAT_ECOLI	2.6.1.1	TRANSFERASE(AMINOTRANSFERASE)
1AMR	AAT_ECOLI	2.6.1.1	TRANSFERASE(AMINOTRANSFERASE)
1AQ7	TRY1_BOVIN	3.4.21.4	SERINE PROTEASE
1AQP	RNP_BOVIN	3.1.27.5	HYDROLASE (PHOSPHORIC DIESTER)
1ARS	AAT_ECOLI	2.6.1.1	TRANSFERASE(AMINOTRANSFERASE)
1ART	AAT_ECOLI	2.6.1.1	TRANSFERASE(AMINOTRANSFERASE)
1ASA	AAT_ECOLI	2.6.1.1	AMINOTRANSFERASE
1ASE	AAT_ECOLI	2.6.1.1	AMINOTRANSFERASE
1AUJ	TRY1_BOVIN	3.4.21.4	HYDROLASE
1AZF	LYC_CHICK	3.2.1.17	HYDROLASE

continued on the next page

[1] In case the protein structure is an enzyme, for other protein classes, e.g. antibodies no enzyme numbers are assigned.

Protein	Swissprot Name	EC number	Description
1B0D	LYC_CHICK	3.2.1.17	HYDROLASE
1B2L	ADH_DROLE	1.1.1.1	OXIDOREDUCTASE
1BEL	RNP_BOVIN	3.1.27.5	HYDROLASE
1BGI	LYC_CHICK	3.2.1.17	HYDROLASE
1BHZ	LYC_CHICK	3.2.1.17	HYDROLASE
1BJU	TRY1_BOVIN	3.4.21.4	SERINE PROTEASE
1BVX	LYC_CHICK	3.2.1.17	HYDROLASE
1BWI	LYC_CHICK	3.2.1.17	HYDROLASE
1BWJ	LYC_CHICK	3.2.1.17	HYDROLASE
1C2D	TRY1_BOVIN	3.4.21.4	HYDROLASE/HYDROLASE INHIBITOR
1C2E	TRY1_BOVIN	3.4.21.4	HYDROLASE/HYDROLASE INHIBITOR
1C3I	MM03_HUMAN	3.4.24.17	HYDROLASE
1CE5	TRY1_BOVIN	3.4.21.4	HYDROLASE
1CG0	PURA_ECOLI	6.3.4.4	LIGASE
1CGJ	CTRA_BOVIN	3.4.21.1	SERINE PROTEASE/INHIBITOR COMPLEX
1CHG	CTRA_BOVIN	3.4.21.1	HYDROLASE ZYMOGEN (SERINE PROTEINASE)
1CIB	PURA_ECOLI	6.3.4.4	LIGASE
1CQ6	AAT_ECOLI	2.6.1.1	TRANSFERASE
1CQ7	AAT_ECOLI	2.6.1.1	TRANSFERASE
1CQ8	AAT_ECOLI	2.6.1.1	TRANSFERASE
1CQR	MM03_HUMAN	3.4.24.17	HYDROLASE
1CSE	SUBT_BACLI	3.4.21.62	COMPLEX(SERINE PROTEINASE-INHIBITOR)
1D6O	FKB1_HUMAN	5.2.1.8	ISOMERASE
1D6R	TRY1_BOVIN	3.4.21.4	HYDROLASE
1D7H	FKB1_HUMAN	5.2.1.8	ISOMERASE
1D7I	FKB1_HUMAN	5.2.1.8	ISOMERASE
1D7X	MM03_HUMAN	3.4.24.17	HYDROLASE
1D8F	MM03_HUMAN	3.4.24.17	HYDROLASE
1D8M	MM03_HUMAN	3.4.24.17	HYDROLASE
1DFJ	RNP_BOVIN	3.1.27.5	COMPLEX (ENDONUCLEASE/INHIBITOR)
1DIF	POL_HV1BR	3.4.23.16	ASPARTIC PROTEINASE
1DPW	LYC_CHICK	3.2.1.17	HYDROLASE
1DPX	LYC_CHICK	3.2.1.17	HYDROLASE
1DY4	GUX1_TRIRE	3.2.1.91	HYDROLASE(O-GLYCOSYL)
1E8L	LYC_CHICK	3.2.1.17	HYDROLASE
1EOW	RNP_BOVIN	3.1.27.5	HYDROLASE
1EX3	CTRA_BOVIN	3.4.21.1	HYDROLASE
1F0V	RNP_BOVIN	3.1.27.5	HYDROLASE/DNA
1F0W	LYC_CHICK	3.2.1.17	HYDROLASE

continued on the next page

Protein	Swissprot Name	EC number	Description
1F10	LYC_CHICK	3.2.1.17	HYDROLASE
1F2S	TRY1_BOVIN	3.4.21.4	HYDROLASE/HYDROLASE INHIBITOR
1FAP	FKB1_HUMAN	5.2.1.8	COMPLEX (ISOMERASE/KINASE)
1FDL	LYC_CHICK	3.2.1.17	COMPLEX (ANTIBODY-ANTIGEN)
1FKB	FKB1_HUMAN	5.2.1.8	ISOMERASE
1FKD	FKB1_HUMAN	5.2.1.8	CIS-TRANS ISOMERASE
1FKF	FKB1_HUMAN	5.2.1.8	ISOMERASE
1FKG	FKB1_HUMAN	5.2.1.8	CIS-TRANS ISOMERASE
1FKH	FKB1_HUMAN	5.2.1.8	CIS-TRANS ISOMERASE
1FKJ	FKB1_HUMAN	5.2.1.8	ROTAMASE
1FKR	FKB1_HUMAN	5.2.1.8	CIS-TRANS ISOMERASE
1FKS	FKB1_HUMAN	5.2.1.8	CIS-TRANS ISOMERASE
1FKT	FKB1_HUMAN	5.2.1.8	CIS-TRANS ISOMERASE
1FQX	POL_HV1BR	3.4.23.16	HYDROLASE
1FXT	UBC1_YEAST	6.3.2.19	LIGASE
1G05	MM03_HUMAN	3.4.24.17	HYDROLASE
1G2K	POL_HV1B5	3.4.23.16	HYDROLASE
1G35	POL_HV1PV	3.4.23.16	HYDROLASE
1G36	TRY1_BOVIN	3.4.21.4	HYDROLASE
1G49	MM03_HUMAN	3.4.24.17	HYDROLASE
1G7H	LYC_CHICK	3.2.1.17	HYDROLASE INHIBITOR/HYDROLASE
1G7I	LYC_CHICK	3.2.1.17	HYDROLASE INHIBITOR/HYDROLASE
1G7J	LYC_CHICK	3.2.1.17	HYDROLASE INHIBITOR/HYDROLASE
1G7L	LYC_CHICK	3.2.1.17	HYDROLASE INHIBITOR/HYDROLASE
1G7M	LYC_CHICK	3.2.1.17	HYDROLASE INHIBITOR/HYDROLASE
1G9I	TRY1_BOVIN	3.4.21.4	HYDROLASE/HYDROLASE INHIBITOR
1GBT	TRY1_BOVIN	3.4.21.4	HYDROLASE(SERINE PROTEINASE)
1GCD	CTRA_BOVIN	3.4.21.1	HYDROLASE(SERINE PROTEINASE)
1GHL	LYC_PHACO	3.2.1.17	HYDROLASE(O-GLYCOSYL)
1GIN	PURA_ECOLI	6.3.4.4	LIGASE
1GNO	POL_HV1B1	3.4.23.16	HYDROLASE (ACID PROTEASE)
1GRC	PUR3_ECOLI	2.1.2.2	TRANSFERASE(FORMYL)
1HBV	POL_HV1B1	3.4.23.16	HYDROLASE (ACID PROTEASE)
1HEL	LYC_CHICK	3.2.1.17	HYDROLASE(O-GLYCOSYL)
1HEU	ADHE_HORSE	1.1.1.1	OXIDOREDUCTASE
1HEW	LYC_CHICK	3.2.1.17	HYDROLASE(O-GLYCOSYL)
1HF4	LYC_CHICK	3.2.1.17	HYDROLASE
1HIH	POL_HV1B1	3.4.23.16	HYDROLASE (ASPARTIC PROTEINASE)
1HOS	POL_HV1B1	3.4.23.16	HYDROLASE(ACID PROTEINASE)

continued on the next page

Protein	Swissprot Name	EC number	Description
1HPS	POL_HV1B1	3.4.23.16	HYDROLASE(ACID PROTEINASE)
1HPV	POL_HV1B5	3.4.23.16	HYDROLASE (ACID PROTEINASE)
1HPX	POL_HV1BR	3.4.23.16	HYDROLASE (ACID PROTEASE)
1HSG	POL_HV1BR	3.4.23.16	HYDROLASE (ACID PROTEINASE)
1HSW	LYC_CHICK	3.2.1.17	HYDROLASE
1HSX	LYC_CHICK	3.2.1.17	HYDROLASE
1HTE	POL_HV1B1	3.4.23.16	HYDROLASE(ACID PROTEINASE)
1HTF	POL_HV1B1	3.4.23.16	HYDROLASE(ACID PROTEINASE)
1HVI	POL_HV1B1	3.4.23.16	HYDROLASE(ACID PROTEASE)
1HVJ	POL_HV1B1	3.4.23.16	HYDROLASE(ACID PROTEASE)
1HVK	POL_HV1B1	3.4.23.16	HYDROLASE(ACID PROTEASE)
1HVL	POL_HV1B1	3.4.23.16	HYDROLASE(ACID PROTEASE)
1IC4	LYC_CHICK	3.2.1.17	PROTEIN BINDING/HYDROLASE
1IC5	LYC_CHICK	3.2.1.17	PROTEIN BINDING/HYDROLASE
1IC7	LYC_CHICK	3.2.1.17	PROTEIN BINDING/HYDROLASE
1JA2	LYC_CHICK	3.2.1.17	HYDROLASE
1JA4	LYC_CHICK	3.2.1.17	HYDROLASE
1JA6	LYC_CHICK	3.2.1.17	HYDROLASE
1JA7	LYC_CHICK	3.2.1.17	HYDROLASE
1JIR	TRY1_BOVIN	3.4.21.4	HYDROLASE
1JIS	LYC_CHICK	3.2.1.17	HYDROLASE
1JIT	LYC_CHICK	3.2.1.17	HYDROLASE
1JIY	LYC_CHICK	3.2.1.17	HYDROLASE
1JJ0	LYC_CHICK	3.2.1.17	HYDROLASE
1JJ1	LYC_CHICK	3.2.1.17	HYDROLASE
1JJ3	LYC_CHICK	3.2.1.17	HYDROLASE
1JPO	LYC_CHICK	3.2.1.17	HYDROLASE
1JRS	TRY1_BOVIN	3.4.21.4	HYDROLASE (SERINE PROTEASE)
1JRT	TRY1_BOVIN	3.4.21.4	HYDROLASE (SERINE PROTEASE)
1JUY	PURA_ECOLI	6.3.4.4	LIGASE
1K1I	TRY1_BOVIN	3.4.21.4	HYDROLASE
1K1J	TRY1_BOVIN	3.4.21.4	HYDROLASE
1K1L	TRY1_BOVIN	3.4.21.4	HYDROLASE
1K1M	TRY1_BOVIN	3.4.21.4	HYDROLASE
1K1N	TRY1_BOVIN	3.4.21.4	HYDROLASE
1K1O	TRY1_BOVIN	3.4.21.4	HYDROLASE
1K1P	TRY1_BOVIN	3.4.21.4	HYDROLASE
1KIP	LYC_CHICK	3.2.1.17	COMPLEX (IMMUNOGLOBULIN/HYDROLASE)
1KIQ	LYC_CHICK	3.2.1.17	COMPLEX (IMMUNOGLOBULIN/HYDROLASE)

continued on the next page

Protein	Swissprot Name	EC number	Description
1KIR	LYC_CHICK	3.2.1.17	COMPLEX (IMMUNOGLOBULIN/HYDROLASE)
1KSZ	PURA_ECOLI	6.3.4.4	LIGASE
1LCN	LYC_CHICK	3.2.1.17	HYDROLASE
1LKR	LYC_HUMAN	3.2.1.17	HYDROLASE
1LMA	LYC_CHICK	3.2.1.17	HYDROLASE(O-GLYCOSYL)
1LPI	LYC_CHICK	3.2.1.17	HYDROLASE
1LSA	LYC_CHICK	3.2.1.17	HYDROLASE(O-GLYCOSYL)
1LSB	LYC_CHICK	3.2.1.17	HYDROLASE(O-GLYCOSYL)
1LSC	LYC_CHICK	3.2.1.17	HYDROLASE(O-GLYCOSYL)
1LSD	LYC_CHICK	3.2.1.17	HYDROLASE(O-GLYCOSYL)
1LSE	LYC_CHICK	3.2.1.17	HYDROLASE(O-GLYCOSYL)
1LSF	LYC_CHICK	3.2.1.17	HYDROLASE(O-GLYCOSYL)
1LYZ	LYC_CHICK	3.2.1.17	HYDROLASE (O-GLYCOSYL)
1LZ8	LYC_CHICK	3.2.1.17	HYDROLASE
1LZ9	LYC_CHICK	3.2.1.17	HYDROLASE
1LZB	LYC_CHICK	3.2.1.17	HYDROLASE (O-GLYCOSYL)
1LZC	LYC_CHICK	3.2.1.17	HYDROLASE (O-GLYCOSYL)
1LZH	LYC_CHICK	3.2.1.17	HYDROLASE (O-GLYCOSYL)
1LZT	LYC_CHICK	3.2.1.17	HYDROLASE(O-GLYCOSYL)
1MAY	TRY1_BOVIN	3.4.21.4	HYDROLASE (SERINE PROTEASE)
1MEL	LYC_CHICK	3.2.1.17	COMPLEX (ANTIBODY/ANTIGEN)
1MTS	TRY1_BOVIN	3.4.21.4	SERINE PROTEINASE
1MTU	TRY1_BOVIN	3.4.21.4	SERINE PROTEASE
1MTV	TRY1_BOVIN	3.4.21.4	SERINE PROTEASE
1MTW	TRY1_BOVIN	3.4.21.4	SERINE PROTEASE
1NSG	FKB1_HUMAN	5.2.1.8	COMPLEX (ISOMERASE/KINASE)
1PPE	TRY1_BOVIN	3.4.21.4	HYDROLASE(SERINE PROTEINASE)
1PPH	TRY1_BOVIN	3.4.21.4	HYDROLASE(SERINE PROTEINASE)
1QA0	TRY1_BOVIN	3.4.21.4	HYDROLASE
1QB1	TRY1_BOVIN	3.4.21.4	HYDROLASE
1QB6	TRY1_BOVIN	3.4.21.4	HYDROLASE
1QB9	TRY1_BOVIN	3.4.21.4	HYDROLASE
1QBN	TRY1_BOVIN	3.4.21.4	HYDROLASE
1QBO	TRY1_BOVIN	3.4.21.4	HYDROLASE
1QCP	TRY1_BOVIN	3.4.21.4	HYDROLASE
1QF4	PURA_ECOLI	6.3.4.4	LIGASE
1QF5	PURA_ECOLI	6.3.4.4	LIGASE
1QIO	LYC_CHICK	3.2.1.17	HYDROLASE
1QL7	TRY1_BOVIN	3.4.21.4	SERINE PROTEASE

continued on the next page

Protein	Swissprot Name	EC number	Description
1QL8	TRY1_BOVIN	3.4.21.4	SERINE PROTEASE
1QPF	FKB1_HUMAN	5.2.1.8	ISOMERASE
1QPL	FKB1_HUMAN	5.2.1.8	ISOMERASE
1QTK	LYC_CHICK	3.2.1.17	HYDROLASE
1RAT	RNP_BOVIN	3.1.27.5	HYDROLASE (NUCLEIC ACID,RNA)
1RBB	RNP_BOVIN	3.1.27.5	HYDROLASE (NUCLEIC ACID, RNA)
1RBN	RNP_BOVIN	3.1.27.5	HYDROLASE(NUCLEIC ACID,RNA)
1RHA	RNP_BOVIN	3.1.27.5	HYDROLASE (NUCLEIC ACID,RNA)

Table A.1: *Test set of unbound proteins used for energy based classification of residue flexibility*

Protein	Swissprot Name	Description
1A2P	RNBR_BACAM	RIBONUCLEASE
1AAP	A4_HUMAN	PROTEINASE INHIBITOR (TRYPSIN)
1AH6	HS82_YEAST	CHAPERONE
1AVU	ITRA_SOYBN	SERINE PROTEASE INHIBITOR
1B0C	BPT1_BOVIN	HYDROLASE INHIBITOR
1B8L	PRVB_CYPCA	CALCIUM BINDING PROTEIN
1BGD	CSF3_CANFA	CYTOKINE
1BI0	DTXR_CORDI	REPRESSOR
1BI1	DTXR_CORDI	REPRESSOR
1BPI	BPT1_BOVIN	PROTEINASE INHIBITOR (TRYPSIN)
1C57	CONA_CANEN	SUGAR BINDING PROTEIN
1C6R	CYC6_SCEOB	ELECTRON TRANSPORT
1C76	SAK_STAAM	HYDROLASE
1C78	SAK_STAAM	HYDROLASE
1C79	SAK_STAAM	HYDROLASE
1CJP	CONA_CANEN	LECTIN
1COF	COFI_YEAST	ACTIN-BINDING PROTEIN
1CON	CONA_CANEN	LECTIN(AGGLUTININ)
1CSE	ICIC_HIRME	COMPLEX(SERINE PROTEINASE-INHIBITOR)
1D0D	BPT1_BOVIN	BLOOD CLOTTING INHIBITOR
1D3Z	UBIQ_HUMAN	HYDROLASE
1D6R	IBB1_SOYBN	HYDROLASE
1DFJ	RINI_PIG	COMPLEX (ENDONUCLEASE/INHIBITOR)
1DPR	DTXR_CORDI	TRANSCRIPTION REGULATION
1DQ0	CONA_CANEN	SUGAR BINDING PROTEIN
1DQ1	CONA_CANEN	SUGAR BINDING PROTEIN
1DQ5	CONA_CANEN	SUGAR BINDING PROTEIN
1E65	AZUR_PSEAE	ELECTRON TRANSPORT(COPPER BINDING)
1E67	AZUR_PSEAE	ELECTRON TRANSPORT(COPPER BINDING)

continued on the next page

Protein	Swissprot Name	Description
1E7F	ALBU_HUMAN	PLASMA PROTEIN
1F2S	ITR2_MOMCH	HYDROLASE/HYDROLASE INHIBITOR
1F9J	UBIQ_HUMAN	CHAPERONE
1FAP	FRAP_HUMAN	COMPLEX (ISOMERASE/KINASE)
1FXT	UBIQ_HUMAN	LIGASE
1G6J	UBIQ_HUMAN	GENE REGULATION, CELL CYCLE
1G7H	HV44_MOUSE	HYDROLASE INHIBITOR/HYDROLASE
1G7I	HV44_MOUSE	HYDROLASE INHIBITOR/HYDROLASE
1G7J	HV44_MOUSE	HYDROLASE INHIBITOR/HYDROLASE
1G7M	HV44_MOUSE	HYDROLASE INHIBITOR/HYDROLASE
1G9I	IBB_PHAAU	HYDROLASE/HYDROLASE INHIBITOR
1GIC	CONA_CANEN	LECTIN
1HA2	ALBU_HUMAN	SERUM PROTEIN
1HPT	IPK1_HUMAN	SERINE PROTEASE INHIBITOR
1I3H	CONA_CANEN	SUGAR BINDING PROTEIN
1JW6	CONA_CANEN	SUGAR BINDING PROTEIN
1JZE	AZUR_PSEAE	ELECTRON TRANSPORT
1JZF	AZUR_PSEAE	ELECTRON TRANSPORT
1JZG	AZUR_PSEAE	ELECTRON TRANSPORT
1JZH	AZUR_PSEAE	ELECTRON TRANSPORT
1JZI	AZUR_PSEAE	ELECTRON TRANSPORT
1KPA	HNT1_HUMAN	PROTEIN KINASE C INTERACTING PROTEIN
1KPB	HNT1_HUMAN	PROTEIN KINASE C INTERACTING PROTEIN
1KPC	HNT1_HUMAN	PROTEIN KINASE C INTERACTING PROTEIN
1KPE	HNT1_HUMAN	PROTEIN KINASE INHIBITOR
1KPF	HNT1_HUMAN	PROTEIN KINASE INHIBITOR
1NLS	CONA_CANEN	AGGLUTININ
1NSG	FRAP_HUMAN	COMPLEX (ISOMERASE/KINASE)
1PPE	ITR1_CUCMA	HYDROLASE(SERINE PROTEINASE)
1QGL	CONA_CANEN	LECTIN (AGGLUTININ)
1QNY	CONA_CANEN	LECTIN
1QPV	COFI_YEAST	ACTIN-BINDING PROTEIN

Table A.2: *Test set of unbound proteins used for energy based classification of residue flexibility (proteins with no EC number assigned).*

A.2 Test Cases used for Docking Experiments

Here, the test cases are listed used for the evaluation of the flexibility information and for the enhancements of ElMaR. These test cases have been derived automatically by the methods described in section 7.1.1.

Complex	Unbound 1	Unbound 2	EC number of Complex
1A2W	1BEL	1RAT	3.1.27.5
1A2W	1AQP	1RAT	3.1.27.5
1A2W	1BEL	1RBN	3.1.27.5
1A2W	1BEL	1RHA	3.1.27.5
1A2W	1EOW	1RHA	3.1.27.5
1A2W	1RBN	1RHA	3.1.27.5
1A2W	1RAT	1RBN	3.1.27.5
1A7X	1FKB	1FKF	5.2.1.8
1A7X	1FKB	1FKJ	5.2.1.8
1A7X	1FKF	1FKJ	5.2.1.8
1A7X	1FKB	1FKG	5.2.1.8
1A7X	1FKG	1FKJ	5.2.1.8
1A7X	1FKH	1FKJ	5.2.1.8
1ADE	1CIB	1QF5	6.3.4.4
1ADI	1QF4	1QF5	6.3.4.4
1AFK	1BEL	1RAT	3.1.27.5
1AFK	1AQP	1BEL	3.1.27.5
1AFK	1BEL	1RHA	3.1.27.5
1AFK	1BEL	1RBN	3.1.27.5
1AFK	1BEL	1EOW	3.1.27.5
1AFK	1AQP	1RAT	3.1.27.5
1AFK	1EOW	1RAT	3.1.27.5
1AFK	1RBN	1RHA	3.1.27.5
1AFK	1RAT	1RHA	3.1.27.5
1AFL	1BEL	1RBN	3.1.27.5
1AFL	1AQP	1RHA	3.1.27.5
1AFL	1AQP	1BEL	3.1.27.5
1AFL	1AQP	1EOW	3.1.27.5
1AFL	1EOW	1RHA	3.1.27.5
1AFL	1RBN	1RHA	3.1.27.5
1AFU	1BEL	1RAT	3.1.27.5
1AFU	1BEL	1RHA	3.1.27.5
1AFU	1AQP	1EOW	3.1.27.5
1AFU	1BEL	1EOW	3.1.27.5
1AFU	1AQP	1RAT	3.1.27.5
1AFU	1AQP	1RHA	3.1.27.5
1AFU	1AQP	1BEL	3.1.27.5
1AFU	1EOW	1RHA	3.1.27.5
1AFU	1EOW	1RAT	3.1.27.5

continued on the next page

Complex	Unbound 1	Unbound 2	EC number of Complex
1AFU	1RAT	1RHA	3.1.27.5
1AO6	1E7F	1HA2	
1APN	1C57	1DQ1	
1APN	1C57	1DQ5	
1APN	1C57	1CON	
1APN	1DQ0	1NLS	
1APN	1CON	1DQ1	
1APN	1CON	1QNY	
1APN	1I3H	1NLS	
1APN	1DQ5	1I3H	
1APN	1DQ0	1DQ1	
1APN	1I3H	1QNY	
1APN	1CON	1DQ0	
1APN	1DQ1	1QNY	
1APN	1NLS	1QNY	
1ARG	1ARS	1ASA	2.6.1.1
1ARG	1AMQ	1AMR	2.6.1.1
1ARG	1AMR	1ART	2.6.1.1
1ARG	1AMR	1CQ8	2.6.1.1
1ARG	1AMR	1ASE	2.6.1.1
1ARG	1ARS	1CQ7	2.6.1.1
1ARG	1CQ7	1CQ8	2.6.1.1
1ARG	1ASE	1CQ8	2.6.1.1
1ASM	1AMQ	1AMR	2.6.1.1
1ASM	1AMQ	1CQ8	2.6.1.1
1ASM	1ASA	1ASE	2.6.1.1
1ASM	1ART	1ASE	2.6.1.1
1ASN	1AMR	1CQ8	2.6.1.1
1ASN	1ARS	1ART	2.6.1.1
1ASN	1ASA	1CQ8	2.6.1.1
1ASN	1ASE	1CQ8	2.6.1.1
1B2K	132L	193L	3.2.1.17
1B2K	132L	1B0D	3.2.1.17
1B2K	132L	1DPX	3.2.1.17
1B2K	132L	1LSE	3.2.1.17
1B2K	193L	1AKI	3.2.1.17
1B2K	193L	1B0D	3.2.1.17
1B2K	193L	1DPW	3.2.1.17
1B2K	193L	1HSX	3.2.1.17

continued on the next page

Complex	Unbound 1	Unbound 2	EC number of Complex
1B2K	193L	1JJ0	3.2.1.17
1B2K	193L	1LSD	3.2.1.17
1B2K	193L	1LZ9	3.2.1.17
1B2K	193L	1LZB	3.2.1.17
1B2K	194L	1AZF	3.2.1.17
1B2K	194L	1JIY	3.2.1.17
1B2K	194L	1LSE	3.2.1.17
1B2K	1AKI	1B0D	3.2.1.17
1B2K	1AKI	1BVX	3.2.1.17
1B2K	1AKI	1F10	3.2.1.17
1B2K	1AZF	1BVX	3.2.1.17
1B2K	1AZF	1BWJ	3.2.1.17
1B2K	1AZF	1JJ0	3.2.1.17
1B2K	1AZF	1QIO	3.2.1.17
1B2K	1B0D	1BVX	3.2.1.17
1B2K	1B0D	1BWJ	3.2.1.17
1B2K	1B0D	1DPX	3.2.1.17
1B2K	1B0D	1F0W	3.2.1.17
1B2K	1B0D	1HEL	3.2.1.17
1B2K	1B0D	1LZ9	3.2.1.17
1B2K	1B0D	1LZC	3.2.1.17
1B2K	193L	1BVX	3.2.1.17
1B2K	193L	1HEL	3.2.1.17
1B2K	193L	1JIT	3.2.1.17
1B2K	193L	1JJ1	3.2.1.17
1B2K	193L	1LSA	3.2.1.17
1B2K	193L	1QTK	3.2.1.17
1B2K	194L	1BWI	3.2.1.17
1B2K	193L	1JIY	3.2.1.17
1B2K	194L	1F0W	3.2.1.17
1B2K	194L	1BWJ	3.2.1.17
1B2K	194L	1HSX	3.2.1.17
1B2K	194L	1JIT	3.2.1.17
1B2K	194L	1LPI	3.2.1.17
1B2K	194L	1HEL	3.2.1.17
1B2K	194L	1LZ8	3.2.1.17
1B2K	194L	1QIO	3.2.1.17
1B2K	1AKI	1JIT	3.2.1.17
1B2K	1AKI	1LSD	3.2.1.17

continued on the next page

Complex	Unbound 1	Unbound 2	EC number of Complex
1B2K	1AKI	1LSF	3.2.1.17
1B2K	1AZF	1DPX	3.2.1.17
1B2K	1AZF	1DPW	3.2.1.17
1B2K	1AZF	1HEL	3.2.1.17
1B2K	1AZF	1JIS	3.2.1.17
1B2K	1AZF	1LPI	3.2.1.17
1B2K	1AZF	1LSD	3.2.1.17
1B2K	1AZF	1LZT	3.2.1.17
1B2K	1AZF	1QTK	3.2.1.17
1B2K	1B0D	1BGI	3.2.1.17
1B2K	1B0D	1DPW	3.2.1.17
1B2K	1B0D	1LPI	3.2.1.17
1B2K	193L	1F10	3.2.1.17
1B2K	193L	1LPI	3.2.1.17
1B2K	193L	1LSF	3.2.1.17
1B2K	193L	1QIO	3.2.1.17
1B2K	194L	1BVX	3.2.1.17
1B2K	194L	1JJ1	3.2.1.17
1B2K	1AKI	1JJ0	3.2.1.17
1B2K	1AKI	1LYZ	3.2.1.17
1B2K	1AKI	1LPI	3.2.1.17
1B2K	193L	1BWI	3.2.1.17
1B2K	193L	1JIS	3.2.1.17
1B2K	193L	1JPO	3.2.1.17
1B2K	193L	1LSE	3.2.1.17
1B2K	194L	1HSW	3.2.1.17
1B2K	194L	1LYZ	3.2.1.17
1B2K	194L	1LZB	3.2.1.17
1B2K	1AKI	1BWI	3.2.1.17
1B2K	1AKI	1LZT	3.2.1.17
1B2K	1AKI	1LZC	3.2.1.17
1B2K	132L	1AZF	3.2.1.17
1B2K	132L	1HEL	3.2.1.17
1B2K	132L	1HEW	3.2.1.17
1B2K	132L	1LSA	3.2.1.17
1B2K	132L	1BGI	3.2.1.17
1B2K	132L	1LZB	3.2.1.17
1B2K	132L	1QTK	3.2.1.17
1B2K	132L	1LSD	3.2.1.17

continued on the next page

Complex	Unbound 1	Unbound 2	EC number of Complex
1B2K	193L	1LYZ	3.2.1.17
1B2K	194L	1BGI	3.2.1.17
1B2K	1BGI	1BWI	3.2.1.17
1B2K	1BVX	1BWI	3.2.1.17
1B2K	1BWI	1LSE	3.2.1.17
1B2K	1BWJ	1LZ8	3.2.1.17
1B2K	1DPW	1LSB	3.2.1.17
1B2K	1DPX	1JIY	3.2.1.17
1B2K	1DPX	1LMA	3.2.1.17
1B2K	1F10	1LSF	3.2.1.17
1B2K	1F10	1LZ9	3.2.1.17
1B2K	1F10	1LZC	3.2.1.17
1B2K	1F10	1QIO	3.2.1.17
1B2K	1DPW	1DPX	3.2.1.17
1B2K	1DPW	1LZ9	3.2.1.17
1B2K	1HEL	1LZ9	3.2.1.17
1B2K	1HEL	1QIO	3.2.1.17
1B2K	1HEW	1LSA	3.2.1.17
1B2K	1HEW	1LSE	3.2.1.17
1B2K	1HEW	1LZ8	3.2.1.17
1B2K	1HEW	1LZB	3.2.1.17
1B2K	1LSA	1LSD	3.2.1.17
1B2K	1LSC	1LZ9	3.2.1.17
1B2K	1LSD	1LSE	3.2.1.17
1B2K	1LSD	1LZ9	3.2.1.17
1B2K	1LSF	1LZC	3.2.1.17
1B2K	1LZ8	1LZC	3.2.1.17
1BM0	1E7F	1HA2	
1CGI	1CHG	1HPT	3.4.21.1
1CGI	1GCD	1HPT	3.4.21.1
1LYS	132L	1LZ9	3.2.1.17
1LYS	132L	1LZC	3.2.1.17
1LYS	193L	1LMA	3.2.1.17
1LYS	193L	1LSC	3.2.1.17
1LYS	193L	1LZ9	3.2.1.17
1LYS	194L	1BVX	3.2.1.17
1LYS	194L	1LSB	3.2.1.17
1LYS	194L	1LSE	3.2.1.17
1LYS	1AKI	1HSX	3.2.1.17

continued on the next page

Complex	Unbound 1	Unbound 2	EC number of Complex
1LYS	1B0D	1LSD	3.2.1.17
1LYS	193L	1B0D	3.2.1.17
1LYS	193L	1BWI	3.2.1.17
1LYS	193L	194L	3.2.1.17
1LYS	193L	1JIY	3.2.1.17
1LYS	193L	1LZB	3.2.1.17
1LYS	193L	1QIO	3.2.1.17
1LYS	194L	1JJ1	3.2.1.17
1LYS	194L	1JIY	3.2.1.17
1LYS	194L	1QTK	3.2.1.17
1LYS	194L	1LSF	3.2.1.17
1LYS	1AKI	1BWJ	3.2.1.17
1LYS	1AKI	1F0W	3.2.1.17
1LYS	1AKI	1HEW	3.2.1.17
1LYS	1AKI	1F10	3.2.1.17
1LYS	1AKI	1LZ9	3.2.1.17
1LYS	193L	1AKI	3.2.1.17
1LYS	132L	1JIT	3.2.1.17
1LYS	132L	1LSE	3.2.1.17
1LYS	193L	1F10	3.2.1.17
1LYS	193L	1JIS	3.2.1.17
1LYS	193L	1JJ1	3.2.1.17
1LYS	193L	1LPI	3.2.1.17
1LYS	1BGI	1JJ0	3.2.1.17
1LYS	1BGI	1LSF	3.2.1.17
1LYS	1BVX	1BWI	3.2.1.17
1LYS	1BVX	1HEL	3.2.1.17
1LYS	1BVX	1JJ0	3.2.1.17
1LYS	1BWI	1LMA	3.2.1.17
1LYS	1BWI	1LSF	3.2.1.17
1LYS	1BWI	1LZ9	3.2.1.17
1LYS	1BWJ	1JIS	3.2.1.17
1LYS	1BWJ	1LSB	3.2.1.17
1LYS	1DPX	1HEW	3.2.1.17
1LYS	1DPX	1LSB	3.2.1.17
1LYS	1DPX	1LZ9	3.2.1.17
1LYS	1HEL	1HEW	3.2.1.17
1LYS	1HEL	1LZC	3.2.1.17
1LYS	1HEW	1LSA	3.2.1.17

continued on the next page

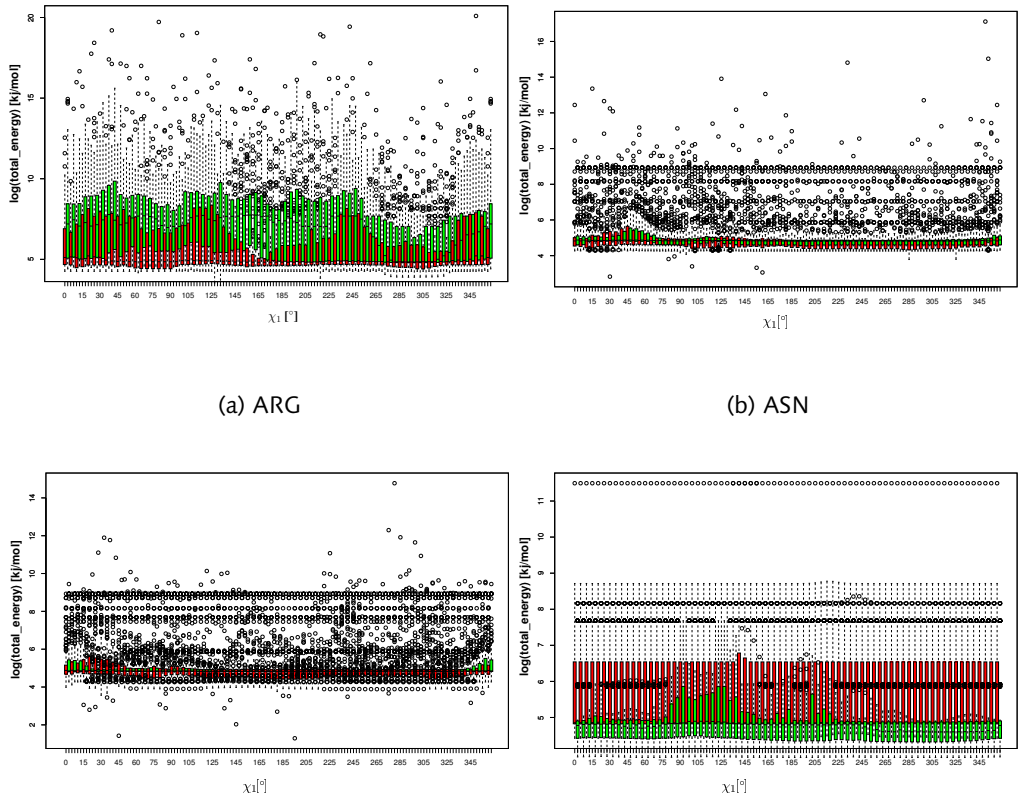
Complex	Unbound 1	Unbound 2	EC number of Complex
1LYS	1LSC	1QIO	3.2.1.17
1LYS	1LSD	1LZC	3.2.1.17
1LYS	1LSF	1LZC	3.2.1.17
1LYS	1LZ8	1LZT	3.2.1.17
1LYS	1LZC	1QIO	3.2.1.17
1LYS	1LZB	1QTK	3.2.1.17
1TPA	1BJU	1BPI	3.4.21.4
1TPA	1AUJ	1BPI	3.4.21.4
2ptc	1AQ7	1BPI	3.4.21.4
2ptc	1AUJ	1BPI	3.4.21.4
2ptc	1BJU	1BPI	3.4.21.4

Table A.3: *Test set of unbound proteins used for energy based classification of residue flexibility. Besides the* PDB *identifier of the complex and the two unbound partners, here also the enzyme number of the complex is given.For some test cases no enzyme number has been assigned or the proteins are no enzymes. Thus, for these no enzyme number is listed in the table.*

Appendix B
Supplementary Material

B.1 Boxplots of Energy Landscapes

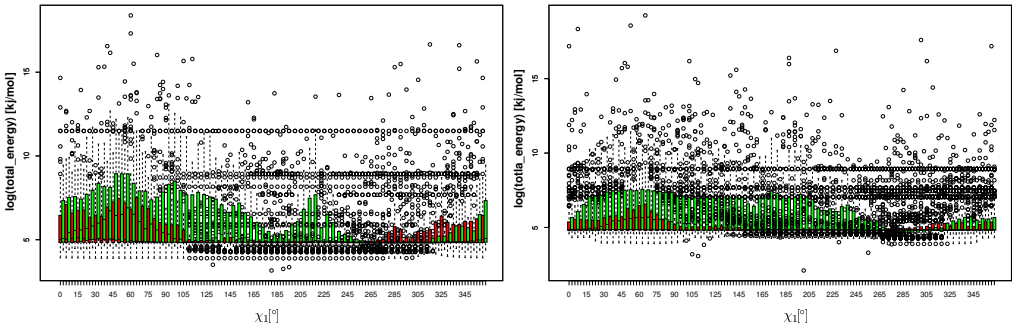
In this section box-plots drawn of the energy landscapes of all residues in the data set for χ_1 are shown. In red the flexible labelled residues, in green rigid residues are given.



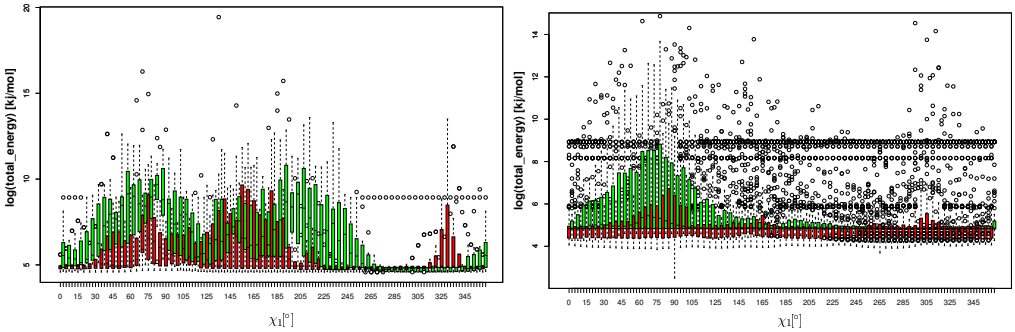
(a) ARG (b) ASN

(c) ASP (d) CYS

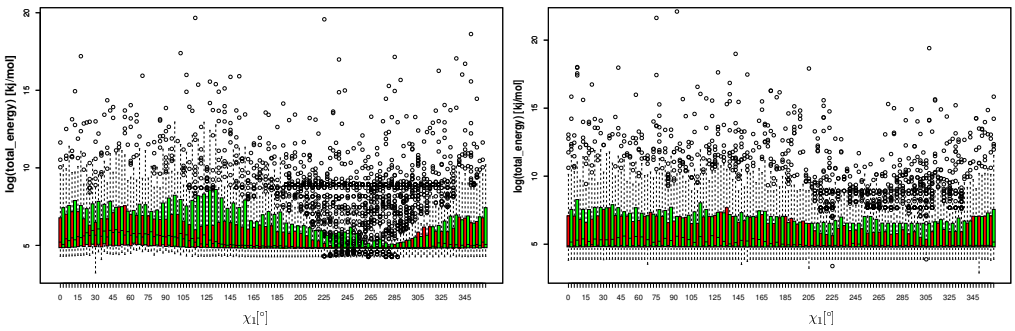
Figure B.1: *Box-plots of energy landscapes of* χ_1.



(e) GLN (f) GLU

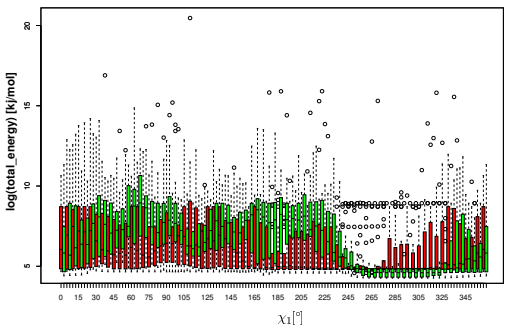


(g) HIS (h) ILE

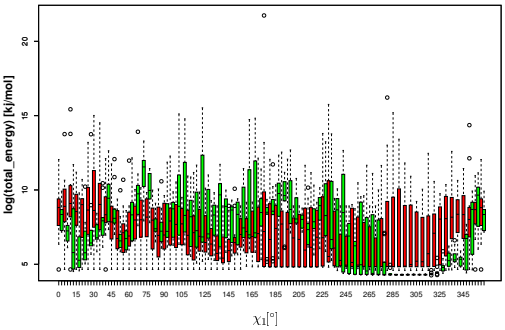


(i) LEU (j) LYS

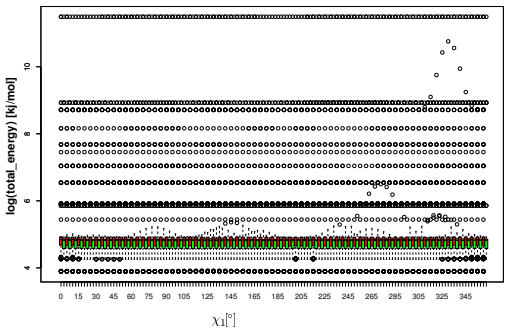
Figure B.1: *Box-plots of energy landscapes of* χ_1.



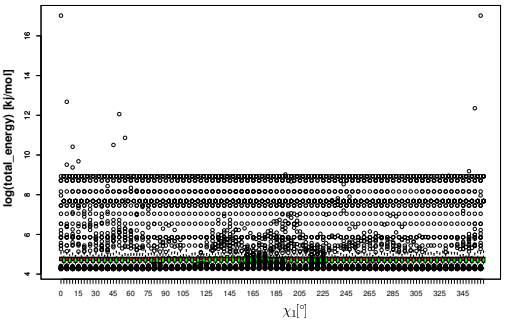
(k) MET



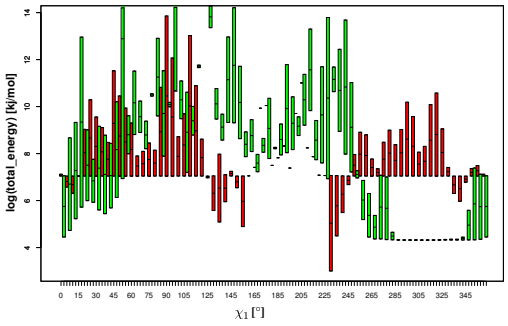
(l) PHE



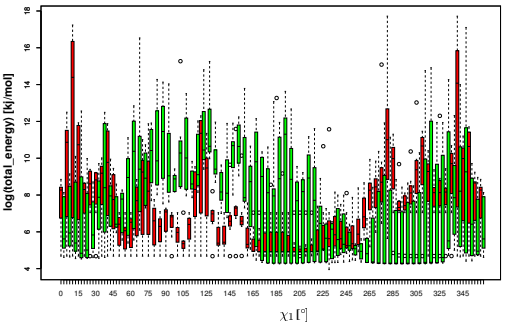
(m) SER



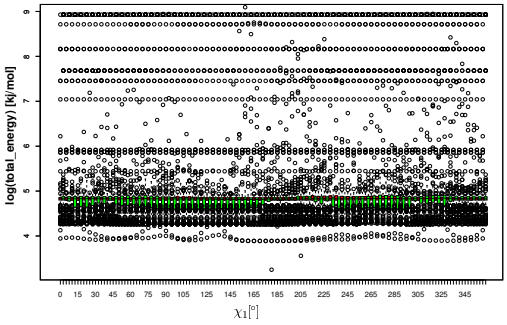
(n) THR



(o) TRP



(p) TYR



(q) VAL

Figure B.1: *Box-plots of energy landscapes of* χ_1.

B.2 Tables of the Normalisation Factor Analysis

In this section the results of the analysis of different normalisation factors (see section 5.2.3) is given. In the following ΔQ is defined as the inter-quantile distance $\Delta Q = Q_3 - Q_1$. The red highlighted numbers represent the optimal normalisation factor of an amino acid.

AS	ROC Area					
	$0.5 * \Delta Q$	$1.2 * \Delta Q$	$1.5 * \Delta Q$	$3 * \Delta Q$	$6 * \Delta Q$	$Q4$
ARG	0.74	0.72	0.72	0.70	0.67	0.5
ASN	0.69	0.68	0.67	0.64	0.64	0.5
ASP	0.64	0.62	0.61	0.58	0.57	0.5
CYS	0.70	0.70	0.70	0.71	0.75	0.5
GLN	0.72	0.69	0.69	0.66	0.65	0.5
GLU	0.60	0.57	0.59	0.55	0.56	0.5
HIS	0.74	0.73	0.69	0.66	0.65	0.5
ILE	0.62	0.62	0.62	0.62	0.61	0.5
LEU	0.65	0.64	0.62	0.62	0.61	0.5
LYS	0.66	0.64	0.63	0.60	0.59	0.5
MET	0.63	0.60	0.61	0.61	0.63	0.5
PHE	0.67	0.64	0.64	0.62	0.62	0.5
SER	0.54	0.50	0.49	0.47	0.47	0.5
THR	0.70	0.68	0.68	0.65	0.63	0.5
TRP	0.88	0.63	0.63	0.63	0.63	0.5
TYR	0.57	0.58	0.59	0.55	0.55	0.5
VAL	0.71	0.71	0.70	0.68	0.67	0.5

Table B.1: *ROC area of different amino acids for χ_1 and for different normalisation factors.*

AS	ROC Area					
	$0.5 * \Delta Q$	$1.2 * \Delta Q$	$1.5 * \Delta Q$	$3 * \Delta Q$	$6 * \Delta Q$	$Q4$
ARG	0.63	0.60	0.58	0.56	0.54	0.5
ASN	0.62	0.58	0.56	0.53	0.53	0.5
ASP	0.51	0.48	0.48	0.47	0.47	0.5
GLN	0.69	0.65	0.64	0.61	0.60	0.5
GLU	0.55	0.53	0.52	0.52	0.51	0.5
HIS	0.75	0.72	0.73	0.69	0.67	0.5
ILE	0.65	0.65	0.64	0.64	0.65	0.5
LEU	0.64	0.62	0.61	0.60	0.59	0.5
LYS	0.55	0.52	0.51	0.50	0.49	0.5
MET	0.51	0.49	0.49	0.48	0.46	0.5
PHE	0.58	0.57	0.56	0.56	0.54	0.5
TRP	0.72	0.70	0.69	0.69	0.68	0.5
TYR	0.63	0.62	0.62	0.61	0.61	0.5

Table B.2: *ROC area of different amino acids for χ_2 and for different normalisation factors.*

AS	ROC Area					
	$0.5 * \Delta Q$	$1.2 * \Delta Q$	$1.5 * \Delta Q$	$3 * \Delta Q$	$6 * \Delta Q$	$Q4$
ARG	0.64	0.62	0.61	0.57	0.57	0.5
GLN	0.55	0.52	0.51	0.49	0.50	0.5
GLU	0.54	0.51	0.50	0.49	0.48	0.5
LYS	0.51	0.50	0.48	0.47	0.46	0.5
MET	0.76	0.76	0.76	0.75	0.73	0.5

Table B.3: *ROC area of different amino acids for χ_3 and for different normalisation factors.*

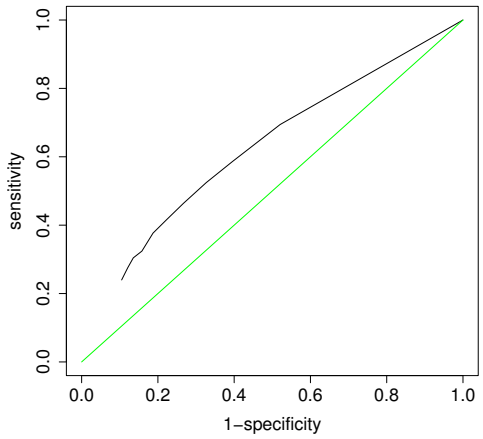
AS	ROC Area					
	$0.5 * \Delta Q$	$1.2 * \Delta Q$	$1.5 * \Delta Q$	$3 * \Delta Q$	$6 * \Delta Q$	$Q4$
ARG	0.61	0.57	0.55	0.53	0.51	0.5
LYS	0.49	0.48	0.47	0.47	0.47	0.5

Table B.4: *ROC area of different amino acids for χ_4 and for different normalisation factors.*

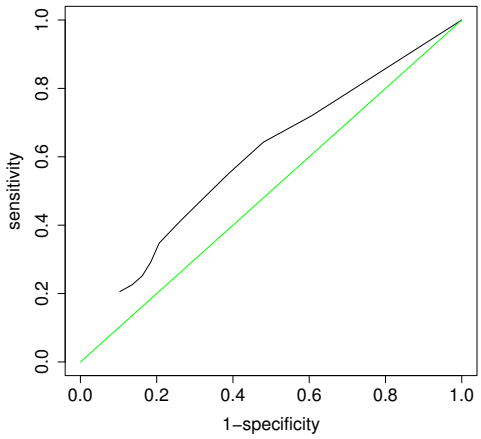
B.3 ROC–Plots

In this section the ROC curves of the different amino acids are given using the energy difference for classifying the flexibility. For each curve the 1-specificity is plotted against the sensitivity (see also section 7.2.1). The diagonal represents the chance line.

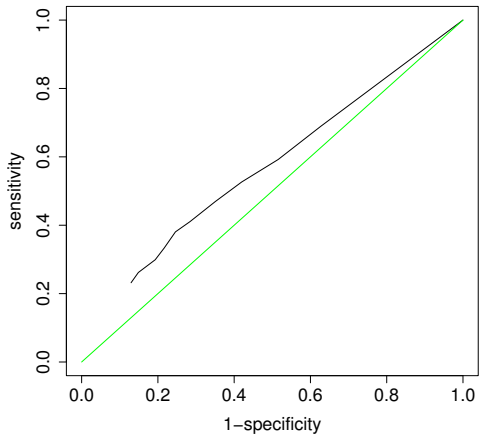
B.3.1 ROC curves for χ_1



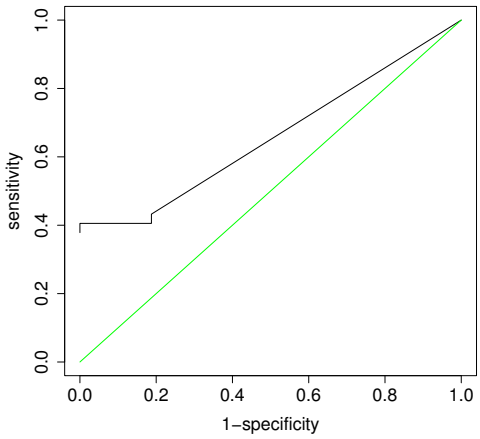
(a) ARG



(b) ASN

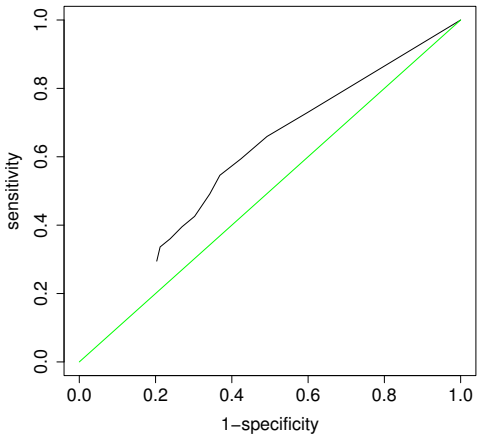


(c) ASP

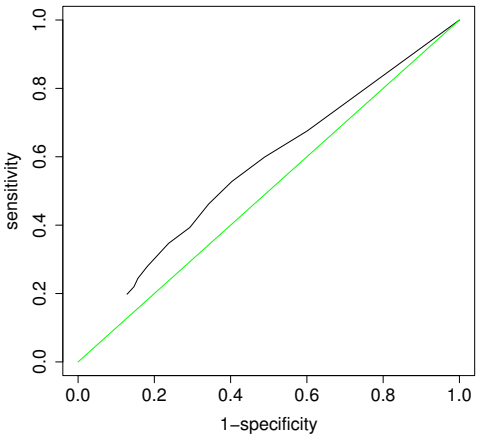


(d) CYS

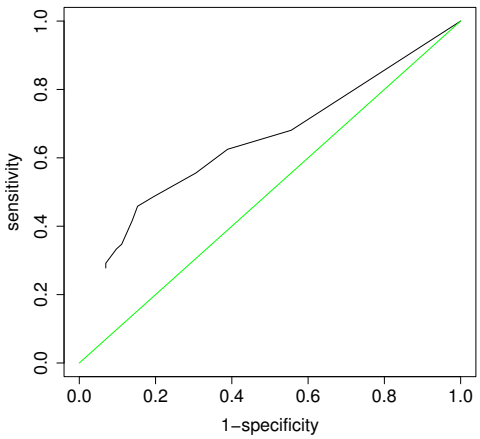
Figure B.2: *ROC curves for all residues and χ_1.*



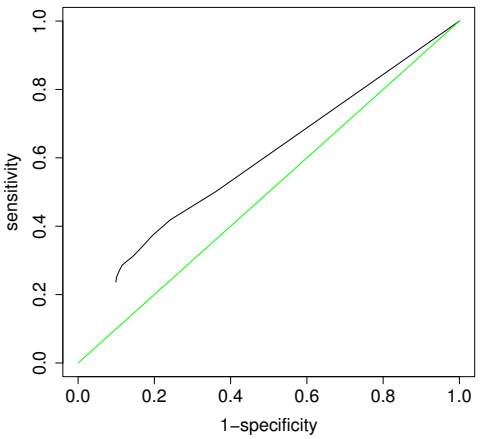
(e) GLN



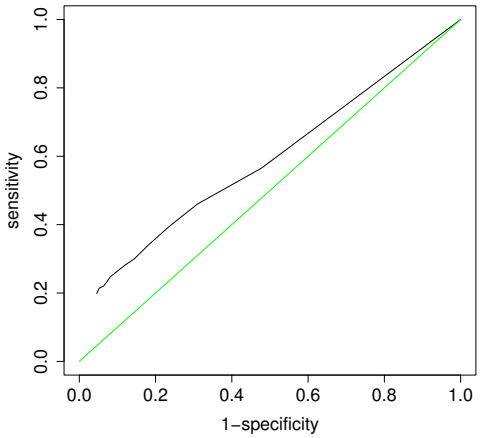
(f) GLU



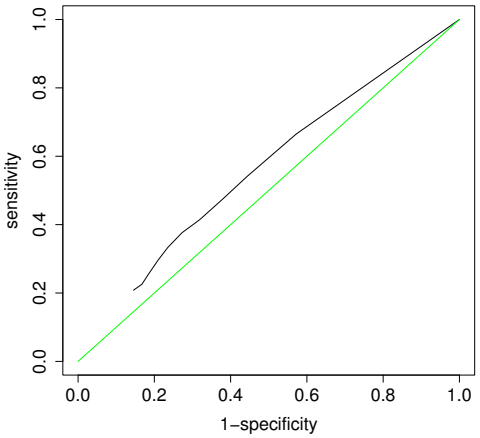
(g) HIS



(h) ILE

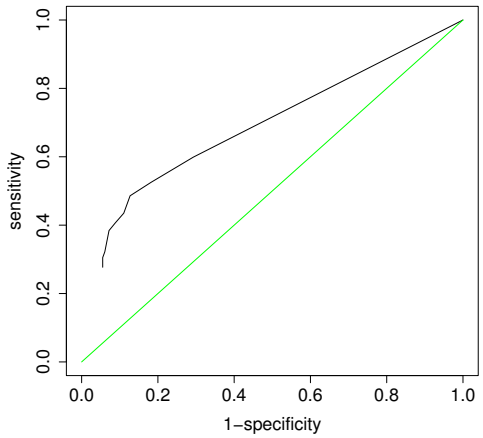


(i) LEU

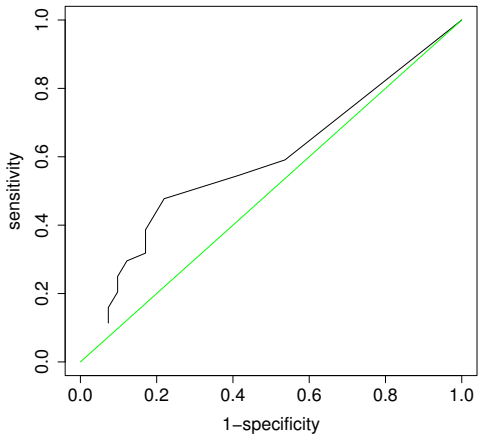


(j) LYS

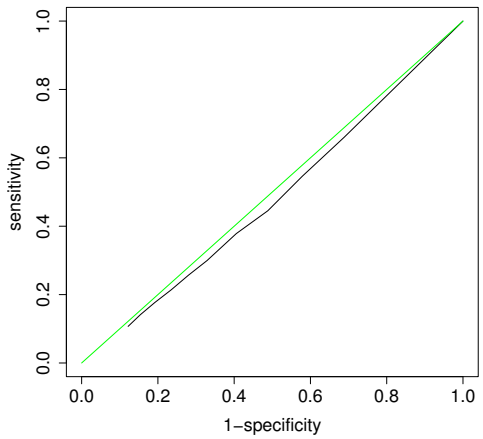
Figure B.2: *ROC curves for all residues and* χ_1.



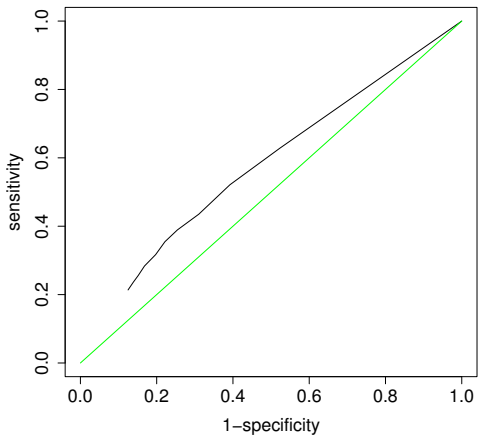
(k) MET



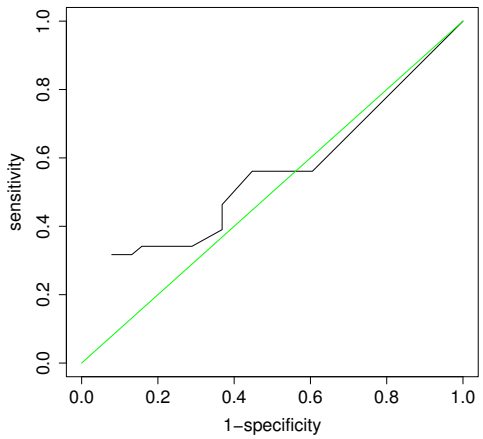
(l) PHE



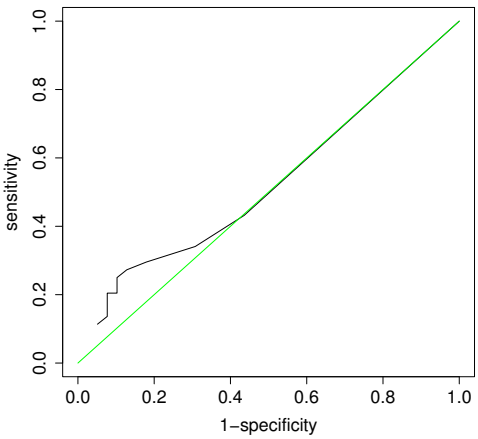
(m) SER



(n) THR

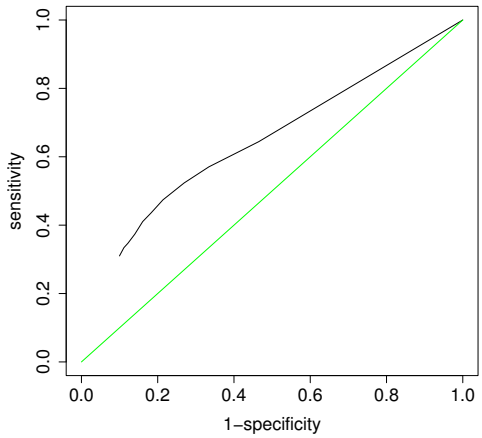


(o) TRP



(p) TYR

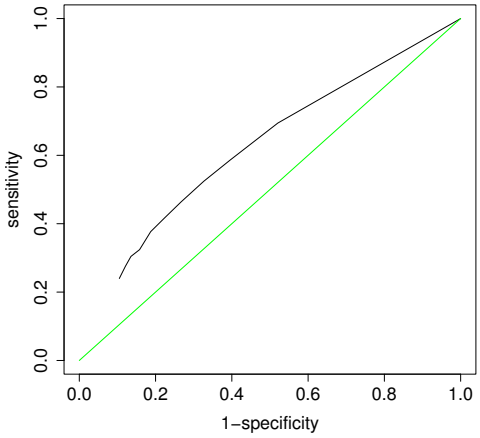
Figure B.2: *ROC curves for all residues and* χ_1.



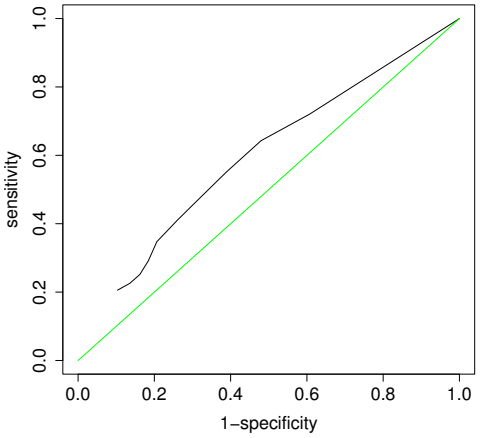
(q) VAL

Figure B.2: *ROC curves for all residues and* χ_1.

B.3.2 ROC curves for χ_2

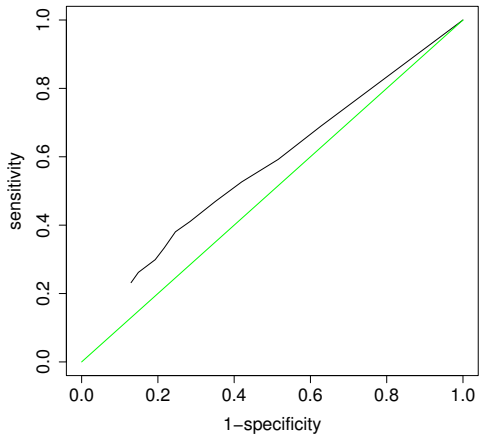


(a) ARG

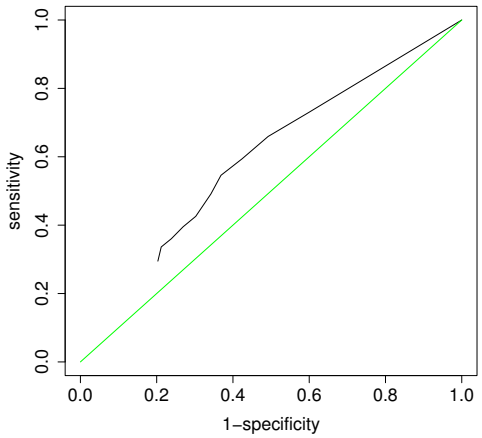


(b) ASN

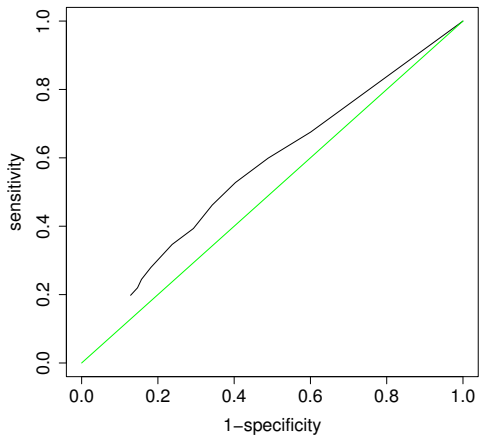
Figure B.3: *ROC curves for all residues and* χ_2.



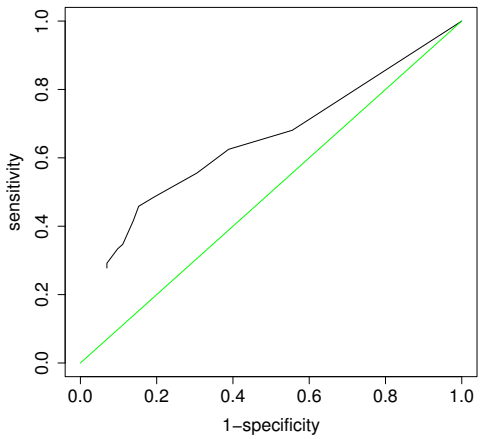
(c) ASP



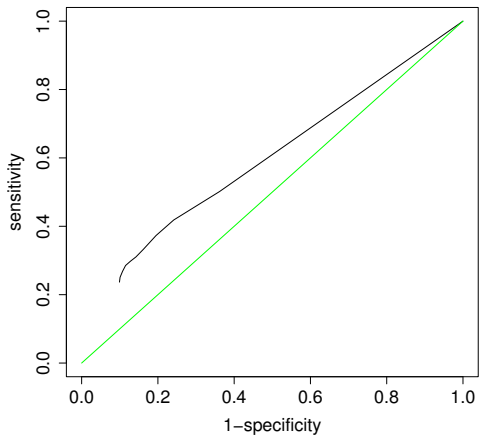
(d) GLN



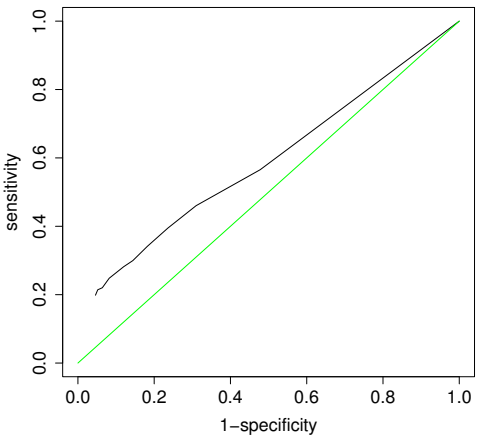
(e) GLU



(f) HIS

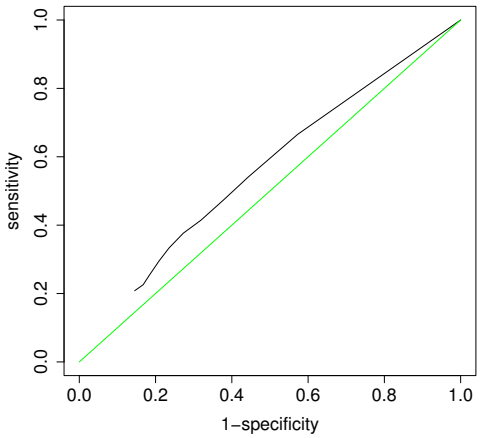


(g) ILE

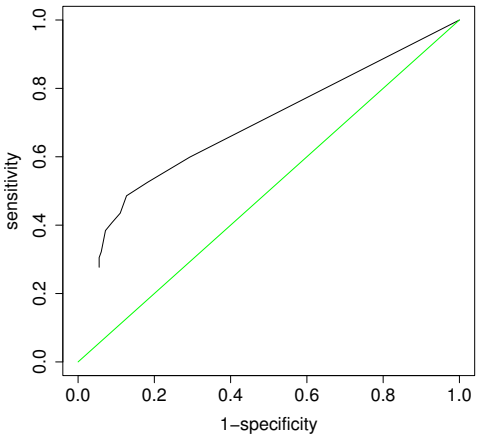


(h) LEU

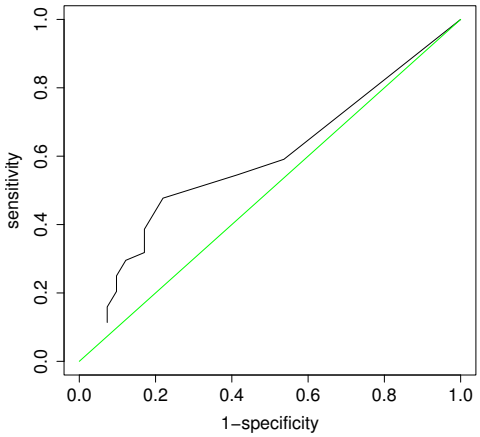
Figure B.3: *ROC curves for all residues and* χ_2.



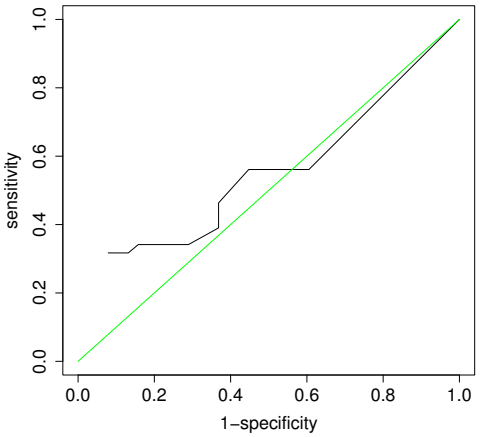
(i) LYS



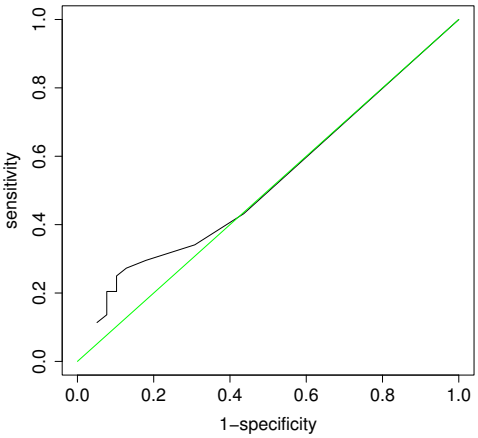
(j) MET



(k) PHE



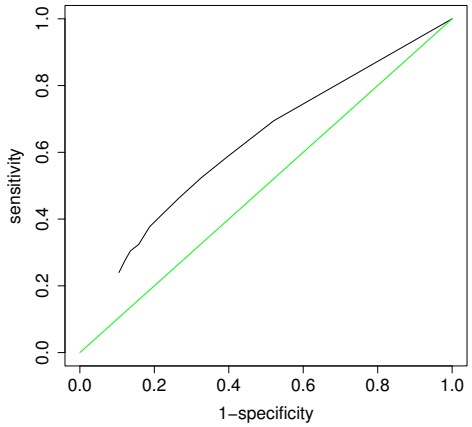
(l) TRP



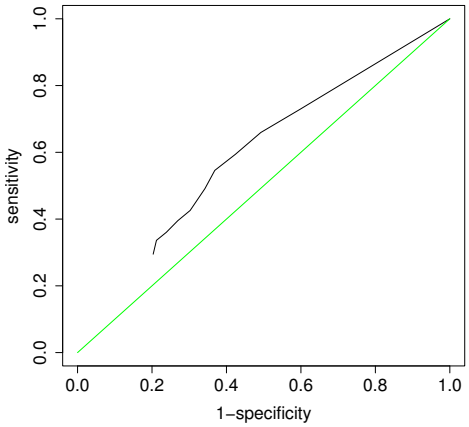
(m) TYR

Figure B.3: *ROC curves for all residues and* χ_2.

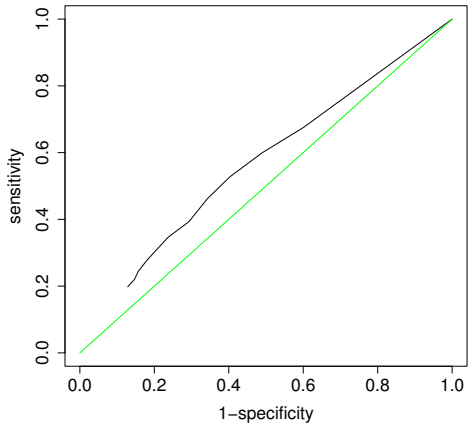
B.3.3 ROC curves for χ_3 and χ_4



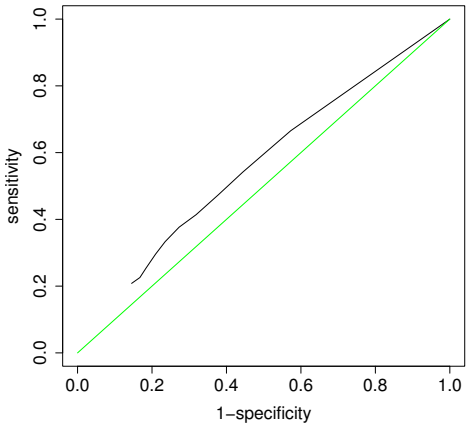
(a) ARG



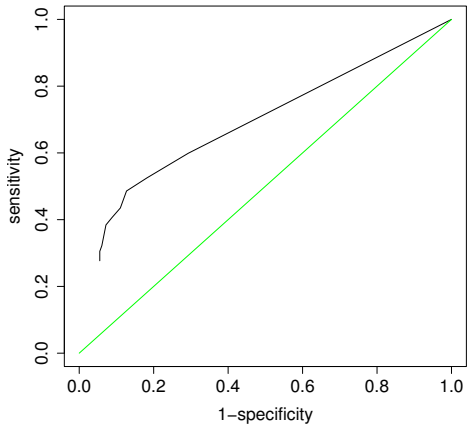
(b) GLN



(c) GLU

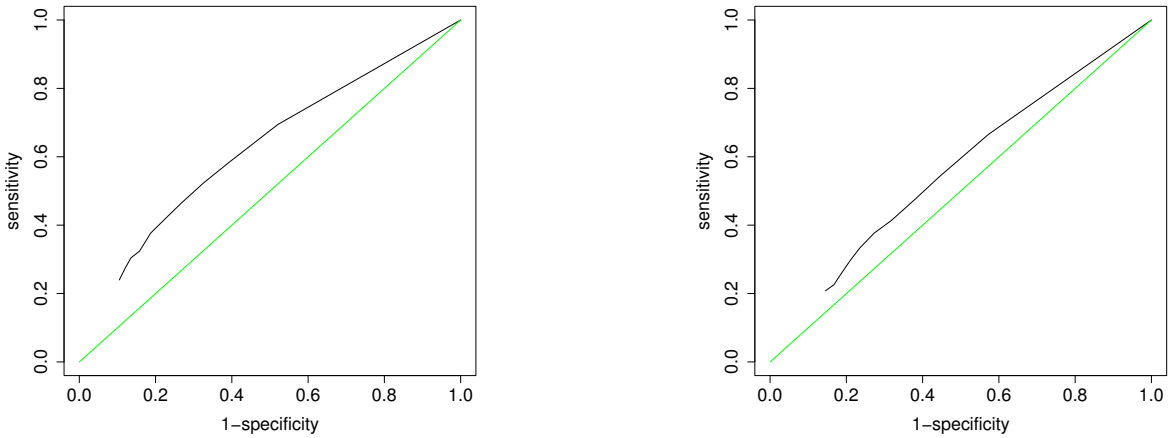


(d) LYS



(e) MET

Figure B.4: *ROC curves for all residues and χ_3.*



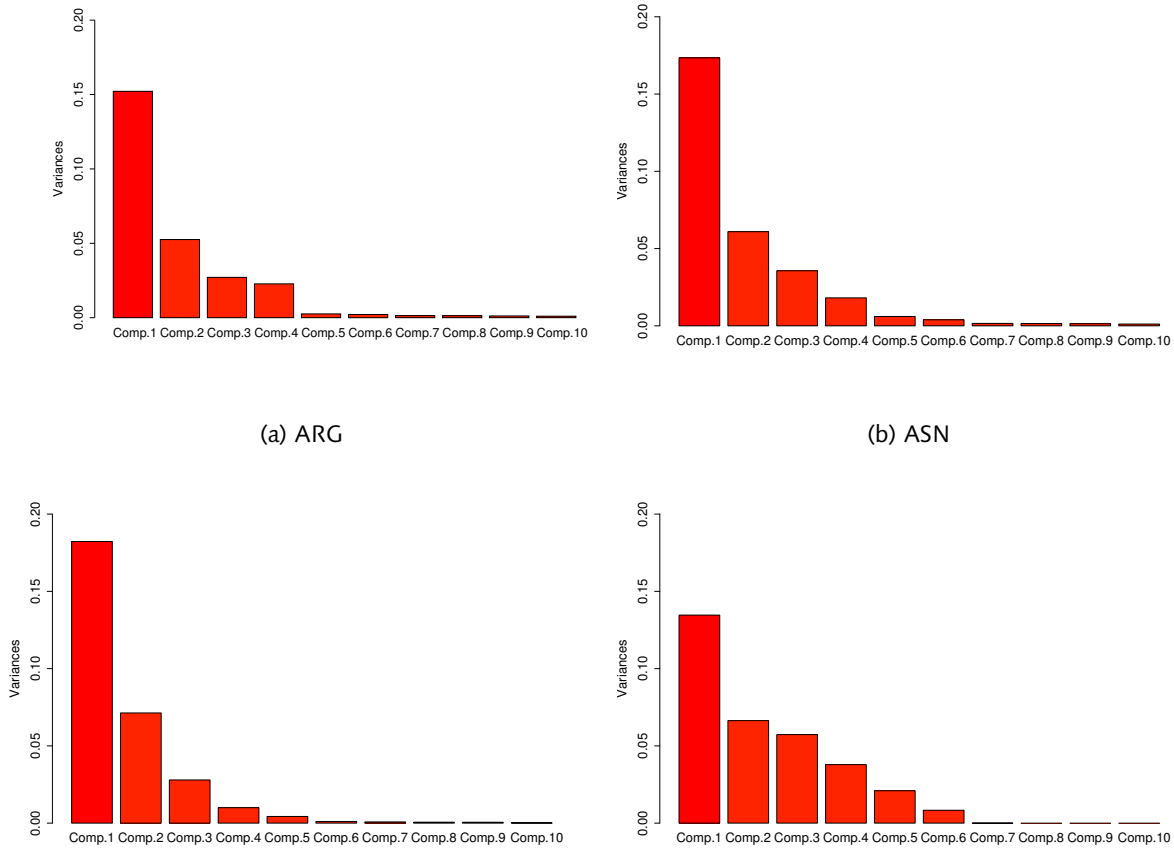
(a) ARG (b) LYS

Figure B.5: *ROC curves for all residues and* χ_4.

B.4 PCA Plots of Features

In this section, the different eigenvalue spectra of each residue type and torsion angle are given. As input, the features extracted from the energy landscapes as well as the other features like e.g. the SAS (cf. section 5.2.2) are taken.

B.4.1 Principle Component Analysis for χ_1



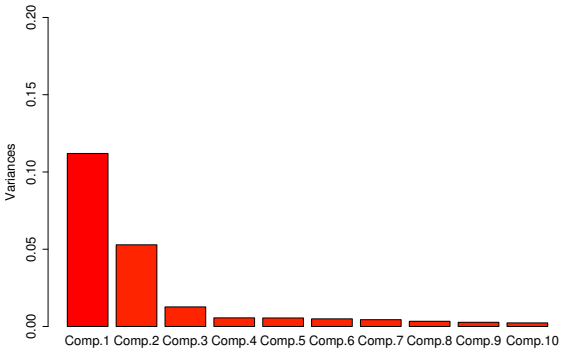
(a) ARG

(b) ASN

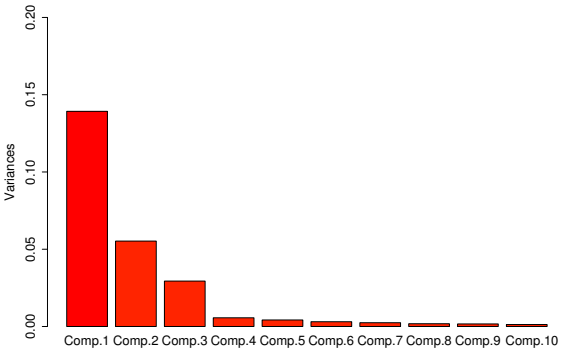
(c) ASP

(d) CYS

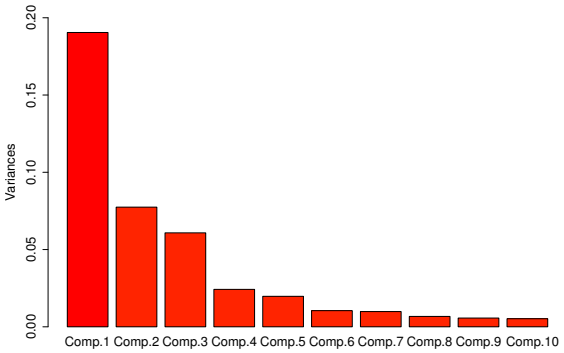
Figure B.6: *PCA Eigenvalue spectra for* χ_1.



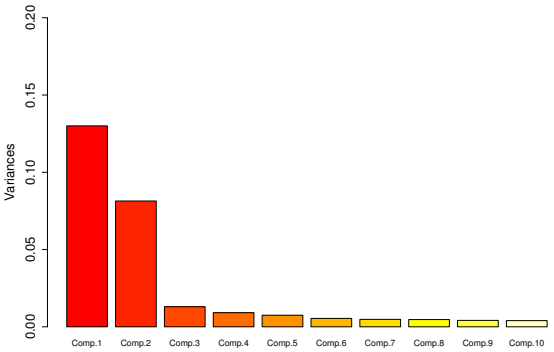
(e) GLN



(f) GLU

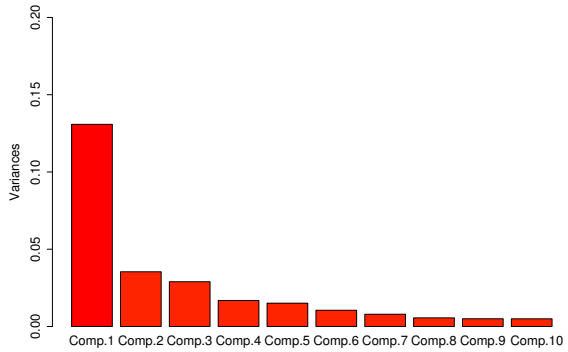


(g) HIS

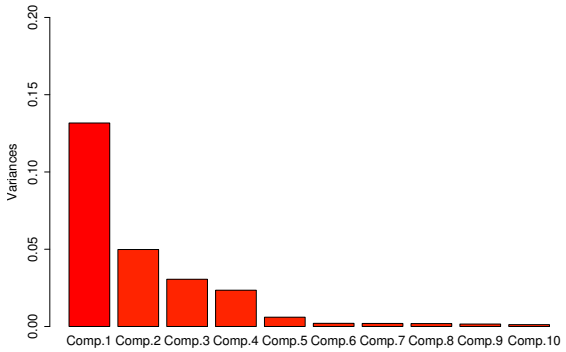


(h) ILE

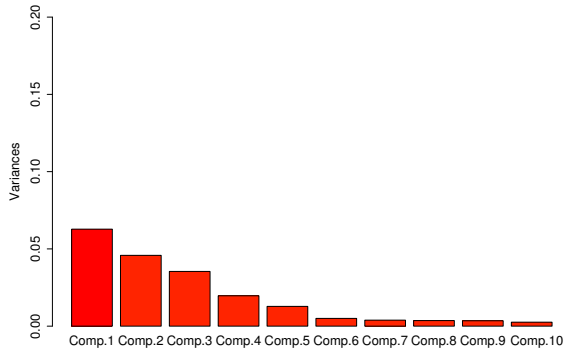
Figure B.6: *(cont.) PCA Eigenvalue spectra for* χ_1.



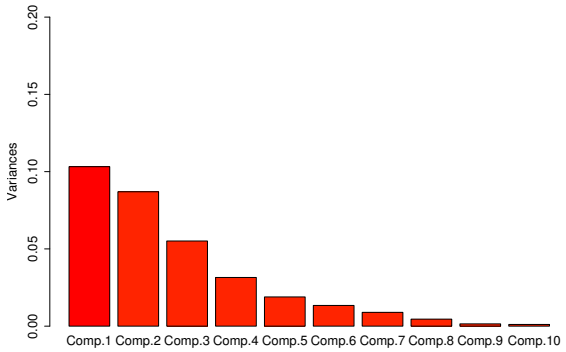
(i) LEU



(j) LYS

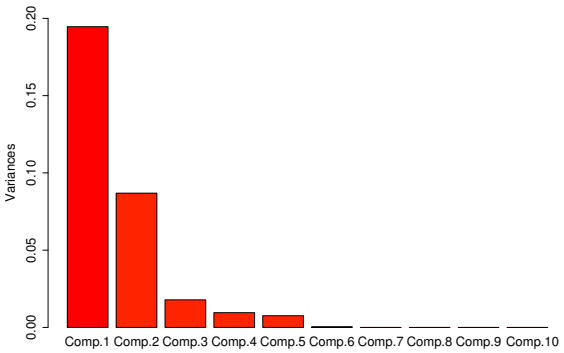


(k) MET

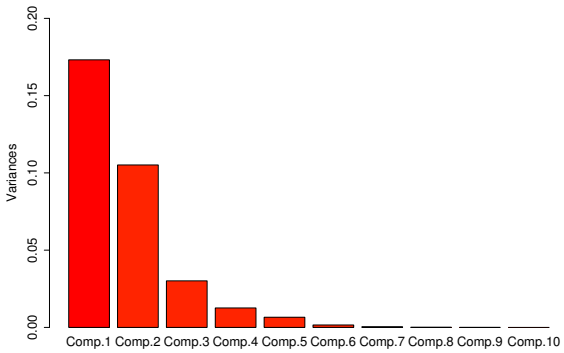


(l) PHE

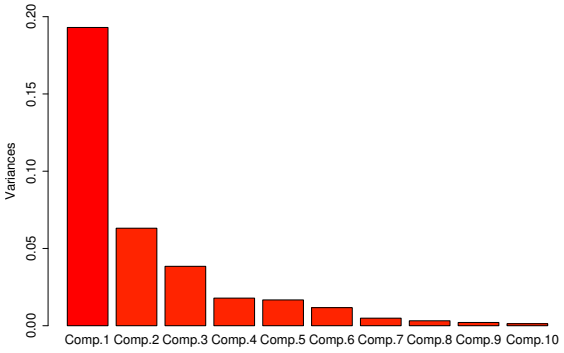
Figure B.6: *(cont.) PCA Eigenvalue spectra for* χ_1.



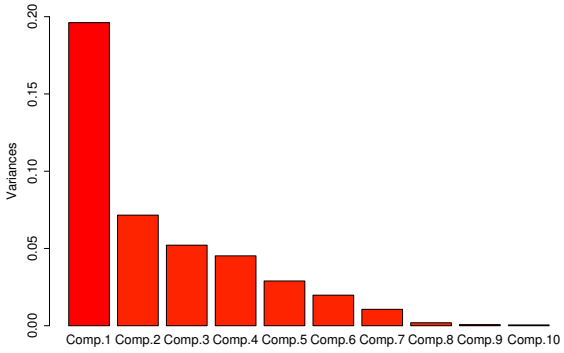
(m) SER



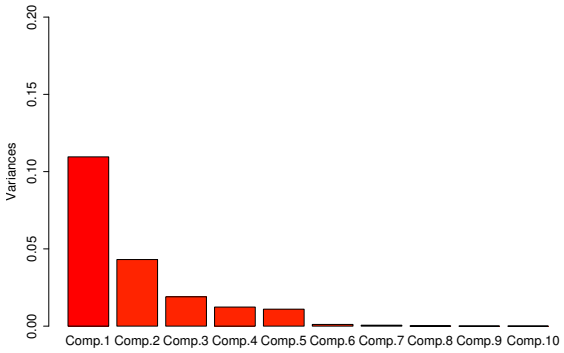
(n) THR



(o) TRP



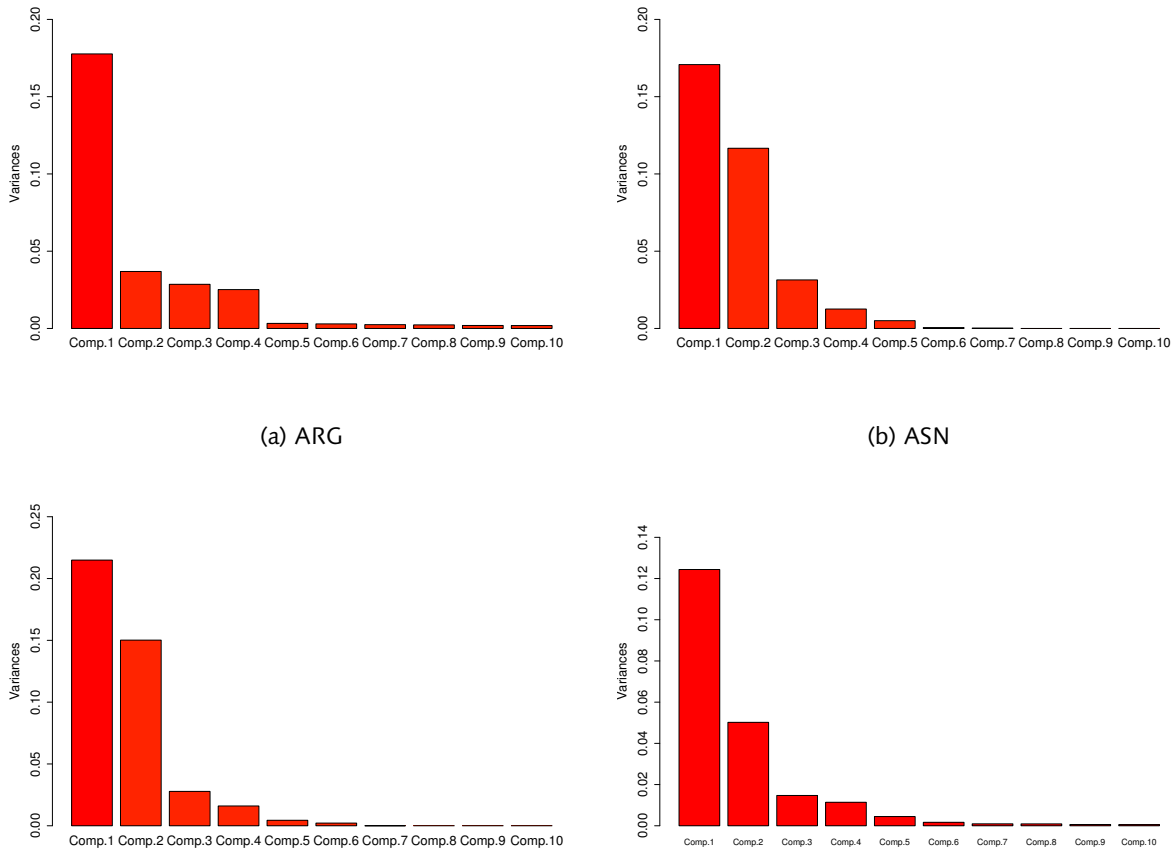
(p) TYR



(q) VAL

Figure B.6: *(cont.) PCA Eigenvalue spectra for* χ_1.

B.4.2 Principle Component Analysis for χ_2



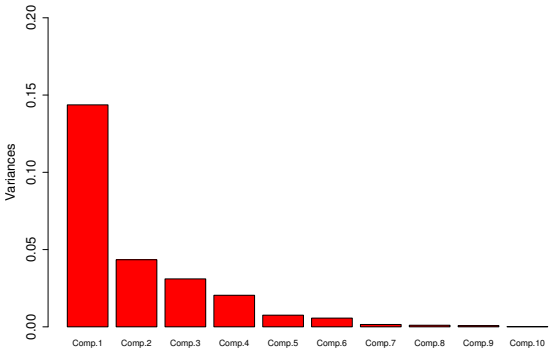
(a) ARG

(b) ASN

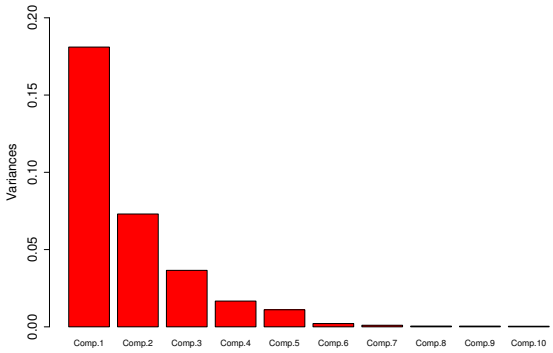
(c) ASP

(d) GLN

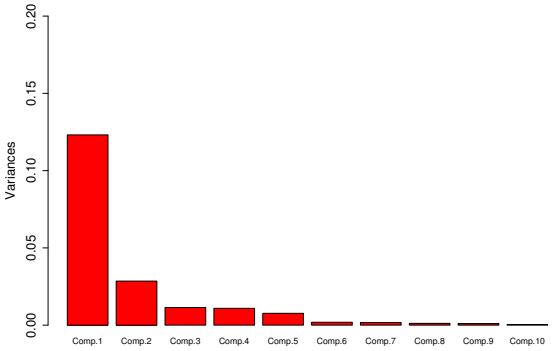
Figure B.7: *PCA Eigenvalue spectra for* χ_2.



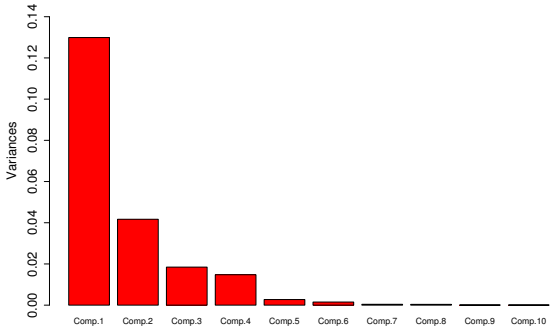
(e) GLU



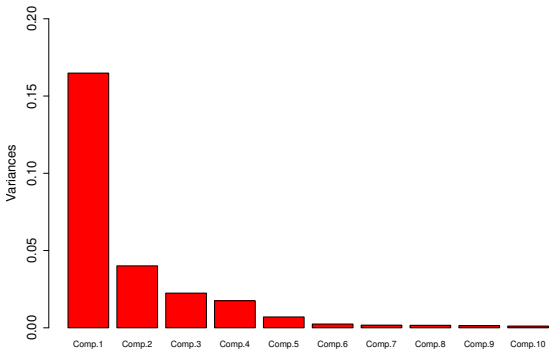
(f) HIS



(g) ILE

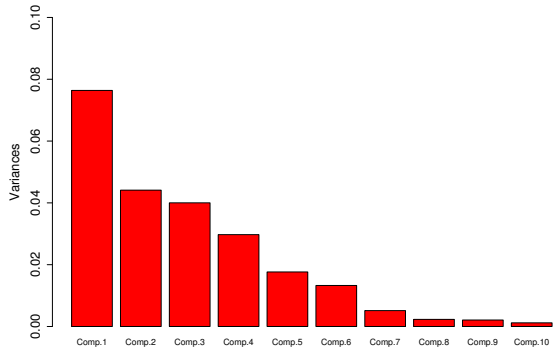


(h) LEU

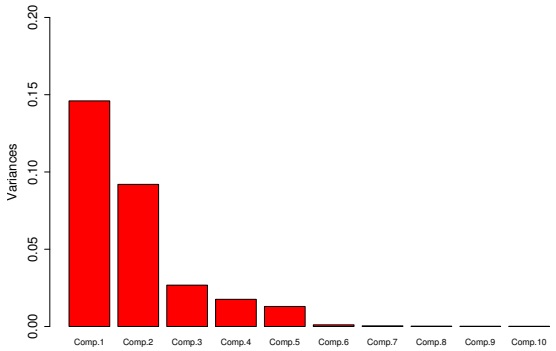


(i) LYS

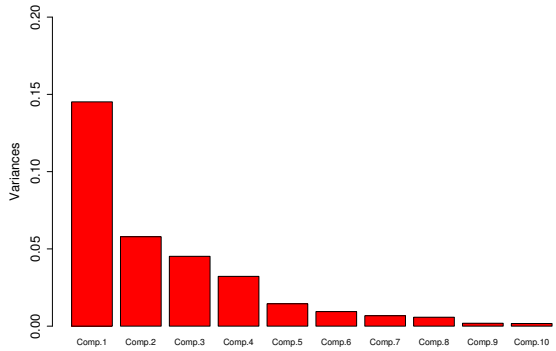
Figure B.7: *(cont.) PCA Eigenvalue spectra for* χ_2.



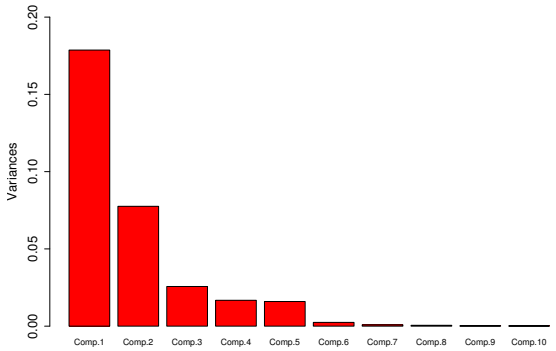
(j) MET



(k) PHE



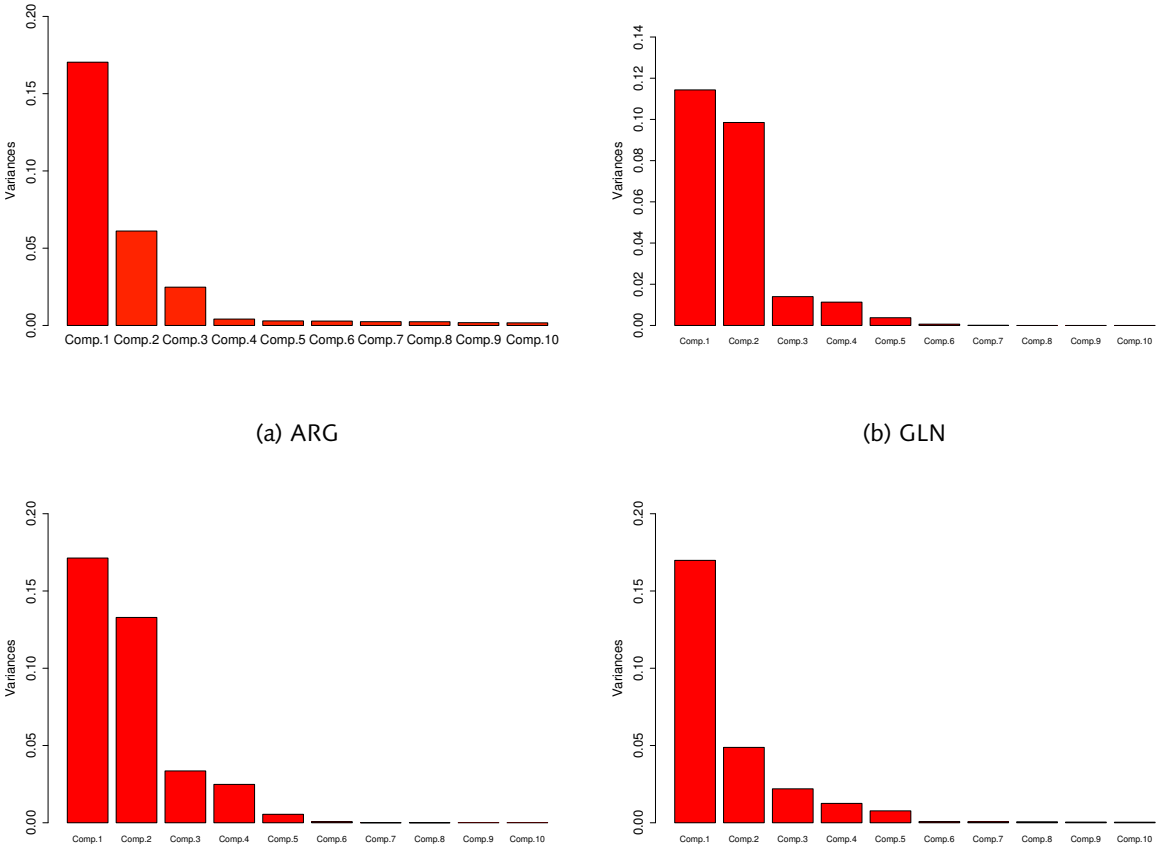
(l) TRP



(m) TYR

Figure B.7: *(cont.) PCA Eigenvalue spectra for* χ_2.

B.4.3 Principle Component Analysis for χ_3 and χ_4



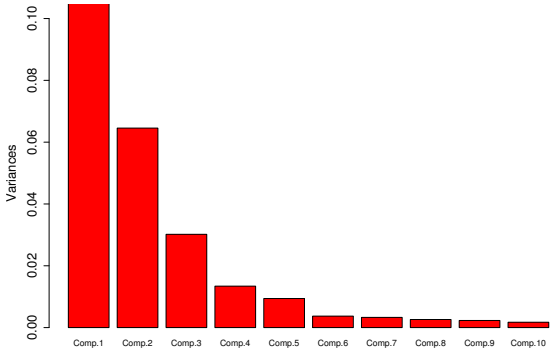
(a) ARG

(b) GLN

(c) GLU

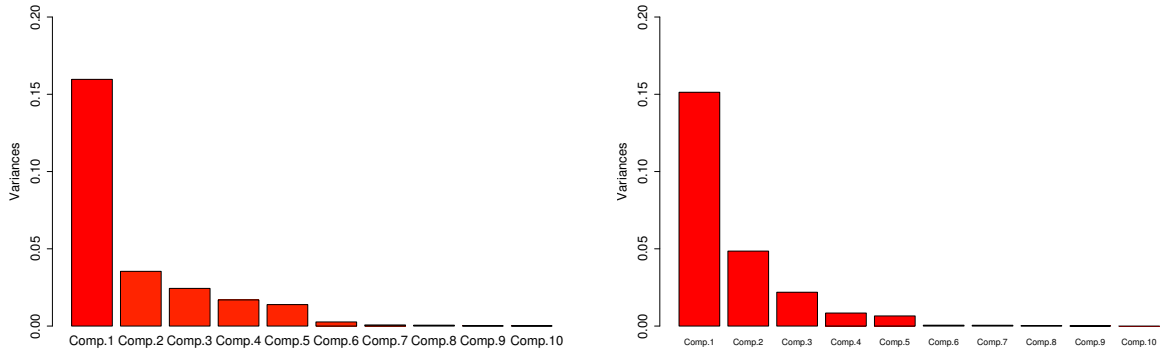
(d) LYS

Figure B.8: *PCA Eigenvalue spectra for* χ_3.



(e) MET

Figure B.8: *(cont.) PCA Eigenvalue spectra for* χ_3.



(a) ARG

(b) LYS

Figure B.9: *PCA Eigenvalue spectra for* χ_4.

B.5 Tables of Classification Results using a SVM

Here, the results from the flexibility classification using a support vector machine (SVM) for the different amino acids and torsion angles are shown. Besides a 10–fold cross evaluation the SVM has been configured to use radial basis functions. For each amino acid and torsion angle a confusion matrix is compiled, showing the percentages of classifying a data example as true positive (flexible), true false (not flexible) and as false positive or false negative. In brackets the absolute numbers are given.

B.5.1 Results for χ_1

		Predicted	
		not flexible	flexible
True	not flexible	78.9% (15)	21.1% (4)
	flexible	23.3% (7)	76.7% (23)

(a) Classification of χ_1 for ARG. A total accuracy of 77.1% is reached.

		Predicted	
		not flexible	flexible
True	not flexible	71.4% (35)	28.6% (14)
	flexible	27.1% (13)	72.9% (35)

(b) Classification of χ_1 for ASN. A total accuracy of 73.3% is reached.

		Predicted	
		not flexible	flexible
True	not flexible	66.2% (47)	33.8% (24)
	flexible	30.3% (20)	69.7% (46)

(c) Classification of χ_1 for ASN. A total accuracy of 70.8% is reached.

		Predicted	
		not flexible	flexible
True	not flexible	100.0% (2)	0.0% (0)
	flexible	20.0% (1)	80.0% (4)

(d) Classification of χ_1 for CYS. A total accuracy of 89% is reached.

		Predicted	
		not flexible	flexible
True	not flexible	73.6% (39)	26.4% (14)
	flexible	17.9% (7)	82.1% (32)

(e) Classification of χ_1 for GLN. A total accuracy of 78.8% is reached.

		Predicted	
		not flexible	flexible
True	not flexible	66.7% (54)	33.3% (27)
	flexible	30.1% (22)	69.9% (51)

(f) Classification of χ_1 for GLU. A total accuracy of 68.4% is reached.

Table B.5: *Classification results for ARG, ASN, ASP, CYS, GLN and GLU for* χ_1.

		Predicted	
		not flexible	flexible
True	not flexible	66.7% (4)	33.3% (2)
	flexible	25.0% (2)	75.0% (6)

(g) Classification of χ_1 for HIS. A total accuracy of 75.3% is reached.

		Predicted	
		not flexible	flexible
True	not flexible	76.3% (45)	23.7% (14)
	flexible	26.9% (18)	73.1% (49)

(h) Classification of χ_1 for ILE. A total accuracy of 75% is reached.

		Predicted	
		not flexible	flexible
True	not flexible	78.3% (36)	21.7% (10)
	flexible	25.0% (13)	75.0% (39)

(i) Classification of χ_1 for LEU. A total accuracy of 76.8% is reached.

		Predicted	
		not flexible	flexible
True	not flexible	64.0% (48)	36.0% (27)
	flexible	28.2% (22)	71.8% (56)

(j) Classification of χ_1 for LYS. A total accuracy of 70% is reached.

		Predicted	
		not flexible	flexible
True	not flexible	80.0% (12)	20.0% (3)
	flexible	17.6% (3)	82.4% (14)

(k) Classification of χ_1 for MET. A total accuracy of 82.5% is reached.

		Predicted	
		not flexible	flexible
True	not flexible	66.7% (2)	33.3% (1)
	flexible	20.0% (1)	80.0% (4)

(l) Classification of χ_1 for PHE. A total accuracy of 78.8% is reached.

		Predicted	
		not flexible	flexible
True	not flexible	60.5% (69)	39.5% (45)
	flexible	27.5% (25)	72.5% (66)

(m) Classification of χ_1 for SER. A total accuracy of 68.4% is reached.

		Predicted	
		not flexible	flexible
True	not flexible	63.8% (44)	36.2% (25)
	flexible	39.2% (40)	60.8% (62)

(n) Classification of χ_1 for THR. A total accuracy of 64.2% is reached.

		Predicted	
		not flexible	flexible
True	not flexible	66.7% (2)	33.3% (1)
	flexible	20.0% (1)	80.0% (4)

(o) Classification of χ_1 for TRP. A total accuracy of 85,7% is reached.

		Predicted	
		not flexible	flexible
True	not flexible	75.0% (3)	25.0% (1)
	flexible	0.0% (0)	100.0% (3)

(p) Classification of χ_1 for TYR. A total accuracy of 81% is reached.

		Predicted	
		not flexible	flexible
True	not flexible	67.1% (47)	32.9% (23)
	flexible	38.1% (32)	61.9% (52)

(q) Classification of χ_1 for VAL. A total accuracy of 68.1% is reached.

Table B.5: *Classification results for HIS, ILE, LEU, LYS, MET, PHE, SER, THR, TRP, TYR and VAL for* χ_1.

B.5.2 Results for χ_2

		Predicted	
		not flexible	flexible
True	not flexible	69.0% (40)	31.0% (18)
	flexible	33.3% (23)	66.7% (46)

(a) Classification of χ_2 for ARG. A total accuracy of 70.6% is reached.

		Predicted	
		not flexible	flexible
True	not flexible	60.7% (74)	39.3% (48)
	flexible	34.1% (29)	65.9% (56)

(b) Classification of χ_2 for ASN. A total accuracy of 63.3% is reached.

		Predicted	
		not flexible	flexible
True	not flexible	67.6% (46)	32.4% (22)
	flexible	28.8% (15)	71.2% (37)

(c) Classification of χ_2 for ASP. A total accuracy of 70.2% is reached.

		Predicted	
		not flexible	flexible
True	not flexible	74.5% (41)	25.5% (14)
	flexible	25.5% (14)	74.5% (41)

(d) Classification of χ_2 for GLN. A total accuracy of 74.9% is reached.

		Predicted	
		not flexible	flexible
True	not flexible	71.7% (38)	28.3% (15)
	flexible	35.1% (26)	64.9% (48)

(e) Classification of χ_2 for GLU. A total accuracy of 70.1% is reached.

		Predicted	
		not flexible	flexible
True	not flexible	69.7% (23)	30.3% (10)
	flexible	27.6% (8)	72.4% (21)

(f) Classification of χ_2 for HIS. A total accuracy of 69.2% is reached.

		Predicted	
		not flexible	flexible
True	not flexible	76.0% (38)	24.0% (12)
	flexible	35.2% (32)	64.8% (59)

(g) Classification of χ_2 for ILE. A total accuracy of 69.7% is reached.

		Predicted	
		not flexible	flexible
True	not flexible	62.4% (68)	37.6% (41)
	flexible	40.5% (53)	59.5% (78)

(h) Classification of χ_2 for LEU. A total accuracy of 63.4% is reached.

		Predicted	
		not flexible	flexible
True	not flexible	81.8% (54)	18.2% (12)
	flexible	31.5% (34)	68.5% (74)

(i) Classification of χ_2 for LYS. A total accuracy of 75.6% is reached.

		Predicted	
		not flexible	flexible
True	not flexible	84.6% (11)	15.4% (2)
	flexible	21.4% (3)	78.6% (11)

(j) Classification of χ_2 for MET. A total accuracy of 79.3% is reached.

Table B.6: *Classification results for ARG, ASN, ASP, GLN, GLU and* χ_2.

True \ Predicted	not flexible	flexible
not flexible	74.1% (20)	25.9% (7)
flexible	40.3% (25)	59.7% (37)

(k) Classification of χ_2 for PHE. A total accuracy of 66.4% is reached.

True \ Predicted	not flexible	flexible
not flexible	81.8% (9)	18.2% (2)
flexible	29.4% (5)	70.6% (12)

(l) Classification of χ_2 for TRP. A total accuracy of 75.7% is reached.

True \ Predicted	not flexible	flexible
not flexible	59.6% (28)	40.4% (19)
flexible	37.2% (16)	62.8% (27)

(m) Classification of χ_2 for TYR. A total accuracy of 62.1% is reached.

Table B.6: *(cont.) Classification results for PHE, TRP and TYR and χ_2.*

B.5.3 Results for χ_3 and χ_4

True \ Predicted	not flexible	flexible
not flexible	66.7% (42)	33.3% (21)
flexible	37.8% (34)	62.2% (56)

(a) Classification of χ_3 for ARG. A total accuracy of 64.5% is reached.

True \ Predicted	not flexible	flexible
not flexible	60.8% (59)	39.2% (38)
flexible	33.3% (21)	66.7% (42)

(b) Classification of χ_3 for GLN. A total accuracy of 65.1% is reached.

True \ Predicted	not flexible	flexible
not flexible	65.6% (40)	34.4% (21)
flexible	37.7% (29)	62.3% (48)

(c) Classification of χ_3 for GLU. A total accuracy of 63.1% is reached.

True \ Predicted	not flexible	flexible
not flexible	83.3% (50)	16.7% (10)
flexible	24.7% (19)	75.3% (58)

(d) Classification of χ_3 for LYS. A total accuracy of 80.8% is reached.

True \ Predicted	not flexible	flexible
not flexible	77.8% (21)	22.2% (6)
flexible	28.6% (10)	71.4% (25)

(e) Classification of χ_3 for MET. A total accuracy of 75.3% is reached.

Table B.7: *Classification results for ARG, GLN, GLU, LYS and MET for the χ_3 torsion angle.*

		Predicted	
		not flexible	flexible
True	not flexible	70.7% (41)	29.3% (17)
	flexible	38.7% (43)	61.3% (68)

(a) Classification of χ_4 for ARG. A total accuracy of 66.6% is reached.

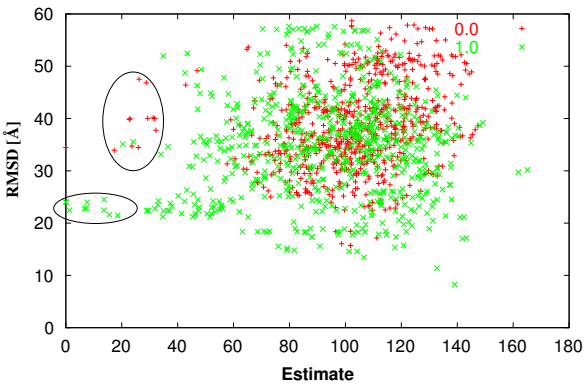
		Predicted	
		not flexible	flexible
True	not flexible	72.0% (54)	28.0% (21)
	flexible	35.3% (41)	64.7% (75)

(b) Classification of χ_4 for LYS. A total accuracy of 69.2% is reached.

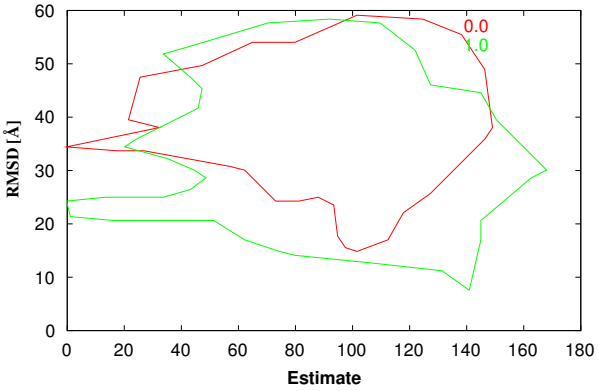
Table B.8: *Classification results for ARG and LYS for χ_4.*

B.6 Comparision Plots of Docking Test Cases

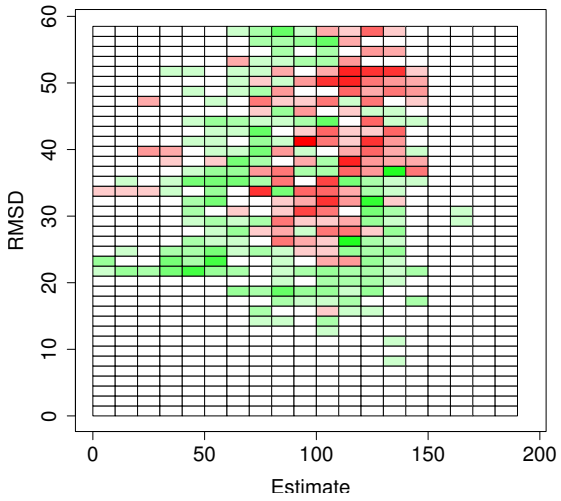
In this section, the plots of the detailed analysis for the test cases `1A2W(1BEL/1RAT)` and `1TPA(1BJU/1BPI)` are given. In both cases, the SVM based flexibility prediction for the first torsion angle and a scaling factor of 1.0 is used.



(a) Comparison plot of the docking results. Here, the flexibility is scaled by $\omega = 1.0$

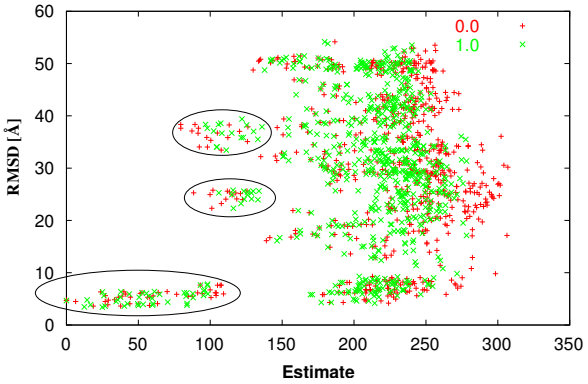


(b) Plot of the outer hulls (red: without flexibility, green: flexibility, $\omega = 1.0$).

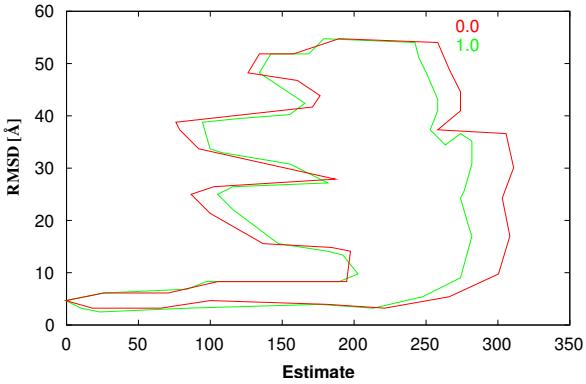


(c) Changes between docking with and without flexibility (green: increase in number of hypotheses, red: decrease in number of hypotheses, white: no or equal changes).

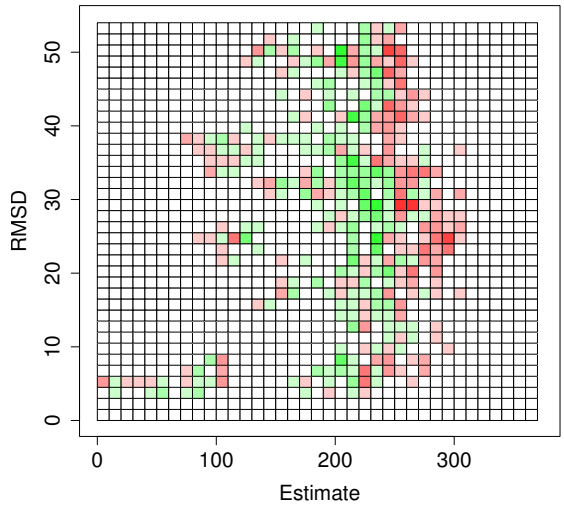
Figure B.10: *Visualisation of the docking results of `1BEL/1RAT`. Here, the same experiment is visualised like described in figure 7.25 but a scaling factor of $\omega = 1.0$ is used for the flexibility. On the top left the hypotheses are superimposed whereas on the top right the outer hulls are shown. Below these figures, the changes within the rectangular area of size 10×1.5 are plotted.*



(a) Comparison plot of the docking results. Here, the flexibility is scaled by $\omega = 1.0$



(b) Plot of the outer hulls. Here, the flexibility is scaled by $\omega = 1.0$



(c) Changes plot of the docking results. Here, the flexibility is scaled by $\omega = 1.0$

Figure B.11: *Visualisation of the docking results of 1BJU/1BPI. Each point in the plots represents one docking hypothesis. Here, the estimated costs are plotted against the RMSD. In red the results without incorporating flexibility are shown. The green coloured points are hypotheses from docking with flexibility information. In this experiment only the flexibility information of the first torsion angle χ_1 is used. The marked parts of the plot show changes effected by the flexibility incorporated.*

Appendix C
Amino Acids

In this part the common twenty amino acids are described. For each amino acid the structure formula and a 3D picture are given. The caption of each figure contains the name, three- and one letter code as well as the chemical features of the side chain.
The structure formula of the amino acids are drawn using the tool BKChem (Kosata, 2003). The 3D pictures are created on basis of PDB files (New York University Scientific Visualization Center, 2003) and are visualised using VMD (Humphrey *et al.*, 1996).

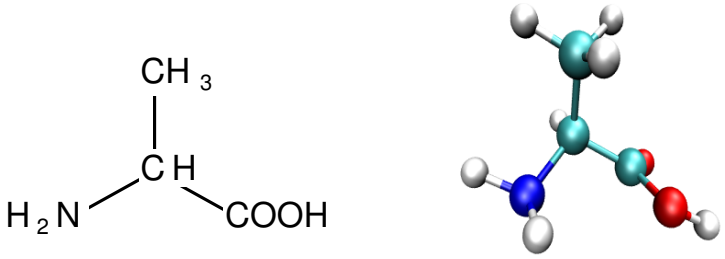


Figure C.1: *Alanine (ALA, A), apolar.*

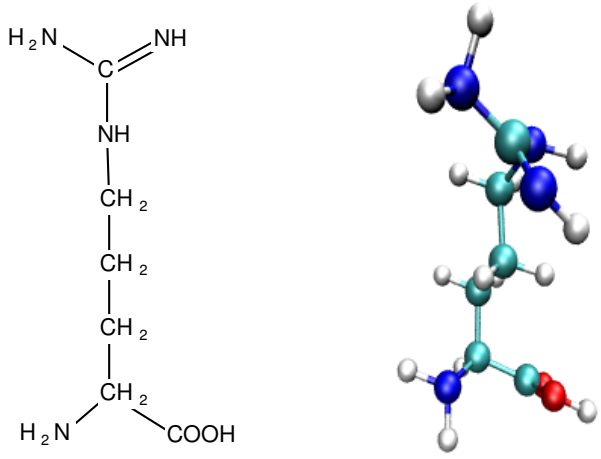


Figure C.2: *Arginine (ARG, R), charged.*

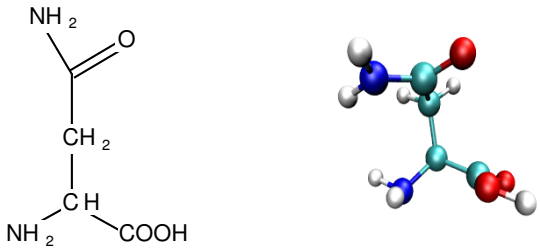


Figure C.3: *Asparagine (ASN, N), uncharged, polar.*

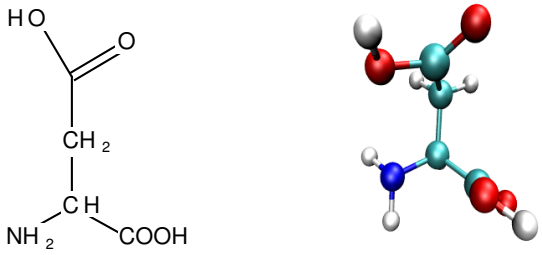


Figure C.4: *Aspartic acid (ASP, D), charged.*

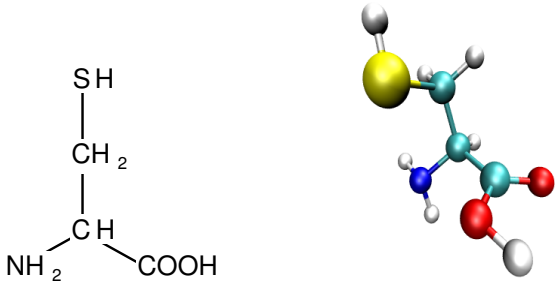


Figure C.5: *Cystein (CYS, C), uncharged, polar.*

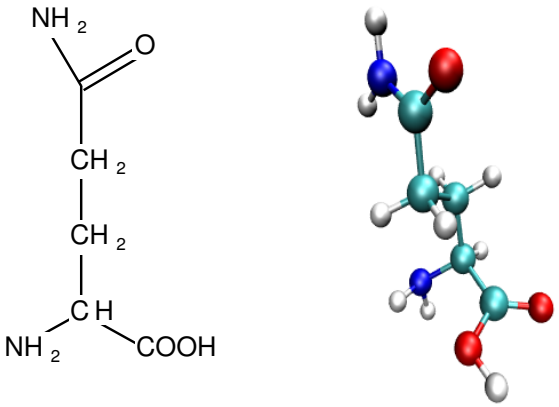


Figure C.6: *Glutamine (GLN, Q), uncharged, polar.*

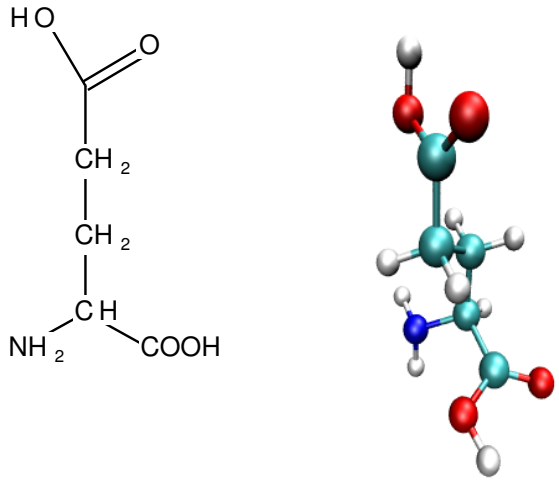


Figure C.7: *Glutamic acid (GLU, E), charged.*

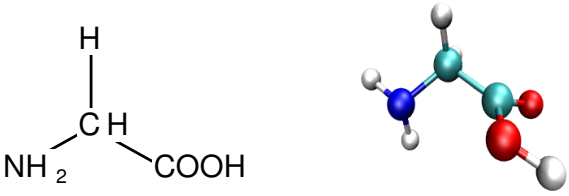


Figure C.8: *Glycine (GLY, G), apolar.*

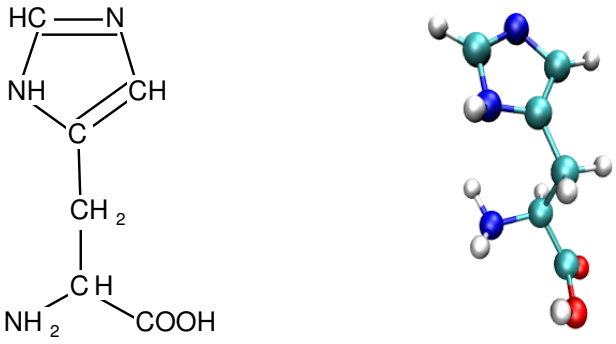


Figure C.9: *Histidine (HIS, H), charged.*

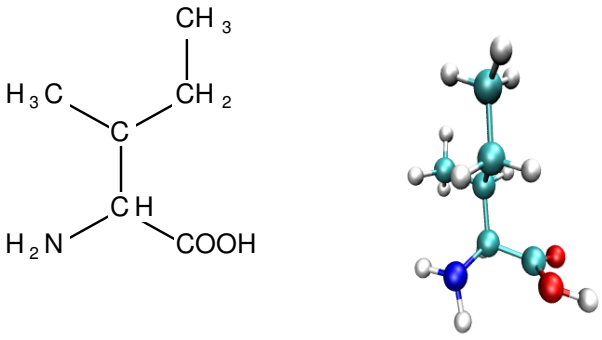


Figure C.10: *Isoleucine (ILE, I), apolar.*

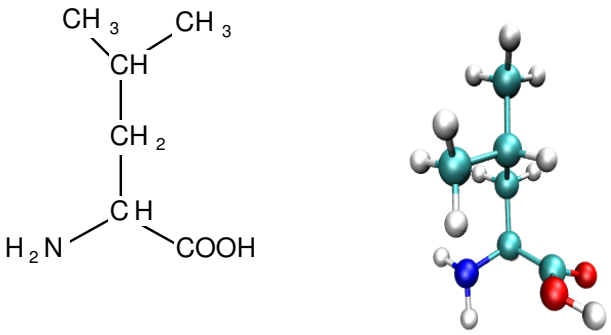


Figure C.11: *Leucine (LEU, L), apolar.*

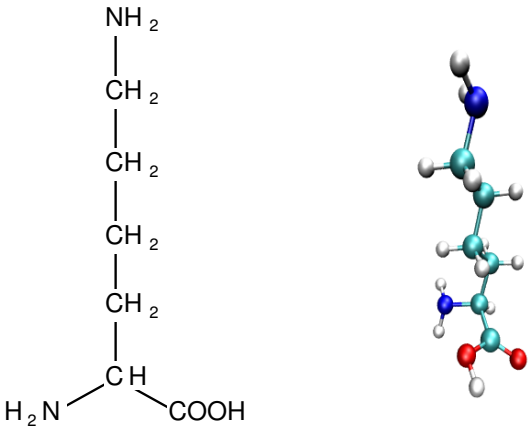


Figure C.12: *Lysine (LYS, K), charged.*

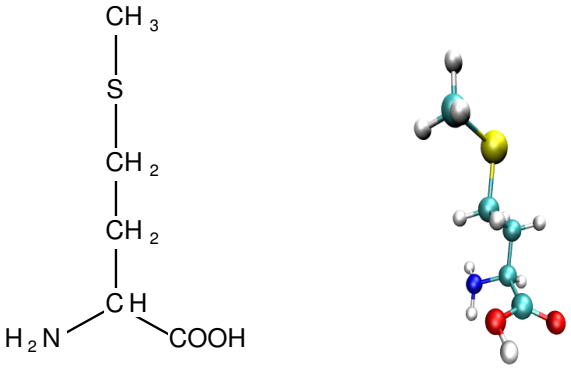


Figure C.13: *Methionine (MET, M), apolar.*

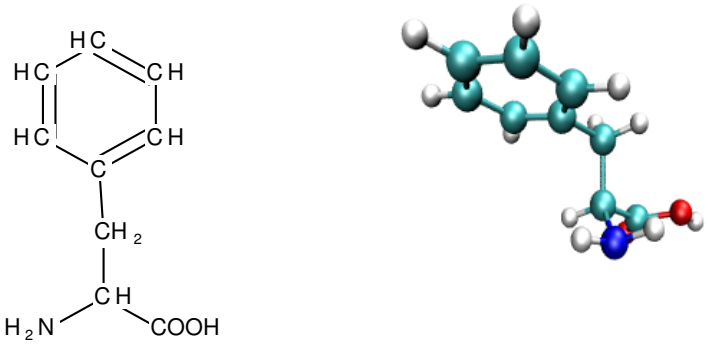


Figure C.14: *Phenylalanine (PHE, F), apolar.*

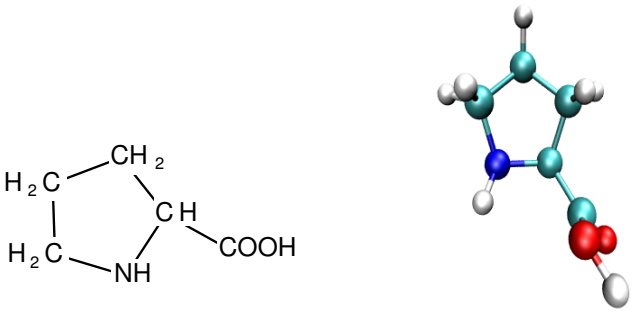


Figure C.15: *Proline (PRO, P), apolar.*

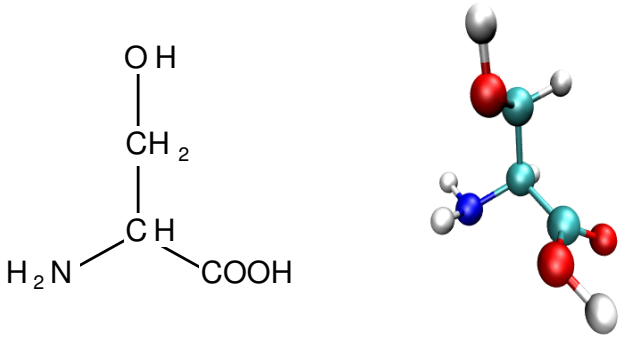


Figure C.16: *Serine (SER, S), uncharged, polar.*

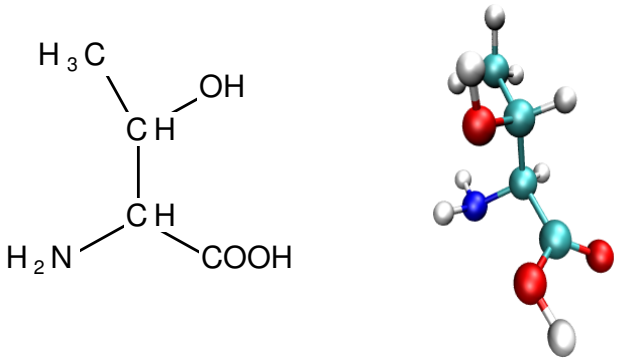


Figure C.17: *Threonine (THR, T), uncharged, polar.*

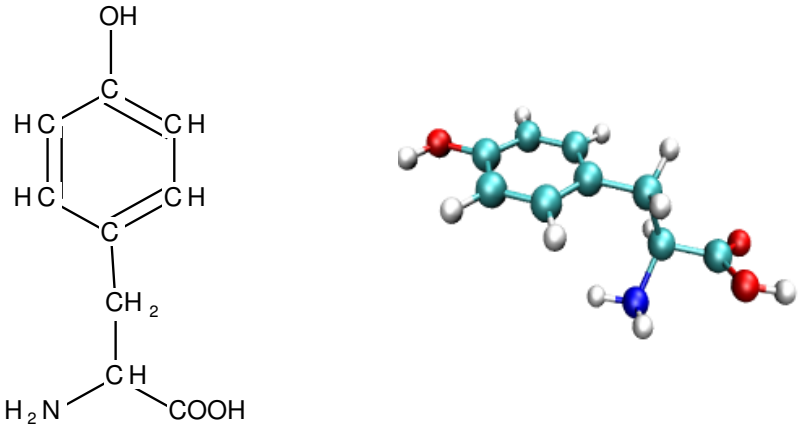


Figure C.18: *Tyrosine (TYR, Y), uncharged, polar.*

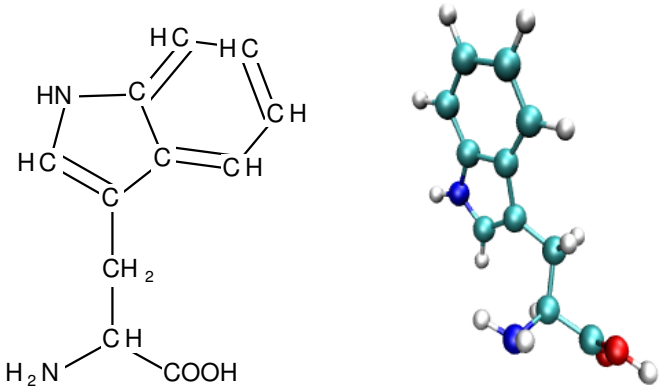


Figure C.19: *Tryptophan (TRP, W), apolar.*

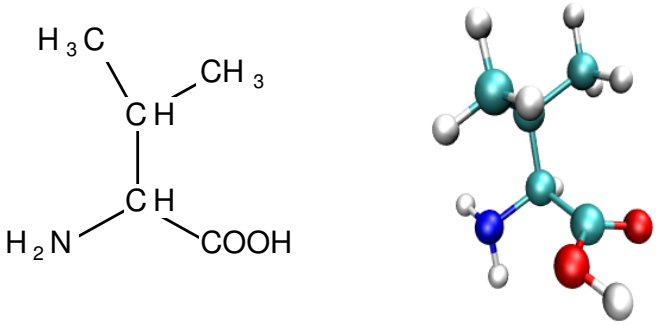


Figure C.20: *Valin (VAL, V), apolar.*

Appendix D
Systems

D.1 Automatic Test Set Generation

In this section the tools and applications used within the automatic processing line for the protein structures are described (see also Fig. 7.2 and 7.3). First the control structure of the system is outlined then the different programs are presented briefly.

D.1.1 Control Structures for Automatic Test Set Generation

The whole pipelined system is controlled by the java build-tool **ANT** (Loughran, 2002). The ANT tool is similar to the gnu tool "make". It runs along a so called *build file* processing targets defined within the file. Dependencies are defined to set the order of processing the targets. The build file is an XML file using a set of special tags for describing the targets, dependencies, etc. As one aim of this processing line is to be able to re-run parts of it to incrementally update information, the ANT build file is compiled from an instruction file using an XSLT style sheet (see Fig. D.2). Figure D.3 shows part of the style sheet used for setting up the build-file. Figure D.1 shows an example instruction file whereas in figure D.4 the resulting build-file is presented.

The transformations is performed as follows: The XSLT processor (here **saxon** (Kay, 2001) is used) is given the style sheet and the XML instruction file as input. The transformation

```
<?xml version="1.0"?>
<dataset>
   <overview>
      <entry>initial</entry>
   </overview>
</dataset>
```

Figure D.1: *XML instruction file used for generating an ANT build file.*

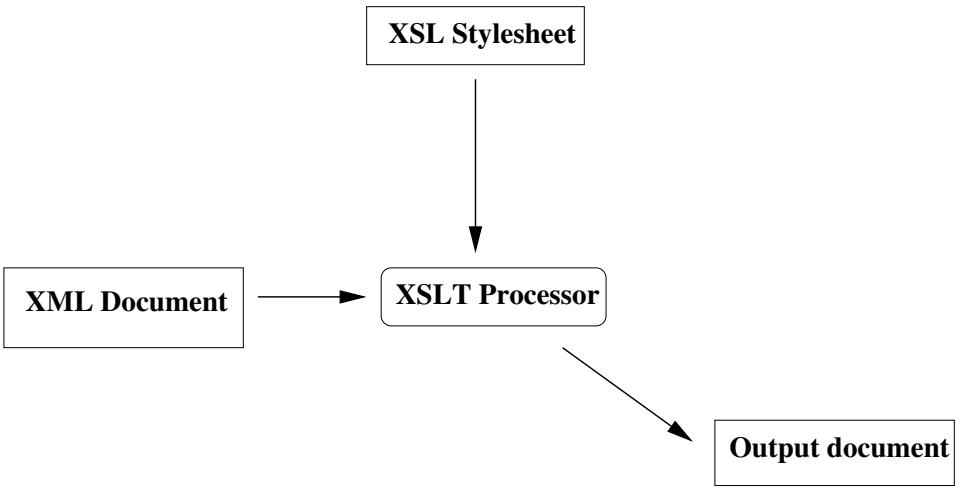


Figure D.2: *Scheme of XSLT transformation: The source XML document is transformed by an XSLT processor using the given style sheet. The processor outputs a new document usually containing a rearranged selection of the source document. The output must not be necessarily an XML document.*

processor then selects and rearranges the input according to the style sheet. The rearranged and selected data is then printed to the output document, here the ANT build file. For a detailed description about XSLT transformations see Kay (Kay, 2001).

The ANT build file (see Fig. D.4) contains a base set of tags. Every ANT build file is a "project" and therefore the root node of the build file is the tag <project>. Within the <target> ... </target> element a build instruction can be described. In the opening tag additional attributes can be defined. The most important ones are "name", "depends" and "unless". Whereas the "name" attribute enables to name the target the other two attributes are used to control the processing order of the targets. The "depends"–attribute holds dependencies within the target, e.g. the target *test cases* depends on *pdb2mysql*. The "unless" attribute controls the execution of a target. The target "pdb2mysql" will be executed unless a parameter called *pdb2mysql* is set.

The body of the target holds certain instruction what should happen if the target is executed. Here, the instruction <exec> is used to call the different modules of the pipeline system. The attribute "failonerror" handles possible errors. If it is set true, the build process will stop immediately if an error occurs. Arguments can be passed to a script called within the "exec" element by using the <arg value=" "> construct.

D.1.2 Module descriptions

In this section the different programs used within the modules are described. Within the system a large number of tools and programs are used implemented in different programming and script languages like C++ (Stroustrup, 1998), perl (Wall *et al.*, 2002) or bash shell scripts. The programs or tools are grouped into modules. Every module is wrapped by a

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:output method="xml" indent="yes"/>

<xsl:template match="dataset">
<xsl:variable name="id">
      <xsl:value-of select="//overview/entry"/>
</xsl:variable>
<project name="{$id}" default="init">
  <description>
     processing <xsl:value-of select="//overview/entry"/>
  </description>

  <target name="pdb2mysql"
          unless="pdb2mysql" description="run pdb2mysql">
    <exec executable="/vol/elmar/src/pspl/pspl/run_pdb2mysql.pl"
          failonerror="true">
      <arg value="{$id}"/>
    </exec>
  </target>

  <target name="testcase"
          depends="pdb2mysql"
          description="testcases are build every time new">

    <exec executable="/vol/elmar/src/pspl/pspl/run_insert_twochain_testcases.pl
          failonerror="true">
        <arg value="{$id}"/>
    </exec>
    <exec executable="/vol/elmar/src/pspl/pspl/run_pdbataglance.sh"
          failonerror="true" >
        <arg value="{$id}"/>
    </exec>
  </target>
```

Figure D.3: *Part of the XSLT style sheet used to compile a build file for ANT. Here, the style sheet for the test set generation is shown.*

```
  <target name="jointc" depends="testcase">
    <exec executable="/vol/elmar/src/pspl/pspl/run_pubtc.sh"
          failonerror="true"/>
    <exec executable="/vol/elmar/src/pspl/pspl/run_intersecttestcase.pl"
          failonerror="true"/>
  </target>

  <target name="init" depends="jointc"/>

  <target name="init_first">
    <exec executable="/vol/elmar/src/pspl/pspl/run_mirror.sh"
          failonerror="true"/>
    <exec executable="/vol/elmar/src/pspl/pspl/run_pdb2mysqlfirst.pl"
          failonerror="true"/>
    <exec executable="/vol/elmar/src/pspl/pspl/run_pdbataglance.sh"
          failonerror="true"/>
    <exec executable="/vol/elmar/src/pspl/pspl/run_insert_twochain_testcases.pl
          failonerror="true"/>
    <exec executable="/vol/elmar/src/pspl/pspl/run_intersecttestcase.pl"
          failonerror="true"/>
    <exec executable="/vol/elmar/src/pspl/pspl/run_pubtc.sh"
          failonerror="true"/>
   </target>
</project>
</xsl:template>
</xsl:stylesheet>
```

Figure D.3: *(cont.) Part of the XSLT style sheet used to compile a build file for ANT. Here, the style sheet for the test set generation is shown.*

```
<?xml version="1.0" encoding="utf-8"?>
<project name="initial" default="init">
   <description>
          processing initial
   </description>
   <target name="pdb2mysql"
           unless="pdb2mysql"
           description="run pdb2mysql">
     <exec executable="/vol/elmar/src/pspl/pspl/run_pdb2mysql.pl"
           failonerror="true">
        <arg value="initail"/>
     </exec>
   </target>
   <target name="testcase"
           depends="pdb2mysql"
           description="testcases are build every time new">
     <exec executable=
            "/vol/elmar/src/pspl/pspl/run_insert_twochain_testcases.pl"
           failonerror="true">
         <arg value="initail"/>
     </exec>
     <exec executable="/vol/elmar/src/pspl/pspl/run_pdbataglance.sh"
           failonerror="true">
        <arg value="initail"/>
     </exec>
   </target>
   <target name="jointc"
           depends="testcase">
     <exec executable="/vol/elmar/src/pspl/pspl/run_pubtc.sh"
           failonerror="true"/>
     <exec executable="/vol/elmar/src/pspl/pspl/run_intersecttestcase.pl"
           failonerror="true"/>
   </target>
   <target name="init"
           depends="jointc"/>
   <target name="init_first">
     <exec executable="/vol/elmar/src/pspl/pspl/run_mirror.sh"
           failonerror="true"/>
     <exec executable="/vol/elmar/src/pspl/pspl/run_pdb2mysqlfirst.pl"
           failonerror="true"/>
     <exec executable="/vol/elmar/src/pspl/pspl/run_pdbataglance.sh"
           failonerror="true"/>
     <exec executable=
             "/vol/elmar/src/pspl/pspl/run_insert_twochain_testcases.pl"
           failonerror="true"/>
     <exec executable="/vol/elmar/src/pspl/pspl/run_intersecttestcase.pl"
           failonerror="true"/>
     <exec executable="/vol/elmar/src/pspl/pspl/run_pubtc.sh"
           failonerror="true"/>
   </target>
</project>
```

Figure D.4: *ANT build file automatic test set generation. It can be run as a batch job or incrementally to update only new* PDB *structures.*

perl script to control the execution and to fetch additional information and data from the database. Another benefit of wrapping the modules is that they can be tested easily. These scripts are named by the prefix "run_" (see Fig. D.4) and is given usually the 4 letter identifier (PDB Id) as argument. The first part of the automated pipeline system has been described in section 7.1.1. It contains the following scripts for processing the PDB and compiling the test cases:

mirror Perl tool mirror (McLoughlin, 2003) is used to update the local PDB repository against the master database (PDB).

pdb2mysql Extracts meta information from the PDB file and stores it in the database (e.g. number of chains, resolution, residue identifiers, etc.).

The second part of the pipeline contains seven modules, the "PDB Structure Check", the "Additional Metainformation", two modules for calculating the flexibility: the rotamer library, the docking system and the evaluation module "IPHEX". The "PDB Structure Check" module is used to validate the PDB structures and to add hydrogen atoms to the protein models. It contains therefore the following tools:

pdbchecker The pdbchecker tests the integrity of the given protein structure. This tool is implemented using the C++ library BALL (Kohlbacher, 2000).

addHydrogens Part of the "structure check" module. It is used to add hydrogens to the protein structure. In a first step hydrogen atoms are placed roughly using the add_hydrogens method from the BALL library. In a second step the model is optimised by geometric minimisation using the AMBER force field and a steepest gradient minimiser.

dssp The dssp (Kabsch & Sander, 1983) program is used to calculate secondary structure elements of a given protein structure.

The second element in the pipeline is the "Additional Metainformation" module. It contains different applications to calculate information needed within the other modules like secondary structure or SAS:

sequence alignment This tool is used to calculate an alignment between the residue numbers given in the sequence section of the PDB file and the atoms section. In some cases these entries are not synchronised and therefore structure comparisons (e.g. needed for the statistical analysis) on residue level are difficult.

SAS Here, the solvent accessible surface area is calculated. It is used to discriminate between residue on the surface and in the core of the protein.

Contact site In this tool three different methods are used to identify the contact site of a protein complex. The first method uses distance calculations between the chains of a protein complex. The second method bases on overlapping voxels as calculated within the docking algorithm ELMAR. The third method refines the results of the first method and analyses atom–atom contacts.

In order to calculate flexibility information two separate modules are set up. The first module calculates flexibility information on basis of statistical analysis of protein structures. The algorithm used here is described in the thesis of Koch (Koch, 2003). The second module uses a classification approach to receive flexibility information of the residues. The methods applied here are described in chapter 5.

Adjacent to the flexibility calculations the docking system ELMAR can be utilised. It is outlined briefly in chapter 4. The system has been developed by S.Neumann. A more detailed description of the docking algorithm and the different parts of the system are given in his thesis (Neumann, 2003).

The last module of the pipeline is the IPHEx system. IPHEx is used to enhance the scoring of the ELMAR system. In chapter 6 an approach to search for improved weights used within the ELMAR scoring function is described. Furthermore, in the section D.3 additional informations about IPHEx are given.

D.2 Implementation of the Incorporation of Flexibility Information into ELMAR

In this section, the technical realisation of the incorporation of the flexibility information into ELMAR is outlined. As mentioned in section 7.1.1, the whole pipeline for processing the proteins to derive flexibility information is back ended by a database. All flexibility predictions are stored there, too. Thus, the flexibility information can be queried easily from the database. In ELMAR within the "Final Docking"module, two queries are run to fetch the required information. The queries are given in figure D.5.

```
select IFNULL(chi1, 0) from flexibility
    where Res_Id = %3q:Res_Id
      AND Entry = %1q:Entry
      AND Chain_Id = %2q:Chain_Id

select IFNULL(%0:func(IFNULL(chi1, 0)),%1:defaultvalue) from flexibility
    where Entry = %2q:Entry
```

Figure D.5: *SQL queries used for incorporating the flexibility predictions into* ELMAR.

The first query fetches the flexibility information. If no information is present (represented by the NULL value in the corresponding field) for a specific residue, the assumption is made that it is not flexible. In this case ELMAR treads the residue like a rigid body docking algorithm as fall-back. The second query is used for the normalisation and scaling of the flexibility values. Both queries are placed within a configuration file, so that a change of the flexibility source can be adapted without recompiling the ELMAR system.

D.3 IPHEx

In this section the usage of the IPHEx system and the technical details will be outlined. As already mentioned in section 6.2, the IPHEx system consists of two parts: the visualisation of the protein structures using the ViWish (Klein *et al.*, 1996) and a navigation panel including the adaptation of the weights. As a back end IPHEx uses a database to receive hypotheses and to store adapted weights. The database is also used for mapping weights onto protein classes (e.g. grouped by enzyme numbers) as the protein class information is also stored there.

In principle the two parts of IPHEx must not run on the same computer but the user interfaces should be run on the same display. This is because IPHEx uses TCL/Tk commands for communication between the two modules. Controlling the visualisation is therefore quite easy. For instance, translation or rotating a hypothesis can be done simply by passing the translation or rotation vector via the TCL interpreter (see Fig. D.6). The navigation module

```
Tcl_VarEval(interp, "send viwish ","pdb",
            protein id," 2", " translate ", transvec, NULL)
```

Figure D.6: *Communication between the* IPHEx *modules via TCL script. Here, the translation vector transvec is send to the protein named "protein id". The string translate is a build in command of* ViWish.

contains on the one hand an interface for navigating through the set of hypotheses and to give feedback. Here, also the adapted weights are plotted for each feedback iteration. In case of navigating backwards, the already given feedback is highlighted by changing the colour of the corresponding button. This enables the user to remember scores and possibly re-score the actual hypothesis[1].

On the other hand the adaption of weights is placed within this module. The adaption is performed as already described in section 6.2. A set of parameters control the adaptation of the weights. By default these are set as given in table D.1. Three other parameters

Parameter	default value	command line option
learning rate	0.01	-l $< value >$
gradient of F function	0.25	-g $< value >$
number of hypotheses to be scored	20	-h $< value >$

Table D.1: *Parameters and default values of* IPHEx

control the behaviour of IPHEx. For testing one can choose that at the end of session,

[1] Please note that this is only possible within a feedback iteration. After having adapted the weights, the internal feedback list is reseted.

```
iphexgui <PDB Id of Protein 1> <Chain Id of Protein 1> <PDB Id of Protein 2>
         <Chain Id of Protein 2> <commandline>
         -l <learning rate>
         -g <Gradient>
         -h <number of hypotheses>
         -? help
         -s single session mode, do not write weights to database
         -e do evaluation
         -u <user>
```

Figure D.7: *Command line options for* IPHEx*.*

the modified weights should not be stored within the database (-s). Also an evaluation of changes between the original set of hypotheses and the re-ranked after the feedback session can be turned on and off (-e). This parameter also controls the mapping of the actually changed weights onto the whole set of hypotheses belonging to the same enzyme family as the test set currently under investigation. The whole command line is given in figure D.7. At the moment IPHEx does not support feedback from different users at the same time. The -u switch creates within the output directory a new subdirectory where the results of the actual session can be saved. So several user can score the same set of hypotheses simultaneously within interfering each other.

Besides the command line options, the IPHEx system is additionally set up by a configuration file. Here, global settings like the database host to choose or a save path for the results can be defined. Also, all queries used for interacting with the data base backend are listed here, enabling quick changes without re–compiling the sources (e.g. in case database tables are renamed or moved).

Curriculum Vitae

Frank Gerrit Zöllner

Address

Universität Bielefeld
Universitätsstr. 25
D-33615 Bielefeld
Germany
Phone: +49-(0)521-106-2951
Email: fzoellne@techfak.uni-bielefeld.de
Homepage: `www.techfak.uni-bielefeld.de/~fzoellne/`

Personal Details

Gender: Male
Date of birth: 18th of February, 1976
Place of birth: Bielefeld, Germany
Present Citizenship: German
Parents: Ulrike and Gerhard Zöllner

Education

08/1982–07/1986	Primary School "Buschkampschule", Bielefeld, Germany
08/1986–06/1995	Secondary School "Hans–Ehrenberg–Schule", Bielefeld, Germany. Leaving with the A-Level.
10/1996–06/2001	Undergraduate Studies Bielefeld University in "Computing in the natural sciences" Specialisation: Applied Computer Science, Organic and Bioorganic Chemistry. Thesis: *Scoring the flexibility of amino acid side chains within proteins using empirical force fields*, Supervisors: Dr.-Ing. S. Neumann and Prof. Dr. F. Kummert
since 07/2001	Ph.D. Student Bielefeld University, Germany Graduate College "Bioinformatics" Project title: *Modelling amino acid side chain within Modellierung von Aminosäureseitenketten in spatial neighborhood and scoring of their flexibility by force fields*, Supervisor: Prof. Dr. G. Sagerer I intend to complete the Ph.D. in June, 2004.

Working Experience

09/1995–09/1996	Civil Service: Working in hospital "Städtische Kliniken Rosenhöhe", Bielefeld, Germany
03/1997–06/1998	part time job as medical orderly at Ev. Altenpfleegzentrum Ernst–Barlach–Haus, Bielefeld
02/1998–12/2000	Student helper, conceptual design & organisation of lecture "Orientierungsschwerpunkte Informatik"
07/1998–12/2000	Student helper within Research Focus 360, project A1
since 07/2001	Research Assitant at Bielefeld University, Germany

Teaching Experience

2002	Seminar "Structure and dynamic of proteins"
2002	Lab Course "Applied computer science and proteins"
2003	accompanying Tutorial of lecture "Pattern Analysis"

Language Knowledge

German	native
French	fair
English	fluently

Journal Publications

[1] K. Koch, F. Zöllner, S. Neumann, F. Kummert, and G. Sagerer. Comparing bound and unbound protein structures using energy calculation and rotamer statistics. *In Silico Biology*, 2:32, 2002.

[2] Frank Zöllner, Steffen Neumann, Franz Kummert, and Gerhard Sagerer. Database driven test case generation for protein-protein docking. *Bioinformatics*, 2004. to appear.

Conference Contributions

[1] K. Koch, F. Zöllner, and G. Sagerer. Building a new rotamer library for protein-protein docking using energy calculations and statistical approaches. In *Computer Science and Biology: Proceedings of the German Conference on Bioinformatics*, pages 201–202, Braunschweig, Oktober 2001.

[2] Kerstin Koch, Steffen Neumann, Frank Zöllner, and Gerhard Sagerer. Side chain flexibility for 1:n protein-protein docking. Poster 92A, ISMB 2002, Edmonton, 2002.

[3] Steffen Neumann, Frank Zöllner, Kerstin Koch, Franz Kummert, and Gerhard Sagerer. Elmar: A protein docking system using flexibility information. In *European Conference on Computational Biology 2002, Poster Abstracts*, pages 169–170, Saarbrücken, October 2002.

[4] Frank Zöllner, Steffen Neumann, Kerstin Koch, Franz Kummert, and Gerhard Sagerer. Calculating residue flexibility information from statistics and energy based prediction. In *European Conference on Computational Biology 2002, Poster Abstracts*, pages 275–276, Saarbrücken, October 2002.

[5] Frank Zöllner, Steffen Neumann, Kerstin Koch, Franz Kummert, and Gerhard Sagerer. IPHEx: A System For Evaluating Protein Docking Hypothesis Using User Feedback. In Catherine Christophe, Hans-Peter Lenhof, and Marie-France Sagot, editors, *Proceedings of the Eurpean Conference on Computational Biology, Poster Abstracts*, pages PS–28, Paris, September 2003.

Thesis

[1] F. Zöllner. Bewertung der Flexibilität von Aminosäureseitenketten in Proteinkonformationen durch empirische Energiefelder. Master's thesis, Bielefeld, University, 2001.

Bielefeld, July 18, 2004

List of Figures

List of Tables

Bibliography

[Ackermann *et al.*, 1998] Ackermann, F., Hermann, G., Posch, S. & Sagerer, G. (1998) Estimation and filtering of potential protein-protein docking positions. *Bioinformatics,* **14** (2), 196–205.

[Allinger, 1977] Allinger, N. L. (1977) Conformational analysis 130. mm2. a hydrocarbon force field utilizing v1 and v2 torsional terms. *Journal of the American Chemical Society,* **99**, 8127–8134.

[Althaus *et al.*, 2002] Althaus, E., Kohlbacher, O., Lenhof, H.-P. & Müller, P. (2002) A combinatorial approach to protein docking with flexible side-chains. *Journal of Computational Biology,* **9**, 597 – 612.

[An *et al.*, 1998] An, J., Nakama, T., Kubota, Y. & Sarai, A. (1998) 3DinSight: an integrated relational database and search tool for structure, function and property of biomolecules. *Bioinformatics,* **14**, 188–195.

[Atkins & Gesteland, 2002] Atkins, J. F. & Gesteland, R. (2002) The 22nd amino acid. *Science,* **296**, 1409–1410.

[Aubertin *et al.*, 2002] Aubertin, T., Boghossian, N. P., Burchardt, A., Hildebrandt, A., Klein, H., Kerzmann, A., Kohlbacher, O., Lehnhof, H.-P., Moll, A., Müller, P. & Strobel, S. (2002). Ball reference manual. http://www.mpi-sb.mpg.de/BALL/doc_1.0b/html/AmberFF.html.

[Bairoch & Apweiler, 2000] Bairoch, A. & Apweiler, R. (2000) The swiss-prot protein sequence database and its supplement trembl in 2000. *Nucleic Acids Research,* **28**, 45–48.

[Baldi & Brunak, 2001] Baldi, P. & Brunak, S. (2001) *Bioinformatics: the machine learning approach*. Adaptive Computation and Machine Learning, 2nd edition,, MIT Press, Cambridge,Massachusetts–London,England. section 6.7.

[Bäni, 2002] Bäni, W. (2002) *Wavelets*. Oldenbourg, Munich Vienna.

[Bauckhage *et al.*, 2003] Bauckhage, C., Käster, T., Pfeiffer, M. & Sagerer, G. (2003) Content-Based Image Retrieval by Multimodal Interaction. In *Proc. of the 29th Annual Conference of the IEEE Industrial Electronics Society* pp. 1865–1870, Roanoke, VA.

[Betts & Sternberg, 1999] Betts, M. J. & Sternberg, M. J. (1999) An analysis of conformational changes on protein-protein association: implications for predictive docking. *Protein Engineering,* **12** (4), 271–283.

[Bhat *et al.*, 2001] Bhat, T. N., Bourne, P., Feng, Z., Gilliland, G., Jain, S., Ravichandran, V., Schneider, B., Schneider, K., Thanki, N., Weissig, H. & Westbrook, J. (2001) The pdb data uniformity project. *Nucleic Acids Research,* **29** (1), 214–218.

[Bilban *et al.*, 2002] Bilban, M., Buehler, L. K., Head, S., Desoye, G. & Quaranta, V. (2002) Defining signal thresholds in dna microarrays: exemplary applications for invasive cancer. *BMC Genomics,* **3** (19).

[Boser *et al.*, 1992] Boser, B. E., Guyon, I. M. & Vapnik, V. N. (1992) A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, (Haussler, D., ed.), pp. 144–152 ACM Press.

[Bower *et al.*, 1997] Bower, M., Cohen, F. E. & Dunbrack, R. L. (1997) Prediction of proteins side-chain rotamers from a backbone-dependend rotamer library: a new homology modelling tool. *Journal of Molecular Biology,* **267** (5), 1268–1282.

[Bradley, 1997] Bradley, A. P. (1997) The use of the Area under the ROC Curve in the Evaluation of Machine Learning Algorithms. *Pattern Recognition,* **30** (7), 1145–1159.

[Brandon & Tooze, 1999] Brandon, C. & Tooze, J. (1999) *Introduction to Protein Structure*. 2nd edition,, Garland Publishing Inc.

[Brooks *et al.*, 1983] Brooks, B. R., Bruccelori, R. E., Olafson, B. D., States, D. J., Swaminathan, S. & Karplus, M. (1983) Charmm: a program for macromolecular energy, minimization, and dynamics calculations. *Journal of Computational Chemistry,* **4** (2), 187–217.

[Chang & Lin, 2001] Chang, C.-C. & Lin, C.-J. (2001) Training nu-support vector classifiers: theory and algorithms. *Neural Computation,* **13** (9), 2119–2147.

[Chen & Weng, 2002] Chen, R. & Weng, Z. (2002) Docking unbound proteins using shape complementarity, desolvation, and electrostatics. *Proteins: Structure, Function, and Genetics,* **47** (3), 281–294.

[Christianini & Shawe-Taylor, 2000] Christianini, N. & Shawe-Taylor, J. (2000) *An Introduction to Support Vector Machines and other kernel–based learning methods*. Cambridge University Press, Cambridge, UK.

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

[Colonna-Cesari *et al.*, 1986] Colonna-Cesari, F., Perahia, D., Karplus, M., Eklund, H., Bränden, C. & Tapia, O. (1986) Interdomain motion in liver alcohol dehydrogenase. structural and energetic analysis of the hinge bending mode. *J. Biol. Chem,* **261**, 15273–15280.

[Comeau *et al.*, 2004] Comeau, S. R., Gatchell, D. W., Vajda, S. & Camacho, C. J. (2004) ClusPro: an automated docking and discrimination method for prediction of protein complexes. *Bioinformatics,* **20** (1), 45–50.

[Connolly, 1983*a*] Connolly, M. L. (1983*a*) Solvent–accessible surfaces of proteins and nucleic acids. *Science,* **221**, 709.

[Connolly, 1983*b*] Connolly, M. L. (1983*b*) Analytical molecular surface calculation. *Journal of Applied Chrystallography,* **16**, 548–558.

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

[Cyrus, 1976] Cyrus, C. (1976) The nature of the accessible and buried surfaces in proteins. *Journal of Molecular Biology,* **105**, 1–14.

[Dimitriadou *et al.*, 2004] Dimitriadou, E., Hornik, K., Leisch, Friedrichand Meyer, D. & Weingessel, A. (2004) *e1071: Misc Functions of the Department of Statistics (e1071)*. TU Wien. `http://cran.at.r-project.org/src/contrib/Descriptions/e1071.html`.

[Donoho, 1995] Donoho, D. L. (1995) De–noising via soft thresholding. *IEEE Transaction on Information Theory,* **41** (3), 613–627.

[Dosztanyi & Torda, 2001] Dosztanyi, Z. & Torda, A. E. (2001) Amino acid similarity matrices based on force fields. *Bioinformatics,* **17** (8), 686–699.

[Drummond & Holte, 2000] Drummond, C. & Holte, R. C. (2000) Explicitly representing excepted cost: an alternative to roc representation. In *International Conference on Knowledge Discovery and Data Mining*. Paper No. 331.

[Dunbrack & Karplus, 1993] Dunbrack, R. L. & Karplus, M. J. (1993) Backbone-dependent rotamer library for proteins. application to side-chain prediction. *Journal of Molecular Biology,* **230**, 543–574.

[Echols *et al.*, 2003] Echols, N., Milburn, D. & Gerstein, M. (2003) MolMolvDB: analysis and visualisation of conformational change and structural flexibility. *Nucleic Acids Research,* **31** (1), 478–482.

[Egan, 1975] Egan, J. P. (1975) *Signal detection theory and ROC analysis*. Academic Pr., New York.

[Ewing *et al.*, 2001] Ewing, T. J. A., Makino, S., Skillman, A. G. & Kuntz, I. D. (2001) Dock 4.0: search strategies for automated molecular docking of flexible molecular databases. *Journal of Computer Aided Molecular Design,* **15**, 411–428.

[Faber & Matthews, 1990] Faber, H. & Matthews, B. (1990) A mutant t4 lysozyme displays five different crystal conformations. *Nature,* **348**, 263–266.

[Fernández-Recio *et al.*, 2002] Fernández-Recio, J., Totrov, M. & Abagyan, R. (2002) Soft protein-protein docking in internal coordinates. *Protein Science,* **11** (2), 280–291.

[Fischer, 1894] Fischer, E. (1894) Einfluss der Konfiguration auf die Wirkung der Enzyme. In *Berichte der deutschen chemischen Gesellschaft* vol. 27,. pp. 2985–2993.

[Foster & Fawcett, 1997] Foster, P. & Fawcett, T. (1997) Analysis and visualization of classifier performance: comparison under imprecise class and cost distributions. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining* pp. 43–48 AAAI Press, Menlo Park,CA.

[Gerstein *et al.*, 1994] Gerstein, M., Lesk, A. & Chothia, C. (1994) Structural mechanisms for domain movements in proteins. *Biochemistry,* **33**, 6739–7649.

[Goodman, 1998] Goodman, J. M. (1998) *Chemical Applications of Molecular Modelling*. The Royal Society of Chemistry. Chapter 2, Introduction to Force Fields.

[Greenbaum *et al.*, 2003] Greenbaum, D., Jansen, R. & Gerstein, M. (2003) Analysis of mRNA expression and protein abundance data: an approach for the comparison of the enrichment of features in the cellular population of proteins and transcripts. *Bioinformatics,* **18** (4), 585–596.

[Halperin *et al.*, 2002] Halperin, I., Ma, B., Wolfson, H. & Nussinov, R. (2002) Principles of docking: an overview of search algorithms and a guide to scoring functions. *Proteins: Structure, Function, and Genetics,* **47** (4), 409–443.

[Hubbard *et al.*, 1991] Hubbard, S. J., Campell, S. & Thornton, J. M. (1991) Molecular recognition conformational analysis of limited proteolytic sites and serine proteinase protein inhibitors. *Journal of Molecular Biology,* **220**, 507–530.

[Humphrey *et al.*, 1996] Humphrey, W., Dalke, A. & Schulten, K. (1996) Vmd - visual molecular dynamics. *Journal of Molecular Graphics,* **14**, 33–38.

[Ihaka & Gentleman, 1996] Ihaka, R. & Gentleman, R. (1996) R: a language for data analysis and graphics. *Journal of Computational and Graphical Statistics,* **5** (3), 299–314.

[IUPAC-IUB Commission on Biochemical Nomenclature (CBN), 1967] IUPAC-IUB Commission on Biochemical Nomenclature (CBN) (1967). Abbreviations and symbols for the description of the conformation of polypeptide chains. `http://www.chem.qmw.ac.uk/iupac/misc/noGreek/ppep1.html`. Link 11.12.2001.

[Janin & Wodak, 1978] Janin, J. & Wodak, S. (1978) Conformation of amino acid side-chains in proteins. *Journal of Molecular Biology,* **125**, 357–386.

[Jiang *et al.*, 2002] Jiang, F., Lin, W. & Rao, Z. (2002) Softdock: understanding of molecular recognition through a systematic docking study. *PROTEIN ENGINEERING -OXFORD-,* **15** (4), 257–264.

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

[Kabsch & Sander, 1983] Kabsch, W. & Sander, C. (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers,* **22**, 2577–2637.

[Kämpfe *et al.*, 2002] Kämpfe, T., Käster, T., Pfeiffer, M., Ritter, H. & Sagerer, G. (2002) INDI – Intelligent Database Navigation by Interactive and Intuitive Content-Based Image Retrieval. In *IEEE 2002 International Conference on Image Processing* vol. III, pp. 921–924 IEEE, Rochester, USA.

[Kanehisa & Goto, 2000] Kanehisa, M. & Goto, S. (2000) Kegg: kyoto encyclopedia of genes and genomes. *Nucleic Acids Research,* **28**, 27–30.

[Karplus & Schulz, 1985] Karplus, P. A. & Schulz, G. E. (1985) Prediction of chain flexibility in proteins. *Naturwissenschaften,* **72**, 212–213.

[Kay, 2001] Kay, M. (2001) *XSLT*. Programmer to programmer, 2. edition,, Wrox Press, Birmingham.

[Kini & Evans, 1992] Kini, R. M. & Evans, H. J. (1992) Comparison of protein models minimized by the all-atom and united-atom models in the amber force field: correlation of rms deviation with the crystallographic r factor and size. *Journal of Biomolecular Structure & Dymanics,* **10** (2), 265–279.

[Klein *et al.*, 1996] Klein, T., Ackermann, F. & Posch, S. (1996) viwish: a visualisation server for protein modelling and docking. *Gene–COMBIS,* **Gene 183**, GC51–GC58.

[Koch, 2003] Koch, K. (2003). *Statistical analysis of amino acid side chain flexibility for 1:n Protein-Protein docking*. Dissertation Universität Bielefeld, Technische Fakultät.

[Koch *et al.*, 2002] Koch, K., Zöllner, F., Neumann, S., Kummert, F. & Sagerer, G. (2002) Comparing bound and unbound protein structures using energy calculation and rotamer statistics. *In Silico Biology,* **2**, 32.

[Koch *et al.*, 2001] Koch, K., Zöllner, F. & Sagerer, G. (2001) Building a new rotamer library for protein-protein docking using energy calculations and statistical approaches. In *Computer Science and Biology: Proceedings of the German Conference on Bioinformatics* pp. 201–202, Braunschweig.

[Kohlbacher, 2000] Kohlbacher, O. (2000). *New approaches to protein docking*. PhD thesis, University Saarbrücken.

[Kohlbacher *et al.*, 2001] Kohlbacher, O., Burchardt, A., Moll, A., Hildebrandt, A., Bayer, P. & Lenhof, H.-P. (2001) Structure prediction of protein complexes by a nmr-based protein docking algorithm. *Journal of Biomolecular NMR,* **20**, 15–21.

[Kosata, 2003] Kosata, B. (2003). Bkchem – a free chemical drawing program. `http://www.nongnu.org/bkchem/`. Link 10.11.2003.

[Koshland, 1958] Koshland (1958) Application of a theory of enzyme specificity to protein synthesis. *Proc. Natl. Acad. Sci. USA.,* **44**, 98–104.

[Lazaridis & Karplus, 2001] Lazaridis, T. & Karplus, M. (2001) Effective energy functions for protein structure prediction. *Current Opinion in Structural Biology,* **10**, 139–145.

[Leach, 1996] Leach, A. R. (1996) *Molecular Modelling Principles and Applications*. Pearson Education Limited.

[Leach & Lemon, 1998] Leach, A. R. & Lemon, A. P. (1998) Exploring the conformational space of protein side chains using dead-end elemination an the a* algorithm. *Proteins: Structure, Function, and Genetics,* **33**, 227–239.

[Lee & Richards, 1971] Lee, B. & Richards, F. (1971) The interpretation of protein structures: estimation of static accessibility. *Journal of Molecular Biology,* **55**, 379–400.

[Lengauer *et al.*, 1999] Lengauer, T., Rarey, M. & Zimmer, R. (1999) Bioinformatik - diagnose von krankheiten und entwicklung von wirkstoffen mit hilfe des computers. *Spektrum der Wissenschaft,* **Dossier: Software**, 38–42.

[Lenhof, 1995] Lenhof, H.-P. (1995) An algorithm for the protein docking problem. In *Bioinformatics: From Nucleic Acids and Proteins to Cell Metabolism*, (Schomburg, D. & Lessel, U., eds),. pp. 125–139.

[Lenhof, 1997] Lenhof, H.-P. (1997) New contact measures for the protein docking problem. In *Proc. of the First Annual International Conference on Computational Molecular Biology RECOMB 97* pp. 182–191.

[Lii & Allinger, 1991] Lii, J.-H. & Allinger, N. L. (1991) The mm3 force field for amides, polypeptides and proteins. *Journal of Computational Chemistry,* **12**, 186–199.

[Lorber *et al.*, 2002] Lorber, D. M., Udo, M. K. & Shoichet, B. K. (2002) Protein-protein docking with multiple residue conformations and residue substitutions. *PROTEIN SCIENCE,* **11** (6), 1393–1408.

[Loughran, 2002] Loughran, S. (2002). Ant in anger: using apache ant in a production development system. `http://ant.apache.org/ant_in_anger.html`.

[Lovell *et al.*, 2000] Lovell, S. C., Word, M., Richardson, J. S. & Richardson, D. C. (2000) The penultimate rotamer library. *Proteins: Structure, Function, and Genetics,* **40**, 389–408.

[Mallat, 1989] Mallat, S. G. (1989) A theory for multiresolution signal decomposition: The wavelet Representation. *IEEE Transactions on Pattern Analysis and machine intelligence,* **11** (7).

[Mao & McCammon, 1984] Mao, B. & McCammon, J. A. (1984) Structural study of hinge bending in l-arabinose-binding protein. *Journal of Biological Chemistry,* **259**, 4964–4970.

[Martínez-Cruz *et al.*, 2003] Martínez-Cruz, L. A., Rubio, A., Martínez-Chantar, M. L., Labarga, A., Barrio, I., Podhorski, A., Segura, V., Campo, J. L. S., Avila, M. A. & Mato, J. M. (2003) GARBAN: genomic analysis and rapid biological annotation of cDNA microarray and proteomic data. *Bioinformatics,* **19** (16), 2158–2160.

[Mc Gregor *et al.*, 1987] Mc Gregor, M., Islam, S. & Sternberg, M. J. E. (1987) Analysis of the relationship between side-chain conformation and secondary structure in globular proteins. *Journal of Molecular Biology,* **198**, 295–310.

[McCammon *et al.*, 1976] McCammon, J. A., Gelin, B. R., Karplus, M. & Wolynes, P. G. (1976) The hinge-bending mode in lysozyme. *Nature,* **262**, 325–326.

[McLoughlin, 2003] McLoughlin, L. (2003). Mirror 2.9 reference manual. `http://sunsite.org.uk/packages/mirror/`. Link 29.10.2003.

[Metz & Pan, 1999] Metz, C. E. & Pan, X. (1999) "Proper" binormal roc curves: theory and maximum–likelihood eximation. *Journal of Mathematical Psychology,* **43**, 1–33.

[Morris *et al.*, 1996] Morris, G. M., Goodsell, D. S., Huey, R. & Olson, A. J. (1996) Distributed automated docking of flexible ligands to proteins: parallel applications of autodock 2.4. *Journal of Computer-Aided Molecular Design,* **10**, 293–304.

[Murzin *et al.*, 1995] Murzin, A. G., Brenner, S. E., Hubbard, T. & Chothia, C. (1995) Scop: a structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology,* **247**, 536–540.

[Nagata *et al.*, 2002] Nagata, H., Mizushima, H. & Tanaka, H. (2002) Concept and prototype of protein-ligand docking simulator with force feedback technology. *Bioinformatics,* **18** (1), 140–146.

[Najmanovich *et al.*, 2000] Najmanovich, R., Kuttner, J., Sobolev, V. & Edelmann, M. (2000) Side-chain flexibility in proteins upon ligand binding. *Proteins: Structure, Function, and Genetics,* **39**, 261–268.

[NC-IUBMB, 1992] NC-IUBMB (1992). Nomenclature Committee of the International Union of Biochemistry and Molecular Biology, Enzyme Nomenclature. `http://www.chem.qmul.ac.uk/iubmb/enzyme/EC1/cont1aa.html`. Link 21.05.2003.

[Neumann, 2003] Neumann, S. (2003). *Soft volume models for protein-protein docking*. Dissertation Universität Bielefeld, Technische Fakultät.

[Neumann *et al.*, 2002] Neumann, S., Zöllner, F., Koch, K., Kummert, F. & Sagerer, G. (2002) Elmar: a protein docking system using flexibility information. In *European Conference on Computational Biology 2002, Poster Abstracts* pp. 169–170, Saarbrücken.

[New York University Scientific Visualization Center, 2003] New York University Scientific Visualization Center (2003). Mathmol library. `http://www.nyu.edu/pages/mathmol/library/`. Link 10.11.2003.

[Norrby *et al.*, 1996] Norrby, P.-O., Petterson, I., Liljefors, T. & Gundertofte, K. (1996) A comparison of conformational energies calculated by several molecular mechanics methods. *Journal of Computational Chemistry,* **17**, 429–449.

[Norrel *et al.*, 1999] Norrel, R., Petrey, D., Wolfson, H. J. & Nussinov, R. (1999) Examination of shape complementarity in docking of unbound proteins. *Proteins: Structure, Function, and Genetics,* **36** (3), 307–317.

[Ogata & Umeyana, 1998] Ogata, K. & Umeyana, H. (1998) The role played by enviromental residues on sidechain torsional angles within homologous families of proteins: a new method of sidechain modeling. *Proteins: Structure, Function, and Genetics,* **31** (4), 355–369.

[Orengo *et al.*, 1997] Orengo, C., Michie, A., Jones, S., Jones, D., Swindells, M. & Thornton, J. (1997) Cath- a hierarchic classification of protein domain structures. *Structure,* **5** (8), 1093–1108.

[Orengo *et al.*, 1999] Orengo, C. A., Pearl, F. M. G. & Bray, J. E. (1999) The cath database provides insights into protein structure/function relationships. *Nucleic Acids Research,* **27** (1), 275–279.

[Pacios, 2001] Pacios, L. F. (2001) Distinct molecular surfaces and hydrophobicity of amino acid. *JOURNAL OF CHEMICAL INFORMATION AND COMPUTER SCIENCES,* **41** (5), 1427–1435.

[Pearlstein & FitzGerald, 1996] Pearlstein, R. & FitzGerald, P. (1996). Pdb–at–a–glance. `http://cmm.info.nih.gov/modeling/pdb_at_a_glance.html`. Link 11.12.2001.

[Pillardy *et al.*, 2001] Pillardy, J., Czaplewski, C., Liwo, A., Lee, J., Ripoll, D. R., Kazmierkiewicz, R., Oldziej, S., Wedemeyer, W. L., Gibson, K. D., Arnautova, Y. A., Saunders, J. & Ye, Y.-J. (2001) Recent improvements in prediction of protein structure by global optimazation of a potential energy function. *Proceedings of the National Academy of Sciences,* **98** (5), 2329–2333.

[Ponder & Richards, 1987] Ponder, J. W. & Richards, F. M. (1987) Tertiary templates for proteins: use of packing criteria in the enumeration of allowed sequences for different structural classes. *Journal of Molecular Biology,* **193**, 775–791.

[Ponstingl *et al.*, 2000] Ponstingl, H., Henrick, K. & Thornton, J. M. (2000) Discriminating between homodimeric and monomeric proteins in the crystalline state. *Proteins,* **41**, 47–57.

[Rarey, 1996] Rarey, M. (1996). *Rechnergestützte Vorhersage von Rezeptor-Ligand Wechselwirkungen*. PhD thesis, GMD-Bericht 268.

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

[Research Collaboratory for Structural Bioinformatics (RCSB), 2003] Research Collaboratory for Structural Bioinformatics (RCSB) (2003). Pdb current holdings. `http://www.rcsb.org/pdb/holdings.html`. Link 6.1.2004.

[Rini *et al.*, 1990] Rini, J., Schulze-Gahmen, U. & Wilson, I. (1990) Structural evidence for induced fit as a mechanism for antibody–antigen recognition. *Science,* **255**, 959–965.

[Rosen *et al.*, 2000] Rosen, J., Phillips, A. T., Oh, S. Y. & Dill, K. A. (2000) A method for parameter optimiziation in computational biology. *Biophysical Journal,* **79**, 2818–2824.

[Rui *et al.*, 1998] Rui, Y., Huang, T. S. & Mehrotra, S. (1998) Relevance feedback techniques in interactive content-based image retrieval. In *Proc. of Storage and Retrieval for Image and Video Databases (SPIE)* pp. 25–6.

[Salton & McGill, 1983] Salton, G. & McGill, M. J. (1983) *Introduction to Modern Information Retrieval*. McGraw-Hill Advanced Computer Science Series.

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

[Schomburg, 2003] Schomburg, D. (2003). Brenda, the comprehensive enzyme information system. `http://www.brenda.uni-koeln.de`. Link 16.04.2003.

[Schrauber *et al.*, 1993] Schrauber, H., Eisenhaber, F. & Argos, P. (1993) Rotamers: to be or not to be? An analysis of amino acid side-chain conformations in globular proteins. *Journal of Molecular Biology,* **230**, 592–612.

[Scott *et al.*, 1999] Scott, W. R., Hünenberger, P. H., Tironi, I. G., Mark, A. E., Billeter, S. R., Fennen, J., Torda, A. E., Huber, T., Krüger, P. & van Gunsteren, W. F. (1999) The gromos biomolecular simulation program package. *The Journal of Physical Chemistry A,* **103**, 3596–3607.

[Stone *et al.*, 2001] Stone, J. E., Gullingsrud, J., Schulten, K. & Grayson, P. (2001) A system for interactive molecular dynamics simulation. In *2001 ACM Symposium on Interactive 3D Graphics*, (Hughes, J. F. & Sequin, C. H., eds), pp. 191–194 ACM SIGGRAPH, New York.

[Stroustrup, 1998] Stroustrup, B. (1998) *The C++ Programming Language*. Addison-Wesley, Reading, Massachusetts, USA.

[Stryer, 1996] Stryer, L. (1996) *Biochemie*. Spektrum.

[Swets, 1988] Swets, J. A. (1988) Measuring the accuracy of diagnostic systems. *Science,* , 1285–1293.

[Tapia & Velazquez, 1997] Tapia, O. & Velazquez, I. (1997) Molecular dynamics simulation of dna with protein's consistent gromos force field and the role of countions' symmetry. *Journal of the American Chemical Society,* **119**, 5934–5938.

[Thornton, 2003] Thornton, J. (2003) The proteome and the metabolome. In *European Conference on Computational Biology 2003* p. ii237, Paris.

[Torgasin, 2003] Torgasin, S. (2003). Untersuchung der Auswirkung von lokalen Konformationsänderungen in Proteinen. Master's thesis ,Bielefeld University.

[Trad *et al.*, 2002] Trad, C. H. d., Fang, Q. & Cosic, I. (2002) Protein sequence comparison based on the wavelet transform approach. *Protein Engineering,* **15** (3), 193–203.

[Tuffery *et al.*, 1997] Tuffery, P., Etchebest, C. & Hazout, S. (1997) Prediction of protein side chain conformations: a study on the influence of backbone accuracy on conformation stability in the rotamer space. *Protein Engineering,* **10** (4), 361–372.

[Ulrich *et al.*, 1997] Ulrich, P., Scott, W., van Gunsteren, W. F. & Torda, A. E. (1997) Protein structure prediction force fields: parametrization with quasi-newtonian dynamics. *Proteins: Structure, Function, and Genetics,* **27**, 367–384.

[Urzhumtsev *et al.*, 1997] Urzhumtsev, A., Tête-Favier, F., Mitschler, A., Barbanton, J., Barth, P., Urzhumtseva, L., Biellmann, J.-F., Podjarny, A. D. & Moras, D. (1997) A 'specificity' pocket inferred from the crystal structures of the complexes of aldose reductase with the pharmaceutically important inhibitors tolrestat and sorbinil. *Structure,* **5**, 601–612.

[van Erkel & Pattynama, 1998] van Erkel, A. & Pattynama, P. M. T. (1998) Receiver operating characteristic (roc) analysis: basic principles and applications in radiology. *European Journal of Radiology,* **27** (2), 88–94.

[Venter *et al.*, 2001] Venter, J. C., Adams, M. D., Myers, E. W., Li, P. W., Mural, R. J., Sutton, G. G., Smith, H. O., Yandell, M., Evans, C. A., Holt, R. A. *et al.* (2001) The sequence of the human genome. *Science,* **5507**, 1304–1351.

[Verbitsky *et al.*, 1999] Verbitsky, G., Nussinov, R. & Wofson, H. (1999) Flexible structural comparision allowing hinge-bending, swiveling motions. *Proteins: Structure, Function, and Genetics,* **34**, 232–254.

[Wall *et al.*, 2002] Wall, L., Christiansen, T. & Orwant, J. (2002) *Programmieren mit Perl*. 2nd edition,, O'Reilly, Beijing.

[Walls & Sternberg, 1992] Walls, P. H. & Sternberg, M. J. E. (1992) New algorithm to model protein-protein recognition based on surface complementarity. *Journal of Molecular Biology,* **228**, 277–297.

[Ward, 2001] Ward, J. M. (2001) Identification of novel families of membrane proteins from the model plant Arabidopsis thaliana. *Bioinformatics,* **17** (6), 560–563.

[Weber & Steitz, 1987] Weber, I. & Steitz, T. (1987) Structure of a complex of catabolic gene activator protein and cyclic amp refined at 2.5å resolution. *Journal of Molecular Biology,* **198**, 311–326.

[Weiner *et al.*, 1984] Weiner, S. J., Kollmann, P. A., Case, D., Chandra Singh, U., Ghio, C., Alagona, G., Profeta, S. & Weiner, P. (1984) A new force field for molecular mechanical simulation of nucleic acids and proteins. *Journal of the American Chemical Society,* **106**, 765–784.

[Weiner *et al.*, 1986] Weiner, S. J., Kollmann, P. A., Nguyen, D. T. & Case, D. A. (1986) An all atom force field for simulations of proteins and nucleic acids. *Journal of Computational Chemistry,* **7**, 230–252.

[Wilson *et al.*, 1993] Wilson, C., Greoret, L. M. & Agard, D. A. (1993) Modeling side-chain conformation for homologous proteins using an energy-based rotamer search. *Journal of Molecular Biology,* **229**, 996–1006.

[Zhang *et al.*, 1997] Zhang, C., Vasmatzis, G., Cornette, J. L. & DeLisi, C. (1997) Determination of atomic desolvation energies from the structures of crystallized proteins. *Journal of Molecular Biology,* **267**, 707–726.

[Zhang *et al.*, 2003] Zhang, S.-W., Pan, Q., Zhang, H.-C., Zhang, Y.-L. & Wang, H.-Y. (2003) Classification of protein quaternary structure with support vector machine. *Bioinformatics,* **19** (18), 2390–2396.

[Zhao *et al.*, 2001] Zhao, S., Goodsell, D. S. & Olson, A. J. (2001) Analysis of a data set of paired uncomplexed protein structures: new metrices for side-chain flexibility and model evaluation. *Proteins: Structure, Function, and Genetics,* **43**, 271–279.

[Zien *et al.*, 2000] Zien, A., Zimmer, R. & Lengauer, T. (2000) A simple iterative approach to parameter optimization. *Journal of Computational Biology,* **7** (3/4), 483–501.

[Zöllner *et al.*, 2002] Zöllner, F., Neumann, S., Koch, K., Kummert, F. & Sagerer, G. (2002) Calculating residue flexibility information from statistics and energy based prediction. In *European Conference on Computational Biology 2002, Poster Abstracts* pp. 275–276, Saarbrücken.

[Zöllner *et al.*, 2003] Zöllner, F., Neumann, S., Koch, K., Kummert, F. & Sagerer, G. (2003) Iphex: a system for evaluating protein docking hypotheses using user feedback. In *European Conference on Computational Biology 2003* pp. 214–216, Paris.

[Zöllner, 2001] Zöllner, F. (2001). Bewertung der Flexibilität von Aminosäureseitenketten in Proteinkonformationen durch empirische Energiefelder. Master's thesis ,Bielefeld University.

[Zöllner *et al.*, 2004] Zöllner, F., Neumann, S., Kummert, F. & Sagerer, G. (2004) Database driven test case generation for protein-protein docking. *Bioinformatics,* . to appear.

[Zuegg & Gready, 2000] Zuegg, J. & Gready, J. E. (2000) Molecular dynamics simulation of human prion protein including both n-linked oligosaccharides and gpi anchor. *Glycobiology,* **10** (10), 959–974.

Index

X

Z

Versicherung

Hiermit versichere ich, daß ich die vorliegende Dissertation selbständig erarbeitet und keine anderen als die angegebenen Quellen benutzt habe. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche kenntlich gemacht.

Bielefeld, 18. Juli 2004

Frank Gerrit Zöllner