Structure-based algorithms for protein-protein interactions

Georgy Derevyanko

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Physique pour les Sciences du Vivant**

Arrêté ministériel : 7 août 2006

Présentée par

Georgy Derevyanko

Thèse dirigée par **Valentin GORDELY**
et codirigée par **Sergei GRUDININ**

préparée au sein **Groupe des Transporteurs Membranaires de l'Institut de Biologie Structurale**
et de **l'École doctorale de physique**

Structure-based algorithms for protein-protein interactions

Thèse soutenue publiquement le ,
devant le jury composé de :

Martin ENGELHARD
professeur, Max Planck Institut of Molecular Physiology, Rapporteur

Georg BÜLDT
professeur, Moscow Institute of Physics and Technology, Rapporteur

Giuseppe ZACCAI
professeur, Institut Laue-Langevin, Examinateur

Grégory DURAND
PhD, Maître de Conférences, Université d'Avignon et des Pays de Vaucluse , Examinateur

Valentin GORDELIY
professeur, Institut de Biologie Structurale, Directeur de thèse

Sergei GRUDININ
PhD Charge de Recherche, Inria Rhone-Alpes Research Center et Laboratoire Jean Kuntzman, CNRS, Co-Directeur de thèse

Abstract

The phenotype of every known living organism is determined mainly by the complicated interactions between the proteins produced in this organism. Understanding the orchestration of the organismal responses to the external or internal stimuli is based on the understanding of the interactions of individual proteins and their complexes structures. The prediction of a complex of two or more proteins is the problem of the protein-protein docking field. Docking algorithms usually have two major steps: exhaustive 6D rigid-body search followed by the scoring. In this work we made contribution to both of these steps. We developed a novel algorithm for 6D exhaustive search, HermiteFit. It is based on Hermite decomposition of 3D functions into the Hermite basis. We implemented this algorithm in the program for fitting low-resolution electron density maps. We showed that it outperforms existing algorithms in terms of time-per-point while maintaining the same output model accuracy. We also developed a novel approach to computation of a scoring function, which is based on simple logical arguments and avoids an ambiguous computation of the reference state. We compared it to the existing scoring functions on the widely used protein-protein docking benchmarks. Finally, we developed an approach to include water-protein interactions into the scoring functions and validated our method during the Critical Assessement of Protein Interactions round 47.

Dedication

To my parents.

Acknowledgements

I would like to thank my supervisor Valentin Gordely and co-supervisor Sergey Grudinin for their guidance during my thesis. I also thank my parents, brother and all the people I was lucky to work with for their support.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 Role of proteins and their interactions in the cell

The main dogma of biology states: information in a cell flows from DNA to RNA and then to proteins. Although the actual working of the cell is much more comlpex to be contained in any dogma, it captures a large picture of the state-of-art of our understanding of the working of a cell. Proteins in this picture have a crucial place: they are the cogs of the cell machinery. These are the protein expressed in a cell and their interactions that mainly define phenotype of that cell. Therefore gainig new data on the protein interaction in a cell profoundly enhances our capabilities to understand and change living organisms for the needs of the society.

1.1.1 Classification of protein-protein interactions

Commonly used classification of the protein-protein complexes uses the following main criteria: composition of a complex, its lifetime and stability [94].

1.1.1.1 Protein composition and interface

The complex can be comprised of two identical or highly homologous proteins. An example of such a complex is the D-alanyl carrier protein (pdb code 4BPG), which consists of two identical polypeptide chains (Fig. 1.1). The proteins of this class are called homo-oligomers. On the other hand, if the complex consists of two distinct proteins it is named hetero-oligomer. An example of hetero-dimer (Glycosidase CelD bound to artificial affitin E12 protein, [28], pdb code 4CJ0) is shown on Fig 1.2. Homo-oligomers are further divided dependent on the interface of oligomerization. If the two chains bind using the identical binding interace, their complex is called isologous. We can see that D-alanyl carrier protein is an example of isologous homodimer (Fig. 1.2). On the contrary, the homo-oligomeric assemblies where proteins bind through distinct interfaces are called heterologous.

1.1.1.2 Stability of protomers

The proteins that form a complex can be either stable or not on their own *in vivo*. In the first case, when a complex is formed out of stable proteins it is called *non-*

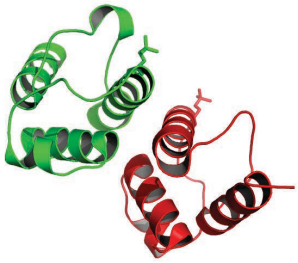


Figure 1.1: Example of a homodimeric complex, D-alanyl carrier protein, pdb code 4BPG.

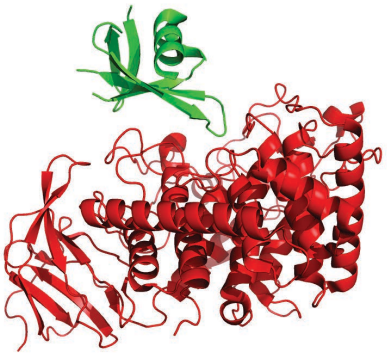


Figure 1.2: Example of a heterodimeric complex, Glycosidase CelD bound to artificial affitin E12 protein, [28], pdb code 4CJ0

obligate. In many cases these proteins are localized in different compartements of a cell. Usually, interactions between receptor-ligand, enzyme-inhibitor, *etc* belong to non-obligate interactions. In the other case when a complex of one or both chains do not exist as folded proteins on their own, it is called *obligate*. An example of such a complex is the Arc repressor dimer, which consists of two peptide chains. Upon dimerization (catalized by DNA [82]) these chains obtain stable secondary structure, that was absent in the monomer.

1.1.1.3 Complex lifetime

If two proteins form a very stable complex they said to interact *permanently*. In many cases obligate complexes, such as Arc repressor dimer, are permanent. On the other hand, if a complex is dissociated and formed continuously *in vivo*, the interaction between its constituting proteins is called *transient*. Many of non-obligate interactions are also transient. However, there are strong transient interactions that require the presense of some molecule for the complex dissociation. An example of a strong transient interaction is the formation of trimeric G protein, for which guanine triphosphate is the dissociation trigger.

1.2 Experimental methods to probe protein-protein interactions

Giving importance and variety of protein-protein interactions in a cell, numerous methods were developed to discover the fact that two or more proteins form a complex by measuring kinetics and free energy of its formation or even by solving the atomic structure of the complex. The openness of the proteomics community also resulted in the construction of many databases of protein-protein interactions. All experimental methods of identification and characterization of protein-protein interactions can be classified using two parameters: scale(high-throuput, individual) and system(*in vivo*, *in vitro*). Usually, a low-throuput approach gives much more information about interaction properties, whereas a high-throuput one only identifies the presense or absense of an interaction.

1.2.1 Yeast two-hybrid method (Y2H)

The Y2H is probably the oldest and most widespread method to identify interaction of proteins *in vivo*, which was scaled up to proteome level. In the pioneering work by Fields S. and Song O. [39] it was shown that if the domain of the transcription activator that directs binding to a promoter (BD) is separated from a domain that activates transcription from this promoter (AD), the transcription is deactivated. Based on this principle, two proteins of interest are fused to BD and AD, respectively, and the transcription of the reporter gene signals if they interact (Fig 1.3).

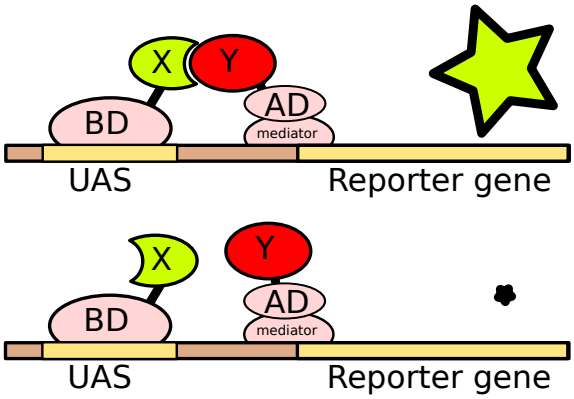


Figure 1.3: Schematic representation of the Y2H method. If the two protein X and Y do interact, expression of the reporter gene is high (yellow star), otherwise AD and BD domains of the promoter are separated and the expression is low. UAS stands for the upstream regulating sequence.

The are at least two scaling schemes: matrix and library approaches [138, 61]. In the first one, clones expressing different proteins X_i are taken and plated each in the rows of wells on a plate. The same set of clones is plated in colums of wells on a plate. Then, when the two yeast cells in each well mate and make a diploid that expresses both proteins X_i and X_j. Afterwards, the expression of a reporter gene in a cell i, j means that proteins X_i and X_j interact. In the library-based approach, clones containig protein X_i are screened agains a library of clones with various proteins, which can be expressed from random cDNA or all open reading

frames of a certain genome. In this case, the diploids expressing interacting proteins are selected agains specific growing media. The proteins that interact with X_i are determined by the DNA sequencing.

1.2.2 Mass spectrometry and tandem affinity purification

In this method, a certain protein is fused with the DNA sequence that expresses the TAP tag (IgG binding domains of *Staphylococcus* protein A and calmodulin binding peptide separated by the TEV protease site [108]). When the DNA construct is expressed in a host, it produces the protein of interest that forms complexes with other proteins of the host. During the purification, these protein complexes bind to the IgG matrix. Other proteins that did not form complexes with the TAP-tagged one, hop through the matrix. Afterwards, using TEV protease, one cleaves of IgG binding domains and purified complexes are eluated from the matrix. The second purification step is the binding to calmodulin-coated beads. The simple representation of the two-step purification is shown on Fig. 1.4. Finally, the eluate is loaded to the SDS-PAGE gel and the resulting bands are cleaved by proteases.

After the purification, one uses mass-spectrometry to identify the fragments of the cleaved protein-protein complexes [31]. Mass-spectrometry identifies particles based on their charge-to-mass ration. It allows to recognize fingerprints of short peptides and therefore identify the proteins using the solution of peptide fragments. A certain advantage of the TAP-MS method over Y2H is that it can detect not only dimeric protein-protein interaction, but also multimeric complexes.

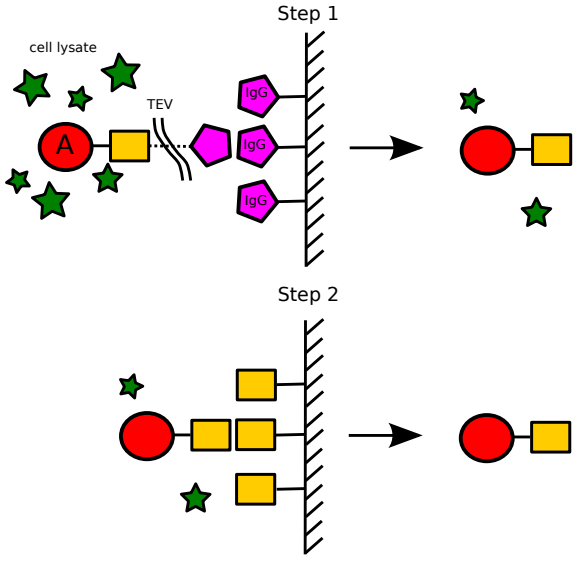


Figure 1.4: Schematic representation of the tandem affinity pyrification method. Contaminants are shown with green stars, IgG binding domain is shown with pink pentagon (as well as the IgG coating) and calmodulin (as well as calmodulin coating) is shown with squares.

1.2.3 Gene co-expression

Recently the methods that allowed measuring expression of the genes on the scale of the whole cell were invented. Using this methodology, one can measure the similarity

of expression profiles over different conditions of the two or more genes that code for interacting proteins. It turns out that they are significantly more similar than the gene expression profiles of random non-interacting proteins [63].

1.2.4 Synthetic lethality

The mutations in the genome of an organism affect its phenotype. In many cases this phenotype alternation is caused by the change in the protein-protein interaction network. By introducing random mutations or deletions in the genes of two proteins, one can monitor survival rate of these cells. The lethality of one such mutation can point to presence of interaction between the two target proteins [95].

1.2.5 Fluorescence resonance energy transfer

The fluorescence resonance energy transfer (FRET) occurs between two molecules: one (donor) in an excited state and the orther (aceptor) in the ground state. The energy is transfered through the dipole-dipole interaction between the molecules and does not involve emission [152]. The probability of such transfer to occur strongly depends on the distance between the donor and the aceptor. However, it is almost independent of the environmental conditions, which makes this method a great tool to study molecular interactions *in vivo*. More specifically, the two fluorophores are fused to the two proteins of interest (Fig. 1.5). One is then excited using a laser and if these molecules form a complex, it transfers energy to the second fluorophore that emits light at a certain wavelength. One of examples of FRET application is the investigation of membrane proteins dimerization *in vivo*, in particular of melatonin receptor types 1A and 1B [9].

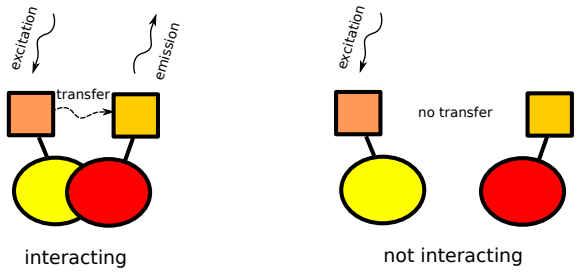


Figure 1.5: Schematic representation of the fluorescence resonance energy transfer method. If the two proteins interact, the fluorophores (blocks) come close and the energy transfer between them becomes possible. Therefore, exciting one fluorophore, one detects the emission from another. Otherwise, if the proteins do not interact, no emission of the second fluorophore is detected upon excitation of the first one.

1.2.6 Isothermal titration calorimetry

This method allows to measure stoichiometry, dissociation constant, enthalpy and entrophy of the binding reaction between two proteins [144]. The experimental setup consists of two thermally isolated chambers. One of the chambers is used as a reference and is filled with water. The other contains one of the interacting proteins. Its interaction partner is titrated in a known amount to the chamber. The thermometer measures temperature difference between these two chambers and controls

heaters to equilibrate their temperature (while maintaining the temperature of the reference chamber constant). The amount of heat spent to make the two chamber isothermic is measured during the experiment. Figure 1.6 shows a schematic example of data obtained from an ITC experiment. The red line shows the peaks of energy transfer that correspond to titration events. Fitting these peaks with the interaction model (blue dashed line on Fig. 1.6) of a titration substance and a substance in the reservoir one can deduce the parameters of the model. In the simplest case of protein-ligand interaction one obtains enthalpy, dissociation constant and stoichiometry of the reaction.

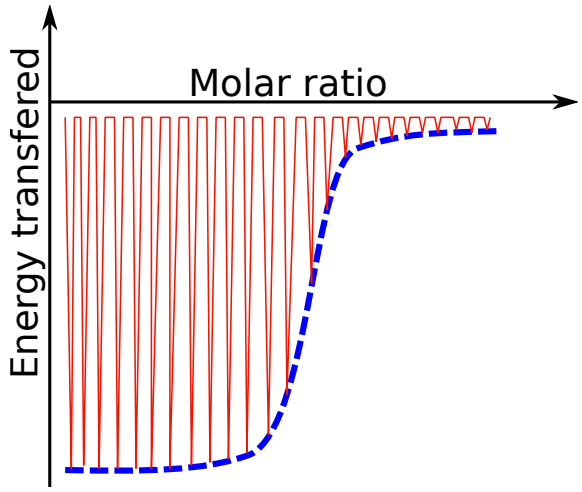


Figure 1.6: Schematic representation of the data obtained from an isothermal titration calorimetry device. The red line shows the energy transfered from the reservoir with the protein to the reference reservoir. Blue dashed line is an example of model fitted to the data.

1.2.7 Nuclear magnetic resonance spectroscopy (NMR)

This method is based on the absorbtion and emission of radio-frequency radiation by the nuclei of certain atoms. The emission and absorbtion spectra depend on the environment of nuclei and therefore the measurement allows to reconstruct the distance matrix between certain atoms in a protein in solution. This method also allows to measure the dynamics of proteins and their interaction. The classical approach is based on the Nuclear Overhauser Effect (NOE). One of the examples of the NMR results is the structure of the Twist protein, a transcription factor that plays a key role in the epithelial-mesenchymal transition and the bromodomain-containing protein 4 [124] (see Fig. 1.7). Due to the complexity of spectra measured during the experiment, this approach is limited to protein complexes of the size up to 300 amino acids.

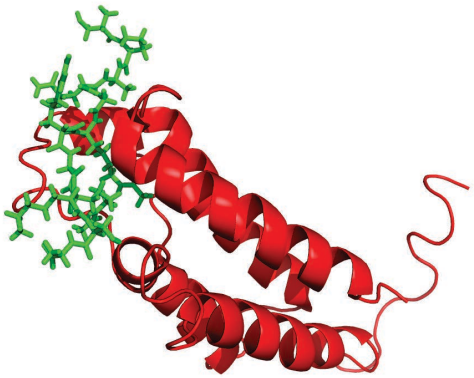


Figure 1.7: The structure of Twist protein(green) with the bromodomain-containing protein 4 (red) measured using NOE-NMR technique [124]. PDB code 2MJV.

1.2.8 Cryo electron microscopy

The broad variety of methods go under the name of Cryo electron microscopy (Cryo-EM). The basic principle underlying this method is the same as in the conventional light microscopy. The difference lies in the radiation source. In Cryo-EM, the electron beam accelerated up to 300kV is used. This allows increasing the resolution that depends on the wavelength of the incident radiation, up to atomic one (wavelength of electrons accelerated by 300kV is about 0.02Å). However, the increased resolution goes with the increased radiation dammage to the sample. Variety of methods are used to diminish ionizing effect of electrons on the biological samples. More precisely, measurement are conducted at low temperatures [34], averaging many identical units [132], single-particle microscopy. The last one is probably the most used one in structural biology.

The single-particle Cryo-EM is based on collecting information from many 2D projections of an object to reconstruct its 3D low-resolution model (electron density map, EDM). Afterwards, the supplementary information provided by NMR of X-Ray crystallography is used to construct an atomistic model of the object.

The 3D reconstruction of the EDM is usually based on the central projection theorem. It states that the Fourier images of 2D projections of a 3D object are the central slices of its 3D Fourier images. The relative positioning of two projections can be derived from the common lines in their Fourier images. These algorithms are implemented in the programs like IMAGIC [140], SPIDER [121], FREALIGN [50] *etc.*

The resolution that can be obtained using this technique is highly dependent on the symmetry of the object measured. For example, highly symmetric icosahedral viruses envelopes were measured with up to 3Å resolution [155]. The other examples of reconstructed protein complexes usually have resolution in the range of 7Å-15Å. This method is the main source of information on the large protein-protein assemblies today, like ribosomes [99] and chperonines [27].

1.2.9 X-ray crystallography

The X-ray crystallography is the most time- and cost- demanding method, but it provides the most detailed atomistic information on the structure of protein-protein complexes. The starting point of this technique is a protein crystal. Obtaining crystal of a certain protein-protein complex is often the major hurdle and sometimes even impossible. Large protein-protein complexes and membrane proteins are especially hard to crystallize. Nontheless, the X-ray crystallography remains a major method to study structures of protein-protein complexes. The method is based on the diffraction of X-rays on the atoms, arranged in a lattice. The crystal of a protein is rotated with respect to the incident beam and the diffraction patterns are measured for each rotation. From these patterns one can reconstruct the absolute values of the Fourier image of a unit cell of the crystal. Afterwards, the phases of the Fourier image have to be reconstructed or measured. One of the most used way to obtain the phasing is the so-called molecular replacement. This means that approximate theoretical molecular structure of the unit cell is fitted into the given dataset. Afterwards, one minimizes discrepancy of the fitted structure with the diffraction pattern and obtains the final one. The quality of the final structure is judged by the R-factor. It shows to what extent the resulting structure explains the diffraction peaks.

1.3 Computational methods to probe protein-protein interactions

The experimental methods described in the previous section give rich information on the protein-protein interactions. However, in many cases the interactions being identified using these techniques are incomplete and contradictory owing to certain limitations and biases of the experimental conditions. To validate and cover the unknown spots in the protein-protein interactome maps, a plethora of computational techniques is used. These methods rely on different assumptions and use different information sources to decipher interaction details of proteins. Despite the rich variety one can approximatelly classify them in two major groups: top-down methods that use whole-organism or even evelutionary information and bottom-up, methods that employ the knowledge of single protein structures. They also differ by the amount of information they provide: ranging from a simple fact of interaction down to the details and precise conformation of the interaction interface.

1.3.1 Top-down approaches

This class of methods use the evolutionary and genomic data to predict if two proteins interact and identify domains that contain the interaction interface.

1.3.1.1 Gene neighbour and gene cluster methods

This pack of methods rely on the assumption that genes encoding for possibly interacting proteins often transcribed as a single operon in procariotes or co-regulated in eukariotes. A simplified example on how co-regulation maintains a certain stechiometry of a complex is shown on Fig. 1.8. For example, it was found that up to 75% of co-regulated genes in bacterial and achaeal genomes interact [30]. The evolution tends to shuffle the order of genes in the distantly related organisms, however co-regulated gene clusters are found to be conserved. A prominent application of this method is the prediction of interaction of exsosome complex, that is capable of degrading viral RNA, and the RNAse P complex by comparing the order of genes in archaeal and eukariotic genomes [72].

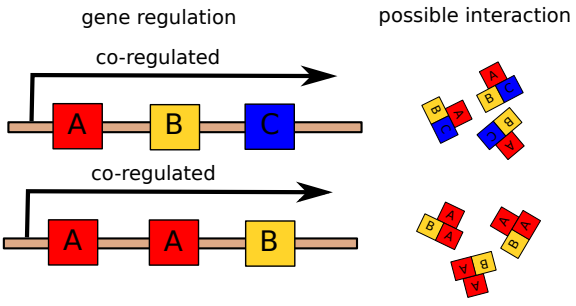


Figure 1.8: Example of a gene co-regulated cluster and the probable complex with the stechiometry set by the co-regulation.

1.3.1.2 Phylogenetic profile methods

These methods are based on the hypothesis that the interacting proteins coevolve and therefore have orthologous proteins among sequenced organisms [10]. If the proteins related to the two proteins in question are present in majority of organisms therefore, they probably constitute a pathway or physically interact. The phylogenetic profiles are constructed for the proteins where their presense indicated by 1 or 0. Then, these profiles are clustered and the proteins belonging to one cluster are assumed functionally related or interacting.

1.3.1.3 Rosetta Stone method

This method relies on the observation that interacting proteins have homologs in other organisms that are fused into one protein. These types of proteins are called Rosetta Stone proteins. This fusion is the limiting case of the co-expression optimization of functionally related proteins. Using this method Marcotte *et all* [81] identified about 7,000 pair of potentially interacting proteins in *E.Coli* and further analysis of the data revealed that around a half of these pairs are functionally related.

1.3.1.4 Co-evolution based methods

During the evolution, mutations in one of the proteins of a complex should be compensated by the mutations in its partner in order to maintain the function (see Fig.1.9). The co-evolution based methods use the similarity measures between phylogenetic trees of two interacting protein families. Studies showed that some implementations of this method can predict up to 50% of real interactions with false positive rate as low as 6.4% [97].

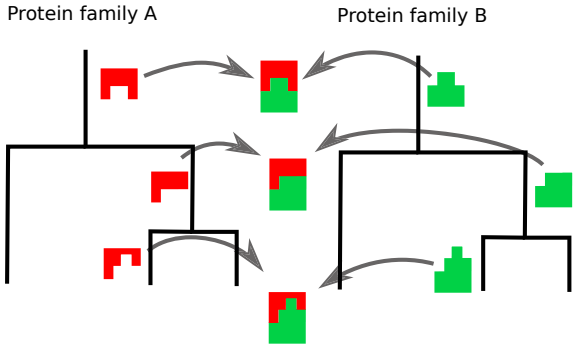


Figure 1.9: Example of a genes co-evolution where evolutionary changes in one protein compensate for the changes in its interaction partner.

1.3.2 Bottom-up approaches

Bottom-up approaches to the prediction of protein-prtoein interactions rely on the known structure of the proteins that constitute the complex. The structures can either be modelled by homology, obtained using evolutionary constraints or taken from such experiments as NMR, X-Ray crystallography or Cryo-EM. These methods are usually applied to infer the missing structural information about the protein-protein complex rather than to predict the fact of interaction. Because bottom-up

approaches use two or more protein structures that they dock into a complex, this class of techniques are usually called docking algorithms. The pioneering work in this field was done by Janin and Wodak [150]. Since then, this field has grown enormously, with its own quality assessment [85, 86, 62]. The two major parts of all the algorithms in this field are: sampling and scoring [110, 55]. As the name suggest, the sampling part generates putative conformations of a complex of proteins and scoring ranks them according to some criteria.

1.3.2.1 Search strategies

Suppose we have two molecules that are assumed to be rigid. One of the molecules (usually the one of a higher molecular weight) is called receptor and is fixed at the origin of the coordinate frame. The other, called ligand, is moved around the receptor. The global search algorithm explores all possible rotations and translations of the ligand movements. Complexity of this problem is $O(N^9)$, where $O(N^3)$ comes from rotational, $O(N^3)$ from the translational degrees of freedom and $O(N^3)$ is the complexity of the integration of the overlap integral that is computed for each rotation and translation of the ligand. A rough estimate gives $\approx 10^{10}$[55] operations. Several approaches allowed reducing this enormous complexity by at least an order of magitude. These are the fast Fourier transform correlation and clever heuristics in the direct search.

1.3.2.1.1 FFT-based rigid body docking The idea to perform correlations in the Fourier space was first used by Katchalski-Katzir and colleagues [68]. In this section I give a short description of this approach with a comprehesive 2D example.

Suppose that proteins receptor ($\mathbf{R}$) and ligand ($\mathbf{L}$) are represented as the numbers on a 3D grid:

$$R(l,m,n) = \begin{cases} 1, (l,m,n) \in \text{surface of } \mathbf{R} \\ \rho, (l,m,n) \in \text{inside of } \mathbf{R} \\ 0, (l,m,n) \in \text{outside of } \mathbf{R} \end{cases} \qquad L(l,m,n) = \begin{cases} 1, (l,m,n) \in \text{surface of } \mathbf{L} \\ \delta, (l,m,n) \in \text{inside of } \mathbf{L} \\ 0, (l,m,n) \in \text{outside of } \mathbf{L} \end{cases} \tag{1.1}$$

where $\rho \ll -1$ and $0 < \delta < 1$, according to the original work by Katchalski-Katzir [68]. The shape complementarity score will therefore read:

$$C(p,q,r) = \sum_{l,m,n=1}^{N} R(l,m,n) \times L(l+p, m+q, n+r), \tag{1.2}$$

where the periodic boundary conditions are applied. Function C defined on the same grid gives complementarity score that depends on the shift of the ligand (p,q,r). This function can be computed using the FFT algorithm as follows:

$$C(p,q,r) = \mathrm{FFT}^{-1}\left[\overline{\mathrm{FFT}[\mathbf{R}]} \times \mathrm{FFT}[\mathbf{L}]\right],$$

where FFT^{-1} stands for the inverse fast Fourier transform and overline for the complex conjugate. This algorithm allows to rapidly sample translational degrees of freedom. The rotations of the protein are separated from the translations in the outer loop of the algorithm. Fig. 1.10 shows a comprehensible example of 2D calculations using this approach.

An example of the docking procedure is shown on Fig. 1.10. The first row shows the picture of receptor and its Fourier image. The first column shows different ligand poses. The border outlined with the dark-blue and the inner space with light-blue. The numbers placed on the grid are described by Eq. 1.1. The second column shows the Fourier images of receptor and individual ligand poses. Third column shows the phases of Fourier image of the convolution $\mathrm{FFT}[C(p,q,r)] = \overline{\mathrm{FFT}[\mathbf{R}]} \times \mathrm{FFT}[\mathbf{L}]$. The phases of the inverse Fourier transform of the data from the third column are shown in the fourth one, it was computed according to Eq. 1.3.2.1.1. Finally, in the last column the pose with the best score is shown, here the ligand is shown with the light-blue and the receptor is depicted with the dark-blue color. The images of Fourier transforms were built using the projection of complex values to the HSV colorspace [100] with the amplitude fixed to $V = 100.0$.

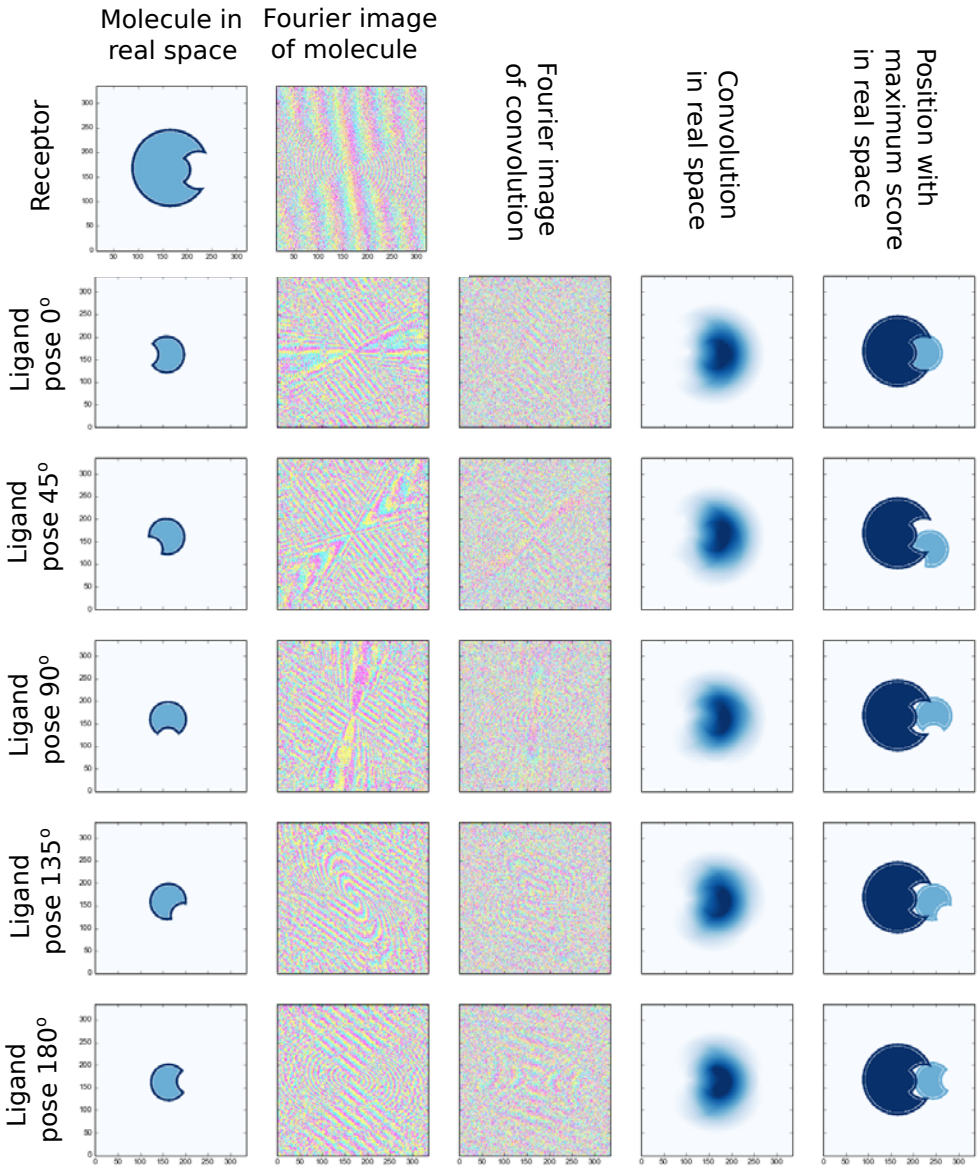


Figure 1.10: Example of a FFT-based docking procedure.

The algorithms that use FFT-based search strategies outnumber the algorithms relying on the other types of search strategies, to name a few: FTDock[44], GRAMM[139], ZDOCK[103], PIPER[76], *etc.* This technique is also used to accelerate the search not only in the space of translations but also in the rotational space [47, 111, 110] and even in 5D rotation-translational space .

1.3.2.1.2 Sophisticated scoring during exhaustive search In modern algorithms based on the idea described above, a simple shape complementarity is usually coupled to the information about binding site [13], electrostatic energy [44], atomic desolvation effect [25], knowledge-based potentials [87], *etc.* The general idea behind the incorporation of additional scoring approaches can be shown on the example of the ZDOCK program [87]. The scoring function that is used in addition to shape complementarity in this algorithm contains the desolvation term, electrostatics and knowledge-based scoring potentials. To estimate the desolvation energy, the authors used atomic contact energies (ACE)[25]. It is defined as the free energy change of breaking two protein atom contacts with water and forming a new contact between these two atoms. The pairwise shape complementarity (PSC) and ACE are represented on the grid in the following way:

$$R_{PSC} = L_{PSC} = \begin{cases} 3 & \text{surface of a protein} \\ 3^2 & \text{protein core} \\ 0 & \text{empty space} \end{cases}$$

$$\mathrm{Re}\left[R_{DE}\right] = \mathrm{Re}\left[L_{DE}\right] = \begin{cases} \text{sum of ACE and PSC scores for nearby atoms} & \text{empty space} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathrm{Im}\left[R_{DE}\right] = \mathrm{Im}\left[L_{DE}\right] = \begin{cases} 1 & \text{grid point is the nearest grid point of an atom} \\ 0 & \text{otherwise} \end{cases},$$

where R stands for receptor and L - for the ligand. The final score is:

$$S_{PSC+DE} = \mathrm{Re}\left[R_{PSC} \times L_{PSC}\right] + \frac{1}{2}\mathrm{Im}\left[R_{DE} \times L_{DE}\right]$$

The electrostatic energy calculation is performed similarily:

$$R_{PSC+ELEC} = L_{PSC+ELEC} = \begin{cases} 3 & \text{surface of a protein} \\ 3^2 & \text{protein core} \\ 0 & \text{empty space} \end{cases}$$

$$\mathrm{Im}\left[R_{PSC+ELEC}\right] = \begin{cases} \beta \times (\text{electric potential of receptor}) & \text{empty space} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathrm{Im}\left[L_{PSC+ELEC}\right] = \begin{cases} -1 \times (\text{atom charge}) & \text{grid point is the nearest grid point of a ligand atom} \\ 0 & \text{otherwise} \end{cases}$$

Finally, the total score is computed as follows:

$$S_{PSC+DE+ELEC} = \mathrm{Re}\left[R_{PSC+ELEC} \times L_{PSC+ELEC}\right] + \frac{1}{2}\mathrm{Im}\left[R_{DE} \times L_{DE}\right]$$

We see that using not only real part of the grid values but also their imaginary part, one can compute electrostatics, desolvation and shape complementarity using two convolution calculations.

1.3.2.1.3 Direct search in Cartesian space In this type of approaches the representation of the protein is also grid based, but simpler than in the FFT-based ones. Grid values are either 1 if the grid is occupied by the protein or 0 otherwise. The algorithms from this category use boolean logic and heuristic rules to speed up the search [64, 134].

1.3.2.1.4 Local shape matching Algorithms implemented in such programs as LZerD [146], GAPDOCK [45], PatchDock [35], *etc* represent protein as a set of surface patches. Efficient search algorithms match the surface patches on two proteins and recover the rotation and position of a candidate docking pose from the patches.

1.3.2.1.5 Randomized search strategies An example of this type of approaches is the package RosettaDock[48]. It generates random starting positions of a ligand around receptor. Afterwards, it minimizes the score while moving the ligand along the line connecting their centers of masses. RosettaDock also uses different representations of the protein at two distinct stages of minimization to reduce the number of degrees of freedom: coarse-grained and full-atom ones. The same methodology is used by the programs like ATTRACT [153], HADDOCK [33], *etc.* Another example of randomized search approach is the SwarmDock algorithm [80]. This program uses population-based search algorithm, called particle swarm optimization. Each copy of the protein-protein complex being optimized is an agent. During optimization of each agent it shares information about its state with the neighbouring agents and they change the parameters of optimization accordingly. The particle swarm optimization algorithm is especially well suited for exploring energy landscapes with large number of local minima.

1.3.2.2 Scoring candidate conformations

Despite the vast variety of the methods to obtain the scoring functions, we can group them into two major classes – physics-based SFs and statistical SFs. The first class of SFs is constructed as a weighted sum of terms, such as desolvation [149], electrostatic interactions [122], hydrogen bonds [38], hydrophobic interactions [119], etc., given as $E = \sum_i \alpha_i E_i$. Then, the weights α_i are usually tuned to match some experiments or to attain a minimum of the SF on a set of known structures of protein complexes. On the other hand, statistical SFs are developed based on the observation that the distances between the atoms in experimentally determined structures follow the Boltzmann distribution [40]. More precisely, using ideas from statistical theory of liquids, effective potentials between atoms are extracted using the inverse Boltzmann relation: $E_{ij}(r) = -k_B T \ln \frac{P_{ij}(r)}{Z}$, where k_B is the Boltzmann

constant, $P_{ij}(r)$ denotes the probability to find two atoms of types i and j at a distance r, and Z denotes the probability distribution in the reference state. The latter is the thermodynamic equilibrium state of the protein when all interactions between the atoms are set to zero. The score of a protein conformation is then given as a sum of effective potentials between all pairs of atoms. Although this concept is old (it originates from the work of Tanaka and Scheraga [131], Miyazawa and Jernigan [89] and Sippl [126]), it is still under debates [135, 127, 73, 11]. Particularly, the computation of the reference state is a challenging problem and only recently some attempts to rigorously justify and compute it have been made [51]. Some scoring functions from this class were obtained without the computation of the reference state. Among those we should mention SF obtained using linear programming [137, 136, 145], quadratic programming, support vector machines [16, 53, 83], and iterative techniques [56, 57].

Here I describe some examples of knowledge-based scoring functions, used in DFIRE [156], ATTRACT [153] and ZDOCK [87] algorithms. The approach chosen by the authors of DFIRE scoring function uses the ideal-gas reference state that allowed them to unify the folding and docking scoring functions. The pair distribution function has the following dependence on the number of observed pairs:

$$N_{obs}(i,j,r) = \frac{1}{V} N_i N_j g_{ij}(r) 4\pi r^2 \delta r$$

Where $N_{obs}(i,j,r)$ is the number of observed atoms of i-th and j-th types at the distance r. The potential of mean-force is connected to the pair distribution function: $u(i,j,r) = -RT \ln g_{ij}(r)$. When the interaction $u(i,j,r)$ is set to zero, one obtains the distribution in the reference state $N_{exp}(i,j,r) = \frac{1}{V} N_i N_j 4\pi r^2 \delta r$. However, due to the finite size of the protein, the correction coefficient α was introduced:

$$N_{exp}(i,j,r) = \frac{1}{V} N_i N_j 4\pi r^{\alpha} \delta r$$

The potential is assumed to have finite range and the equation for the potential is simplified by employing the pairwise distribution in the reference state at the potential cutoff distance:

$$N_{exp}(i,j,r_{cut}) = N_{obs}(i,j,r_{cut}) = \frac{1}{V} N_i N_j 4\pi r_{cut}^{\alpha} \delta r_{cut}$$

Finally, the form of the potential is the following:

$$u(i,j,r) = \begin{cases} -\nu RT \ln \frac{N_{obs}(i,j,r)}{\left(\frac{r}{r_{cut}}\right)^{\alpha} \left(\frac{\delta r}{\delta r_{cut}}\right) N_{obs}(i,j,r_{cut})}, & r < r_{cut} \\ 0 & otherwise \end{cases}$$

where $\nu = 0.0157$, R is the gas constant, T was set to 300K and $\alpha = 1.61$. The δr and δr_{cut} are the widths of bins at the distances r and r_{cut} correspondingly. The coefficient ν was tuned to maximize correllation between the experimental and the predicted stability upon the mutations of monomeric proteins. The exponential factor α is determined using the uniformly distributed points in a sphere for each structure. The radii of the spheres were set to cR_g, where R_g is the gyration radius of a protein and c was tuned to equal the number of pairs within the r_{cut} for the reference state and the experimental structures. Such a choice of the reference

state implies that the information about the protein-protein contacts is negligible compared to the information on the interaction withing proteins. This fact let the authors to unify folding and docking scoring functions.

D. Kozakov *et. al.* developed the "decoys as reference state" (DARS) potential for scoring protein-protein interactions. The key idea was to dock the proteins in the training set using only shape complementarity term in the scoring of conformations. These structures simulate the absense of interactions between two proteins and were used as the reference state. However due to computational complexity, the authors chose only 22 protein-protein complexes to derive the reference state.

Zacharias and his team used a different approach: they used only the Leenard-Jones type of interaction and the electrostatic interaction with distance-dependent dielectric constant $\epsilon = 15r$. They estimated the parameters of interaction potential using the similar approach as in the work by Miyazawa and Jernigan [90].

1.3.2.3 Knowledge-based potentials in rigid-body search

During the rigid-body search algorithms typically filter out about $\approx 10^4$ conformations. Most of them are later refined by the scoring procedure. However, due to a simple scoring function used during the search, the conformations that are close to the solution could be underrepresented in the output of rigid-body docking algorithms. Therefore, two approaches exist to incorporate knowledge-based scoring functions into exhaustive 6D search procedure. One of them, used in the ZDOCK3.0 program [87] integrates atomic contact potential into 6D search procedure in the following way. Suppose one has 2N atom types. For the ligand, N functions on a grid are defined in the following way:

$$\mathrm{Re}\,[L_i] = \begin{cases} 1 & \text{if grid cell is occupied by a ligand atom type i} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathrm{Im}\,[L_i] = \begin{cases} 1 & \text{if grid cell is occupied by a ligand atom type i+1} \\ 0 & \text{otherwise} \end{cases}$$

The N functions for the receptor are:

$$\mathrm{Im}\,[R_i] = \begin{cases} \sum e_{i,j} & \text{Neighbours within } r_{cut} \\ 0 & \text{non-neighbour atoms} \end{cases}$$

$$\mathrm{Re}\,[R_i] = \begin{cases} \sum e_{(i+1),j} & \text{Neighbours within } r_{cut} \\ 0 & \text{non-neighbour atoms} \end{cases},$$

where $e_{i,j}$ is the value of the contact potential between atom types i and j and r_{cut} is the contact radius. The sum of contacts looks as follows:

$$E = \sum_{i=1}^{2N} \sum_{j=1}^{2N} e_{ij} n_{ij} = \sum_{k=1}^{N} \left[\sum_{x,y,z} L_i \times R_i \right]$$

Thus, to compute the contact potential energy one has to perform N forward Fourier transforms and one backward (due to additivity of energy).

Despite the usage of both complex and real parts during the computation of contact potentials N can be around 10, which slows drastically the rigid-body docking algorithm. In order to reduce the number of Fourier transforms Kozakov *et. al.* [76] proposed to decompose the interaction matrix $e_{i,j}$ into the eigenvectors:

$$e_{i,j} = \sum_{p}^{P} \lambda_p u_{p,i} u_{p,j}$$

where P depends on the allowed error rate. Ususally the aproximation of the pairwise potential energy using the grid yelds 10% error in energy, therefore the truncation of the decomposition is well justified. The functions for the receptor and the ligand look like:

$$R_p = \begin{cases} \sum_i u_{p,i} & \text{Over all neighbours } i \text{ within } r_{cut} \\ 0 & \text{no neighbour atoms} \end{cases}$$

$$L_p = \begin{cases} u_{p,j} & \text{If atom of type j is in the cell} \\ 0 & \text{otherwise} \end{cases}$$

This approach allows to compute only four Fourier correlations with the same error as in the previous algorithm.

1.3.2.4 Modelling of water molecules

An important part of a protein-protein interface constitute the water-mediated interactions. One pronounced example of the protein-protein complex where water molecules play great role is the barnase-barstar complex. In the interface of this complex 18 water molecules are fully burried mediating a considerable amount of sidechain-sidechain interactions [19]. Water molecules also play a role in the interaction between protein and drug-like molecules [12, 58].

In spite of the importance of the water molecules prediction for the drug desing, numerous works are devoted to predict positions of water molecules around a known protein structure [41, 148, 114]. However, the amount of papers attempting to predict the explicit solvation of protein-protein interfaces is substantially less [65, 20, 66, 5].

Historically, the first approach to account for the water-mediated interactions during protein-protein docking was the use of the *solvated rotamers* [65]. The key idea of this method is to attach the water molecules to the functional groups of the residues and the backbone and treat different solvation modes as rotamers. Fig 1.11 shows an examples of water molecules placement around aromatic nitrogen in histidine.

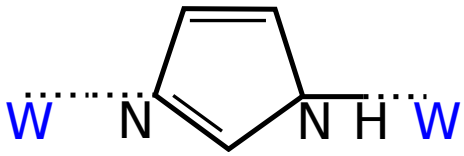


Figure 1.11: Water molecules placement around aromatic nitrogen in histidine residue functional group.

The water molecules placement around the protein was derived by the authors from X-Ray crystallographic structures solved at high resolution. To avoid combinatoric explosion of the number of solvated rotamers they restricted the placement of water molecules to the non-adjacent sites in a protein.

The other approach was implemented in the algorithm WATGEN [20]. First, the hydrogens are added to the interacting proteins. These hydrogens, that interact with the atoms of a protein are called donors. Afterwards, all possible positions of water molecules are generated and those which have clashes with the atoms of the two proteins are discarded. Then, the water sites are selected based on the number of interactions they are involved into. Additionally, two water sites closer than 0.5Å are considered equal. Finally, geometry of the water hydrogens is optimized to maximize the number of interactions.

1.3.2.5 Multimeric docking

Methods and algorithms described previously are applicable mainly to the problem of doking dimeric protein-protein complexes. However, many proteins in a cell form multimeric assemblies that play crucial role in recycling of the proteins in a cell, folding of the proteins, translation, transcription and many other vital cellular processes. There are two major ways in deciphering the structure of a multimeric complex starting from its subunits. The first type of programs do not rely on any other information except for the possible symmetry of a complex and the structures of its subunits. The second one uses low-resolution electron density maps, obtained from cryo-EM or small angle neutron scattering experiments.

The first class of methods deals with complexes of two general types: symmetric and nonsymmetrical. The number of packages that can do symmetry-based multimeric docking is quite small: SymmDock [120], can build complexes with cyclic symmetry; Rosetta program protocol, that uses Monte-Carlo approach and can take into account cyclic, dihedral, helical and icosahedral symmetries [7]; M-ZDOCK [101] reduces rotational search space assuming cyclic symmetry and relies on FFT-based docking approach, *etc.*

A few programs can dock proteins into nonsymmetrical complexes: CombDock [60] uses combinatorial approach, generating all pairwise pairs and finding among them tripples with the optimal score; Kim and Hummer algorithm [70], which uses Monte-Carlo approach and coarse-graining the protein models, HADDOCK [33]; ATTRACT [153]; Multi-LZerD [36] and DockTrina [107].

However, despite the variety of methods, they still perform poorly even on a simple benchmark [107]. The most reliable and widely-used technique to obtain atomic structure of a multimeric assembly is the docking of subunits into a low-resolution electron density map, usually obtained from cryo-EM experiments.

1.3.2.5.1 Docking proteins into low-resolution electron density map For this task, a number of software packages have been developed. Most notable of them are Situs [151, 22], NORMA [130], EMFit [115], UROX [125], etc. Despite the differences in the implementation, all algorithms maximize some score that shows the goodness of the fitting using a certain optimization algorithm. An excellent review on different types of the scoring functions used for cryo-EM density fitting is given by Vasishtan and Topf [143]. According to them, one of the most popular

scoring functions is the cross-correlation function (CCF) between the EDM and the density of the fitted protein.

Given a protein structure that is described by its electron density $f(\mathbf{r})$, and an EDM obtained from e.g. a cryo-EM experiment described by the function $g(\mathbf{r})$, we can minimize the square root discrepancy between them. Precisely, this discrepancy is given by

$$S = \int d\mathbf{r} \, \left(\hat{T}\hat{R}f(\mathbf{r}) - g(\mathbf{r})\right)^2, \tag{1.3}$$

where $\hat{T}$ and $\hat{R}$ are the operators of the translation and the rotation respectively, applied to the density $f(\mathbf{r})$. We can rewrite the scoring function S as

$$S = \int d\mathbf{r} \, \left(\hat{T}\hat{R}f(\mathbf{r})\right)^2 + \int d\mathbf{r} \, g^2(\mathbf{r}) - 2\int d\mathbf{r} \, \hat{T}\hat{R}f(\mathbf{r})g(\mathbf{r}) \tag{1.4}$$

Therefore, the minimization of the score S is equivalent to the maximization of the CCF:

$$\text{CCF} = \int d\mathbf{r} \, \hat{T}\hat{R}f(\mathbf{r})g(\mathbf{r}) \tag{1.5}$$

with respect to the parameters of the operators $\hat{T}$ and $\hat{R}$. This scoring function has been used in the majority of the algorithms and software packages that perform the fitting into the EDM [151, 125, 130].

Another widely used scoring function is the Laplacian-filtered cross-correlation function (LCCF). It originated from the observation that a human performing a manual fitting a structure into an EDM tends to match the isosurfaces of the densities rather than the densities themselves,

$$\text{LCCF} = \int d\mathbf{r} \, \left(\triangle\hat{T}\hat{R}f(\mathbf{r})\right)\left(\triangle g(\mathbf{r})\right) \tag{1.6}$$

This scoring function works better than CCF for low resolution maps ($\sim 10-30$ Å) [151] and was used for the first time in the CoAn/CoFi algorithm [147]. Other scoring functions that e.g. penalise symmetry-induced protein-protein contacts, or make use of protein-protein docking potentials, etc., have also been developed [143].

1.4 Aim of work

This work deals with the protein-protein docking problem. The aim of the work was to propose and validate new algorithms in the area of 6D exhaustive search and in the field of ranking the docking predictions. Particularly, the goal was to invent and test new ways to solve general global rigid-body search approaches that could be more effective than the state-of-art in the field. In order to convincingly demonstrate their applicability, they have to be applied to one of the challenging problems, like fitting of cryo-EM electron density maps or rigid-body protein-protein docking and compared to the existing programs in the field. The scoring method has to be applicable to a wide range of problems: from drug-like molecules docking to crystallographic water prediction. It has to be free of methodological difficulties as in the case of statistical potentials, where the debates about the validity of the reference state computations are still going. This algorithm should be derived from the basic logical statements and have acceptable convergence properties.

Chapter 2

Hermite fitting

2.1 Introduction

As was already mentioned in the introduction, an important class of algorithms in computer science and structural biology deals with the exhaustive search in the six-dimensional space of translations and rotations of a rigid body.

Modern exhaustive search algorithms either implement the fast 3D translational search using the fast Fourier transform (FFT) [22, 67, 43, 151, 125] or the fast 3D rotational search by means of the spherical harmonics decomposition and the FFT [75] or even the fast 5D rotational search [74, 112]. Exhaustive search is also widely used as a preliminary step preceding the local search or flexible refinement procedures. Thus, the quality and the speed of the exhaustive search algorithms have a great impact on the solution of the vast variety of problems. Therefore, we believe that new directions of research on this topic are very important and highly valuable.

In this section, we present the new HermiteFit algorithm that uses the orthogonal Hermite functions to perform exhaustive search in the six-dimensional space of rigid-body motions. We apply this method to the problem of fitting of a high resolution X-ray structure of a protein subunit into the cryo-electron microscopy (cryo-EM) density map of a protein complex. As a part of the new method, we developed an algorithm for the rotation of the decomposition in the Hermite basis and another algorithm for the conversion of the Hermite expansion coefficients into the Fourier basis. We demonstrate the ability of our algorithm to compete with the well–established approaches by using two examples of different difficulty, the PniB conotoxin peptide and the GroEL complex. The first example illustrates encoding principles and demonstrates the influence of the encoding quality on the goodness of fit. The second example is the gold standard of all electron density map fitting algorithms. Our approach allows to analytically assess the quality of encoding of the Hermite basis using an estimation of the crystallographic R-factor. We then compare this estimation with the one computed numerically for the PniB conotoxin density map. Finally, we compare the speed and the fitting accuracy of our algorithm with the two popular programs, the ADP_EM fitting method and the *colores* program from the Situs package and demonstrate that HermiteFit spends less running time per one search point compared to the two other methods while attaining a similar accuracy.

The HermitFit algorithm can be straightforwardly applied to a broad class of

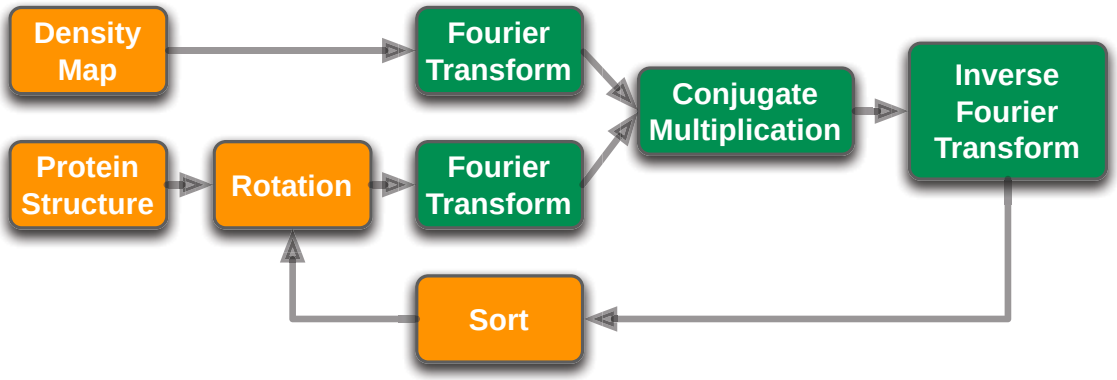
problems in different fields of research. For example, one of the bottlenecks of the algorithms for molecular replacement in crystallography is the computation of the Fourier coefficients (structure factors) of a molecule [92]. This operation is to be precise and fast. However, the exact analytical evaluation of the structure factors is too costly [118] when recomputing them for each rotation of the molecule. Therefore, currently one uses the Sayre–Ten Eyck approach to compute the Fourier coefficients [133]. Unfortunately, one has to be very careful tuning the parameters of the electron density model and the grid cell size to obtain the desired precision [93, 4]. Unlike the Sayre–Ten Eyck, our algorithm offers the analytical expression for the structure factors of the Hermite decomposition of a molecule. Finally, our approach allows to analytically estimate the quality of encoding using, e.g., crystallographic R-factors.

2.2 Methods

2.2.1 Summary of the standard fitting algorithm

The standard FFT-based 3D fitting algorithm operates according to the workflow shown in Figure 2.2.1 [67, 43, 22]. The input of this algorithm is a protein atomic structure determined experimentally by, e.g., X-ray crystallography or nuclear magnetic resonance (NMR) experiments. Another input is an experimental EDM determined by means of, e.g., cryo-EM. First, the algorithm decomposes the experimental EDM into the Fourier basis using the fast Fourier transform algorithm. Then, it rotates the protein structure to a certain orientation $\mathbf{r}$ and decomposes the electron density of the rotated structure into the Fourier basis.

Figure 2.1: Flowchart of the standard fitting algorithm based on the Fourier correlations. Green blocks correspond to the operations in the Fourier space.



The electron density is typically computed as a sum of Gaussians centred on non-hydrogen atoms of the protein. Afterwards, the algorithm exhaustively explores translational degrees of freedom of the rotated protein with respect to the EDM. For every translation $\mathbf{t}$, it determines the corresponding score, which is usually given by the correlation between the two densities. This procedure is equivalent to computing the convolution of two functions,

$$\mathrm{CCF}(\mathbf{r}, \mathbf{t}) = \int d\mathbf{x}\, f(\mathbf{r}, \mathbf{x} - \mathbf{t}) g(\mathbf{x}), \tag{2.1}$$

where $f(\mathbf{r}, \mathbf{x} - \mathbf{t})$ is the density of the protein rotated by $\mathbf{r}$ and translated by $\mathbf{t}$, and $g(\mathbf{x})$ is the experimental electron density map. To speed up this step, the algorithm computes the values of the Fourier transform of the CCF for all translational degrees of freedom at once, using the convolution theorem. Finally, the algorithm computes the inverse Fourier transform (IFT) of the convolution, generates a new rotation of the protein structure, and returns to the second step. This procedure is repeated until all rotational degrees of freedom of the protein with respect to the EDM are explored (see Fig. 2.2.1). The solution of the fitting problem is then given by $(\mathbf{r}_{max}, \mathbf{t}_{max}) = \mathrm{argmax}_{\mathbf{r},\mathbf{t}} \{\mathrm{CCF}(\mathbf{r}, \mathbf{t})\}$.

The bottleneck of the standard algorithm is the re-projection of the protein electron density into the Fourier space after each rotation. To overcome it, we propose to encode the electron density of the protein structure in the orthogonal Hermite basis, prior to performing the rotational search. This allows to speed up the

Operation	Complexity	Loop multiplier
Decomposition of the step function	$O(M^3 \log M^3)$	1
Decomposition of the Gaussian	$O(N_{atoms} N^3)$	
Construction of the rotation matrix	$O(N_{rot} N^4)$	
Rotation	$O(N^4)$	N_{rot}
Evaluation of the Hermite series	$O(M^3 \cdot N + M^2 \cdot N^2 + M \cdot N^3)$	
Multiplication	$O(M^3)$	
Inverse Fourier Transform	$O(M^3 \log M^3)$	

Table 2.1: Complexity of the Hermite fitting algorithm. Here, M denotes the order of the Fourier decomposition; N is the order of the Hermite decomposition; N_{atoms} is the number of atoms in the protein; N_{rot} – the number of rotations to be sampled.

projection of the protein density into the Fourier space. Since only the members of the Fourier family of linear transforms can replace $O(N^2)$ operations of a convolution in a time domain by $O(N)$ operations in a frequency domain [128], we still need to perform the convolution in the Fourier space. Figure 2.2.1 shows the workflow of the proposed algorithm. Computational complexity of this algorithm is listed in Table 2.1.

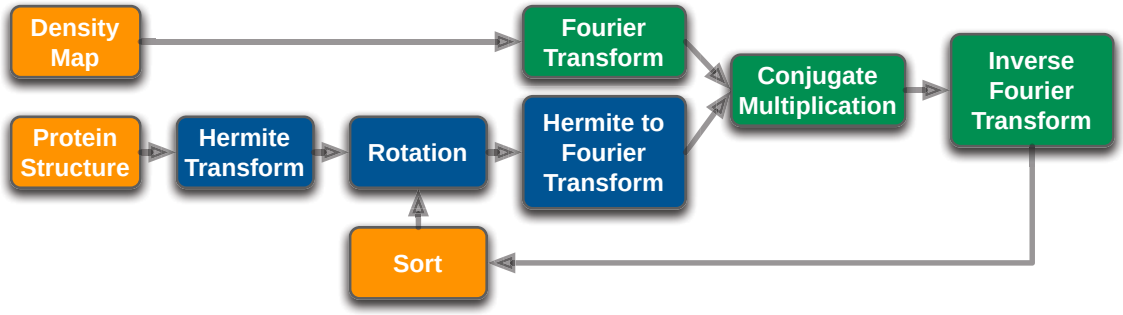


Figure 2.2: Flowchart of HermiteFit, the new fitting algorithm based on the Hermite expansions. Green blocks correspond to the operations in the Fourier space. Blue blocks correspond to the operations in the Hermite space.

2.2.2 Hermite functions

Orthogonal Hermite function of order n is defined as:

$$\psi_n(x;\lambda) = \frac{\sqrt{\lambda}}{\sqrt{2^n n! \sqrt{\pi}}} \exp(-\frac{\lambda^2 x^2}{2}) H_n(\lambda x), \tag{2.2}$$

where $H_n(x)$ is the Hermite polynomial and λ is the scaling parameter. In Fig. 2.2.2 we show several orthogonal Hermite functions of different orders with different parameters λ. These functions form an orthonormal basis set in $L^2(\mathbb{R})$. A 1D function $f(x)$ decomposed into the set of 1D Hermite functions up to an order N reads

$$f(x) = \sum_{i=0}^{N} \hat{f}_i \psi_i(x;\lambda) \tag{2.3}$$

Here, $\hat{f}_i$ are the decomposition coefficients, which can be determined from the orthogonality of the basis functions $\psi_i(x;\lambda)$. Decomposition in Eq. 2.3 is called the *band-limited decomposition* with $\psi_i(x;\lambda)$ basis functions. To decompose the EDM and the protein structures, we employ the 3D Hermite functions:

$$\psi_{n,l,m}(x,y,z;\lambda) = \psi_n(x;\lambda)\psi_l(y;\lambda)\psi_m(z;\lambda), \tag{2.4}$$

which form an orthonormal basis set in $L^2(\mathbb{R}^3)$. A function $f(x,y,z)$ represented as a band-limited expansion in this basis reads

$$f(x,y,z) = \sum_{i=0}^{N}\sum_{j=0}^{N-i}\sum_{k=0}^{N-i-j} \hat{f}_{i,j,k}\psi_{i,j,k}(x,y,z;\lambda) \tag{2.5}$$

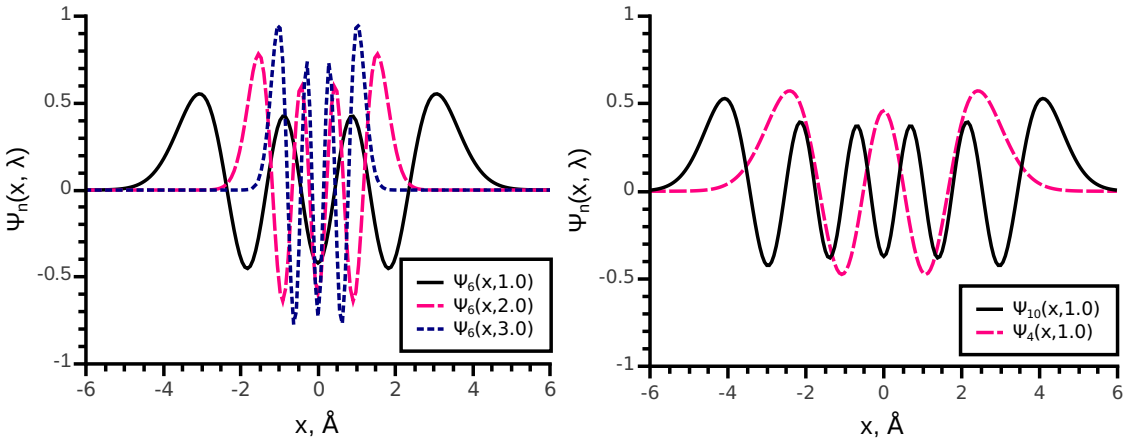


Figure 2.3: Left: 1D Hermite functions of order six for three different scaling parameters λ. Right: 1D Hermite functions of two different orders for the scaling parameter $\lambda = 1$.

Figure 2.2.2 shows that Hermite functions are very similar to the cosine functions near the coordinate axis origin.

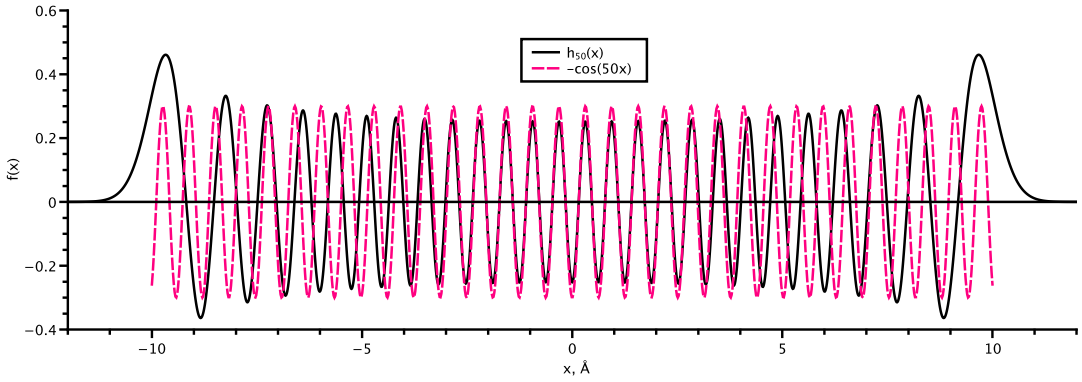


Figure 2.4: Hermite function of the order 50 (solid line) and cosine function with the frequency equal to 50 (dashed line).

2.2.3 Decomposition of electron densities into the orthogonal Hermite basis

One of the advantages of the orthogonal Hermite basis is that we can derive the exact analytical expression for the decomposition coefficients of a molecular structure. This allows to rapidly obtain the exact decompositions without costly numerical integration over the 3D space. In our algorithm, the electron density of the protein ($f(x)$ in Eq. 2.1, upon which rotation and translation operators act) is expanded in the Hermite basis using the Gaussian model. More precisely, we model the electron density of a single atom in the molecular structure as a Gaussian centred at the atomic position $\mathbf{r}_0^{(i)}$ with the squared variance equal to $\alpha^2/2$. Then, the electron density of the whole molecular structure is given by the following sum:

$$M(\mathbf{r}) = \sum_{i=1}^{N_{atoms}} e^{-|\mathbf{r}-\mathbf{r}_0^{(i)}|^2/\alpha^2}, \tag{2.6}$$

where $\mathbf{r}_0^{(i)}$ is the position of the i-th atom, $\alpha/\sqrt{2}$ is the variance of the Gaussian distribution, and $\mathbf{r} = (x, y, z) \in \mathbb{R}^3$ is the sampling volume. Normally, each Gaussian should be weighted with a coefficient corresponding to electron distribution of a particular atom. However, we omit the weights in our approximation. In the section 2.2.4, we provide analytical expressions (Eqs. 2.8 and 2.14) for the decomposition coefficients of $M(\mathbf{r})$ in the 1D and the 3D cases.

2.2.4 Shifted Gaussian expansion

Here we provide the derivation of the expansion coefficients of a shifted Gaussian of the following form:

$$g(\mathbf{r}) = e^{-\frac{|\mathbf{r}-\mathbf{r}_0|^2}{\alpha^2}} \tag{2.7}$$

into the orthogonal Hermite basis. The well known property of this basis (as well as of any orthogonal basis) is the following:

$$\begin{aligned} \text{if } f(x,y,z) &= f^{(1)}(x)f^{(2)}(y)f^{(3)}(z) \\ \text{and } f^{(k)}(t) &= \sum_{i=0}^{N} \hat{f}_i^{(k)} \psi_i(t;\lambda) \\ &\text{then} \\ \hat{f}_{i,j,k} &= \hat{f}_i^{(1)} \hat{f}_j^{(2)} \hat{f}_k^{(3)} \end{aligned} \tag{2.8}$$

First, we derive the decomposition of a 1D Gaussian into the 1D orthogonal Hermite basis. Then, using property (2.8) we obtain the decomposition of a 3D Gaussian into the 3D orthogonal Hermite basis. More specifically, the 1D Gaussian function reads as:

$$g(x) = e^{-\frac{(x-\xi)^2}{\alpha^2}} \tag{2.9}$$

Its decomposition coefficients are equal to:

$$\hat{g}_n(\xi;\lambda,\alpha) = \int g(x)\psi_n(x;\lambda)\,dx = \frac{n!\sqrt{\lambda}e^{-\frac{\xi^2}{\alpha^2}\left(1-\frac{1}{\alpha^2\beta^2}\right)}}{\sqrt{2^n n!\sqrt{\pi}}} \sum_{m=0}^{[\frac{n}{2}]} \frac{(-1)^m}{m!(n-2m)!} \int e^{-\beta^2\left(x-\frac{\xi}{\alpha^2\beta^2}\right)^2} (2\lambda(x-\frac{\xi}{\alpha^2\beta^2}) + \frac{2\lambda\xi}{\alpha^2\beta^2})^{n-2m} dx, \tag{2.10}$$

where $\beta^2 = \frac{\lambda^2}{2} + \frac{1}{\alpha^2}$. From now on we will, for brevity, write $\hat{g}_n$ instead of $\hat{g}_n(\xi;\lambda,\alpha)$. Changing the variables $t = x - \frac{\xi}{\alpha^2\beta^2}$ and denoting $a = \frac{\xi}{\alpha^2\beta^2}$, we obtain:

$$\hat{g}_n = \frac{n!\sqrt{\lambda}e^{-\frac{\xi^2}{\alpha^2}\left(1-\frac{1}{\beta^2}\right)}}{\sqrt{2^n n!\sqrt{\pi}}} \sum_{m=0}^{[\frac{n}{2}]} \frac{(-1)^m (2\lambda)^{n-2m}}{m!(n-2m)!} \int e^{-\beta^2 t^2}(t+a)^{n-2m} dx \tag{2.11}$$

Next, we decompose the sum $(t+a)^k$ using Newton's formula:

$$(t+a)^k = \sum_{i=0}^{k} \left(\begin{array}{c} k \\ i \end{array} \right) t^i a^{k-i} \tag{2.12}$$

Thus, the integral in Eq. 2.11 will read:

$$\int e^{-\beta^2 t^2}(t+a)^{n-2m} dx = \sum_{i=0,\,i-even}^{n-2m} \frac{(n-2m)!}{2^i\left(\frac{i}{2}\right)!(n-2m-i)!}\sqrt{\pi}\beta^{-1-i}a^{n-2m-i} \tag{2.13}$$

Substituting it to the formula for $\hat{g_n}$ and denoting $\sum_{i=0,\,i-\text{even}}^{n-2m} = \sum_{l=0}^{[\frac{n-2m}{2}]}$ $(i=2l)$, we obtain the following expression for the coefficients:

$$\hat{g}_n(\xi;\lambda,\alpha) = e^{-\frac{\xi^2}{\alpha^2}\left(1-\frac{1}{\alpha^2\beta^2}\right)} \sqrt{\frac{n!\sqrt{\pi}\lambda}{2^n}} \sum_{m=0}^{[\frac{n}{2}]} \sum_{l=0}^{[\frac{n-2m}{2}]} \frac{(-1)^m 2^{n-2m-2l}\lambda^{n-2m}}{l!(n-2m-2l)!m!}\beta^{-2n+4m+2l-1}\left(\frac{\xi}{\alpha^2}\right)^{n-2m-2l} \tag{2.14}$$

Finally, using Eq. 2.8 we obtain a decomposition of the 3D Gaussian into the 3D Hermite basis. We should note that in order to avoid the rounding error, one should begin the summation with the Gaussians that are located father from the origin.

2.2.5 Expansion of a function defined on a grid

In many docking algorithms the parwise interaction of particles is approximated as the set of functions on the grid. Therefore in many important cases the protein description goes beyond the sum of gaussians as in Eq. 2.6. In this section we

provide a way to directly obtain decomposition of a function $f(x, y, z)$ defined on a regular grid. We can represent this function as a sum:

$$f(x, y, z) = \sum_{i,j,k} f(x_i, y_j, z_k)\eta_{ijk}(x, y, z)$$

where $\eta_{ijk}(x, y, z)$ is a step-function in the position of ijk-th grid cell. Fig 2.2.5 shows the function η that begins at point a and has the width h. To derive the decomposition of the general step-function in 3D and with arbitrary shift a we have to begin with the basic 1D step-function that is fixed in $a = 0$ point.

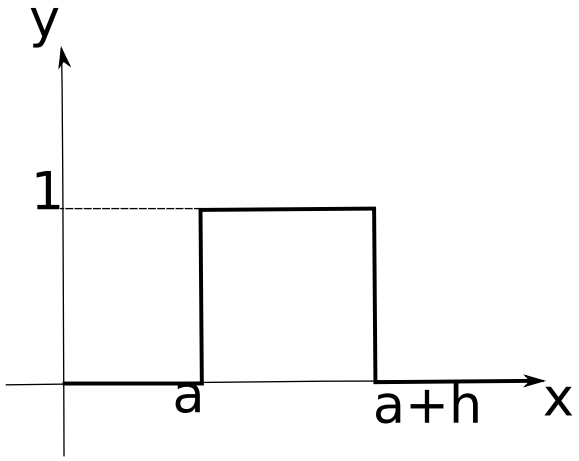


Figure 2.5: Shifted 1D step-function η.

The expression for function is the following:

$$\eta(x) = \begin{cases} 1, & 0 \leq x < h \\ 0, & \text{otherwise} \end{cases} \tag{2.15}$$

The decomposition coefficients of the shifted η read:

$$\eta(x - a) = \sum_{i}^{N} \alpha_i \psi_i(x; \lambda) \tag{2.16}$$

$$\alpha_i = \int_{-\infty}^{+\infty} \eta(x - a)\psi_i(x; \lambda)\ dx = \int_{a}^{h+a} \psi_i(x; \lambda)\ dx \tag{2.17}$$

We can obtain the coefficients α_n by simple integration:

$$\alpha_n = \frac{1}{\sqrt{2^n n! \sqrt{\pi}\lambda}} \int_{\lambda a}^{\lambda(h+a)} \exp(-\frac{t^2}{2}) H_n(t)\ dt \tag{2.18}$$

Using the well known result for the finite integral of the Hermite polynomials:

$$\int z^{q-1} e^{-pz^2} H_n(z) dz = n! \sum_{k=0}^{\left[\frac{n}{2}\right]} \frac{(-1)^{k-1} 2^{n-2k-1} z^{n-2k+q} \left(pz^2\right)^{k-\frac{n+q}{2}} \Gamma\left(\frac{n+q}{2} - k,\, pz^2\right)}{k!(n-2k)!} \tag{2.19}$$

with parameters $q = 1$ and $p = 1/2$ we obtain:

$$\int e^{-\frac{z^2}{2}} H_n(z)dz = \tag{2.20}$$

$$n!\,(\mathrm{Sgn}[z])^{n+1} \quad \sum_{k=0}^{\left[\frac{n}{2}\right]} \frac{(-1)^{k-1}\Gamma\left(\frac{n+1}{2}-k,\,\frac{z^2}{2}\right)}{2^{3k-\frac{3n}{2}+\frac{1}{2}}\,k!(n-2k)!} + C \tag{2.21}$$

After deriving the missing coefficients C_n we obtain the following expression for the coefficients α_n:

$$\alpha_n = \frac{2^n\sqrt{n!}}{\sqrt{2\sqrt{\pi}\lambda}} \sum_{k=0}^{\left[\frac{n}{2}\right]} \frac{(-1)^{k-1}}{\left(2\sqrt{2}\right)^{2k} k!(n-2k)!}$$

$$\left[(\mathrm{Sgn}[h+a])^{n+a}\,\Gamma\left(\frac{n+1}{2}-k,\,\frac{(\lambda h+\lambda a)^2}{2}\right) - (\mathrm{Sgn}[a])^{n+a}\,\Gamma\left(\frac{n+1}{2}-k,\,\frac{(\lambda a)^2}{2}\right) - \right.$$

$$\left.\left((\mathrm{Sgn}[h+a])^{n+1} - (\mathrm{Sgn}[a])^{n+1}\right)\Gamma\left(\frac{n+1}{2}-k,\,0\right)\right]$$

Next, we are moving to the 3D case:

$$\begin{aligned}
\eta(x,y,z) &= \begin{cases} 1, & 0 \le x < h_x \bigwedge 0 \le y < h_y \bigwedge 0 \le z < h_z \\ 0, & \text{otherwise} \end{cases} \\
\eta(x,y,z) &= \eta(x;\,h_x)\eta(y;\,h_y)\eta(z;\,h_z) \\
\eta(x-a_x,\,y-a_y,\,z-a_z;h_x,h_y,h_z) &= \sum_k^N \sum_j^N \sum_i^N \alpha_i\alpha_j\alpha_k\psi_i(x;\lambda)\psi_i(y;\lambda)\psi_k(z;\lambda)
\end{aligned}$$

As we see we have to multiply the coefficients of individual 1D-function with the shifts corresponding to the grid cell position along individual axes. We begin summation staring from the cells that farther from the frame origin because rounding error substantially influences the final result.

2.2.6 Laplacian filter in the Hermite basis

For mid- to low- resolution maps the Laplacian-filtered cross-correlation function gives a better match compared to the CCF [151]. In the Hermite basis, the Laplacian filter has a particularly simple form. Using the well-known recurrence relation for the derivatives of the Hermite functions, we can easily derive the following relation for the second derivative of a 1D basis function:

$$\frac{d^2}{dx^2}\psi_n(x;\lambda) = \frac{\lambda^2}{2}\left(\sqrt{n(n-1)}\psi_{n-2}(x;\lambda) + (2n+1)\psi_n(x;\lambda) + \sqrt{(n+1)(n+2)}\psi_{n+2}(x;\lambda)\right) \tag{2.22}$$

A similar relationship holds for the coefficients of the decomposition:

$$\hat{h}''_n = \frac{\lambda^2}{2}\left(\sqrt{n(n-1)}\hat{h}_{n-2} + (2n+1)\hat{h}_n + \sqrt{(n+2)(n+1)}\hat{h}_{n+2}\right), \tag{2.23}$$

where $\hat{h}_n$ and $\hat{h''}_n$ are the n-th order decomposition coefficients of the original basis and its Laplacian representation, respectively. For $n < 0$ and $n > N$ we let $\hat{h}_n = 0$ and $\hat{h''}_n = 0$. Due to the properties of the Laplace operator and the 3D Hermite decomposition, the contribution of the derivatives along each axis are additive. The derivation of the formula for the 3D decomposition derivative is straightforward and we omit it for brevity.

2.2.7 Rotation of the Hermite decomposition

Recently, Park et al. [96] presented the method to perform an in-plane rotation of a 2D orthogonal Hermite band-limited decomposition. Here, we extend their method for the 3D case. Let us first consider the decomposition of a 2D function into a 2D orthogonal Hermite function basis:

$$f(x,y) = \sum_{n=0}^{N} \sum_{m=0}^{N-m} \hat{f}_{n,m} \psi_n(x;\lambda)\psi_m(y;\lambda) \tag{2.24}$$

The decomposition of a function $f^\theta(x,y)$ rotated clock-wise by an angle θ reads

$$f^\theta(x,y) = \sum_{m=0}^{N} \sum_{k=0}^{m} (\sum_{n=0}^{m} \hat{f}_{n,m-n} S^m_{k,n}) \psi_k(x;\lambda)\psi_{m-k}(y;\lambda), \tag{2.25}$$

where coefficients $S^m_{k,n}$ are computed using the following recurrent formulas [96]:

$$\begin{aligned}
S^{m+1}_{q,n} &= \sqrt{\frac{n}{m-q+1}}\sin(\theta) S^m_{q,n-1} + \sqrt{\frac{m-n+1}{m-q+1}}\cos(\theta) S^m_{q,n} \\
S^{m+1}_{q,0} &= \sqrt{\frac{m+1}{m-q+1}}\cos(\theta) S^m_{q,0} \\
S^{m+1}_{m+1,n} &= \sqrt{\frac{n}{m+1}}\cos(\theta) S^m_{m,n-1} - \sqrt{\frac{m-n+1}{m+1}}\sin(\theta) S^m_{m,n} \\
S^{m+1}_{m+1,0} &= -\sin(\theta) S^m_{m,0}
\end{aligned} \tag{2.26}$$

The key idea that allows to generalize these formulas to a 3D decomposition is that we can factorize a rotation in 3D space into 3 independent in-plane rotations around three different axes, and then rotate each 2D decomposition using Eq. 2.25. Let us consider the following 3D decomposition:

$$f(x,y,z) = \sum_{n=0}^{N} \psi_n(x;\lambda) \sum_{m=0}^{N-n} \sum_{l=0}^{N-m-n} \hat{f}_{n,m,l} \psi_m(y;\lambda)\psi_l(z;\lambda) \tag{2.27}$$

If we rotate this decomposition about x axis, this rotation will be equivalent to N rotations of different 2D decompositions in the yz-plane:

$$f_n(y,z) = \sum_{m=0}^{N-n} \sum_{l=0}^{N-m-n} \hat{f}_{n,m,l} \psi_m(y;\lambda)\psi_l(z;\lambda) \tag{2.28}$$

This observation means that in order to perform such rotation, we need to recompute rank-3 tensor of coefficients $\hat{f}_{n,m,l}$ slice by slice N times using Eq. 2.25. Figure 2.2.7 illustrates three subsequent rotations of tensor $\hat{f}_{n,m,l}$. Each rotation of the coefficients in one plane corresponds to a multiplication of these coefficients with a rotation matrix. Therefore, a 3D rotation defined with three Euler angles is equivalent to three sequential rotations of coefficients in three planes.

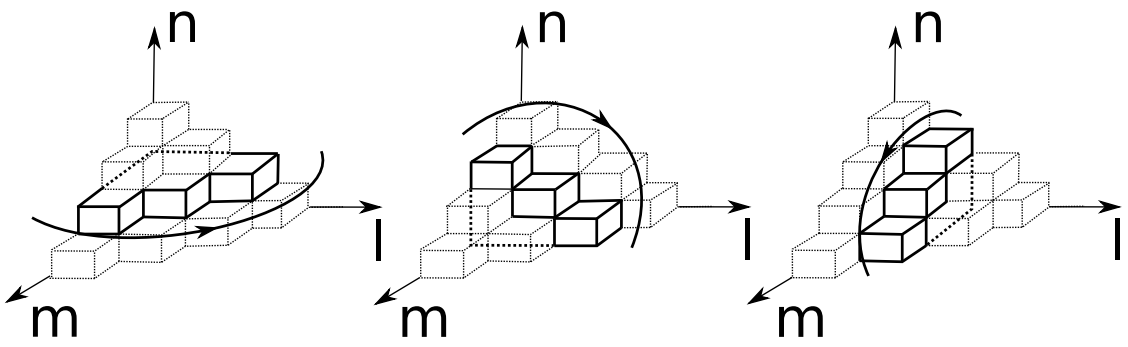


Figure 2.6: Sequential rotations of coefficients $\hat{f}_{n,m,l}$ about different axes. The rotated layer is shown with the solid cubes, other coefficients are shown with the dashed cubes.To perform the complete rotation of the decomposition about one axis, we rotate each layer of coefficients about the corresponding axis in the space of coefficients.

2.2.8 Transition from the Hermite to the Fourier basis

In order to perform a fast convolution as in Eq. 2.1, we convert the decomposition coefficients from the Hermite basis into the Fourier basis. This allows to use the fast convolution algorithm based on the Fourier convolution theorem, which was first introduced in protein-protein docking studies [67, 43] and then also applied in the EDM fitting [22, 151, 125]. The key idea of this algorithm is to compute the Fourier transform of the values of a scoring function on a grid, $\mathrm{CCF}(\mathbf{r},\mathbf{t}) = \int f(\mathbf{r},\mathbf{x})g(\mathbf{r},\mathbf{x}-\mathbf{t})d\mathbf{x}$, using the convolution theorem:

$$F\left[f * g\right] = \overline{F}[f]F[g], \tag{2.29}$$

i.e. to multiply the complex conjugated coefficients of the Fourier transform of the protein electron density with the coefficients of the Fourier transform of the EDM. Then, we obtain $\mathrm{CCF}(\mathbf{r},\mathbf{t})$ by taking the inverse Fourier transform of $F\left[f * g\right]$,

$$\mathrm{CCF}(\mathbf{r},\mathbf{t}) = \mathrm{IFT}\left(\overline{F}[f]F[g]\right) \tag{2.30}$$

Now we explain how we convert the decomposition coefficients from the Hermite basis into the Fourier basis. Consider the decomposition of a function $f(\mathbf{r})$ in the 3D Hermite basis with the decomposition coefficients $\hat{f}_{i,j,k}$ (Eq. 2.5). Orthogonal Hermite functions are the eigenfunctions of the continuous Fourier transform:

$$\int \psi_n(x;\lambda)e^{-2\pi i\omega x}\,dx = (-i)^n\psi_n(\omega;\frac{2\pi}{\lambda}) \equiv \tilde{\psi}_n(\omega;\lambda), \tag{2.31}$$

where ω is the frequency in the reciprocal space. In order to compute Fourier coefficients of $f(\mathbf{r})$ up to order M, we first compute the Fourier transforms of the basis functions $\psi_i(x;\lambda)$, $\psi_j(y;\lambda)$, and $\psi_k(z;\lambda)$ using Eq. 2.31. After, we substitute these coefficients into Eq. 2.5 and obtain the following expression for $\tilde{f}_{l,m,n}$, the Fourier coefficients of $f(\mathbf{r})$:

$$\tilde{f}_{l,m,n} = \frac{1}{L_xL_yL_z}\sum_{i=0}^{N}\sum_{j=0}^{N-i}\sum_{k=0}^{N-i-j}\hat{f}_{i,j,k}\tilde{\psi}_i(\frac{l}{L_x};\lambda)\tilde{\psi}_j(\frac{m}{L_y};\lambda)\tilde{\psi}_k(\frac{n}{L_z};\lambda) \tag{2.32}$$

These values can be computed in $O(M^3\cdot N + M^2\cdot N^2 + M\cdot N^3)$ steps (see section 2.2.9).

2.2.9 Fast summation

Here we explain the fast summation in Eq. 2.33:

$$\tilde{f}_{l,m,n} = \sum_{i=0}^{N} \sum_{j=0}^{N-j} \sum_{k=0}^{N-i-j} \hat{f}_{i,j,k} \tilde{\psi}_{i,l} \tilde{\psi}_{j,m} \tilde{\psi}_{k,n}, \tag{2.33}$$

with indexes $l, m, n \in [0, M]$. The summation in this formula can be performed with less operations than a naive estimation $O(M^3N^3)$ suggests. We perform the fast summation by splitting the equation into three consecutive sums:

$$\widetilde{T^1_{i,j,n}} = \sum_{k=0}^{N-i-j} \hat{f}_{i,j,k} \tilde{\psi}_{k,n} \tag{2.34}$$

$$\widetilde{T^2_{i,m,n}} = \sum_{j=0}^{N-i} \widetilde{T^1_{i,j,n}} \tilde{\psi}_{j,m} \tag{2.35}$$

$$\tilde{f}_{l,m,n} = \sum_{i=0}^{N} \widetilde{T^2_{i,m,n}} \tilde{\psi}_{i,l} \tag{2.36}$$

It is easy to see that the construction of $\widetilde{T^1_{i,j,n}}$ matrix takes $O(MN^3)$ operations, the construction of $\widetilde{T^2_{i,m,n}}$ matrix takes $O(M^2N^2)$ operations, and the final summation takes $O(M^3N)$ operation. In the common use case ($N = 15$, $M \gg N$) the last sum takes much more time than the other two. To optimize it, we used the Gauss method to multiply complex numbers and expressed the whole sum as a generalized matrix product of three real-valued matrices. To implement these operations, we used the ATLAS library.

2.2.10 Implementation details and running time

We chose to demonstrate the potential of the Hermite basis by implementing the rigid-body fitting of an atomistic structure of a protein in an electron density map of low resolution. The HermiteFit algorithm was implemented using the C++ programming language and compiled using g++ with -O3 optimization. The running times of the tested algorithms are measured on a single core of an Intel® Xeon® CPU X5650 @ 2.67GHz processor with 24 GB of RAM on a Linux 64-bit operating system.

Our fitting method typically samples some 10^{10} rigid-body configurations. Therefore, it is practical to group its fitting solutions into clusters. There are multiple ways to measure the similarity between rigid-body solutions. For example, the pair-wise root–mean–square deviation (RMSD) is a fast and well-accepted similarity measure. Thus, we clustered the fitting solutions using the rigid-body clustering algorithm implemented with the RigidRMSD library [106] as follows. First, the fitting solution with the best score (yet unassigned to any cluster) is taken as the seed for the new cluster. Second, the pair-wise RMSDs between the seed and all other predictions are measured and the predictions with the RMSD lower than a certain threshold are put into the cluster. Finally, these two steps are iterated until all fitting predictions are assigned to corresponding clusters.

2.3 Analysis

This section provides analytical and numerical analysis of the density encoding in the Hermite basis. More specifically, we provide the choice of optimal model parameters and assess the quality of encoding.

2.3.1 Choice of parameters of the method

Orthogonal Hermite functions (2.2) decay exponentially after a certain distance and thus can encode information only within some interval. We can estimate this interval using the formula for the last root of a Hermite polynomial, $\xi_{1,N} \approx \frac{\sqrt{1+2N}}{\lambda}$ [109], which gives an approximation for the half-size of the bounding box that we can successfully encode:

$$L_{box}/2 \lesssim \frac{\sqrt{1+2N}}{\lambda} \tag{2.37}$$

On the other hand, orthogonal Hermite functions are the eigenfunctions of the continuous Fourier transform (Eq. 2.31). Therefore, Hermite decomposition of order N can encode only a certain interval of frequencies. Using the same approximation as in the case of the real-space interval, we obtain the following equation for the maximum encoding frequency:

$$\omega_{max} = \frac{\lambda}{2\pi}\sqrt{2N+1} \tag{2.38}$$

In case of the the Fourier series expansion on an interval $(0, L_{box})$, we can use the same estimation for the maximum encoding index M_{max} by setting $M_{max} = 2L_{box}\omega_{max}$. Resolution R of an X-Ray electron density map is defined by the size of the reciprocal lattice as $R = 1/(2\omega_{max})$, or, equivalently, $R = L_{box}/M_{max}$. Therefore, using resolution of the map R and the order of the Fourier series expansion M, we can estimate the lower bound on the Hermite scaling parameter λ required to encode all the reflexes of the electron density diffraction pattern to be

$$\lambda \gtrsim \frac{\pi}{\max(R, L_{box}/M)\sqrt{2N+1}} \tag{2.39}$$

Here, we bounded the actual resolution by L_{box}/M, because this will be the limit allowed by the finite Fourier series of order M.

The two inequalities (2.37 and 2.39) give approximate bounds on the scaling parameter λ, provided that we know the size of the box L_{box} containing a protein density and the resolution of the map R. Using these inequalities, we obtain the following relationship between parameters λ and N:

$$\frac{\pi}{\sqrt{2N+1}\max(R, L_{box}/M)} \lesssim \lambda \lesssim 2\frac{\sqrt{1+2N}}{L_{box}}, \tag{2.40}$$

which is valid for sufficiently large values of N. Nonetheless, we can use the following empirical estimation for the optimal value of λ at any N:

$$\lambda_{opt} \approx \frac{\pi}{2\max(R, L_{box}/M)\sqrt{2N+1}} + \frac{\sqrt{1+2N}}{L_{box}} \tag{2.41}$$

Using dimensionless relative parameters λL_{box} and L_{box}/R we may rewrite the previous expression as

$$\lambda L_{box} \approx \frac{\pi \min(L_{box}/R, M)}{2\sqrt{2N+1}} + \sqrt{1+2N} \tag{2.42}$$

If at a given expansion order N there is no such parameter λ that satisfies inequality (2.40), then the protein representation might involve information loss. Therefore, we can estimate the minimum order N_{min} of the Hermite expansion that allows this inequality to have solutions to be

$$N_{min} \approx \frac{\pi}{4} \min\left(\frac{L_{box}}{R}, M\right) \tag{2.43}$$

Validity of the provided estimates and the graphical representation of the real–space and the reciprocal–space bounds on parameter λ will be demonstrated in the following sections.

The maximum order of the Fourier expansion M_{max} can be estimated from the resolution and the size of the density map as $R = L_{box}/M_{max}$. However, when finding the global maximum of the cross-correlation function, we need to sample the space of possible translations of a protein with respect to the EDM with a step several times finer than the EDM resolution R. In protein crystallography, it is the common practice to set the sampling step size to $R/3$ [4]. In principle, we can use the same reasoning in choosing the optimal number of rotations N_{rot}. When using spherical harmonics, the angular search step usually equals to the resolution of the basis, $2\pi/N$ [46]. In case of the Hermite basis, we propose to use the same criterion.

2.3.2 The transfer matrix

Below we describe an analytical model of encoding by the Hermite basis for the one-dimensional case. Suppose we have a function $f(x)$ that describes an electron density of a non-periodic object. Without loss of generality, we assume that this function is defined on a 1D interval of $(-L_{box}/2; +L_{box}/2)$. This function has the following decomposition into Fourier series:

$$\tilde{f}_k^{exact} = \frac{1}{L_{box}} \int_{-L_{box}/2}^{+L_{box}/2} f(x) e^{-2\pi i k x / L_{box}} \, dx \tag{2.44}$$

We will refer to Fourier coefficients obtained using this expression as *exact*. The original function is then recovered by the inverse Fourier transform:

$$f(x) = \sum_{k=-\infty}^{+\infty} \tilde{f}_k^{exact} e^{2\pi i k x / L_{box}} \tag{2.45}$$

On the other hand, our algorithm computes *approximate* Fourier coefficients using the Hermite to Fourier transform:

$$\tilde{f}_k^{approx} = \frac{1}{L_{box}} \sum_{n=0}^{N} \hat{f}_n \tilde{\psi}_n\left(\frac{k}{L_{box}}; \lambda\right) \tag{2.46}$$

Assuming that function $f(x)$ is zero outside of the bounding interval, Hermite coefficients $\hat{f}_n$ can be written as the finite integral:

$$\hat{f}_n = \int_{-L_{box}/2}^{+L_{box}/2} f(x)\psi_n(x;\lambda)\ dx \tag{2.47}$$

Now, we can express the *approximate* Fourier coefficients as a linear combination of the *exact* ones:

$$\tilde{f}_k^{approx} = \sum_{l=-\infty}^{+\infty} T_{k,l}\tilde{f}_l^{exact}, \tag{2.48}$$

where the *transfer matrix* $T_{k,l}$ reads as:

$$T_{k,l} = \frac{1}{L_{box}} \sum_{n=0}^{N} \tilde{\psi}_n(k;\lambda) \int_{-L_{box}/2}^{+L_{box}/2} \psi_n(x;\lambda) e^{2\pi i l x/L_{box}} dx \tag{2.49}$$

The transfer matrix acts as a linear filter in the reciprocal space and demonstrates how the input function is distorted by the finite size N of the Hermite basis. We should note that, generally, its values are complex numbers. This matrix can also be seen as a product of two matrices,

$$\mathbf{T} = \mathbf{F^{(1)}}\mathbf{F^{(2)}}, \tag{2.50}$$

where the first matrix is a scaled Fourier transform of the basis functions,

$$F_{kn}^{(1)} = \tilde{\psi}_n(k;\lambda)/\sqrt{L_{box}} \tag{2.51}$$

and the second matrix is a scaled Fourier series of the basis functions,

$$F_{nl}^{(2)} = \int_{-L_{box}/2}^{+L_{box}/2} \psi_n(x;\lambda) e^{2\pi i l x/L_{box}} dx/\sqrt{L_{box}} \tag{2.52}$$

Figure 2.3.2 shows the absolute values of matrices $\mathbf{F^{(1)}}$ and $\mathbf{F^{(2)}}$ computed with $\lambda = 0.55$ and $L_{box} = 23$ Å. The values of the Fourier series $\mathbf{F^{(2)}}$ were computed numerically using adaptive quadrature. The dashed blue line shows the maximum encoding frequency ω_{max}, according to Eq. 2.38, and bounds the encoding region. The solid black line on the right plot demonstrates the maximum order of the Hermite expansion (Eq. 2.37), after which the Fourier series encode mainly the frequencies near ω_{max}. This is because on a finite interval $(-L_{box}/2, +L_{box}/2)$, high-order Hermite basis functions become orthogonal to low-order Fourier basis functions.

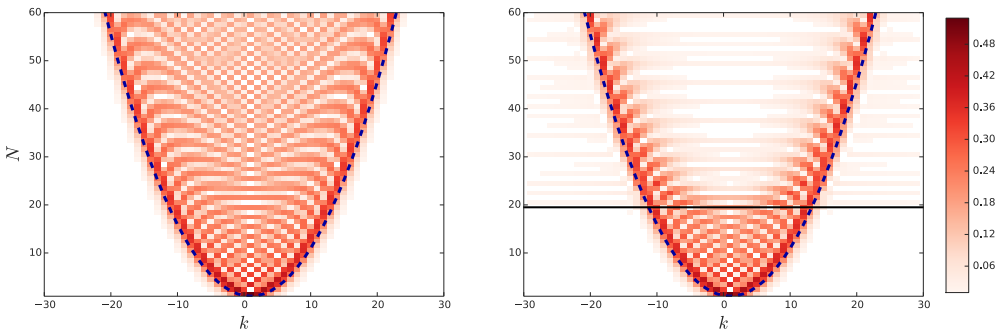


Figure 2.7: Absolute values of two matrices $\mathbf{F^{(1)}}$ and $\mathbf{F^{(2)}}$ that give the transfer matrix as their product (Eq. 2.50). These matrices are computed with the scaling parameter $\lambda = 0.55$ and the input box size of $L_{box} = 23.0$ Å, which mimics the first fitting example shown below. Left: $\mathbf{F^{(1)}}$, the scaled Fourier transform of a 1D Hermite function as given by Eq. 2.51. Right: $\mathbf{F^{(2)}}$, the scaled Fourier series of a 1D Hermite function as given by Eq. 2.52. The dashed blue line highlights the maximum encoded frequency according to Eq. 2.38. The solid black line on the right plot shows the maximum Hermite decomposition order N_{max}, at which the two matrices are still identical (Eq. 2.37).

Figure 2.3.2 shows several examples of the absolute values of the transfer matrix components for three different values of the Hermite scaling parameter λ and three values of the Hermite decomposition order N. The size of the transfer matrix was limited to 60×60 and the box size L_{box} was set to 23 Å. The ideal transfer matrix should be identity, which is the case only at $N \to \infty$, as we demonstrate below. We see, however, that the transfer matrix at small values of λ encodes only low-order reflexes. The index of the last encoded reflex can be estimated from Eq. 2.38 as $k_{max} = \sqrt{2N+1}\lambda L_{box}/(2\pi)$. With the increase in order N and parameter λ, the number of encoded frequencies rises. At the same time, increasing the scaling parameter λ makes the quality of encoding of all the frequencies worse, as we see in the right column. Therefore, it is very important to tune the value of λ according to the class of input functions, such that the quality of encoding becomes optimal. Below we will assess encoding quality by means of the crystallographic R-factor.

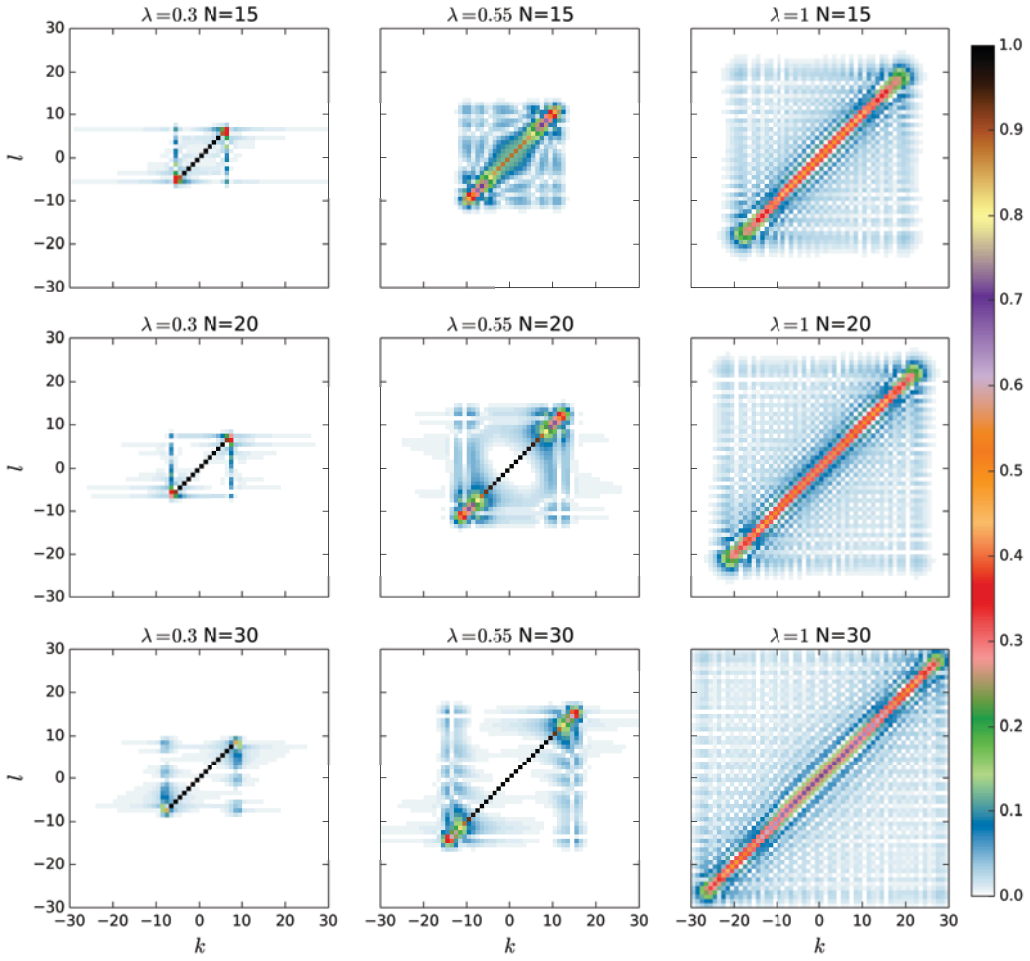


Figure 2.8: Nine examples of the absolute values of the transfer T-matrices for three different values of λ and three different values of the Hermite decomposition order N. The number of Fourier coefficients is $M = 60$, and the input box size is $L_{box} = 23.0$ Å, which mimics the first fitting example shown below. Hermite decomposition orders are $N \in \{15, 20, 30\}$, parameter λ takes the values of 0.3 Å^{-1}, 0.55 Å^{-1}, and 1.0 Å^{-1}. The first column corresponds to the relative λL_{box} value of 6.9, the middle column corresponds to the relative λL_{box} value of 12.65, and the right column to the relative λL_{box} value of 23. Notably, at low values of λ the transfer matrix encodes only small order reflexes. The index of the last reflex can be estimated from Eq. 2.38 as $k_{max} = \frac{\sqrt{2N+1}\lambda L_{box}}{2\pi}$. Increasing the value of λ, the number of encoded frequencies rises. However, at the same time, the quality of encoding of low frequencies worsens, as can be seen from the values at the diagonal.

2.3.3 Asymptotic behaviour of the transfer matrix

Here we demonstrate that the transfer matrix asymptotically achieves the Kronecker delta function at $N \to \infty$. Recall the Mehler's formula [84]:

$$\sum_{n=0}^{N} u^n \psi_n(x)\psi_n(y) = \frac{1}{\sqrt{\pi(1-u^2)}} \exp\left(-\frac{1-u}{1+u}\frac{(x+y)^2}{4} - \frac{1+u}{1-u}\frac{(x-y)^2}{4}\right) \quad (2.53)$$

If we rewrite the transfer matrix in the following way:

$$T_{k,l} = \frac{1}{L_{box}} \int_{-L_{box}/2}^{+L_{box}/2} dx \ \sum_{n=0}^{N} (-i)^n \psi_n(\frac{k}{L_{box}}; \frac{2\pi}{\lambda}) e^{2\pi i l x / L_{box}} \psi_n(x;\lambda), \quad (2.54)$$

and use the fact that

$$\psi_n(x;\lambda) \equiv \sqrt{\lambda}\psi_n(\lambda x), \quad (2.55)$$

we see that we can use the Mehler's formula to compute the limit

$$\lim_{N\to\infty} \sum_{n=0}^{N} (-i)^n \psi_n(\frac{k}{L_{box}}; \frac{2\pi}{\lambda}) \psi_n(\frac{l}{L_{box}}; \lambda) \quad (2.56)$$

After a simple derivation, we obtain the final result:

$$\lim_{N\to\infty} T_{k,l} = \frac{1}{L_{box}} \int_{-L_{box}/2}^{+L_{box}/2} e^{2\pi i l x / L_{box}} e^{-2\pi i k x / L_{box}} dx, \quad (2.57)$$

which is exactly the Kronecker delta function.

2.3.4 Encoding quality

There are several ways to evaluate the quality of a model encoding with the subsequent reconstruction. For example, in the optimal control theory [18], the quality of a linear filter is estimated using a certain norm of the transfer matrix. However, in crystallography, the most used quality criterion is the crystallographic R-factor [129]:

$$R = \frac{\sum_l \left| \left| \tilde{F}_l^{exact} \right| - \left| \tilde{F}_l^{mod} \right| \right|}{\sum_l \left| \tilde{F}_l^{exact} \right|}, \quad (2.58)$$

where F^{exact} and F^{mod} are the exact Fourier coefficients of a molecule and the coefficients computed from the Hermite coefficients, respectively. This quantity is a widely used measure of agreement between a crystallographic model and the corresponding experimental X-ray diffraction data. In the case of an ideal electron density encoding, R-factor is equal to zero. In protein crystallography, models with R-factors less than 0.2 are regarded as good when working at a middle resolution.

Equations for the transfer matrix allow to estimate the R-factor values for certain classes of electron density distributions. As described above (Eq. 2.6), we use the Gaussian distribution to model the electron density of an atom. Exact Fourier coefficients of a molecule with N_{atoms} atoms at positions $\mathbf{r}_i$ are then given as:

$$\tilde{f}_{l,m,n}^{exact}(\mathbf{s}) = \alpha^3 \pi^{\frac{3}{2}} \sum_{i=1}^{N_{atoms}} e^{-\alpha^2\pi^2 \mathbf{s}_{lmn}^2} e^{-2i\pi \mathbf{r}_i \mathbf{s}_{lmn}}, \quad (2.59)$$

where $s_{l,m,n}$ is the wave vector, $\mathbf{s}_{l,m,n} = (l/L_x, m/L_y, n/L_z)$, with L_x, L_y, and L_z the dimensions of the bounding box along the corresponding axes. Similarly, one-dimensional exact Fourier coefficients of the Gaussian function are given as:

$$\tilde{f}_l^{exact} = \alpha\pi^{\frac{1}{2}} \sum_{i=1}^{N_{atoms}} e^{-\alpha^2 l^2 / L_{box}^2} e^{-2i\pi r_i l / L_{box}} \tag{2.60}$$

To see how the Hermite basis encodes Gaussian densities with various level of detail, we built models of electron density map with different parameters α. The width of the Gaussian determines the resolution of the density map according to:

$$R = \frac{\pi\alpha}{2} \tag{2.61}$$

The derivation of this formula follows the one well known in crystallography, which describes the extinction of diffraction reflexes. For the sake of completeness of the thesis, we provide its derivation in the section 2.3.5.

To estimate R-factor for certain model parameters, we assume that the input electron density is given as a sum of Gaussians with variance of $\alpha/\sqrt{2}$ equispaced at a distance α. Figure 2.3.4 shows analytical R-factors in one dimension computed using Eqs. 2.48 and 2.60 as a function of the Hermite decomposition order N and the scaling parameter λ. We bounded the input and output frequencies by $M = 30$ Fourier coefficients. The size of the input interval L_{box} is set to 23.0 Å to mimic the alpha-conotoxin PnIB peptide (pdb code 1AKG) decomposition used in the fitting example below.

We should stress that due to the properties of the Hermite functions, the whole model is scale–invariant. More precisely, if we keep the product λL_{box} constant, then the relative shape of the Hermite basis functions would not change. Also, if we scale L_{box} and α simultaneously, then the value of R-factor is unchanged. Therefore, it is useful to provide relative resolutions computed as R/L_{box}. Figure 2.3.4 (Left) shows R-factors for the Gaussian parameter $\alpha = 0.2$ Å, corresponding to the absolute input signal resolution of $R = 0.31$ Å and the relative resolution of $R/L_{box} = 0.014$. However, in this case, the actual absolute resolution is cut at $L_{box}/M = 0.77$ Å, which corresponds to the relative resolution of 0.033. Figure 2.3.4 (Middle) shows R-factors computed using the Gaussian parameter $\alpha = 1.0$ Å, corresponding to the absolute input signal resolution of $R = 1.57$ Å and the relative resolution of $R/L_{box} = 0.068$. Figure 2.3.4 (Right) shows R-factors computed using the Gaussian parameter $\alpha = 5.0$ Å, corresponding to the absolute input signal resolution of $R = 7.85$ Å and the relative resolution of $R/L_{box} = 0.34$. The estimate on the optimal parameter λ (Eq. 2.41) is plotted with the red dashed line. The real-space bound on the optimal parameter λ (Eq. 2.37) is shown with the orange dashed line. The reciprocal-space bound on the optimal parameter λ (Eq. 2.39) is shown with the blue dashed line. We see that lowering the resolution of the input signal, R-factors decrease, as can be expected from general considerations. We can also see that the the lower (Eq. 2.37) and the upper (Eq. 2.39) bounds on the optimal scaling parameter λ follow the isolines of the R-factor map. Therefore, their mean given by Eq. 2.37 provides a reasonable estimation on the optimal value of λ.

Figure 2.3.4 shows R-factors as a function of input signal resolution R for three different Hermite decomposition orders N, 15, 20, and 30. R-factors were estimated in the same way as in the previous case. More precisely, we assumed the same

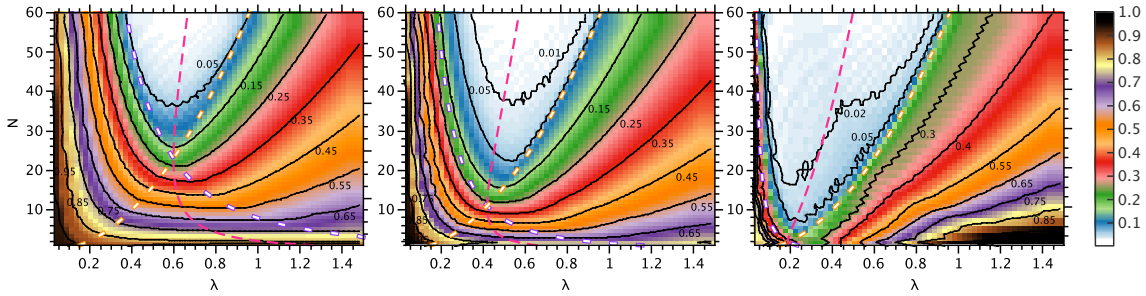


Figure 2.9: Analytical R-factors in one dimension as a function of Hermite decomposition order N and scaling parameter λ computed at three different resolutions. The input signal is modelled as a sum of Gaussians (Eq. 2.6) with the variance of $\alpha/\sqrt{2}$ equispaced at a distance α. The number of Fourier coefficients is $M = 30$, and the input box size is $L_{box} = 23.0$ Å. These values are chosen to mimic the 1AKG peptide decomposition. The estimate on the optimal parameter λ (Eq. 2.41) is plotted with the red dashed line. The real-space bound on the optimal parameter λ (Eq. 2.37) is shown with the orange dashed line. The reciprocal-space bound on the optimal parameter λ (Eq. 2.39) is shown with the blue dashed line. **Left**: The Gaussian parameter $\alpha = 0.2$ Å, corresponding to the absolute input signal resolution of $R = 0.31$ Å and the relative resolution of $R/L_{box} = 0.014$. However, in this case, the actual absolute resolution is cut at $L_{box}/M = 0.77$ Å, which corresponds to the relative resolution of 0.033. **Middle**: The Gaussian parameter $\alpha = 1.0$ Å, corresponding to the absolute input signal resolution of $R = 1.57$ Å and the relative resolution of $R/L_{box} = 0.068$. **Right**: The Gaussian parameter $\alpha = 5.0$ Å, corresponding to the absolute input signal resolution of $R = 7.85$ Å and the relative resolution of $R/L_{box} = 0.34$.

shape of input electron density and then used Eqs. 2.48 and 2.60 to compute the analytical R-factors. For these plots, we computed the optimal scaling parameter λ using Eq. 2.37. Parameter L_{box} and the size of the transfer matrix M were constant and equal to 23 Å and 30, correspondingly. As in the previous figure, these values are chosen to mimic the alpha-conotoxin PnIB peptide decomposition used in the fitting example below. The scale of the top horizontal axis gives the absolute resolution for $L_{box} = 23$ Å. The scale of the bottom horizontal axis gives the relative resolution. In order to compute the absolute resolution, its values need to be multiplied by the chosen value of L_{box}. As expected, the values of R-factors diminish as the resolution becomes lower. This is because at low resolutions, low-frequency columns of the transfer matrix become more important. In the limiting cases of zero and infinite resolutions, R-factor can be computed directly from the transfer matrix as a certain norm of $T - I$. For the infinite resolution limit, it is given as L_1 norm of the central column of matrix $T - I$. For the zero resolution limit, R-factor is given by the entry-wise L_1 norm of $T - I$, $R = \sum_{i,j} |T_{i,j} - \delta_{i,j}|$. Figure 8 also shows an estimation of R-factors for the 3D case. It is based on the assumption that the Hermite decomposition encoding in 3D behaves similar to the 1D case, with the number of coefficients scaled as $N_{1D} = \sqrt[3]{N_{3D}}$.

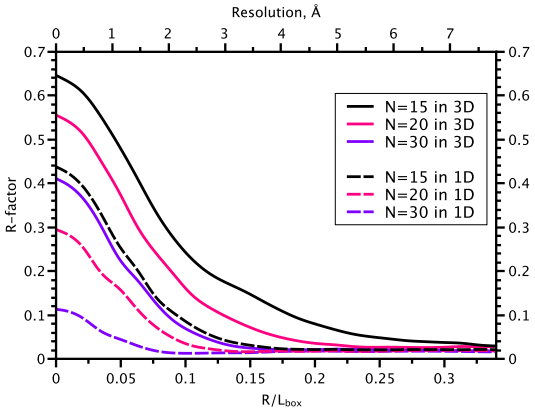


Figure 2.10: Analytical R-factors in one and three dimensions as a function of relative resolution R/L_{box}. The absolute resolution at box size $L_{box} = 23$ Å is shown in the top horizontal axis. Plots for three different Hermite expansions orders are shown, $N \in \{15, 20, 30\}$. Parameters L_{box} and M were constant and equal to 23 Å and 30, correspondingly. Scaling parameter λ was estimated using Eq. 2.41.

2.3.5 Resolution model

To illustrate the connection between parameter α in the model of electron density (Eq. 2.6) and the resolution of the X-ray diffraction pattern, we use the simplest model. More precisely, we model the electron density as the array of Gaussians in a perfect 1D lattice perpendicular to the incoming radiation beam. Parameter α then plays the role similar to the temperature B-factors. X-ray diffraction intensity depends on the angle between the incoming beam and the direction to the detector θ as:

$$I \propto \left| \int dx \; f(x) \exp\left(2\pi i x \frac{\sin\theta}{\lambda}\right) \right|^2 \tag{2.62}$$

where λ is the wavelength of the incoming radiation. Using the model density (Eq. 2.6), we obtain:

$$I \propto \left| \alpha\sqrt{\pi} e^{-(\pi \frac{\sin\theta}{\lambda}\alpha)^2} \int dx \; \rho(x) \exp\left(2\pi i x \frac{\sin\theta}{\lambda}\right) \right|^2, \tag{2.63}$$

where $\rho(x)$ is the sum of delta functions at the atomic positions. Therefore, the extinction of the diffraction peaks is proportional to $\left| e^{-(\pi \frac{\sin\theta}{\lambda}\alpha)^2} \right|^2$, where we neglect the quadratic factor before the exponent. According to the definition used in crystallography, resolution is the inter-planar distance in the real space corresponding to the last observable peak in the reciprocal space. Unfortunately, the index of the last peak depends on the detector's noise and strongly depends on the characteristics of the measurement device. Therefore, to give qualitative estimation on the dependence of resolution on the model parameter α, we assume that the last observable peak is the one whose intensity decreases approximately by the factor e^2. The corresponding angle then reads:

$$\sin\theta_{max} = \frac{\lambda}{\pi\alpha} \tag{2.64}$$

Therefore, the minimum inter-planar distance, or, the resolution is given by Bragg's law as:

$$R = \pi\alpha/2 \tag{2.65}$$

2.4 Results and Discussion

We tested and verified our algorithm using two examples of different difficulty. The first example is a small polypeptide alpha-conotoxin PnIB. We generated the EDM for this example from the coordinates of the polypeptide. The second example is the fitting the GroEL domains into the electron density map of the GroEL complex.

2.4.1 Alpha-conotoxin PnIB

First, we explored the relationship between encoding quality and the quality of the fitting. For this purpose, we chose the small 16–residue polypeptide alpha-conotoxin PnIB. We downloaded the X-ray crystal structure of alpha-conotoxin PnIB (PDB code 1AKG) [54] from the PDB database [14] and simulated the electron density map (2mFo-DFc) using the Uppsala electron density server [71] with the resolution $R = 1.1$ Å. We computed the protein density according to Eq. 2.6 with the Gaussian width $\alpha = 1.0$ Å using only the non-hydrogen atoms of the standard amino acids. We rotated the initial 1AKG structure by the arbitrarily chosen Euler angles equal to 76, 234, and 56 degrees, respectively, and used it as the input for the fitting workflow. We used $N_{rot} = 500$ (corresponding to an angular step of 36°) rotations represented with uniformly distributed Euler angles spanning the space of $2\pi \times \pi \times 2\pi$. The order of the Hermite expansion was set to $N = 15$, which is the minimum expansion order allowed at this resolution according to Eq. 2.43. The order of the Fourier expansion was twice the order of the Hermite expansion, $M = 30$ for each dimension.

To see how the encoding quality influences the fitting algorithm, we studied the dependence of the decomposition on the scaling parameter λ. We chose a range of λ parameters between 0.05 and 2.0. For each λ, we computed the best fitting score along with the average fitting score. Fitting results are shown in Fig. 2.4.1. We see that by choosing λ small, we neglect the details of the protein structure (Fig. 2.4.1 A) and therefore, we can not discriminate between different orientations of the protein (maximum score for $\lambda = 0.05$ is very close to the average score). When choosing λ sufficiently large, we obtain satisfactory discriminative power to find the near-native position of the protein (Fig. 2.4.1 C,D). We also see that, e.g., for $\lambda = 0.5$, the difference between the maximum and the average score is much larger than in the case of $\lambda = 0.05$. Also, when we take λ too large, we can not encode the whole protein (Fig. 2.4.1 E). The red dashed line on Fig. 2.4.1 shows R-factors computed with Eq. 2.58. We see that the choice of parameter λ influences the R-factors and thus determines the quality of the fitting. Notably, the minimum of the R-factor curve corresponds to the maximum of the fitting score.

Due to the strong influence of the scaling parameter λ on the discrimination power of the algorithm, we estimated its optimal value to gain the maximum separation between the score of the correct pose and the average score. Provided that the box that contains all the rotations of the peptide has the size $L_{box} = 23$ Å and setting the resolution of the EDM $R = 1.1$ Å, Eq. 2.41 gives an estimate on the optimal value of the scaling parameter $\lambda_{opt} \approx 0.50$. Fig. 2.4.1 shows that this estimation corresponds to the best discrimination between the near-native and all other structures, which can be deduced from the maximum separation between the score of the prediction and the average score. RMSD between the prediction and the solution at this value of λ is 1.03 Å. We should note that the RMSD can be

decreased by taking a finer angular search step.

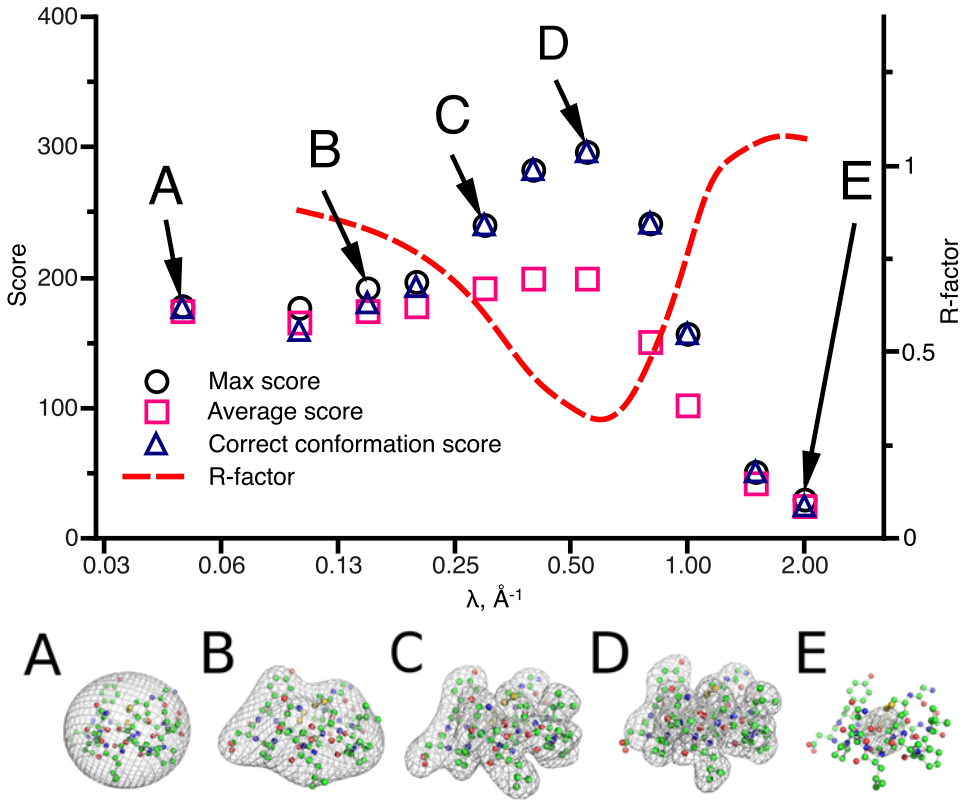


Figure 2.11: Test of the fitting algorithm on artificially generated EDM for the alpha-conotoxin PnIB (PDB code 1AKG). Here, we plotted the dependence of four parameters, the maximum score, the average score, the score of the near-native conformation and the crystallographic R-factor on the scaling parameter λ. Isosurface of the Hermite decomposition at protein model density equal to $(\rho_{max} + \rho_{min})/2$ and several values of λ are shown in sub-plots A ($\lambda = 0.05$), B ($\lambda = 0.15$), C ($\lambda = 0.3$), D ($\lambda = 0.55$) and E ($\lambda = 2.0$).

2.4.2 GroEL complex

Here, we demonstrate that our approach obtains essentially the same results as other programs, provided that the scoring function if the same (LCCF in this case). For this purpose, we use a classical test for a fitting algorithm, the GroEL complex map. We downloaded the EDM of the GroEL complex from the Electron Microscopy Data Bank (EMDB), code EMD-2001 with resolution of 8.5 Å. Then, we downloaded the crystal structure of the GroEL subunits from the PDB database. We used the GroEL-GroES complex (PDB code 1AON), from which we extracted the chain A, centered it and arbitrary rotated to exclude any bias. We chose the sampling grid size according to the resolution and the size of the EDM. The EDM was first padded with zeros and then transformed to the Fourier basis using the FFT algorithm. The number of coefficients in the Fourier decomposition M was equal to $105 \times 107 \times 119$. The angular search step was set to 30°. We used the Hermite expansion order of $N = 15$, which is larger than the minimum expansion order allowed at this resolution, $N_{min} \approx 9$ (see Eq. 2.43). We sampled the rotations using the spiral

algorithm [117], which generates an equispaced distribution of points on a sphere. Unlike in the previous example, due to the lower resolution of the GroEL EDM, here we fitted Laplacian filtered protein density into the Laplacian filtered EDM.

After the 6D exhaustive search, we clustered the solutions using the clustering threshold of 10 Å and kept the top 14 poses. All the 14 poses corresponded to the individual chains of the complex, which comprises 2 heptameric rings structure. Fig. 2.4.2 shows the result of the fitting. We compared the fitted model with the model provided by the authors of the EDM (PDB entry code 4AAU). The average RMSD between the chains due to flexible deformations measured using Cα atoms was 3.0 Å. More precisely, we super-posed the corresponding chains of both models using rigid-body transformations and then measured RMSD between them. Overall, the average RMSD between Cα atoms was 5.35 Å. This includes both the discrepancy between corresponding chains in the assembly due to flexible deformations and because of the rigid body misfit. The average distance between the centers of mass of the corresponding chains was 2.64 Å (Table 2.2).

Algorithm	RMSD C_α, Å	RMSD centers of masses, Å
ADP_EM	4.61	2.29
Colores	5.42	2.52
HermiteFit	5.35	2.64

Table 2.2: Comparison of the models obtained using HermiteFit, Colores and ADP_EM algorithms with the model obtained by the authors of the electron density map (PDB entry 4AAU). For each pair of models, RMSD was measured using the C_α-atoms and the centers of mass of the corresponding chains and then averaged over all chains comprising the assembly.

2.4.3 Runtime of Hermite- to Fourier- space transition

The use of the fast Fourier transform was the inevitable step in every fitting algorithm up until now. Instead, we introduced the basis from which we can transform a decomposition into the Fourier basis avoiding evaluation of the FFT on a grid. When the grid becomes large, the asymptotic complexity of our algorithm becomes $O(M^3 N)$ (see Eq. 2.32). It is comparable to the complexity of the fast Fourier transform algorithm, $O(M^3 \log M)$. Intuitively, at large orders of the Fourier expansion M, our algorithm should be faster compared to the FFT. However, prefactors preceded the complexities of the two algorithms are different. Thus, we conducted a numerical experiment to compare the actual running times. Fig. 2.4.3 shows the time needed to compute the FFT on a cubic grid of size M and the time needed to transform a Hermite expansion of order $N = 15$ to the same Fourier grid. We can see that, generally, at large values of M, $M \gtrapprox 100$, the transition from the Hermite into the Fourier space is faster compared to the speed of the FFT. Also, the timing of the transition grows evenly with respect to M in contrast to the timing of the FFT. One has to take into account that we compared our algorithm with the highly optimized FFTW3 library [42]. Probably, additional optimization of HermiteFit could improve performance even further. One of the ways to speed up the transition will be to use the Fast Hermite Transform instead of the naive matrix multiplication [78]. This implementation will be the subject of our future work.

Figure 2.12: Result of the fitting chain A of the GroEL-GroES X-Ray structure (PDB entry 1AON) to the GroEL complex electron density map (EMD-2001). Two heptameric rings are shown in different colors. The average RMSD measured using the C_α-atoms between the two closest chains in the fitted structure and the flexibly refined structure provided by the authors of the EDM (PDB entry 4AAU) is 5.35 Å .

2.4.4 Comparison with Situs and ADP_EM

We compared the HermiteFit algorithm with two popular existing fitting methods, the *colores* program from the Situs package [22] and the ADP_EM fitting tool [46]. These two packages represent the two major approaches to the problem of exhaustive search in the six-dimensional space of rigid-body motions. *Colores*, a widely used CCF-based fitting tool, rapidly scans the translational degrees of freedom using the fast Fourier transform. The rotations, though, are sampled exhaustively by enumerating a list of equispaced distributed rotations on a sphere. ADP_EM choses points in real space, places there the atomic structure and then rotationally matches it to the EDM using the Fast Rotational Matching algorithm. The authors of the ADP_EM compared their algorithm with the 5D rotational matching and found that the 3D rotational matching works faster in practice [46].

For the comparison, we normalized the running time of the fitting algorithms by the sizes of the search space. For *colores* and HermiteFit, the size of the search space is equal to the number of grid cells (M^3 for a cubic grid in the HermiteFit algorithm) multiplied by the number of sampled angles. The size of the search space of the ADP_EM algorithm is the number of points in real space times the number of cells of the angular grid. The latter is built from uniformly sampled Euler angles on a grid of $2\pi \times \pi \times 2\pi$. The size of the angular grid is determined by the order N_{exp} of a spherical harmonics expansion and equals to $4N_{exp}^3$. For *colores* and HermiteFit, we used the angular search step of 30°. The resolution of the EDM for *colores* and HermiteFit was set to 8.5 Å. The Fourier grid that was used by *colores* and the HermiteFit algorithm had dimensions $105 \times 107 \times 119$. For ADP_EM, we used the spherical harmonics expansion order of $N_{exp} = 16$.

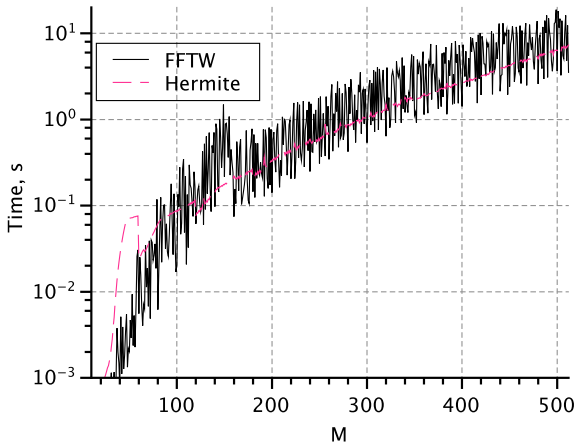


Figure 2.13: Running times of the Hermite to Fourier space transition performed using our algorithm and the FFT algorithm on a cubic grid of $M \times M \times M$ as a function of the Fourier expansion order M. We used the FFTW3 library [42] with the double precision real discrete Fourier transform using the flag FFTW_ESTIMATE to measure the speed of the FFT. The order of the Hermite expansion was $N = 15$.

Table 2.3 shows the normalized times of the complete 6D search for the three algorithms in the case of fitting the GroEL subunit into the 8.5 Å GroEL electron density map. Judging by the total running time, ADP_EM has a big advantage over the two other algorithms, which exhaustively search all the space of possible translations. However, in terms of running time per one search point, the HermiteFit algorithm is more effective than the other two. Interestingly, *colores* spends about half of the total search time on the computation of the Fourier coefficients of the rotated protein. Therefore, it was very important for us to speed up this step. Nonetheless, all three tested algorithms have their own advantages and drawbacks. For example, ADP_EM can use smart heuristics to contract the number of search points in the real space. However, its sample points in the space of rigid body rotations are distributed non-uniformly. In particular, near the poles rotations are sampled more densely, making this sampling scheme less effective [117]. On the other hand, the HermiteFit algorithm along with the *colores* algorithm sample the rotational space nearly uniformly using the spiral algorithm while the translational space sampling also remains uniform. We would like to stress that the absolute runtimes (shown in Table 2.3) are not very informative. In particular, they dramatically depend on the choice of the FFT library, code optimization, the choice of compiler and compilation options, etc. However, this comparison clearly demonstrates that the new approach paves the way to speed up one of the bottlenecks of fitting methods, the projection of the rotated structure into the Fourier space.

To assess the fitting quality of the tested methods, we measured the RMSDs between the obtained models and the structure obtained by the authors of the electron density map (PDB entry 4AAU). Table 2.2 shows the comparison of the measured RMSDs for ADP_EM, *colores* and HermiteFit. We used two different criteria for the measurements. First, we measured the average RMSD between α-carbons. Second, we measured the average distance between the centers of mass of the corresponding chains. ADP_EM produced a model with RMSD of 4.61 Å from the solution, RMSDs for *colores* and HermiteFit were 5.42 Å and 5.35 Å, respectively. Clearly,

Table 2.2 demonstrates that the tested algorithms produce equal quality models. However, results of ADP_EM are slightly better, presumably because of the finer rotational sampling.

Algorithm	Num of rot-space points	Num of trans-space points	Runtime, s	Time per point, $\times 10^{-7}$s
ADP_EM	16384	23186	139	3.6
Colores	4416	1336965	1454	2.5
HermiteFit	4416	1336965	917	1.5

Table 2.3: Comparison of the HermiteFit algorithm with the Colores and ADP_EM algorithms. The comparison criterion was chosen to be the total running time and the running time per one point of the search space.

Chapter 3

Scoring functions for protein-protein docking

3.1 Introduction

As was described in the introduction, scoring method is used to filter out false-positive predictions of rigid-body docking step and refine those close to the native conformation of a protein-protein complex. Therefore the method of scoring has a decisive role in success of the whole docking workflow.

In this section we propose a new method to derive scoring functions. We base our method on separation of the native structures and computationally generated non-native conformations of a complex (decoys). Most of previously used algorithms solving this problem separate all the decoys from all the decoys simultaneously. The key new idea behind our algorithm is that the decoys of a particular complex should be separated from its native structure only. However the form of the scoring potentials should be the same for all the dataset. Based on these prepositions we show that this problem leads to the well-defined convex quadratic optimization problem. We measure the performance of the scoring functions obtained on the commonly used benchmarks. We show that our algorithm has inherent stability against overfitting. Also, due to the properties of the basis we used our scoring potentials show some interesting coarse-graining properties.

Given the widely recognized importance of water molecules at the protein-protein interface we also developed potentials for the prediction of water molecules relying on the general ideology of the scoring potentials we developed for the protein-protein contacts.

3.2 Methods

3.2.1 Problem Formulation

Consider N native proteins configurations P_i^{nat}, $i = 1...N$. For each protein complex number i we generate D decoys, P_{ij}^{nonnat}, $j = 1...D$, where the first index runs over different protein complexes and the second index runs over decoys. Our goal is to find a *scoring functional* F, defined for all possible protein-protein complex structures (the set $\mathbb{P}$), such that for each native complex i and its nonnative decoy j the following inequality holds:

$$F(P_i^{nat}) < F(P_{ij}^{nonnat}) \tag{3.1}$$

There are many ways to construct the functional F (Eq. 3.1). To outline its form, we are relying on the following assumptions:

1. Functional F depends only on the interface between the proteins. We define the interface as a set of all atom pairs at a distance smaller than a certain cutoff distance r_{max}, such that the first atom in each pair belongs to the first protein and the second atom in each pair belongs to the second protein.

2. The protein is represented as a set of discrete interaction sites that are located at the centers of the atomic nuclei. All interaction sites are divided into M types according to the properties of the corresponding atomic nuclei. In this study we choose $M = 20$.

3. Functional F depends only on the distribution of the distances between the interaction sites (the number of site pairs at a certain distance),

$$F(P) = F(n^{11}(r), .., n^{kl}(r), .., n^{mm}(r)) = F(n(r)), \tag{3.2}$$

   where $n^{kl}(r)$ is the *number density of site-site pairs* separated by a distance r, with site k located on the first protein, and site l located on the second protein. For homogeneous systems, such as liquids, functions $n^{kl}(r)$ can be expressed via site-site radial distribution functions $g^{kl}(r)$, which can be obtained experimentally, as $n^{kl}(r) = 4\pi r^2 \rho g^{kl}(r) N_a$, where ρ is the number density and N_a is the total number of atoms in the system [52]. However, for proteins this is not the case.

4. F is a linear functional, $F(\alpha n_1(r) + \beta n_2(r)) = \alpha F(n_1(r)) + \beta F(n_2(r))$.

One of the simplest functionals $F(n(r))$ fulfilling these assumptions can be written as:

$$F(n(r)) \equiv F(n^{11}(r), .., n^{kl}(r), .., n^{MM}(r)) = \sum_{k=1}^{M} \sum_{l=k}^{M} \int_0^{r_{max}} n^{kl}(r) U^{kl}(r)\, dr \tag{3.3}$$

It contains unknown functions $U^{kl}(r)$ that can be determined from the training set of native protein complexes. From now on, we will call these functions *scoring potentials*.[1] Once the scoring functions are known, to compute the value of F we

[1] Though the scoring function (Eq. 3.3) is similar by the structure to e.g. the excess internal energy [52], our scoring potentials $U^{kl}(r)$ are not equal to the potential energy functions between sites k and l.

need to specify site-site number densities $n^{kl}(r)$. In practice, we calculate them as a sum of all $k-l$ distances in a given protein complex using the equation:

$$n^{kl}(r) = \sum_{ij} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(r-r_{ij})^2}{2\sigma^2}}, \tag{3.4}$$

where each distance distribution is represented with a Gaussian centered at r_{ij} with the variance of σ^2. The sum is taken over all $k-l$ site pairs i and j separated by the distance r_{ij} smaller than r_{max}, with site k located on the first protein of the complex, and site l located on the second protein. In the limiting case of the variance tending to zero, Eq. 3.4 turns into a sum over Dirac delta functions. In our study we assume the value of σ to be fixed for all site-site distributions. However, if one has an additional information about individual distance distributions, e.g. Debye-Waller factors, molecular dynamics trajectories, etc., it can be used for more precise parametrization of the variance or even instead of the Gaussian approximation in Eq. 3.4. Finally, we compute the score of each conformation using equation [2]:

$$\text{Score} = \sum_{ij} \Upsilon^{kl}(r_{ij}) \tag{3.5}$$

where the sum is taken over all pairs of atoms i and j separated by the distance r_{ij} smaller then r_{max}, with atom i of type k located on the first protein of the complex, and atom j of type l located on the second protein. The function $\Upsilon^{kl}(r)$ is the Gauss transform of the scoring potential $U^{kl}(x)$:

$$\Upsilon^{kl}(r) = \frac{1}{\sqrt{2\pi\sigma^2}} \int_0^{r_{max}} e^{-\frac{(x-r)^2}{2\sigma^2}} U^{kl}(x)\, dx \tag{3.6}$$

3.2.2 Expansion of $U(r)$ and $n(r)$ in an orthogonal basis

Given a set of functions $\xi_p(r)$ orthogonal on the interval $[r_1; r_2]$ with a nonnegative weight function $\Omega(r)$ such that

$$\int_{r_1}^{r_2} \xi_{p_1}(r)\xi_{p_2}(r)\Omega(r)\, dr = \delta_{p_1 p_2}, \tag{3.7}$$

where $\delta_{p_1 p_2}$ is the Kronecker delta function, scoring potentials $U_{kl}(r)$ and number densities $n_{kl}(r)$ can be expanded on the interval $[r_1; r_2]$ as:

$$U_{kl}(r) = \sum_p w_p^{kl} \xi_p(r) \sqrt{\Omega(r)}, \qquad r \in [r_1; r_2] \tag{3.8}$$

$$n_{kl}(r) = \sum_p x_p^{kl} \xi_p(r) \sqrt{\Omega(r)}, \qquad r \in [r_1; r_2] \tag{3.9}$$

[2] Generally, if the distance distributions have a non-Gaussian shape, $n^{kl}(r) = \sum_{ij} f(r - r_{ij})$, functions $\Upsilon^{kl}(r)$ will be computed as a convolution $\Upsilon^{kl} = f * U^{kl}$.

Expansion coefficients w_p^{kl} and x_p^{kl} can be determined from the orthogonality condition (Eq. 3.7) as

$$w_p^{kl} = \int_{r_1}^{r_2} U_{kl}(r)\xi_p(r)\sqrt{\Omega(r)}\ dr \tag{3.10}$$

$$x_p^{kl} = \int_{r_1}^{r_2} n_{kl}(r)\xi_p(r)\sqrt{\Omega(r)}\ dr \tag{3.11}$$

Using expansions Eq. 3.8 and Eq. 3.9, the functional $F(n(r))$ can be rewritten as:

$$F(n(r)) = \sum_{k,l}^{M} \int_{0}^{r_{max}} \sum_{p_1}\sum_{p_2} w_{p_1}^{kl} x_{p_2}^{kl} \xi_{p_1}(r)\xi_{p_2}(r)\Omega(r)\ dr \tag{3.12}$$

In this study we use two types of functions $\xi_p(r)$ orthogonal on the interval $[0; 10]$ with a unit weight, (i) shifted Legendre polynomials and (ii) traditionally used shifted rectangular functions. These two types of functions are plotted in Fig. 3.1. Other types of orthogonal functions can also be used. If the functions $\xi_p(r)$ are chosen to be negligibly small outside the interval $[0; r_{max}]$ or if their interval of orthogonality $[r_1; r_2]$ coincides with the interval $[0; r_{max}]$, as is the case for two sets of our functions, then the scoring functional $F(n(r))$ can be expanded up to the order P as:

$$F(n(r)) \approx \sum_{k=1}^{M}\sum_{l=k}^{M}\sum_{p}^{P} w_p^{kl} x_p^{kl} = (\mathbf{w}\cdot\mathbf{x}), \qquad \mathbf{w},\mathbf{x} \in \mathbb{R}^{P\times M\times(M+1)/2} \tag{3.13}$$

We will refer to the vector $\mathbf{w}$ as to the *scoring vector* and to the vector $\mathbf{x}$ as to the *structure vector*. Equations 3.4 and 3.11 provide the projection from a protein complex structure into the *scoring space* $\mathbb{R}^{P\times M\times(M+1)/2}$. Using these formulas, we can project structural information of each protein complex into a certain structure vector $\mathbf{x}$ on $\mathbb{R}^{P\times M\times(M+1)/2}$.

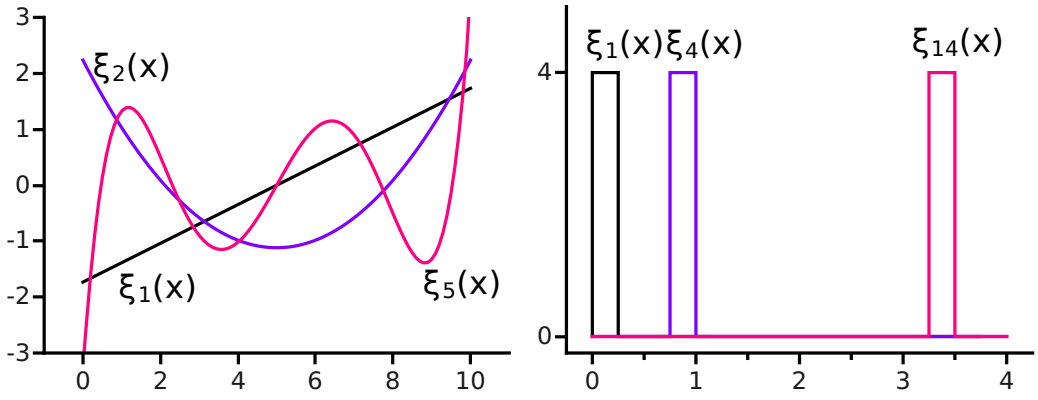


Figure 3.1: Two types of orthogonal functions. Left: shifted Legendre polynomials orthogonal on the interval $[0; 10]$. Right: shifted rectangular functions.

3.2.3 Geometrical interpretation and connection to quadratic programming

Using the expansion of the scoring functional F provided by Eq. 3.13, we can reformulate the scoring problem (Eq. 3.1) as follows – given N native structure vectors $\mathbf{x}_i^{nat}$ and $N \times D$ nonnative structure vectors $\mathbf{x}_{ij}^{nonnat}$, find a scoring vector $\mathbf{w} \in \mathbb{R}^{P \times M \times (M+1)/2}$ such that:

$$\forall i = 1...N, \forall j = 1...D \quad (\mathbf{x}_i^{nat} \cdot \mathbf{w}) < (\mathbf{x}_{ij}^{nonnat} \cdot \mathbf{w}), \tag{3.14}$$

or, equivalently,

$$\forall i = 1...N, \forall j = 1...D \quad ([\mathbf{x}_{ij}^{nonnat} - \mathbf{x}_i^{nat}] \cdot \mathbf{w}) > 0), \tag{3.15}$$

which is a set of $N \times D$ *half-space equations* in $\mathbb{R}^{P \times M \times (M+1)/2}$. Each of the half-spaces is defined by a plane in $\mathbb{R}^{P \times M \times (M+1)/2}$ with the common normal $\mathbf{w}$. Thus, *finding the scoring vector is equivalent to finding the common normal* $\mathbf{w}$ to the planes in Eq. 3.15. Geometrical representation of three groups of structure vectors separated by three parallel hyperplanes with the common normal $\mathbf{w}$ is given in Fig. 3.2.

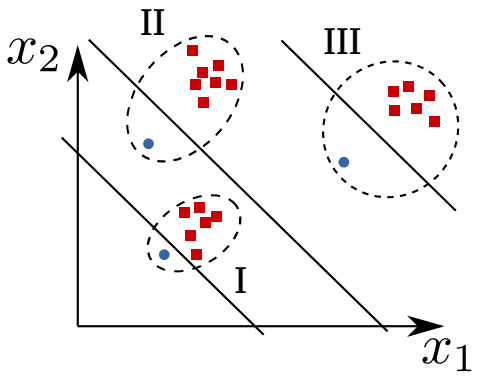


Figure 3.2: Structure vectors for three complexes are shown. Native structure vectors are plotted as blue circles. Nonnative structure vectors are plotted as red squares. Native structure vectors in each complex are separated from nonnative ones by three hyperplanes with a common normal. This normal is the scoring vector $\mathbf{w}$ we are aiming to find.

In the training set, some decoy structures can be very close to the native structures. In practice, we define the native structure as a structure with ligand root-mean-square deviation (lRMSD) smaller than 2 Å. Therefore, for each complex we may have several native structure vectors along with several nonnative structure vectors. Now the question is – how do we determine the set of separating hyperplanes shown in Fig. 3.2 with common normal $\mathbf{w}$? To answer this question we first consider two special cases presented below.

3.2.3.1 Case I. Existence of many solutions

In Fig. 3.3A we present an example of a single complex when infinitely many hyperplanes can separate two classes of structure vectors. A similar example can be easily

constructed for the case with multiple complexes. In case of two classes of vectors, Vapnik proposed to use a special kind of separator, the so-called *optimal separating hyperplane* [142], which is unique and maximizes the distance to the closest point from either class. We can generalize this idea and formulate the following *quadratic programming optimization* problem:

$$\begin{array}{ll} \text{Minimize (in } \mathbf{w},\ \mathrm{b_j}) & \frac{1}{2}\mathbf{w}\cdot\mathbf{w} \\ \text{Subject to} & y_{ij}\left[\mathbf{w}\cdot\mathbf{x}_{ij}-b_j\right]-1\geq 0,\ i=1...N,\ j=1...D \end{array} \tag{3.16}$$

where $y_{ij}=-1$ when the structure vector $\mathbf{x}_{ij}$ is native and $y_{ij}=1$ otherwise. Now we are ready to formulate

Lemma 1. *If exists such a linear scoring functional of form of Eq. 3.13 that correctly discriminates the native structure vectors for all complexes (Eq. 3.15), then the optimal scoring vector is unique and given by the solution of problem (Eq. 3.16).*

Remark. *The scoring vector is optimal in the sense that it maximizes the separation between native and nonnative structure vectors.*

Generally, such a linear scoring functional (with a fixed value of the expansion order P) may not exist, as demonstrated below. Therefore, we will have to modify the optimization problem (Eq. 3.16).

3.2.3.2 Case II. No solution exists

In Fig. 3.3B we present an example when no hyperplane can separate the two classes of the structure vectors of a single complex. For this case, Cortes and Vapnik proposed to relax the condition for the optimal separating hyperplane [29], including an additional term. This term minimizes the sum of penalties for misclassified vectors. We again generalize this idea and introduce for each decoy set $j=1...D$ *slack variables* ξ_{ij}, which are positive for misclassified structure vectors and zero otherwise. A non-zero value of ξ_{ij} allows the structure vector x_{ij} to overcome the inequality condition in Eq. 3.16 at a cost proportional to the value of ξ_{ij} (see Fig. 3.3B). The new *soft-margin* quadratic optimization problem reads:

$$\begin{array}{ll} \text{Minimize (in } \mathbf{w},\ b_j,\ \xi_{ij}): & \frac{1}{2}\mathbf{w}\cdot\mathbf{w}+\sum_{ij}C_{ij}\xi_{ij} \\ \text{Subject to:} & y_{ij}\left[\mathbf{w}\cdot\mathbf{x}_{ij}-b_j\right]-1+\xi_{ij}\geq 0,\ i=1...N,\ j=1...D \\ & \xi_{ij}\geq 0 \end{array} \tag{3.17}$$

The solution of this problem provides a trade-off between how large will be the separation between two classes of the structure vectors of each complex and how many misclassified vectors will be in the solution. Parameters C_{ij} can be regarded as *regularization parameters*. The solution of Eq. 3.17 tends to maximize the structure vector separation for small values of C_{ij} or minimize the number of misclassified structure vectors for large values of C_{ij}. We choose parameters C_{ij} to be different for native and nonnative structure vectors of each complex because fewer native structure vectors should have the larger weight (see for instance [6]). The following lemma provides the foundation for the numerical scheme used in this work:

Lemma 2. *The optimal scoring vector is unique and given by the solution of problem (Eq. 3.17).*

Remark. *Here, the scoring vector is optimal in the sense that it maximizes the separation between native and nonnative structure vectors and minimizes the number of misclassified vectors. Regularization parameters C_{ij} in Eq. 3.17 tune the importance of either factors.*

The proof of lemmas (1, 2) can be found, e.g., in [21]. Overall, the formulation of the optimization problem (Eq. 3.17) is very similar to the formulation of soft-margin *support vector machine* (SVM) problem [29]. Therefore, to solve this problem (Eq. 3.17) we will use techniques developed for SVM.

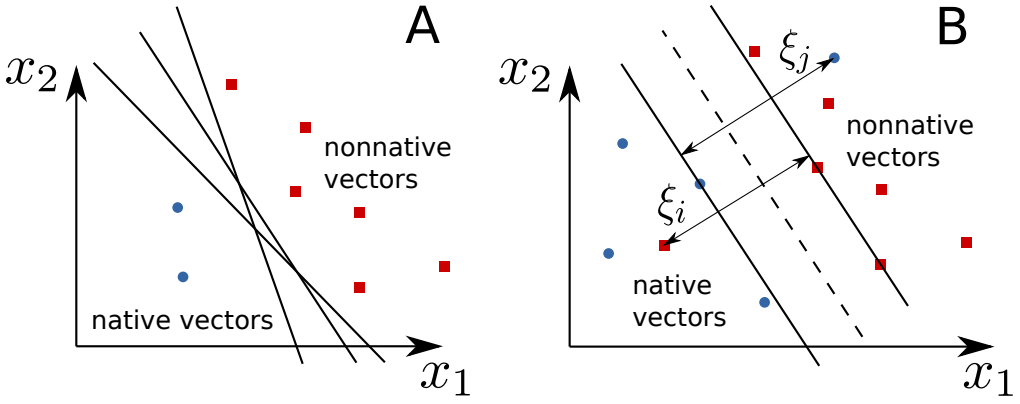


Figure 3.3: Two classes of structure vectors for a single complex are shown. Native structure vectors are plotted as blue circles. Nonnative structure vectors are plotted as red squares. A) The case when infinitely many hyperplanes can separate the two classes. B) The case when no optimal separating hyperplane exists. Slack variables ξ_i and ξ_j for misclassified structure vectors are added, which are the distances to the corresponding margin hyperplanes. The optimal hyperplane, which maximizes the separation between the two classes, is plotted as a dashed line. Two margin hyperplanes are plotted as solid lines.

3.2.4 Algorithm

3.2.4.1 Dual problem

Properties and solutions of quadratic optimization problems similar to the one stated above (Eq. 3.17) have been extensively studied in the theory of convex programming [17, 142]. For instance, using the *Lagrangian formalism* , the optimization problem (Eq. 3.17) can be converted into its dual form (Appendix A), and the resulting dual optimization problem is *convex*:

$$\begin{array}{ll} \text{Maximize } \mathcal{L}(\lambda_{ij}): & \mathcal{L}(\lambda_{ij}) = \sum_{ij}\lambda_{ij} - \frac{1}{2}\sum_{ij}\sum_{kl} y_{ij}y_{kl}\lambda_{ij}\lambda_{kl}\mathbf{x}_{ij}\cdot\mathbf{x}_{kl} \\ \text{Subject to:} & 0 \le \lambda_{ij} \le C_{ij} \\ & \sum_i y_{ij}\lambda_{ij} = 0, \quad \forall j \end{array} , \qquad (3.18)$$

where the maximization is performed with respect to the *Lagrange multipliers* λ_{ij}. This dual problem is similar to the the soft-margin SVM optimization problem [29]. The difference lies in the constraints. For the soft margin SVM, conditions on the parameters written in the same two-indexed form as in Eq. 3.18, are $\sum_{ij} y_{ij}\lambda_{ij} = 0$.

Vectors $\mathbf{x}_{ij}$ for which $\lambda_{ij} > 0$ are called *support vectors*. Once the dual problem (Eq. 3.18) is solved and the Lagrange multipliers λ_{ij} are found, we can express the solution of the original primal problem (Eq. 3.17) (the scoring vector) as a linear combination of the support vectors:

$$\mathbf{w} = \sum_{\text{support vectors}} y_{ij}\lambda_{ij}\mathbf{x}_{ij} \tag{3.19}$$

The dual representation (Eq. 3.18) of the original primal quadratic problem (Eq. 3.17) allows us to break down the original large quadratic optimization problem into a series of smaller sub-problems. Below we describe an algorithm that solves the dual optimization problem (Eq. 3.18) using a decomposition technique.

3.2.4.2 Block sequential minimal optimization algorithm

Due to its enourmus size, the quadratic optimization problem can not easily be solved by standard techniques. The quadratic form in Eq. 3.18 involves a matrix with number of elements proportional to the squared number of the training structure vectors. This matrix often exceeds the size of available RAM, for instance, explicit storage of the matrix used in the current study requires about 20GB of memory. Nonetheless, algorithms that deal with large datasets are widely used in machine learning. More precisely, various decomposition techniques have been developed to reduce the requirements of optimization solvers to the size of available RAM [141, 104]. Here, we employ a *block-decomposition technique* and propose the *block sequential minimal optimization* (BSMO) algorithm. Briefly, we partition the training set into N blocks, each block containing one native structure vector with its D nonnative structure vectors. Then, for each block i, we iteratively optimize each pair of Lagrange multipliers (λ_1, λ_2), preserving the equality constraint $y_1\lambda_1 + y_2\lambda_2 = const$. To do this, we write the Lagrangian (Eq. 3.18) as a function of λ_1 and λ_2:

$$\mathcal{L}(\lambda_1, \lambda_2) = \frac{1}{2}\eta\lambda_2^2 - \eta\lambda_2\lambda_2^{old} + \lambda_2 y_2(y_2 - y_1) + \lambda_2 y_2(\mathbf{x}_{i1} - \mathbf{x}_{i2}) \cdot \mathbf{w}^{old} + \text{Const.} \tag{3.20}$$

with

$$\eta = 2\mathbf{x}_{i1} \cdot \mathbf{x}_{i2} - \mathbf{x}_{i1} \cdot \mathbf{x}_{i1} - \mathbf{x}_{i2} \cdot \mathbf{x}_{i2} \tag{3.21}$$

Then, we analytically maximize this Lagrangian with respect to λ_1 and λ_2 according to the *sequential minimal optimization* (SMO) algorithm [104]. After the minimization, we obtain new values of λ_1 and λ_2. We provide more details about the SMO algorithm in Appendix B. After each iteration, we recompute the current scoring vector $\mathbf{w}^{new}$ (see Eq. 3.19) according to:

$$\mathbf{w}^{new} = \mathbf{w}^{old} + \Delta\lambda_1 y_1 \mathbf{x}_{i1} + \Delta\lambda_2 y_2 \mathbf{x}_{i2} \tag{3.22}$$

For each block i, we continue the iterative optimization of the Lagrangian (Eq. 3.20) until the relative change in its value between two successive inner cycles of iterations is less than the desired tolerance. Each inner cycle consists in the optimization of all pairs of Lagrange multipliers for a given block i. Globally, we terminate the optimization when the relative change in the value of the Lagrangian (Eq. 3.18)

between two successive outer cycles is less than the desired tolerance. Each outer cycle consists in the optimization of all blocks of the training set. The flowchart of the BSMO algorithm is presented in Figure 3.2.4.2.

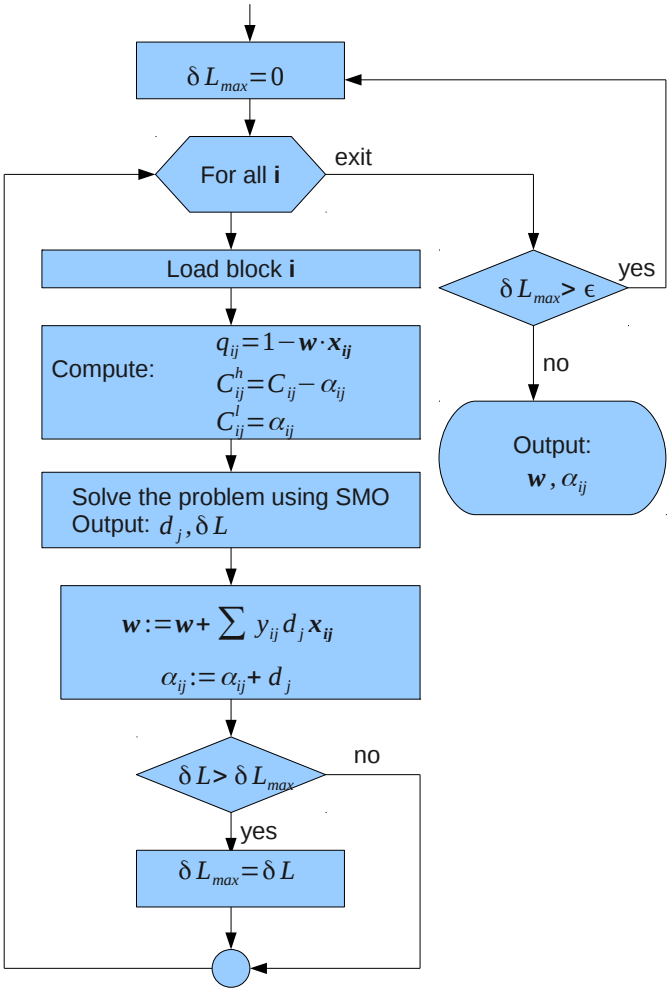


Figure 3.4: The flowchart of block sequential minimal optimization algorithm.

As it is seen from Eq. 3.21, our BSMO algorithm requires only scalar products of the structure vectors within the same block. Therefore, it is sufficient to load each block into RAM sequentially, which results in memory efficiency of our method. Precisely, RAM required for our implementation of the block-decomposition solver is N^2 times less compared to the direct quadratic problem solvers.

3.2.5 Training database for protein-protein interactions

In the present study we used the training database of 851 non-redundant protein-protein complex structures prepared by Huang and Zou [56]. This database contains protein-protein complexes extracted from the PDB [14] and includes 655 homodimers and 196 heterodimers. We updated three PDB structures from the original

training database: 2Q33 supersedes 1N98, 2ZOY supersedes 1V7B, and 3KKJ supersedes 1YVV. The training database contains only crystal dimeric structures determined by X-ray crystallography at resolution better than 2.5 Å. Each chain of the dimeric structure has at least 10 amino acids, and the number of interacting residue pairs (as defined as having at least 1 heavy atom within 4.5 Å) is at least 30. Each protein-protein interface consists only of 20 standard amino acids. No homologous complexes were included in the training database. Two protein complexes were regarded as homologues if the sequence identity between receptor-receptor pairs and between ligand-ligand pairs was $> 70\%$. Finally, Huang and Zou [56] manually inspected the training database and left only those structures that had no artifacts of crystallization.

Our algorithm requires as input native and nonnative structure vectors (see, e.g., Eq. 3.15). Native structure vectors can be computed from the native protein-protein contacts in the training database using Eq. 3.11. However, for the computation of the nonnative structure vectors for each protein-protein complex from the training database, we need to generate decoys for each complex. Since our optimization algorithm is very general and has no special requirements for nonnative protein-protein contacts, we generated them by "rolling" a smaller protein (ligand) over the surface of a bigger protein (receptor) using HEX protein docking software [113, 2]. To do so, we initialized HEX exhaustive search algorithm with the radial search step of 1.5 Å and expansion order of the shape function equal to 31. We used only the shape complementarity energy function from HEX (i.e., electrostatic contribution was omitted). Afterwards, we clustered HEX docking results with a root mean square (RMS) threshold of 8 Å. The top 200 clusters, ranked by HEX surface complementarity function, plus the native protein-protein complex conformation (giving in total 201 structures) were then used to compute the distance distribution functions (Eq. 3.4). Then, we computed the structure vectors using Eq. 3.11 and labeled them as "native" if the root mean square deviation (RMSD) of the corresponding ligand was < 2 Å from its native position. Otherwise, the structure vector was labeled as "nonnative" or "decoy". On average, we obtained about 2.5 native structure vectors (and, correspondingly, about 198.5 nonnative structure vectors) per protein-protein complex. To each structure vector $\mathbf{x}_{ij}$ we assigned a regularization parameter C_{ij} according to

$$\begin{aligned} C_{ij}^{native} &= CD_j^{nonnative}/D_j \\ C_{ij}^{nonnative} &= CD_j^{native}/D_j \end{aligned} \tag{3.23}$$

We repeated the same procedure for each protein-protein complex from the training database. We used $M = 20$ atom-centered interaction sites based on the atom types definitions provided by Huang and Zou [56]. These atom types were defined by the classification of all heavy atoms in 20 standard amino acids according to their element symbol, aromaticity, hybridization, and polarity. These 20 atom types result in total of 210 pair potentials.

Our training set has several proteins homologous to the ones from the two widely used docking benchmarks, Rosetta, and Zdock, which we employ below to validate our results. We define two protein complexes to be homologous if for each chain in the first complex there is a chain in the second complex with sequence identity more than 60%. We determined the sequence identity using the FASTA36 program [98]. Below, we benchmarked our scoring function while both excluding homologs

from the training set and leaving it unchanged. The comparison of the benchmark results in these two cases is shown in supplementary Tables S1 and S2.

3.2.5.1 Training database for water-protein interactions

To obtain potentials for the water molecules we employed the same set of complexes prepared by Huang and Zou [56]. We added a new atom type to the set of 20 atom types for docking corresponding to the water oxygen. We generated a cube with dimensions $130 \times 130 \times 130\mathring{A}$ of TIP3P water molecules and equlibrated it using molecular dynamics at room temperature with periodic boundary conditions. Afterwards native structures and decoys from the training set were immersed into this water-box. In the case when the size of the protein was more than $130\mathring{A}$ we periodically continued the water box. Then, in case of native structures initial water molecules presented in the crystallographic structures were retained. In the case of decoys we removed all the original water molecules. The placed bulk molecules were then removed if they clashed with either the proteins or retained water molecules. The clashing distance was set to be $2\mathring{A}$. We also removed each water molecules farther than $12\mathring{A}$ from all protein atoms. Although we aimed to obtain all pairwise potentials we did not use water-water potential for the predictions, because it influences the equilibrium structure of bulk water, thus in principle after obtaining this potential we had to recalculate water box. However this procedure goes beyond our method of obtaining scoring potentials.

3.3 Results and Discussion

3.3.1 Overfitting and Convergence

Various methods of derivation of the knowledge-based potentials usually produce results biased towards the training data set. Typically, such algorithms maximize the predictive accuracy of the corresponding potential on a set of training data, which does not mean that the same potential will perform equally well on a new set of data. Indeed, fitting the potential to the training data set also fits the noise in the data. Thus, very often a knowledge-based potential memorizes noisy features of the training data instead of deducing general predictive concepts from it. This phenomenon is usually referred to as *overfitting* [32]. A clear indication of an "overfitted" potential can be, for example, the need for post-smoothing techniques applied to the initial knowledge-based potential, as in [56, 88]. "Overfitting" is clearly not desirable. In order to avoid it, many optimization techniques (regularization, cross-validation, etc.) have been successfully proposed to penalize the initial objective function with various additional terms [8, 69]. These terms serve to achieve a better predictive accuracy on the off-training data based on the predictions of the training data.

To avoid overfitting, we used two techniques – *regularization* and *cross-validation*. Regularization penalizes the initial objective function with various additional terms [8, 69]. We introduced two regularization parameters, σ for the width of the Gaussian distribution of distances in Eq. 3.4, and C for the hinge loss function in Eq. 3.17. To find the best values of these parameters we used the following cross-validation procedure. First, we divided the training set into two parts, consisting of 650 complexes (temporary test set) and 200 complexes (temporary training set). Then, for each value of σ and C, we obtained the scoring potentials using the temporary training set and verified it on the temporary test set. Finally, we chose those values of σ and C that correspond to the maximum number of guessed structures in the temporary test set. We define the structure as guessed if its native complex has the score better than all of its decoys. Figure 3.5 shows the predictive performance of the scoring potential on the two sets as a function of σ and C. Obviously, the maximum predictive performance on the training set is achieved at the highest values of C. However, the validation on the test set highlights the best choice of values of C and σ. These values are $C = 10^6 \dots 10^7$ and $\sigma = 0.4$ Å (Fig. 3.5B).

Figure 3.6 shows the convergence of the success rate on the training set with the number of iterations of the training BSMO algorithm. The success rate was measured as the number of guessed structures divided by the total number of protein-protein complexes. We can see a fast convergence of the method. In principle, a hundred optimization steps is sufficient to obtain the final result. We have also observed that increasing the regularization parameter C leads to the slower convergence and vice versa. We should note that thanks to the convexity of our optimization problem, its solution is unique and does not depend on the starting point and the optimization method used.

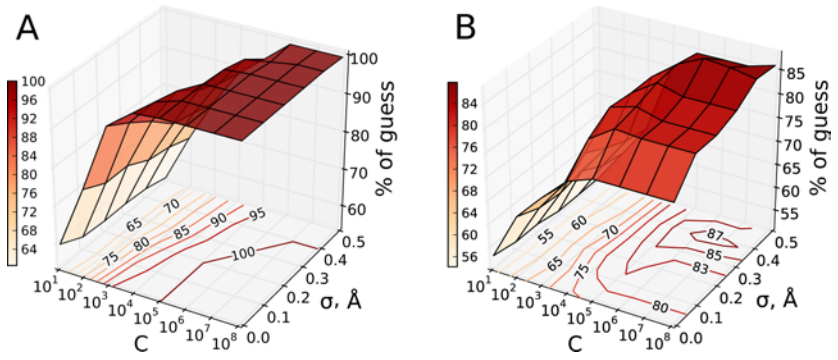


Figure 3.5: Predictive performance of the scoring potential as a function of the smoothing parameter σ and the regularization parameter C. A) Performance obtained if the scoring functions are trained on the whole database and verified on the same database. B) Performance obtained if the scoring functions are trained on 200 protein complexes and verified on the other 650 complexes from the training database. Here the best performance is obtained with $\sigma = 0.4$ Å and $C = 10^6 \ldots 10^7$.

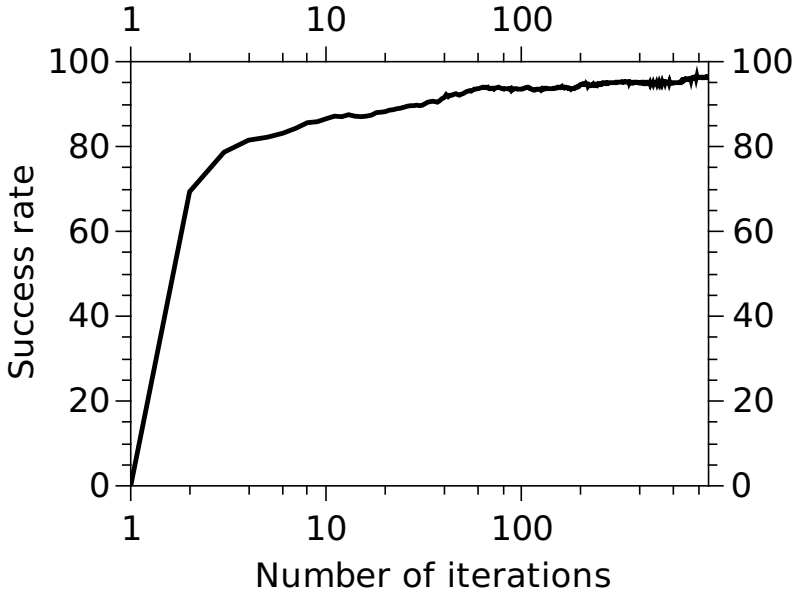


Figure 3.6: Performance rate of the scoring potentials on the training set versus the number of iterations of the BSMO algorithm. The scoring potentials were obtained using the Legendre basis and the whole training set, without excluding any homologous proteins. Parameters σ and C were set to the optimal values of $\sigma = 0.4$ Å and $C = 10^5$.

3.3.2 Extracted Potentials

Our method can in principle use any type of orthogonal polynomials to decompose the structural statistics and reconstruct the potentials. However, since rectangular

functions are the most widely used to collect statistics, we employed this basis as a reference. Then, we also used the Legendre basis, orthogonal on the interval [0;10]. We chose this basis because of its simplicity, in particular because the weight function for this basis does not depend on the distance.

From now on, we call the obtained scoring potentials the Convex Protein Protein (ConvexPP) potentials. Figure 3.7 shows typical scoring potentials derived using two different orthogonal bases. Obtained potentials are smooth by construction, thanks to the smooth Gaussian kernel in Eq. 3.6. According to the plot, the shape of the potentials does not depend on the basis set that was used to derive it. This is the consequence of the global convergence of the optimization problem (see lemma 2). We can also see that the obtained potentials tend to zero as the interaction distance increases. On the other hand, all the potentials approach zero at short distances. This is due to the absence of statistics for the native structures at short distances and the result of the $\mathbf{w} \cdot \mathbf{w}$ term in optimization problem 3.17. We discuss this behaviour in more detail below.

Due to the Gaussian smoothing of statistics, it is sufficient to use the expansion order of $P = r_{max}/\sigma$. For $\sigma = 0.4$ Å and $r_{max} = 10$ Å, the estimate on the number of basis functions is $P = 25$. However, due to the adjustment of σ with the cross-validation procedure, in our experiments we used a larger expansion order, $P = 40$. Figure 3.8 demonstrates how the resulting potentials depend on the expansion order. We should note that the decompositions of orders above 25 are almost indistinguishable and thus are not shown. Indeed, increasing the order of the polynomial P decreases the distance between two consecutive zeros in the polynomial basis. Therefore, at a constant value of σ, the integral in Eq. (3.11) tends to zero with the growth of the value of P due to the oscillatory behaviour of the basis polynomials. Such behaviour of the integral confines all the useful information about the distributions and the scoring potentials in the first few coefficients of the polynomial decomposition. The number of these coefficients depends solely on the value of σ and does not change with the training set or the value of parameter C.

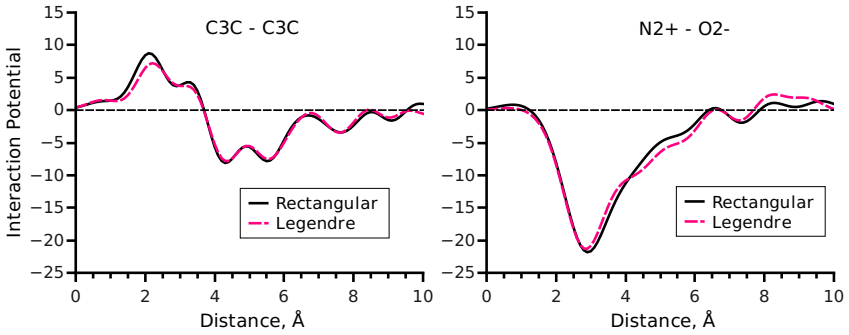


Figure 3.7: The scoring potentials trained in two different polynomial bases. Dashed lines correspond to the scoring potentials that were obtained using the Legendre basis functions. Solid lines correspond to the potentials that were obtained using the rectangular basis functions. Left: Potential between aliphatic carbons bonded to carbons or hydrogens only. Right: Potential between a guanidine nitrogen with two hydrogens and an oxygen in carboxyl groups.

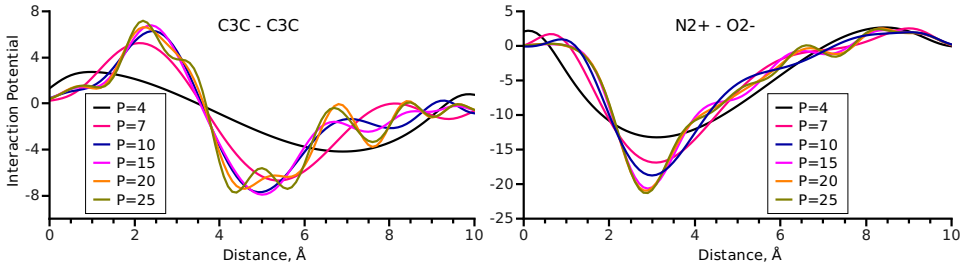


Figure 3.8: Dependence of the extracted scoring potentials on the order of the decomposition in the Legendre basis. After order $P = 25$, the potentials are indistinguishable from each other and thus not shown for clarity. Left: Potential between aliphatic carbons bonded to carbons or hydrogens only. Right: Potential between a guanidine nitrogen with two hydrogens and an oxygen in carboxyl groups.

3.3.3 Protein-Protein docking benchmark version 3.0

First we tested the ConvexPP scoring function on the protein-protein docking benchmark version 3.0. It consists of 124 crystallographic structures of protein-protein complexes from PDB database [59]. They are divided into three groups: rigid, medium and difficult cases. The division criteria is the scale of conformational changes of the proteins upon binding: from minor changes in rigid cases to major in difficult. The nonredundancy of the benchmark was set at the level of family-family pairs. That means that if a complex in Benchmark v3.0 is formed of the protein of family A and another one of family B, then there are no more family A - family B complex in the benchmark. The assignment of a protein to a family was taken according to SCOP database [91].

The decoys for scoring were generated using ZDOCK3.0 [103] with the sampling step equal to 6 degrees (we call this set of docking position *ZDOCK benchmark* below). They were downloaded from the ZLAB website [3]. The docking program ZDOCK3.0 generates the rigid-body protein-protein docking predictions with the corresponding scores. Scoring function used in this program includes shape complementarity, statistical pair potentials and electrostatics. To compare our scoring function to the well established one downloaded the decoy set reranked by ZRANK [102]. It is the program for reranking the ZDOCK3.0 predictions. In addition to the factors used in ZDOCK3.0, it computes detailed electrostatics, estimates desolvation and uses additional Van-der-Waals potential to re-score the decoys.

The benchmark 3.0 has several complexes homologous to certain protein complexes in the training set. Therefore, to see the effect of training set and test set similarity we trained our potential both excluding homologs from the training set and leaving it unchanged. Table 3.1 shows results of ZDOCK3.0, ZRANK and our scoring functions on the ZDOCK benchmark.

Complex	ZDock				ZRank				ConvexPP			
	Rank	L_{rmsd}	I_{rmsd}	F_{nat}	Rank	L_{rmsd}	I_{rmsd}	F_{nat}	Rank	L_{rmsd}	I_{rmsd}	F_{nat}
Rigid-Body												
1AHW	54	8.30	1.58	0.60	175	2.68	1.26	0.72	547	2.14	0.91	0.79
1BVK	-	-	-	-	-	-	-	-	-	-	-	-
1DQJ	-	-	-	-	-	-	-	-	-	-	-	-
1E6J	1	3.80	1.59	0.64	12	6.06	2.35	0.40	35	4.21	2.26	0.48
1JPS	1	3.90	1.04	0.70	254	4.32	1.26	0.65	62	4.14	1.11	0.78
1MLC	5	4.61	1.14	0.36	54	4.61	1.14	0.36	5	4.70	1.12	0.39
1VFB	997	10.89	2.48	0.30	798	0.89	2.48	0.30	1239	10.89	2.48	0.30
1WEJ	2	2.44	0.79	0.91	41	2.44	0.79	0.91	1	4.13	1.30	0.75
2FD6	15	18.67	2.42	0.73	9	9.76	2.42	0.80	8	15.65	2.16	0.80
2I25	1534	7.92	2.21	0.36	1	4.45	1.87	0.33	83	4.45	1.87	0.33
2VIS	8	27.81	2.02	0.63	150	6.31	2.18	0.57	617	23.89	2.37	0.43
1BJ1	19	6.29	1.19	0.62	1	4.22	0.97	0.88	2	2.82	0.98	0.86
1FSK	1	2.69	1.04	0.91	3	1.39	0.65	0.93	3	3.98	1.39	0.81
1I9R	40	3.07	1.53	0.79	493	3.07	1.53	0.79	462	16.85	2.30	0.48
1IQD	169	5.25	1.01	0.60	3	3.08	0.88	0.73	16	4.20	0.97	0.67
1K4C	587	7.42	1.67	0.43	1615	9.12	1.64	0.45	242	5.78	1.31	0.62
1KXQ	14	2.04	1.28	0.70	1	1.75	0.93	0.93	1	3.06	1.04	0.88
1NCA	14	2.85	0.92	0.83	150	1.75	0.97	0.76	12	4.50	1.38	0.86
1NSN	473	4.95	2.00	0.50	728	2.41	1.06	0.79	636	4.95	2.00	0.50
1QFW	192	5.05	1.24	0.71	310	4.21	1.41	0.77	1315	5.12	1.35	0.73
1QFW	192	5.05	1.24	0.71	310	4.21	1.41	0.77	1315	5.12	1.35	0.73
2JEL	1239	6.12	1.90	0.55	223	6.12	1.90	0.55	957	6.83	2.30	0.31
1AVX	**11**	**7.49**	**1.61**	**0.56**	**7**	**6.73**	**1.86**	**0.54**	**3**	**4.85**	**2.23**	**0.39**
1AY7	**74**	**3.82**	**2.13**	**0.52**	**468**	**3.82**	**2.13**	**0.52**	**185**	**5.73**	**1.82**	**0.45**
1BVN	**16**	**2.60**	**1.54**	**0.43**	**1**	**3.74**	**1.85**	**0.46**	**3**	**4.09**	**1.74**	**0.50**
1CGI	**89**	**4.27**	**2.34**	**0.43**	**14**	**4.27**	**2.34**	**0.43**	**61**	**3.20**	**2.30**	**0.49**
1D6R	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
1DFJ	2	5.12	2.08	0.55	1	3.82	1.87	0.52	1	5.97	2.42	0.50
1E6E	5	2.97	2.00	0.42	15	2.98	1.72	0.53	9	4.01	2.41	0.42
1EAW	1	9.32	2.48	0.46	42	9.32	2.48	0.46	1	2.60	1.03	0.70
1EWY	21	3.16	1.74	0.56	9	3.32	1.88	0.61	21	3.16	1.74	0.56
1EZU	-	-	-	-	-	-	-	-	-	-	-	-
1F34	62	7.51	2.34	0.41	34	5.95	2.46	0.49	38	3.41	1.45	0.54
1HIA	-	-	-	-	-	-	-	-	-	-	-	-
1MAH	3	2.77	1.12	0.72	1	2.77	1.12	0.72	1	3.64	1.26	0.69
1N8O	92	4.60	1.51	0.60	1	5.15	1.51	0.68	1	2.94	1.24	0.74
1OPH	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
1PPE	**1**	**1.84**	**0.77**	**0.79**	**1**	**2.25**	**0.86**	**0.83**	**1**	**4.62**	**1.52**	**0.71**
1R0R	**70**	**1.83**	**0.71**	**0.74**	**178**	**1.32**	**0.74**	**0.60**	**2**	**8.36**	**2.46**	**0.40**
1TMQ	**71**	**4.92**	**2.08**	**0.35**	**61**	**3.61**	**1.49**	**0.60**	**8**	**6.11**	**1.97**	**0.45**
1UDI	-	-	-	-	-	-	-	-	-	-	-	-
1YVB	38	12.34	2.32	0.54	6	7.33	1.92	0.71	8	7.33	1.92	0.71
2B42	10	6.17	1.17	0.89	1	6.17	1.17	0.89	8	9.44	2.23	0.43

2MTA	748	4.86	2.41	0.59	1352	2.96	1.81	0.59	125	3.76	1.88	0.50
2O8V	-	-	-	-	-	-	-	-	-	-	-	-
2PCC	920	6.29	2.19	0.44	975	5.67	2.17	0.39	458	6.29	2.19	0.44
2SIC	1	6.47	1.44	0.52	3	8.89	2.40	0.43	3	5.92	1.19	0.73
2SNI	**178**	**4.14**	**1.44**	**0.53**	**230**	**6.34**	**2.16**	**0.42**	**8**	**8.57**	**2.45**	**0.57**
2UUY	22	12.46	2.11	0.74	380	13.94	2.39	0.66	233	13.72	2.38	0.66
7CEI	**3**	**4.33**	**1.36**	**0.65**	**1**	**0.68**	**2.44**	**0.44**	**4**	**8.41**	**2.46**	**0.65**
1A2K	-	-	-	-	-	-	-	-	-	-	-	-
1AK4	-	-	-	-	-	-	-	-	-	-	-	-
1AKJ	175	6.35	2.15	0.53	637	4.22	1.70	0.65	395	3.30	1.54	0.60
1AZS	123	11.64	1.86	0.69	8	2.49	1.74	0.65	1	11.04	1.88	0.69
1B6C	2	7.35	2.37	0.74	1	3.47	2.38	0.87	2	3.67	2.23	0.90
1BUH	353	4.13	1.56	0.60	18	3.42	1.81	0.63	1	3.42	1.81	0.63
1E96	**24**	**5.56**	**1.99**	**0.54**	**219**	**9.54**	**2.38**	**0.42**	**261**	**6.03**	**2.18**	**0.58**
1EFN	-	-	-	-	-	-	-	-	-	-	-	-
1F51	237	3.41	1.82	0.60	368	3.41	1.82	0.60	16	3.41	1.82	0.60
1FC2	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
1FQJ	-	-	-	-	-	-	-	-	-	-	-	-
1GCQ	922	1.98	1.19	0.71	1171	1.98	1.19	0.71	118	1.98	1.19	0.71
1GHQ	-	-	-	-	-	-	-	-	-	-	-	-
1GLA	12	3.38	1.52	0.32	919	3.38	1.52	0.32	57	4.91	2.22	0.37
1GPW	1	5.05	2.03	0.58	39	7.13	2.39	0.50	1	7.10	2.44	0.39
1HE1	43	6.92	2.20	0.47	2	8.68	2.35	0.47	301	5.93	1.74	0.38
1I4D	-	-	-	-	-	-	-	-	-	-	-	-
1J2J	**1897**	**5.59**	**2.12**	**0.55**	**277**	**5.59**	**2.12**	**0.55**	**86**	**5.59**	**2.12**	**0.55**
1K74	1	5.86	2.29	0.42	1	6.11	2.35	0.52	8	7.90	2.02	0.48
1KAC	11	4.47	2.07	0.40	438	5.35	2.22	0.36	287	4.47	2.21	0.36
1KLU	-	-	-	-	-	-	-	-	-	-	-	-
1KTZ	397	6.04	1.15	0.63	1486	4.25	1.54	0.74	282	5.39	1.25	0.63
1KXP	**40**	**4.94**	**1.79**	**0.49**	**1**	**6.29**	**2.09**	**0.46**	**1**	**7.43**	**2.17**	**0.51**
1ML0	1	2.27	1.25	0.76	1	5.05	2.07	0.51	1	4.47	1.89	0.61
1QA9	-	-	-	-	-	-	-	-	-	-	-	-
1RLB	25	8.50	2.05	0.55	2	9.11	1.93	0.66	7	9.11	1.93	0.66
1S1Q	1195	2.35	1.53	0.58	420	2.35	1.53	0.58	766	2.35	1.53	0.58
1SBB	-	-	-	-	-	-	-	-	-	-	-	-
1T6B	351	6.12	1.49	0.52	166	5.90	1.87	0.48	89	5.91	2.03	0.64
1XD3	135	6.68	2.21	0.55	62	4.87	2.49	0.30	6	4.87	2.49	0.30
1Z0K	19	4.60	1.99	0.74	1	5.52	1.99	0.74	11	4.59	1.68	0.56
1Z5Y	77	5.69	1.97	0.50	46	5.80	2.27	0.41	8	6.58	1.97	0.50
1ZHI	**71**	**4.80**	**1.32**	**0.71**	**119**	**8.39**	**1.79**	**0.65**	**78**	**9.90**	**1.96**	**0.61**
2AJF	-	-	-	-	-	-	-	-	-	-	-	-
2BTF	655	5.61	2.31	0.31	398	5.61	2.31	0.31	655	6.00	2.20	0.33
2HLE	1	3.38	2.11	0.42	9	5.95	2.43	0.42	9	6.84	2.35	0.35
2HQS	1092	8.94	2.30	0.37	1162	8.94	2.30	0.37	576	8.94	2.30	0.37
2OOB	-	-	-	-	-	-	-	-	-	-	-	-

Medium

1BGX	-	-	-	-	-	-	-	-	-	-	-	-
1ACB	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**

1IJK	444	7.43	1.65	0.31	376	5.02	1.35	0.38	124	6.42	1.83	0.25
1KKL	1002	6.23	2.50	0.44	1774	6.23	2.50	0.44	325	6.23	2.50	0.44
1M10	-	-	-	-	-	-	-	-	-	-	-	-
1NW9	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
1GP2	-	-	-	-	-	-	-	-	-	-	-	-
1GRN	**1365**	**7.10**	**2.49**	**0.00**	**676**	**7.10**	**2.49**	**0.00**	**1785**	**7.10**	**2.49**	**0.00**
1HE8	-	-	-	-	-	-	-	-	-	-	-	-
1I2M	-	-	-	-	-	-	-	-	-	-	-	-
1IB1	-	-	-	-	-	-	-	-	-	-	-	-
1K5D	1111	8.04	2.03	0.29	466	8.04	2.03	0.29	1185	8.04	2.03	0.29
1N2C	-	-	-	-	-	-	-	-	-	-	-	-
1WQ1	-	-	-	-	-	-	-	-	-	-	-	-
1XQS	314	6.91	2.47	0.34	19	6.91	2.47	0.34	199	5.60	2.28	0.38
2CFH	237	5.20	2.12	0.36	1	3.83	1.86	0.47	1	5.20	2.12	0.36
2H7V	525	13.69	2.47	0.44	98	3.69	2.47	0.44	8	13.69	2.47	0.44
2HRK	-	-	-	-	-	-	-	-	-	-	-	-
2NZ8	-	-	-	-	-	-	-	-	-	-	-	-
Difficult												
1E4K	-	-	-	-	-	-	-	-	-	-	-	-
2HMI	-	-	-	-	-	-	-	-	-	-	-	-
1FQ1	-	-	-	-	-	-	-	-	-	-	-	-
1PXV	-	-	-	-	-	-	-	-	-	-	-	-
1ATN	-	-	-	-	-	-	-	-	-	-	-	-
1BKD	-	-	-	-	-	-	-	-	-	-	-	-
1DE4	-	-	-	-	-	-	-	-	-	-	-	-
1EER	-	-	-	-	-	-	-	-	-	-	-	-
1FAK	-	-	-	-	-	-	-	-	-	-	-	-
1H1V	-	-	-	-	-	-	-	-	-	-	-	-
1IBR	-	-	-	-	-	-	-	-	-	-	-	-
1IRA	-	-	-	-	-	-	-	-	-	-	-	-
1JMO	-	-	-	-	-	-	-	-	-	-	-	-
1R8S	-	-	-	-	-	-	-	-	-	-	-	-
1Y64	-	-	-	-	-	-	-	-	-	-	-	-
2C0L	-	-	-	-	-	-	-	-	-	-	-	-
2OT3	-	-	-	-	-	-	-	-	-	-	-	-
Homologs included	Top1: 8.1% (10/124) Top10: 16.1% (20/124)				Top1: 12.9% (16/124) Top10: 22.6% (28/124)				Top1: 10.5% (13/124) Top10: 27.4% (34/124)			
Homologs excluded									Top1: 12.1% (15/124) Top10: 29.0% (36/124)			

Table 3.1: ZDock benchmark 3.0 results. Proteins homologous to the ones in the training set are shown with the bold font. Absense of hits among first 2000 predictions is shown with hyphens.

We reranked top 2000 decoys generated by ZDOCK3.0 using our scoring potentials. A hit is a predicted near-native decoy with iRMSD (RMSD of C_α atoms of the predicted interaction interface residues after superposition onto the crystallized complex) less than 2.5 Å. The number of hits when only the top one prediction considered (Top1) obtained by ZRANK is higher than that obtained by ConvexPP

potentials (15 vs 12 hits). Although if we consider top 8 predictions our scoring function outperforms ZRANK (32 vs 26 hits) and gives the same number of hits for top 5 to 8 predictions. Excluding homologs from the training set results in a slight improvement of the results (Table C.1).

Figure 3.9 shows ROC curves (success rate vs the number of top predictions considered). We see that ConvexPP scoring functions outperform ZRANK and ZDOCK if the number of considered predictions is more than eight.

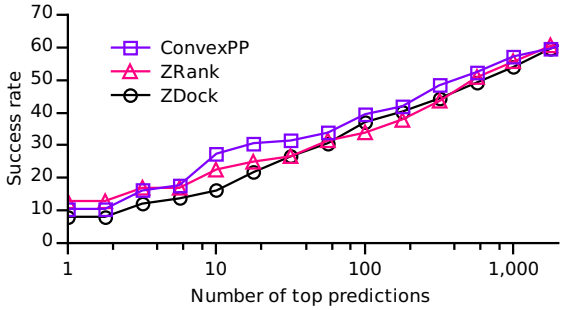


Figure 3.9: Dependence of the success rate on ZDOCK benchmark on the number of top predictions in consideration for the three methods.

3.3.4 Rosetta Benchmark

Baker, Gray *et al* generated the Rosetta benchmark using 54 complexes of the ZDOCK Benchmark 0.0 [24] using a flexible docking protocol, which is a part of the RosettaDock suite [49]. The first step in the protocol is the random translation and rotation of one of the proteins constituting the complex. Afterwards, the side chain is optimized simultaneously with the rigid body displacement. Finally, the full-atom minimization is done to refine the conformation. For each complex, Baker and Gray generated 1000 decoys following the described protocol. The resulting decoys with the corresponding scores assigned by the RosettaDock program can be obtained from [1]. We calculated the success rate of RosettaDock using the same quality criteria as in Critical Assessment of PRediction of Interactions [85, 86, 62]. The Rosetta benchmark contains 5 complexes homologous to the ones present in the training set. Therefore we trained our scoring function using training sets with and without these homologs (Table S2). Table 3.2 compares the results of RosettaDock[49], ITScore-PP[56] and our ConvexPP scoring functions.

Complex	RosettaDock					ITScore-PP					ConvexPP				
	Quality	Rank	L_{rmsd}	I_{rmsd}	F_{nat}	Quality	Rank	L_{rmsd}	I_{rmsd}	F_{nat}	Quality	Rank	L_{rmsd}	I_{rmsd}	F_{nat}
1ACB	**3**	**3**	**9.08**	**3.41**	**0.13**	**3**	**4**	**9.04**	**3.39**	**0.11**	**2**	**1**	**4.31**	**1.13**	**0.78**
1A0O	2	1	10.29	4.18	0.57	2	1	10.29	4.18	0.57	3	1	11.33	5.79	0.35
1AHW	2	1	6.41	2.37	0.51	3	4	6.46	3.00	0.49	2	1	4.82	2.31	0.44
1ATN	3	1	5.83	2.79	0.49	3	1	9.49	4.70	0.38	1	1	2.34	0.71	0.77
1AVW	**2**	**1**	**6.02**	**2.09**	**0.67**	**2**	**1**	**5.07**	**1.81**	**0.71**	**2**	**1**	**6.01**	**1.38**	**0.75**
1AVZ	3	37	8.08	4.05	0.29	3	22	11.47	5.55	0.35	3	20	8.41	4.41	0.12
1BQL	1	1	1.57	0.86	0.64	1	5	1.81	0.72	0.65	1	1	1.40	0.96	0.89
1BRC	2	4	3.77	1.21	0.75	2	1	3.77	1.21	0.75	2	1	7.24	1.62	0.76
1BRS	2	1	4.78	1.73	0.64	3	1	8.68	4.46	0.33	3	1	8.70	3.79	0.31
1BTH	3	4	18.21	5.54	0.30	3	2	5.60	2.35	0.42	3	1	5.59	2.67	0.44
1BVK	3	1	7.91	3.93	0.20	3	1	7.18	3.54	0.20	3	1	7.75	3.45	0.22
1CGI	**2**	**2**	**3.79**	**1.86**	**0.50**	**3**	**8**	**6.01**	**2.37**	**0.42**	**3**	**3**	**6.44**	**2.11**	**0.38**
1CHO	3	1	6.31	2.19	0.46	3	1	10.32	3.76	0.18	2	1	6.35	1.81	0.66
1CSE	2	6	10.10	3.12	0.56	2	1	8.81	2.66	0.71	3	1	7.87	2.29	0.40
1DFJ	2	1	5.69	2.66	0.59	2	1	5.69	2.66	0.59	2	1	5.63	2.55	0.67
1DQJ	3	1	6.71	3.35	0.31	3	5	5.00	2.12	0.34	3	1	14.01	6.06	0.38
1EFU	3	44	5.98	3.78	0.16	3	26	7.83	4.21	0.10	3	20	5.90	4.65	0.11
1EO8	3	1	10.73	5.54	0.31	3	31	6.12	3.36	0.15	3	1	10.90	3.29	0.42
1FBI	2	1	2.79	1.37	0.54	2	1	3.64	1.86	0.51	3	1	11.03	4.23	0.36
1FIN	3	364	9.88	5.38	0.12	3	109	8.26	4.19	0.12	3	200	8.27	6.06	0.10
1FQ1	3	1	11.37	5.43	0.31	3	1	9.33	5.09	0.41	3	1	6.79	4.02	0.43
1FSS	1	1	3.07	0.97	0.74	2	1	4.46	1.42	0.46	2	1	3.89	1.54	0.63
1GLA	2	4	5.96	1.85	0.65	3	7	11.96	3.83	0.35	2	1	6.20	1.70	0.84
1GOT	3	12	7.86	3.95	0.19	3	4	9.71	4.26	0.19	3	2	12.91	3.21	0.17
1IAI	3	8	6.60	3.42	0.22	2	1	4.18	1.61	0.62	2	1	3.82	1.62	0.33
1IGC	1	1	2.15	0.63	0.85	1	1	2.15	0.63	0.85	1	1	1.92	0.56	1.00
1JHL	3	3	8.56	4.51	0.26	2	2	6.09	2.66	0.56	3	2	7.65	3.94	0.31
1MAH	2	1	3.72	1.19	0.70	3	1	8.66	2.77	0.26	2	1	3.71	1.15	0.78
1MDA	3	1	8.77	3.59	0.19	3	1	9.84	3.92	0.26	2	2	6.65	2.38	0.52
1MEL	2	1	8.47	2.62	0.50	2	4	8.47	2.62	0.50	2	1	7.89	2.84	0.52
1MLC	2	7	4.91	1.37	0.52	3	1	15.36	3.45	0.20	3	1	15.41	3.04	0.24
1NCA	2	1	3.06	1.53	0.61	1	1	1.24	0.64	0.75	2	1	7.62	2.13	0.66
1NMB	1	1	0.90	0.44	0.80	1	6	2.66	0.76	0.85	1	1	2.66	0.57	1.00
1PPE	**1**	**1**	**1.38**	**0.52**	**0.73**	**3**	**1**	**7.42**	**2.39**	**0.28**	**1**	**1**	**1.38**	**0.54**	**0.89**
1QFU	2	1	3.02	1.10	0.64	1	4	2.93	1.00	0.69	1	1	3.89	0.98	0.67
1SPB	1	1	1.06	0.62	0.68	1	1	1.47	0.70	0.69	1	1	1.04	0.54	0.82
1STF	1	1	1.91	0.68	0.89	1	2	1.57	0.54	0.91	1	1	1.57	0.51	0.94
1TAB	2	1	4.16	1.30	0.74	2	1	4.39	1.37	0.76	2	1	4.11	1.47	0.68
1TGS	2	1	2.48	1.44	0.59	2	1	2.26	1.38	0.64	3	1	8.41	3.55	0.44
1UDI	2	1	3.35	1.43	0.63	2	1	2.08	1.01	0.74	2	1	2.08	1.09	0.71
1UGH	1	1	1.78	0.86	0.67	2	1	4.64	1.90	0.46	2	1	4.61	1.96	0.60
1WEJ	3	15	9.28	2.92	0.44	2	2	6.90	2.55	0.62	3	1	9.37	4.79	0.23
1WQ1	3	1	5.53	2.46	0.34	2	1	3.38	1.92	0.39	2	2	3.83	1.59	0.58
2BTF	3	1	10.03	3.23	0.22	1	1	1.52	0.60	0.75	1	1	1.31	0.60	0.90

2JEL	2	1	4.82	2.22	0.52	2	1	6.55	1.98	0.65	2	1	6.42	1.26	0.86
2KAI	1	2	2.02	0.97	0.67	2	2	2.48	1.05	0.70	1	8	2.26	0.89	0.88
2PCC	3	2	9.19	3.39	0.24	3	1	9.38	3.84	0.29	3	1	9.40	4.31	0.48
2PTC	1	4	0.82	0.44	0.80	2	4	5.86	1.82	0.70	2	1	5.30	1.16	0.74
2SIC	2	1	5.18	1.53	0.85	2	1	5.18	1.53	0.85	1	1	4.84	0.96	0.82
2SNI	**2**	**1**	**6.70**	**2.01**	**0.60**	**1**	**1**	**2.88**	**0.99**	**0.73**	**2**	**1**	**7.38**	**2.00**	**0.61**
2TEC	1	1	2.46	0.81	0.74	1	1	2.59	0.85	0.78	1	1	1.97	0.59	0.80
2VIR	3	1	7.53	4.19	0.26	2	2	5.74	2.17	0.70	2	1	5.78	1.03	0.67
3HHR	3	50	9.84	3.95	0.26	3	3	8.17	4.03	0.30	3	1	9.41	3.47	0.33
4HTC	2	1	3.81	1.54	0.61	3	1	5.95	2.25	0.42	2	1	4.04	1.50	0.76
	Top1: 66.7% (36/54)					Top1: 59.3% (32/54)					Top1: 83.3% (45/54)				
Homologs	Top1 and quality 1: 16.7% (9/54)					Top1 and quality 1: 11.1% (6/54)					Top1 and quality 1: 20.4% (11/54)				
included	Top1 and quality 1 or 2: 48.1% (26/54)					Top1 and quality 1 or 2: 38.9% (21/54)					Top1 and quality 1 or 2: 57.4% (31/54)				
	Top10 and quality 1 or 2: 61.1% (33/54)					Top10 and quality 1 or 2: 57.4% (31/54)					Top10 and quality 1 or 2: 63.0% (34/54)				
											Top1: 81.5% (44/54)				
Homologs											Top1 and quality 1: 16.7% (9/54)				
excluded											Top1 and quality 1 or 2: 55.6% (30/54)				
											Top10 and quality 1 or 2: 61.1% (33/54)				

Table 3.2: Rosetta unbound benchmark results. Proteins homologous to the ones in the training set are shown with the bold font.

Table 3.2 shows that our potentials significantly improve Top1 prediction rate over ITScore-PP and RosettaDock scoring functions while also outperforming them according to the other criteria (Top1 and quality 1 *etc.*). We computed the percentage of the structures for which the first acceptable prediction was ranked within the top predictions for each complex and plotted it on Fig. 3.10. According to the plot our scoring function outputs the plausible structure (quality $\geq$3) for more complexes than ITScore-PP and RosettaDock. Unlike the results on the ZDock benchmark, the results on the Rosetta unbound benchmark slightly decrease when we remove homologous complexes from the training set. Among the prediction quality criteria it is the number of predicted high quality structures that changed the most. On the other hand Top1 prediction rate stayed almost the same. This observation signalizes that the number of predicted high-quality structures is amenable to overfitting. Therefore unlike the Top1 criterion, it can not serve as a reliable measure of a scoring function predictive power. Table C.2 shows the per-complex comparison of two scoring functions trained with and without homologs.

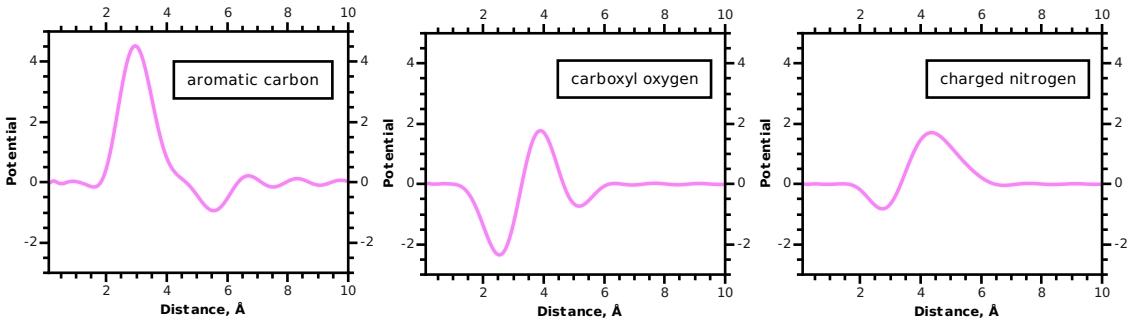


Figure 3.11: Knowledge-based solvation scoring functions $U^k(r)$ between the oxygen of a water molecule and protein atoms as a function of the separation distance. Left: water – aromatic carbon scoring function is plotted. Middle: water – carboxyl oxygen (like in aspartic and glutamic acids) scoring function is plotted. Right: water – charged nitrogen (like in lysins) scoring function is plotted.

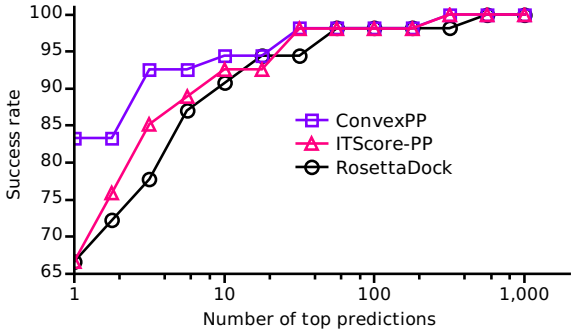


Figure 3.10: The percentage of the structures with plausible predictions (quality higher than 3) that are ranked within a certain number of top predictions. The data for ITScore-PP was taken from original publication [56]. Scoring data for RosettaDock was taken from [1].

3.3.5 Water interaction potentials prediction for CAPRI T47 target

We used the scaled Legendre polynomials as the basis to obtain the potentials for water oxygen interactions. Obtained solvation scoring functions for three atom types (aromatic carbon, carboxyl oxygen, and charged nitrogen) are shown in Figure 3.11. There, one can clearly see the difference between hydrophobic and hydrophilic interactions. Another property of our solvation scoring functions is a same peculiarity at short distances as in the other potentials. Precisely, as there is no training data for distances from 0 to about 2 Å, all obtained scoring functions are close to zero in this range. Therefore, if one would like to use these function for minimization, one might need to adjust them at short distances with some additional information.

The Critical Assessment of Predicted Interaction (CAPRI) [85, 86, 62] is the community-wide competition in prediction of protein-protein complexes. To validate our poteintials for water prediction we took part in predicting the structure of the

target 47 [79]. The participants were asked to predict the interface of the two proteins: DNase domain of colicin E2 and IM2 immunity protein. After docking prediction were performed groups, taking part in this event were invited to predict the positions of the water molecules near the interaction interface of the two proteins. Figure 3.12 shows the crystall structure of this complex that was published after the 23rd round of CAPRI ended.

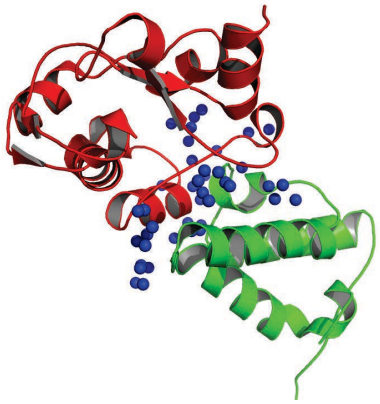


Figure 3.12: The crystallographic X-Ray structure of DNase domain of colicin E2 and IM2 immunity protein (PDB code 3U43). IM2 imunity protein is show in green and DNAse domain in red. Blue spheres denote the positions of interfacial water oxygens. The interfacial water molecule is the one within 3.5Å of both receptor and ligand.

Each contestant was required to submit 10 models that incuded predicted complex and water molecules. We obtained each model as follows. First, we constructed a protein-protein complex by homology using Modeller [37]. Then, we refined the protein-protein interface using our protein-protein docking potentials. Finally, we immersed the complex into the water box and minimized the value of the score, obtained by the summation over all interactions between water pseudo-atoms and other atom types, with respect to the positions of water oxygens.

The criteria of model quality was based on definition of water-mediated contacts between ligand and receptor. A residue of receptor and a residue of ligand were assumed to have water-mediated contact when one of their heavy atoms were closer than 3.5Å to the same water oxygen. The quantity $f^{wmc}(nat)$ was computed as the fraction of native water-mediated contacts, predicted by the model. Additionally the organizers of the competition computed number of native and model interface water molecules and number of clashes in each model. The two water molecules assumed to clash if the distance between their oxygen atoms is less than 2.5Å. All models were then checked for the number of clashes and those where the number of clashing molecules exceeded the number of native interface water molecules were rejected. Other models were ranked according to the criteria in the Table 3.3.

0	Bad	$0 \leq$	$f^{wmc}(nat)$	< 0.1
+	Fair	$0.1 \leq$	$f^{wmc}(nat)$	< 0.3
2+	Good	$0.3 \leq$	$f^{wmc}(nat)$	< 0.5
3+	Excelent	$0.5 \leq$	$f^{wmc}(nat)$	< 0.8
4+	Bad	$0.8 \leq$	$f^{wmc}(nat)$	≤ 1.0

Table 3.3: Classification of model quality according to the fraction of predicted water-mediated contacts.

Among the models that we submitted 9 were of good and one of fair quality according to the criterium described above. Overall our approach was ranked 4th among others competing techniques. The group ranked first (Nakamura *et all*) and the group ranked third (Zou *et all*) were using the water molecules that were derived from the interfaces of the homologs as the initial positions of their predictions. Afterwards they optimized the positions of water molecules using the AMBER forcefield. The second-ranked group led by Zacharias used *ab-initio* water prediction technique. They combined energy-like scoring functions with the well-established forcefield (AMBER), energy minimization and short molecular dynamics runs to generate the final predictions. It is worth to note that our method generated almost no false-positive predictions compared to other competing approaches ([79], SI).

3.3.6 Discussion

3.3.6.1 Short Distances

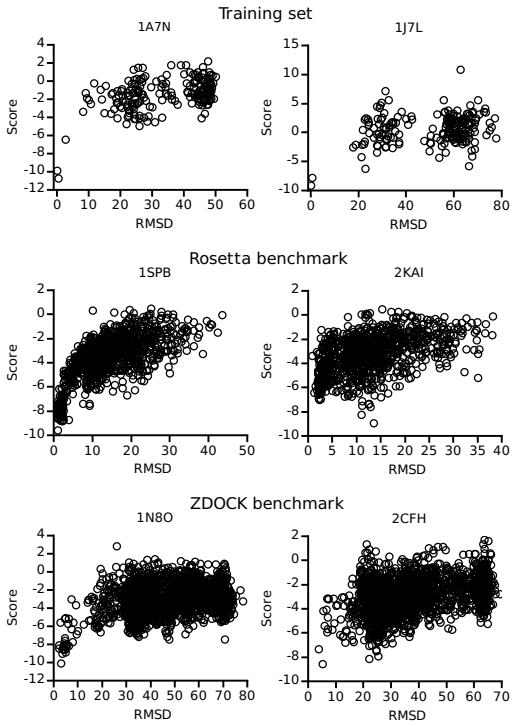


Figure 3.13: The plots of the ConvexPP versus the lRMSD for the decoy structures from the training set (1A7N, 1J7L), Rosetta benchmark (1SPB, 2KAI) and ZDOCK benchmark(1N8O, 2CHF). On the left we show the plots that exhibit funnel-like behaviour near the frame origin. On the right side the plots without obvious funnels are shown.

The key property of a scoring function is the existence of the correlation between the score of a structure and its similarity to the corresponding native structure. Conventionally, the lRMSD is taken as the measure of similarity of the decoys to the native structure. lRMSD is the ligand (the smaller protein in a complex) root mean square deviation of C_α atoms of a decoy relative to the native complex structure when receptors (the larger proteins in the complexes) are superposed. To verify that our potentials indeed correlate with the similarity to the native structures, we plotted the ConvexPP score of each decoy versus the lRMSD for all decoys from the ZDOCK and Rosetta benchmarks. Figure 3.13 shows some typical plots for the complexes from the training set and the two benchmarks. Typically, in the training set we see a wide separation between native and non-native structures. This happens because decoys in the training set have only a few *near-native* structures with lRMSD<10Å. On the contrary, about 28% of the Rosetta decoys are the near-native structures. The ZDOCK benchmark has few near-native decoys compared to Rosetta, only 1.5% of the decoys have the near-native conformations.

For the decoys from the ZDOCK and Rosetta benchmarks, we computed the Pearson correlation between lRMSD and the score using near-native decoys with lRMSD<10Å. In this calculation, we considered only the structures that have at

least 50 such decoys. All 54 complexes from the Rosetta benchmark fulfilled this criterion whereas only 24 complexes from the ZDock benchmark had more than 50 near-native decoys. The average Pearson correlation for the Rosetta benchmark is 0.31 whereas for the ZDock benchmark it is approximately 0.21. We investigated the reason of this discrepancy by looking at the atom-pairs distance distributions of the decoys from the training set, the ZDock and Rosetta benchmarks. For our analysis, we used decoys of the three common structures from these sets, PDB codes 1PPE, 1CGI, 1ACB. For these decoys, we computed the average total number of atom pairs at a certain distance using Eq. 3.4. Then, we normalized these distributions on a reference number of pairs αr^3. We tuned coefficient α in such a way to make each plot approach the value 1.0 when the distance approaches 12.0Å. Figure 3.14 plots normalized atom-pairs distance distributions for the three above-mentioned complexes. From this figure we see that the distance distributions for the Rosetta benchmark are much closer to the native distributions compared to the ZDOCK and training set distributions. We can also see that the Hex docking program [111], which we used for the generation of the training set, produces fewer short-distance atom contacts compared to ZDOCK. Since Rosetta decoys were additionally minimized using the Rosetta scoring function, they do not have short-distance atom contacts and generally their distance distributions resemble the native statistics. Native structures neither have statistics at short distances. Therefore, reconstructed potentials in the vicinity of zero are not reliable and can not provide fair scores for e.g. decoys generated with ZDOCK, since these decoys have many short-distance contacts. Ideally, one needs to additionally penalize short-distance contacts using, e.g., empirical scores that cannot be obtained with statistics from the native structures. However, in this study we do not attempt to provide such additional penalization and only focus on the potentials directly obtained from the training set of the protein structures.

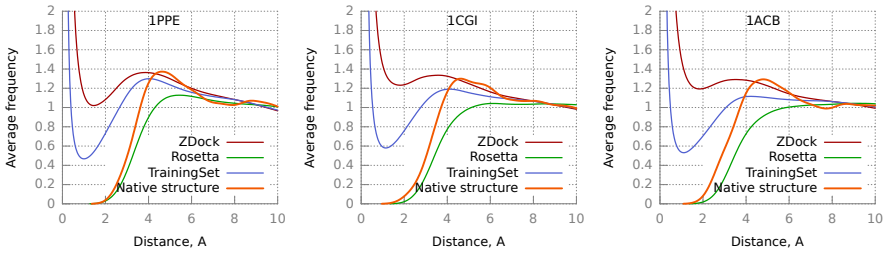


Figure 3.14: The normalized atom-pairs distance distributions for three complexes 1PPE, 1CGI, and 1ACB. For each complex, four plots are shown: average ZDOCK distribution, average Rosetta distribution, average training set distribution and the native distribution. The average is taken over all decoys from the two benchmarks and the training set.

3.3.6.2 Filtering

Some knowledge-based potentials are smoothed with a smoothing filter *a posteriori*. For example, Mitchel et al. [88] and Huang et al. [56] used a "1:2:4:2:1" filter, DOPE potential is smoothed using cubic polynomials [123], etc. On the contrary, our method introduces an assumption about interaction pair distance uncertainty a

priory. More specifically, we collect statistics using gaussian events of a variance σ (Eq. 3.4). We determine the value of the variance from the afore-mentioned cross-validation procedure. Then, according to Eqs. 3.5–3.6, ConvexPP scoring function is smooth by construction. In other words, we do not need to apply a smoothing filter to the obtained potentials, since we introduce the uncertainty when we collect statistics.

Another parameter that indirectly influences the smoothness of the resulting potential is the regularization parameter C (3.23). According to Eq. 3.19, the scoring vector $\mathbf{w}$, from which the scoring potentials are derived, is a weighted sum of the support structure vectors $\mathbf{x}_{ij}$. The more support structure vectors are in the sum, the more regular the scoring vector $\mathbf{w}$ will be. On the other hand, this number equals to the number of non-zero Lagrange multipliers λ_{ij} (Eq. 3.19) , which is uniquely defined by the value of the regularization parameter C [105]. Decreasing C results in the increase of the number of non-zero λ_{ij} therefore resulting in smoother scoring potentials. We also determine the value of this parameter by the cross-validation procedure.

The consistent determination of the two parameters σ and C allows us to obtain smooth potentials (Eq. 3.6) directly as the solution of the optimization problem (Eq. 3.17).

3.3.6.3 Uniqueness of the solution and the reference state

The concept of the statistical knowledge-based potentials is based on the definition of two states: the observed state and the reference state [131, 89, 126]. The observed state is usually the state when a single protein or a complex has the native conformation. It can be derived from the crystal structures. Reference state was introduced as an atom pair distance distribution when the interactions between the atom pairs are absent. The knowledge-based potential is then expressed in terms of these two states as:

$$u_{ij}(r) = -RT \ln \left(\frac{N_{ij}^{obs}(r)/N_{ij}^{obs}}{N_{ij}^{ref}(r)/N_{ij}^{ref}} \right),$$

where $N_{ij}^{ref}(r)$ and $N_{ij}^{obs}(r)$ are the numbers of atomic pairs i, j at a distance r in the reference and observed states, correspondingly, and numbers N_{ij}^{ref} and N_{ij}^{obs} are the total numbers of pairs i, j in these states. Some widely used approaches to derive the reference state for protein folding are the ideal-gas approximation [156], the shuffling of atoms [116], a random-walk chain [154], etc. For protein docking Chuang *et al.* used decoys as the reference state [26], Bernard and Samudrala took the average over the atomic pairs and a cumulative distribution function for all pairs as two reference states [15], etc. The very wide variety of approaches to derive the reference state has its roots in the loose definition and the complexity of the problem.

Recently, the new algorithms that avoid the reference state calculation appeared. We should mention the iterative scheme used by Huang et al.[56] and the neural network classifier by Chae et al. [23]. These algorithms indeed avoid the definition of the reference state. However, they do not guarantee the uniqueness of their solution. On the contrary, we showed that our algorithm converges to the global minimum of the function (Eq. 3.17). Thus, we avoid dependence on the initial guess of the interaction potential.

Chapter 4

Conclusion and outlook

The aim of the work was the development and validation of new algorithms that could lead to certain advances in the fields of exhaustive rigid-body search and scoring of protein-protein complexes conformations. During this work the Hermite fitting algorithm was proposed that is not only competable with the state-of-art approaches to this problem, but adds a new class of algorithms, that operate in Hermite functions space, to the existing ones, that function if spherical harmonics and grid representations of a function. A new algorithm to obtain a scoring functions was proposed that is derived from basic logical considerations about the nature of the training dataset. It avoids common unsolved problems with the reference state and has a valuable property of global convergence. It was applied with success to the problems of protein-protein conformations scoring as well as to the prediction of positions of crystallographic water molecules at the protein-protein interaction interface. It was validated using well established benchmarks and community-wide critical prediction of protein interaction assessement challenge.

4.0.7 Future developments

With the advent of the post-genomic era, the cost of whole-genome sequencing plummets and the number of sequenced organisms grows rapidly. However, the proteomics field still did not step into the phase of the exponential growth. Therefore I believe, that probing protein-protein interactions on the scale of proteome will revolutionize the fields of interactomics, evolution and systems biology. The methods that currently applied to discover protein interaction network are either large-scale, but give big number of false- positives and negatives or fit for the discovering detailed picture of single protein-protein complex. Bridging the gap between these two classes of methods would be a tremendous leap forward in the field of protein-protein interaction prediction.

Therefore it is worth trying to optimize current rigid-body search algorithms to scale the computations up. The parallel computing paradigm and especially general GPU programming could lead to a considerable HermiteFit algorithm runtime reduction.

In the area of scoring functions I believe that the paradigm of predefined atom types should be overcomed. The newly emerging area of dimentionality reduction and deep learning surely will bring new advances to the scoring field.

Another direction of expanding this work is to integrate the developed algorithms into one user-friendly package for the convenient use by those who do not posess

programming skills required to implement them.

Appendix A

Dual optimization problem

Optimization problem (Eq. 3.17) can be solved by the classical method of *Lagrange multipliers* [17, 29]. If we introduce $N \times D$ nonnegative Lagrange multipliers λ_{ij} associated with the first set of inequality constraints from Eq. 3.17 and $N \times D$ nonnegative Lagrange multipliers ν_{ij} associated with the second set of inequality constraints from (3.17), the solution of problem in Eq. 3.17 is equivalent to determining the *saddle point* of the following *Lagrangian* function:

$$\mathcal{L} = \frac{\mathbf{w} \cdot \mathbf{w}}{2} + \sum_{ij} C_{ij}\xi_{ij} - \sum_{ij} \lambda_{ij} \left(y_{ij} \left[\mathbf{w} \cdot \mathbf{x}_{ij} - b_j \right] - 1 + \xi_{ij} \right) - \sum_{ij} \nu_{ij}\xi_{ij} \tag{A.1}$$

with $\mathcal{L} = \mathcal{L}(\mathbf{w}, \mathbf{b}, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\nu})$, where $\mathbf{b} = (b_1, b_2, \ldots, b_D)$, $\boldsymbol{\xi} = (\xi_{11}, \xi_{12}, \ldots, \xi_{ND})$, $\boldsymbol{\lambda} = (\lambda_{11}, \lambda_{12}, \ldots, \lambda_{ND})$, and $\boldsymbol{\nu} = (\nu_{11}, \nu_{12}, \ldots, \nu_{ND})$. At the saddle point, $\mathcal{L}$ has a minimum with respect to $\mathbf{w}$, $\mathbf{b}$ and $\boldsymbol{\xi}$ and a maximum with respect to $\boldsymbol{\lambda}$ and $\boldsymbol{\nu}$. According to the classical *Karush-Kuhn-Tucker* (KKT) conditions [77, 17], which is a generalization of the method of Lagrange multipliers to inequality constraints, the saddle point of the Lagrangian function (Eq. A.1) satisfies four following conditions:

1. Stationarity conditions:

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{ij} y_{ij}\lambda_{ij}\mathbf{x}_{ij} = 0 \tag{A.2}$$

$$\frac{\partial L}{\partial b_j} = \sum_{i} y_{ij}\lambda_{ij} = 0 \tag{A.3}$$

$$\frac{\partial L}{\partial \xi_{ij}} = C_{ij} - \lambda_{ij} - \nu_{ij} = 0 \tag{A.4}$$

2. Complementary slackness conditions:

$$\lambda_{ij} \left(y_{ij} \left[\mathbf{w} \cdot \mathbf{x}_{ij} - b_j \right] - 1 + \xi_{ij} \right) = 0 \tag{A.5}$$

$$\nu_{ij}\xi_{ij} = 0 \tag{A.6}$$

3. Primal feasibility conditions:

$$y_{ij} \left[\mathbf{w} \cdot \mathbf{x}_{ij} - b_j \right] - 1 + \xi_{ij} \geq 0 \tag{A.7}$$

$$\xi_{ij} \geq 0 \tag{A.8}$$

4. Dual feasibility conditions:

$$\lambda_{ij} \geq 0 \tag{A.9}$$
$$\nu_{ij} \geq 0 \tag{A.10}$$

Using equation A.1 along with the aforementioned KKT conditions (Eq. A.2 - Eq. A.10), we can rewrite the original optimization problem (Eq. 3.17) as:

$$\begin{aligned} &\text{Maximize } \mathcal{L}(\lambda_{ij}): \quad \mathcal{L}(\lambda_{ij}) = \textstyle\sum_{ij} \lambda_{ij} - \frac{1}{2} \sum_{ij} \sum_{kl} y_{ij} y_{kl} \lambda_{ij} \lambda_{kl} \mathbf{x}_{ij} \cdot \mathbf{x}_{kl} \\ &\text{Subject to:} \qquad\qquad 0 \leq \lambda_{ij} \leq C_{ij} \\ &\qquad\qquad\qquad\qquad\quad \textstyle\sum_i y_{ij} \lambda_{ij} = 0 \end{aligned}$$

Appendix B

Sequential minimal optimization algorithm

Here we describe how the SMO algorithm [104] solves the problem in Eq. 3.18 for two Lagrange multipliers, λ_1 and λ_2. All quantities that refer to the first multiplier have a subscript 1 and all quantities that refer to the second multiplier have a subscript 2. SMO first computes the constraints on these multipliers and then solves the problem (Eq. 3.18) for the constrained maximum. The inequality constraints in Eq. 3.18 force the two multipliers to lie within a box $[0,\ C_1] \times [0,\ C_2]$, while the equality constraints force the two multipliers to lie on a diagonal line segment:

$$y_1\lambda_1 + y_2\lambda_2 = \gamma \tag{B.1}$$

This equation explains why we need to optimize two Lagrange multipliers simultaneously. Precisely, it is not possible to optimize a single multiplier without breaking the equality constraints in Eq. 3.18 and, subsequently, breaking the constraints (Eq. B.1).

Without loss of generality, SMO first computes the second Lagrange multiplier λ_2 and then expresses the ends of the diagonal line segment in terms of λ_2. The following lower and upper bounds, L_2 and H_2, apply to λ_2:

1. if $y_1 = y_2$:

$$\begin{aligned} L_2 &= \max(0,\ \gamma y_2 - C_1) \\ H_2 &= \min(C_2,\ \gamma y_2) \end{aligned}$$

2. if $y_1 \neq y_2$:

$$\begin{aligned} L_2 &= \max(0,\ \gamma y_2) \\ H_2 &= \min(C_2,\ \gamma y_2 + C_1) \end{aligned}$$

On the next step SMO computes the location of the unconstrained maximum of the Lagrangian with respect to λ_2:

$$\frac{\partial \mathcal{L}(\lambda_1, \lambda_2)}{\partial \lambda_2} = 0 \tag{B.2}$$

The corresponding unconstrained λ_2 will be:

$$\lambda_2^{\text{new}} = \lambda_2^{\text{old}} + y_2 \frac{(\mathbf{x}_2 - \mathbf{x}_1) \cdot \mathbf{w}^{\text{old}} + y_1 - y_2}{\nu} \tag{B.3}$$

Next, SMO computes the constrained maximum by clipping the unconstrained maximum to the ends of the line segment:

$$\lambda_2^{\text{new,clipped}} = \begin{cases} L, & \text{if } \lambda_2^{\text{new}} \leq L \\ \lambda_2^{\text{new}}, & \text{if } L < \lambda_2^{\text{new}} < H \\ H, & \text{if } \lambda_2^{\text{new}} \geq H \end{cases} \tag{B.4}$$

Finally, SMO determines the value of λ_1 from the new clipped value of λ_2:

$$\lambda_1^{\text{new}} \quad = \quad \lambda_1^{\text{old}} - y_1 y_2 (\lambda_2^{\text{new,clipped}} - \lambda_2^{\text{old}}) \tag{B.5}$$

Appendix C

Docking benchmarks results

	With Homologs				Without Homologs			
Complex	Rank	L_{rmsd}	I_{rmsd}	F_{nat}	Rank	L_{rmsd}	I_{rmsd}	F_{nat}
Rigid-Body								
1AHW	547	2.14	0.91	0.79	465	2.14	0.91	0.79
1BVK	-	-	-	-	-	-	-	-
1DQJ	-	-	-	-	-	-	-	-
1E6J	35	4.21	2.26	0.48	63	3.78	1.92	0.68
1JPS	62	4.14	1.11	0.78	127	3.54	0.98	0.70
1MLC	5	4.70	1.12	0.39	2	4.70	1.12	0.39
1VFB	1239	10.89	2.48	0.30	948	10.89	2.48	0.30
1WEJ	1	4.13	1.30	0.75	1	2.20	1.16	0.75
2FD6	8	15.65	2.16	0.80	99	15.65	2.16	0.80
2I25	83	4.45	1.87	0.33	195	7.92	2.21	0.36
2VIS	617	23.89	2.37	0.43	326	23.89	2.37	0.43
1BJ1	2	2.82	0.98	0.86	8	2.82	0.98	0.86
1FSK	3	3.98	1.39	0.81	2	3.52	1.25	0.74
1I9R	462	16.85	2.30	0.48	119	16.85	2.30	0.48
1IQD	16	4.20	0.97	0.67	8	6.90	1.95	0.46
1K4C	242	5.78	1.31	0.62	1177	5.78	1.31	0.62
1KXQ	1	3.06	1.04	0.88	1	3.06	1.04	0.88
1NCA	12	4.50	1.38	0.86	29	4.50	1.38	0.86
1NSN	636	4.95	2.00	0.50	409	4.95	2.00	0.50
1QFW	1315	5.12	1.35	0.73	1274	5.12	1.35	0.73
1QFW	1315	5.12	1.35	0.73	1274	5.12	1.35	0.73
2JEL	957	6.83	2.30	0.31	485	6.83	2.30	0.31
1AVX	**3**	**4.85**	**2.23**	**0.39**	**5**	**4.85**	**2.23**	**0.39**
1AY7	**185**	**5.73**	**1.82**	**0.45**	**126**	**7.85**	**2.46**	**0.45**
1BVN	**3**	**4.09**	**1.74**	**0.50**	**2**	**4.09**	**1.74**	**0.50**
1CGI	**61**	**3.20**	**2.30**	**0.49**	**37**	**3.20**	**2.30**	**0.49**
1D6R	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
1DFJ	1	5.97	2.42	0.50	1	5.97	2.42	0.50
1E6E	9	4.01	2.41	0.42	8	3.87	1.59	0.69

1EAW	1	2.60	1.03	0.70	1	2.60	1.03	0.70
1EWY	21	3.16	1.74	0.56	49	3.16	1.74	0.56
1EZU	-	-	-	-	-	-	-	-
1F34	38	3.41	1.45	0.54	30	3.41	1.45	0.54
1HIA	-	-	-	-	-	-	-	-
1MAH	1	3.64	1.26	0.69	2	4.17	1.48	0.61
1N8O	1	2.94	1.24	0.74	1	2.94	1.24	0.74
1OPH	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
1PPE	**1**	**4.62**	**1.52**	**0.71**	**1**	**3.92**	**1.43**	**0.74**
1R0R	**2**	**8.36**	**2.46**	**0.40**	**4**	**8.36**	**2.46**	**0.40**
1TMQ	**8**	**6.11**	**1.97**	**0.45**	**1**	**6.11**	**1.97**	**0.45**
1UDI	-	-	-	-	-	-	-	-
1YVB	8	7.33	1.92	0.71	9	7.33	1.92	0.71
2B42	8	9.44	2.23	0.43	1	9.44	2.23	0.43
2MTA	125	3.76	1.88	0.50	197	3.76	1.88	0.50
2O8V	-	-	-	-	-	-	-	-
2PCC	458	6.29	2.19	0.44	622	6.29	2.19	0.44
2SIC	3	5.92	1.19	0.73	2	5.92	1.19	0.73
2SNI	**8**	**8.57**	**2.45**	**0.57**	**2**	**8.57**	**2.45**	**0.57**
2UUY	233	13.72	2.38	0.66	346	13.72	2.38	0.66
7CEI	**4**	**8.41**	**2.46**	**0.65**	**1**	**6.39**	**1.94**	**0.65**
1A2K	-	-	-	-	-	-	-	-
1AK4	-	-	-	-	-	-	-	-
1AKJ	395	3.30	1.54	0.60	388	5.59	1.96	0.57
1AZS	1	11.04	1.88	0.69	5	12.49	1.74	0.65
1B6C	2	3.67	2.23	0.90	1	3.67	2.23	0.90
1BUH	1	3.42	1.81	0.63	4	3.42	1.81	0.63
1E96	**261**	**6.03**	**2.18**	**0.58**	**425**	**5.56**	**1.99**	**0.54**
1EFN	-	-	-	-	-	-	-	-
1F51	16	3.41	1.82	0.60	11	3.41	1.82	0.60
1FC2	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
1FQJ	-	-	-	-	-	-	-	-
1GCQ	118	1.98	1.19	0.71	141	1.98	1.19	0.71
1GHQ	-	-	-	-	-	-	-	-
1GLA	57	4.91	2.22	0.37	273	4.91	2.22	0.37
1GPW	1	7.10	2.44	0.39	1	7.10	2.44	0.39
1HE1	301	5.93	1.74	0.38	324	5.93	1.74	0.38
1I4D	-	-	-	-	-	-	-	-
1J2J	**86**	**5.59**	**2.12**	**0.55**	**97**	**5.59**	**2.12**	**0.55**
1K74	8	7.90	2.02	0.48	1	6.65	2.30	0.70
1KAC	287	4.47	2.21	0.36	105	4.47	2.21	0.36
1KLU	-	-	-	-	-	-	-	-
1KTZ	282	5.39	1.25	0.63	735	5.39	1.25	0.63
1KXP	**1**	**7.43**	**2.17**	**0.51**	**4**	**7.43**	**2.17**	**0.51**
1ML0	1	4.47	1.89	0.61	1	4.47	1.89	0.61
1QA9	-	-	-	-	-	-	-	-
1RLB	7	9.11	1.93	0.66	10	9.11	1.93	0.66
1S1Q	766	2.35	1.53	0.58	1187	2.35	1.53	0.58

1SBB	-	-	-	-	-	-	-	-
1T6B	89	5.91	2.03	0.64	1	5.91	2.03	0.64
1XD3	6	4.87	2.49	0.30	8	5.34	1.96	0.40
1Z0K	11	4.59	1.68	0.56	5	4.59	1.68	0.56
1Z5Y	8	6.58	1.97	0.50	8	6.58	1.97	0.50
1ZHI	**78**	**9.90**	**1.96**	**0.61**	**165**	**4.85**	**1.68**	**0.52**
2AJF	-	-	-	-	-	-	-	-
2BTF	655	6.00	2.20	0.33	736	6.00	2.20	0.33
2HLE	9	6.84	2.35	0.35	8	4.11	2.08	0.54
2HQS	576	8.94	2.30	0.37	117	8.94	2.30	0.37
2OOB	-	-	-	-	-	-	-	-

Medium

1BGX	-	-	-	-	-	-	-	-
1ACB	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
1IJK	124	6.42	1.83	0.25	319	5.02	1.35	0.38
1KKL	325	6.23	2.50	0.44	311	6.23	2.50	0.44
1M10	-	-	-	-	-	-	-	-
1NW9	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
1GP2	-	-	-	-	-	-	-	-
1GRN	**1785**	**7.10**	**2.49**	**0.00**	**1758**	**7.10**	**2.49**	**0.00**
1HE8	-	-	-	-	-	-	-	-
1I2M	-	-	-	-	-	-	-	-
1IB1	-	-	-	-	-	-	-	-
1K5D	1185	8.04	2.03	0.29	1656	7.24	2.31	0.19
1N2C	-	-	-	-	-	-	-	-
1WQ1	-	-	-	-	-	-	-	-
1XQS	199	5.60	2.28	0.38	319	5.60	2.28	0.38
2CFH	1	5.20	2.12	0.36	1	5.20	2.12	0.36
2H7V	8	13.69	2.47	0.44	8	13.69	2.47	0.44
2HRK	-	-	-	-	-	-	-	-
2NZ8	-	-	-	-	-	-	-	-

Difficult

1E4K	-	-	-	-	-	-	-	-
2HMI	-	-	-	-	-	-	-	-
1FQ1	-	-	-	-	-	-	-	-
1PXV	-	-	-	-	-	-	-	-
1ATN	-	-	-	-	-	-	-	-
1BKD	-	-	-	-	-	-	-	-
1DE4	-	-	-	-	-	-	-	-
1EER	-	-	-	-	-	-	-	-
1FAK	-	-	-	-	-	-	-	-
1H1V	-	-	-	-	-	-	-	-
1IBR	-	-	-	-	-	-	-	-
1IRA	-	-	-	-	-	-	-	-
1JMO	-	-	-	-	-	-	-	-
1R8S	-	-	-	-	-	-	-	-
1Y64	-	-	-	-	-	-	-	-
2C0L	-	-	-	-	-	-	-	-

2OT3	-	-	-	-	-	-	-	-		
Homologs	Top1: 10.5% (13/124)				Top1: 12.1% (15/124)					
included	Top10: 27.4% (34/124)				Top10: 29.0% (36/124)					

Table C.1: ZDock benchmark results. Proteins homologous to the ones in the training set are shown with bold font. Absense of hits among first 1000 predictions is shown with hyphens.

	Homologs included					Homologs excluded				
Complex	Quality	Rank	L_{rmsd}	I_{rmsd}	F_{nat}	Quality	Rank	L_{rmsd}	I_{rmsd}	F_{nat}
1ACB	**2**	**1**	**4.31**	**1.13**	**0.78**	**2**	**1**	**4.31**	**1.13**	**0.78**
1A0O	3	1	11.33	5.79	0.35	3	1	6.35	3.44	0.47
1AHW	2	1	4.82	2.31	0.44	2	1	4.82	2.31	0.44
1ATN	1	1	2.34	0.71	0.77	1	1	2.34	0.71	0.77
1AVW	**2**	**1**	**6.01**	**1.38**	**0.75**	**2**	**1**	**6.01**	**1.38**	**0.75**
1AVZ	3	20	8.41	4.41	0.12	3	16	8.41	4.41	0.12
1BQL	1	1	1.40	0.96	0.89	2	1	3.78	2.35	0.37
1BRC	2	1	7.24	1.62	0.76	2	1	7.24	1.62	0.76
1BRS	3	1	8.70	3.79	0.31	3	1	8.70	3.79	0.31
1BTH	3	1	5.59	2.67	0.44	3	1	5.59	2.67	0.44
1BVK	3	1	7.75	3.45	0.22	3	1	8.72	3.78	0.22
1CGI	**3**	**3**	**6.44**	**2.11**	**0.38**	**3**	**5**	**6.46**	**2.05**	**0.47**
1CHO	2	1	6.35	1.81	0.66	2	1	6.35	1.81	0.66
1CSE	3	1	7.87	2.29	0.40	3	1	7.87	2.29	0.40
1DFJ	2	1	5.63	2.55	0.67	2	1	5.63	2.55	0.67
1DQJ	3	1	14.01	6.06	0.38	3	1	14.01	6.06	0.38
1EFU	3	20	5.90	4.65	0.11	3	14	5.90	4.65	0.11
1EO8	3	1	10.90	3.29	0.42	3	1	10.90	3.29	0.42
1FBI	3	1	11.03	4.23	0.36	3	1	11.03	4.23	0.36
1FIN	3	200	8.27	6.06	0.10	3	240	8.27	6.06	0.10
1FQ1	3	1	6.79	4.02	0.43	3	1	6.79	4.02	0.43
1FSS	2	1	3.89	1.54	0.63	2	1	3.89	1.54	0.63
1GLA	2	1	6.20	1.70	0.84	2	1	6.20	1.70	0.84
1GOT	3	2	12.91	3.21	0.17	3	2	10.92	3.22	0.13
1IAI	2	1	3.82	1.62	0.33	2	1	3.82	1.62	0.33
1IGC	1	1	1.92	0.56	1.00	1	2	1.92	0.56	1.00
1JHL	3	2	7.65	3.94	0.31	3	3	8.55	3.90	0.38
1MAH	2	1	3.71	1.15	0.78	2	1	3.55	1.16	0.75
1MDA	2	2	6.65	2.38	0.52	2	2	6.65	2.38	0.52
1MEL	2	1	7.89	2.84	0.52	2	1	7.89	2.84	0.52
1MLC	3	1	15.41	3.04	0.24	3	1	15.41	3.04	0.24
1NCA	2	1	7.62	2.13	0.66	2	1	7.62	2.13	0.66
1NMB	1	1	2.66	0.57	1.00	1	1	2.66	0.57	1.00
1PPE	**1**	**1**	**1.38**	**0.54**	**0.89**	**1**	**1**	**1.38**	**0.54**	**0.89**
1QFU	1	1	3.89	0.98	0.67	1	1	3.89	0.98	0.67
1SPB	1	1	1.04	0.54	0.82	1	1	1.04	0.54	0.82

1STF	1	1	1.57	0.51	0.94	1	1	1.57	0.51	0.94
1TAB	2	1	4.11	1.47	0.68	2	1	4.11	1.47	0.68
1TGS	3	1	8.41	3.55	0.44	3	1	8.41	3.55	0.44
1UDI	2	1	2.08	1.09	0.71	2	1	2.08	1.09	0.71
1UGH	2	1	4.61	1.96	0.60	2	1	4.61	1.96	0.60
1WEJ	3	1	9.37	4.79	0.23	3	1	9.37	4.79	0.23
1WQ1	2	2	3.83	1.59	0.58	3	3	8.42	3.29	0.35
2BTF	1	1	1.31	0.60	0.90	1	1	1.31	0.60	0.90
2JEL	2	1	6.42	1.26	0.86	2	1	6.42	1.26	0.86
2KAI	1	8	2.26	0.89	0.88	1	7	2.26	0.89	0.88
2PCC	3	1	9.40	4.31	0.48	3	1	9.40	4.31	0.48
2PTC	2	1	5.30	1.16	0.74	2	1	5.30	1.16	0.74
2SIC	1	1	4.84	0.96	0.82	1	1	4.01	0.79	0.86
2SNI	**2**	**1**	**7.38**	**2.00**	**0.61**	**2**	**1**	**7.38**	**2.00**	**0.61**
2TEC	1	1	1.97	0.59	0.80	1	1	2.10	0.59	0.84
2VIR	2	1	5.78	1.03	0.67	2	1	5.78	1.03	0.67
3HHR	3	1	9.41	3.47	0.33	3	1	9.74	3.74	0.38
4HTC	2	1	4.04	1.50	0.76	2	1	4.04	1.50	0.76
	Top1: 83.3% (45/54)					Top1: 81.5% (44/54)				
Homologs	Top1 and quality 1: 20.4% (11/54)					Top1 and quality 1: 16.7% (9/54)				
included	Top1 and quality 1 or 2: 57.4% (31/54)					Top1 and quality 1 or 2: 55.6% (30/54)				
	Top10 and quality 1 or 2: 63.0% (34/54)					Top10 and quality 1 or 2: 61.1% (33/54)				

Table C.2: Rosetta unbound benchmark results. Proteins homologous to the ones in the training set are shown with bold font. Absense of hits among first 1000 predictions is shown with hyphens.

Bibliography

[1] http: //graylab.jhu.edu/docking/decoys.

[2] http://www.loria.fr/ ritchied/hex/.

[3] http://zlab.umassmed.edu/zlab/.

[4] PV Afonine and A Urzhumtsev. On a fast calculation of structure factors at a subatomic resolution. *Acta Crystallographica Section A: Foundations of Crystallography*, 60(1):19–32, 2004.

[5] Mazen Ahmad, Wei Gu, Tihamér Geyer, and Volkhard Helms. Adhesive water networks facilitate binding of protein interfaces. *Nature Communications*, 2:261, 2011.

[6] Rehan Akbani, Stephen Kwek, and Nathalie Japkowicz. Applying support vector machines to imbalanced datasets. In *In Proceedings of the 15th European Conference on Machine Learning (ECML)*, pages 39–50, 2004.

[7] Ingemar André, Philip Bradley, Chu Wang, and David Baker. Prediction of the structure of symmetrical protein assemblies. *Proceedings of the National Academy of Sciences*, 104(45):17656–17661, 2007.

[8] Sylvain Arlot and Alain Celisse. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4:40–79, 2010.

[9] Mohammed A Ayoub, Angelique Levoye, Philippe Delagrange, and Ralf Jockers. Preferential formation of MT1/MT2 melatonin receptor heterodimers with distinct ligand interaction properties compared with MT2 homodimers. *Molecular Pharmacology*, 66(2):312–321, 2004.

[10] Daniel Barker and Mark Pagel. Predicting functional gene links from phylogenetic-statistical analyses of whole genomes. *PLoS Computational Biology*, 1(1):e3, 2005.

[11] A. Ben-Naim. Statistical potentials extracted from protein structures: Are these meaningful potentials? *The Journal of Chemical Physics*, 107(9):3698–3706, 1997.

[12] A Ben-Naim. Molecular recognition—viewed through the eyes of the solvent. *Biophysical Chemistry*, 101:309–319, 2002.

[13] Efrat Ben-Zeev and Miriam Eisenstein. Weighted geometric docking: Incorporating external information in the rotation-translation scan. *Proteins: Structure, Function, and Bioinformatics*, 52(1):24–27, 2003.

[14] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, TN Bhat, H. Weissig, I.N. Shindyalov, and P.E. Bourne. The protein data bank. *Nucleic Acids Research*, 28(1):235–242, 2000.

[15] Brady Bernard and Ram Samudrala. A generalized knowledge-based discriminatory function for biomolecular interactions. *Proteins: Structure, Function, and Bioinformatics*, 76(1):115–128, 2009.

[16] J. Bernauer, J. Azé, J. Janin, and A. Poupon. A new protein–protein docking scoring function based on interface residue properties. *Bioinformatics*, 23(5):555–562, 2007.

[17] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[18] Stephen P Boyd, Craig H Barratt, Stephen P Boyd, and Stephen P Boyd. *Linear controller design: limits of performance*. Prentice Hall Englewood Cliffs, NJ, 1991.

[19] Ashley M Buckle, Gideon Schreiber, and Alan R Fersht. Protein-protein recognition: Crystal structural analysis of a barnase-barstar complex at 2.0-. ANG. resolution. *Biochemistry*, 33(30):8878–8889, 1994.

[20] Huynh-Hoa Bui, Alexandra J Schiewe, and Ian S Haworth. WATGEN: an algorithm for modeling water networks at protein–protein interfaces. *Journal of Computational Chemistry*, 28(14):2241–2251, 2007.

[21] Cristopher J.C. Burges and David J. Crisp. Uniqueness of the SVM Solution. In *Advances in Neural Information Processing Systems 12*, pages 223–229. The MIT Press, 2000.

[22] P. Chacón and W. Wriggers. Multi-resolution contour-based fitting of macromolecular structures. *Journal of Molecular Biology*, 317(3):375–384, 2002.

[23] Myong-Ho Chae, Florian Krull, Stephan Lorenzen, and Ernst-Walter Knapp. Predicting protein complex geometries with a neural network. *Proteins: Structure, Function, and Bioinformatics*, 78(4):1026–1039, 2010.

[24] R. Chen and Z. Weng. Docking unbound proteins using shape complementarity, desolvation, and electrostatics. *Proteins: Structure, Function, and Bioinformatics*, 47(3):281–294, 2002.

[25] Rong Chen and Zhiping Weng. Docking unbound proteins using shape complementarity, desolvation, and electrostatics. *Proteins: Structure, Function, and Bioinformatics*, 47(3):281–294, 2002.

[26] Gwo-Yu Chuang, Dima Kozakov, Ryan Brenke, Stephen R Comeau, and Sandor Vajda. DARS (Decoys As the Reference State) potentials for protein-protein docking. *Biophysical Journal*, 95(9):4217–4227, 2008.

[27] Yao Cong, Matthew L Baker, Joanita Jakana, David Woolford, Erik J Miller, Stefanie Reissmann, Ramya N Kumar, Alyssa M Redding-Johanson, Tanveer S

Batth, Aindrila Mukhopadhyay, et al. 4.0-Å resolution cryo-EM structure of the mammalian chaperonin TRiC/CCT reveals its unique subunit arrangement. *Proceedings of the National Academy of Sciences*, 107(11):4967–4972, 2010.

[28] Agustín Correa, Sabino Pacheco, Ariel E Mechaly, Gonzalo Obal, Ghislaine Béhar, Barbara Mouratou, Pablo Oppezzo, Pedro M Alzari, and Frédéric Pecorari. Potent and Specific Inhibition of Glycosidases by Small Artificial Binding Proteins (Affitins). *PloS One*, 9(5):e97438, 2014.

[29] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[30] Thomas Dandekar, Berend Snel, Martijn Huynen, and Peer Bork. Conservation of gene order: a fingerprint of proteins that physically interact. *Trends in Biochemical Sciences*, 23(9):324–328, 1998.

[31] Alessandra Di Tullio, Samantha Reale, and Francesco De Angelis. Molecular recognition by mass spectrometry. *Journal of Mass Spectrometry*, 40(7):845–865, 2005.

[32] T. Dietterich. Overfitting and undercomputing in machine learning. *ACM Computing Surveys (CSUR)*, 27(3):326–327, 1995.

[33] Cyril Dominguez, Rolf Boelens, and Alexandre MJJ Bonvin. HADDOCK: a protein-protein docking approach based on biochemical or biophysical information. *Journal of the American Chemical Society*, 125(7):1731–1737, 2003.

[34] Jacques Dubochet, Marc Adrian, Jiin-Ju Chang, Jean-Claude Homo, Jean Lepault, Alasdair W McDowall, and Patrick Schultz. Cryo-electron microscopy of vitrified specimens. *Quarterly Reviews of Biophysics*, 21(02):129–228, 1988.

[35] Dina Duhovny, Ruth Nussinov, and Haim J Wolfson. Efficient unbound docking of rigid molecules. In *Algorithms in Bioinformatics*, pages 185–200. Springer, 2002.

[36] Juan Esquivel-Rodríguez, Yifeng David Yang, and Daisuke Kihara. Multi-LZerD: Multiple protein docking for asymmetric complexes. *Proteins: Structure, Function, and Bioinformatics*, 80(7):1818–1833, 2012.

[37] Narayanan Eswar, Ben Webb, Marc A Marti-Renom, MS Madhusudhan, David Eramian, Min-yi Shen, Ursula Pieper, and Andrej Sali. Comparative protein structure modeling using Modeller. *Current Protocols in Bioinformatics*, pages 5–6, 2006.

[38] A. Fernández and H.A. Scheraga. Insufficiently dehydrated hydrogen bonds as determinants of protein interactions. *Proceedings of the National Academy of Sciences*, 100(1):113–118, 2003.

[39] S Fields and O Song. A novel genetic system to detect protein-protein interactions. *Nature*, 340(6230):245, 1989.

[40] A.V. Finkelstein, A.Y. Badretdinov, and A.M. Gutin. Why do protein architectures have boltzmann-like statistics? *Proteins: Structure, Function, and Bioinformatics*, 23(2):142–150, 2004.

[41] Stefano Forli and Arthur J Olson. A force field with discrete displaceable waters and desolvation entropy for hydrated ligand docking. *Journal of Medicinal Chemistry*, 55(2):623–638, 2012.

[42] Matteo Frigo and Steven G Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005.

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

[44] Henry A Gabb, Richard M Jackson, and Michael JE Sternberg. Modelling protein docking using shape complementarity, electrostatics and biochemical information. *Journal of Molecular Biology*, 272(1):106–120, 1997.

[45] Eleanor J Gardiner, Peter Willett, and Peter J Artymiuk. Protein docking using a genetic algorithm. *Proteins: Structure, Function, and Bioinformatics*, 44(1):44–56, 2001.

[46] José Ignacio Garzón, Julio Kovacs, Ruben Abagyan, and Pablo Chacón. ADP_EM: fast exhaustive multi-resolution docking for high-throughput coverage. *Bioinformatics*, 23(4):427–433, 2007.

[47] José Ignacio Garzon, José Ramón Lopéz-Blanco, Carles Pons, Julio Kovacs, Ruben Abagyan, Juan Fernandez-Recio, and Pablo Chacon. FRODOCK: a new approach for fast rotational protein–protein docking. *Bioinformatics*, 25(19):2544–2551, 2009.

[48] Jeffrey J Gray, Stewart Moughon, Chu Wang, Ora Schueler-Furman, Brian Kuhlman, Carol A Rohl, and David Baker. Protein–protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations. *Journal of Molecular Biology*, 331(1):281–299, 2003.

[49] J.J. Gray, S. Moughon, C. Wang, O. Schueler-Furman, B. Kuhlman, C.A. Rohl, and D. Baker. Protein–protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations. *Journal of Molecular Biology*, 331(1):281–300, 2003.

[50] Nikolaus Grigorieff. FREALIGN: high-resolution refinement of single particle structures. *Journal of Structural Biology*, 157(1):117–125, 2007.

[51] T. Hamelryck, M. Borg, M. Paluszewski, J. Paulsen, J. Frellsen, C. Andreetta, W. Boomsma, S. Bottaro, and J. Ferkinghoff-Borg. Potentials of mean force for protein structure prediction vindicated, formalized and generalized. *PloS One*, 5(11):e13714, 2010.

[52] J.P. Hansen and I.R. McDonald. *Theory of simple liquids*. Academic Press, 2006.

[53] C. Hu, X. Li, and J. Liang. Developing optimal non-linear scoring function for protein design. *Bioinformatics*, 20(17):3080–3098, 2004.

[54] Shu-Hong Hu, John Gehrmann, Paul F. Alewood, David J. Craik, and Jennifer L. Martin. Crystal Structure at 1.1 A Resolution of alpha-Conotoxin PnIB: Comparison with alpha-Conotoxins PnIA and GI. *Biochemistry*, 36(38):11323–11330, 1997.

[55] Sheng-You Huang. Search strategies and evaluation in protein–protein docking: principles, advances and challenges. *Drug Discovery Today*, 2014.

[56] S.Y. Huang and X. Zou. An iterative knowledge-based scoring function for protein–protein recognition. *Proteins: Structure, Function, and Bioinformatics*, 72(2):557–579, 2008.

[57] S.Y. Huang and X. Zou. Inclusion of solvation and entropy in the knowledge-based scoring function for protein- ligand interactions. *Journal of Chemical Information and Modeling*, 50(2):262–273, 2010.

[58] David J Huggins and Bruce Tidor. Systematic placement of structural water molecules for improved scoring of protein–ligand interactions. *Protein Engineering Design and Selection*, 24(10):777–789, 2011.

[59] Howook Hwang, Brian Pierce, Julian Mintseris, Joël Janin, and Zhiping Weng. Protein–protein docking benchmark version 3.0. *Proteins: Structure, Function, and Bioinformatics*, 73(3):705–709, 2008.

[60] Yuval Inbar, Hadar Benyamini, Ruth Nussinov, and Haim J Wolfson. Prediction of multimolecular assemblies by multiple docking. *Journal of Molecular Biology*, 349(2):435–447, 2005.

[61] Takashi Ito, Tomoko Chiba, Ritsuko Ozawa, Mikio Yoshida, Masahira Hattori, and Yoshiyuki Sakaki. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proceedings of the National Academy of Sciences*, 98(8):4569–4574, 2001.

[62] J. Janin. Assessing predictions of protein–protein interaction: the CAPRI experiment. *Protein Science*, 14(2):278–283, 2009.

[63] Ronald Jansen, Dov Greenbaum, and Mark Gerstein. Relating whole-genome expression data with protein-protein interactions. *Genome Research*, 12(1):37–46, 2002.

[64] Fan Jiang and Sung-Hou Kim. "Soft docking": matching of molecular surface cubes. *Journal of Molecular Biology*, 219(1):79–102, 1991.

[65] Lin Jiang, Brian Kuhlman, Tanja Kortemme, and David Baker. A "solvated rotamer" approach to modeling water-mediated hydrogen bonds at protein–protein interfaces. *Proteins: Structure, Function, and Bioinformatics*, 58(4):893–904, 2005.

[66] Panagiotis L Kastritis, Koen M Visscher, Aalt DJ van Dijk, and Alexandre MJJ Bonvin. Solvated protein–protein docking using Kyte-Doolittle-based water preferences. *Proteins: Structure, Function, and Bioinformatics*, 81(3):510–518, 2013.

[67] E. Katchalski-Katzir, I. Shariv, M. Eisenstein, A.A. Friesem, C. Aflalo, and I.A. Vakser. Molecular surface recognition: determination of geometric fit between proteins and their ligands by correlation techniques. *Proceedings of the National Academy of Sciences*, 89(6):2195–2199, 1992.

[68] Ephraim Katchalski-Katzir, Isaac Shariv, Miriam Eisenstein, Asher A Friesem, Claude Aflalo, and Ilya A Vakser. Molecular surface recognition: determination of geometric fit between proteins and their ligands by correlation techniques. *Proceedings of the National Academy of Sciences*, 89(6):2195–2199, 1992.

[69] M. Kearns, Y. Mansour, A.Y. Ng, and D. Ron. An experimental and theoretical comparison of model selection methods. *Machine Learning*, 27(1):7–50, 1997.

[70] Young C Kim and Gerhard Hummer. Coarse-grained models for simulations of multiprotein complexes: application to ubiquitin binding. *Journal of Molecular Biology*, 375(5):1416–1433, 2008.

[71] G.J. Kleywegt, M.R. Harris, J. Zou, T.C. Taylor, A. Wahlby, and T.A. Jones. The Uppsala electron-density server. *Acta Crystallographica Section D: Biological Crystallography*, 60(12):2240–2249, 2004.

[72] Eugene V Koonin, Yuri I Wolf, and L Aravind. Prediction of the archaeal exosome and its connections with the proteasome and the translation and transcription machineries by a comparative-genomic approach. *Genome Research*, 11(2):240–252, 2001.

[73] WA Koppensteiner and M.J. Sippl. Knowledge-based potentials–back to the roots, 1998.

[74] J.A. Kovacs, P. Chacón, Y. Cong, E. Metwally, and W. Wriggers. Fast rotational matching of rigid bodies by fast Fourier transform acceleration of five degrees of freedom. *Acta Crystallographica Section D: Biological Crystallography*, 59(8):1371–1376, 2003.

[75] J.A. Kovacs and W. Wriggers. Fast rotational matching. *Acta Crystallographica Section D: Biological Crystallography*, 58(8):1282–1286, 2002.

[76] Dima Kozakov, Ryan Brenke, Stephen R Comeau, and Sandor Vajda. PIPER: An FFT-based protein docking program with pairwise potentials. *Proteins: Structure, Function, and Bioinformatics*, 65(2):392–406, 2006.

[77] H.W. Kuhn and A.W. Tucker. Nonlinear programming. In *Second Berkeley symposium on mathematical statistics and probability*, volume 1, pages 481–492, 1951.

[78] Gregory Leibon, Daniel N Rockmore, Wooram Park, Robert Taintor, and Gregory S Chirikjian. A fast Hermite transform. *Theoretical Computer Science*, 409(2):211–228, 2008.

[79] Marc F Lensink, Iain H Moal, Paul A Bates, Panagiotis L Kastritis, Adrien SJ Melquiond, Ezgi Karaca, Christophe Schmitz, Marc Dijk, Alexandre MJJ Bonvin, Miriam Eisenstein, et al. Blind prediction of interfacial water positions in CAPRI. *Proteins: Structure, Function, and Bioinformatics*, 82(4):620–632, 2014.

[80] Xiaofan Li, Iain H Moal, and Paul A Bates. Detection and refinement of encounter complexes for protein–protein docking: taking account of macromolecular crowding. *Proteins: Structure, Function, and Bioinformatics*, 78(15):3189–3196, 2010.

[81] Edward M Marcotte, Matteo Pellegrini, Ho-Leung Ng, Danny W Rice, Todd O Yeates, and David Eisenberg. Detecting protein function and protein-protein interactions from genome sequences. *Science*, 285(5428):751–753, 1999.

[82] Amir Marcovitz and Yaakov Levy. Arc-repressor dimerization on DNA: folding rate enhancement by colocalization. *Biophysical Journal*, 96(10):4212–4220, 2009.

[83] O. Martin and D. Schomburg. Efficient comprehensive scoring of docked protein complexes using probabilistic support vector machines. *Proteins: Structure, Function, and Bioinformatics*, 70(4):1367–1378, 2007.

[84] F Gustav Mehler. Ueber die Entwicklung einer Function von beliebig vielen Variablen nach Laplaceschen Functionen höherer Ordnung. *Journal für die reine und angewandte Mathematik*, 66:161–176, 1866.

[85] R. Méndez, R. Leplae, L. De Maria, and S.J. Wodak. Assessment of blind predictions of protein–protein interactions: current status of docking methods. *Proteins: Structure, Function, and Bioinformatics*, 52(1):51–67, 2003.

[86] R. Méndez, R. Leplae, M.F. Lensink, and S.J. Wodak. Assessment of CAPRI predictions in rounds 3–5 shows progress in docking procedures. *Proteins: Structure, Function, and Bioinformatics*, 60(2):150–169, 2005.

[87] Julian Mintseris, Brian Pierce, Kevin Wiehe, Robert Anderson, Rong Chen, and Zhiping Weng. Integrating statistical pair potentials into protein complex prediction. *Proteins: Structure, Function, and Bioinformatics*, 69(3):511–520, 2007.

[88] John BO Mitchell, Roman A Laskowski, Alexander Alex, and Janet M Thornton. BLEEP—potential of mean force describing protein–ligand interactions: I. Generating potential. *Journal of Computational Chemistry*, 20(11):1165–1176, 1999.

[89] S. Miyazawa and R.L. Jernigan. Estimation of effective interresidue contact energies from protein crystal structures: quasi-chemical approximation. *Macromolecules*, 18(3):534–552, 1985.

[90] Sanzo Miyazawa and Robert L Jernigan. Self-consistent estimation of inter-residue protein contact energies based on an equilibrium mixture approximation of residues. *Proteins: Structure, Function, and Bioinformatics*, 34(1):49–68, 1999.

[91] Alexey G Murzin, Steven E Brenner, Tim Hubbard, and Cyrus Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology*, 247(4):536–540, 1995.

[92] J Navaza and E Vernoslova. On the fast translation functions for molecular replacement. *Acta Crystallographica Section A: Foundations of Crystallography*, 51(4):445–449, 1995.

[93] Jorge Navaza. On the computation of structure factors by FFT techniques. *Acta Crystallographica Section A: Foundations of Crystallography*, 58(6):568–573, 2002.

[94] Irene Nooren and Janet M Thornton. Diversity of protein–protein interactions. *The EMBO Journal*, 22(14):3486–3492, 2003.

[95] Siew Loon Ooi, Xuewen Pan, Brian D Peyser, Ping Ye, Pamela B Meluh, Daniel S Yuan, Rafael A Irizarry, Joel S Bader, Forrest A Spencer, and Jef D Boeke. Global synthetic-lethality analysis and yeast functional profiling. *Trends in Genetics*, 22(1):56–63, 2006.

[96] W. Park, G. Leibon, D.N. Rockmore, and G.S. Chirikjian. Accurate image rotation using Hermite expansions. *IEEE Transactions on Image Processing*, 18(9):1988–2003, 2009.

[97] Florencio Pazos and Alfonso Valencia. Similarity of phylogenetic trees as indicator of protein–protein interaction. *Protein Engineering*, 14(9):609–614, 2001.

[98] W.R. Pearson, T. Wood, Z. Zhang, and W. Miller. Comparison of DNA sequences with protein sequences. *Genomics*, 46(1):24–36, 1997.

[99] Pawel A Penczek, Marek Kimmel, and Christian MT Spahn. Identifying conformational states of macromolecules by eigen-analysis of resampled cryo-EM images. *Structure*, 19(11):1582–1590, 2011.

[100] Emilia Petrisor. Visualizing complex-valued functions with Matplotlib and Mayavi.

[101] Brian Pierce, Weiwei Tong, and Zhiping Weng. M-ZDOCK: a grid-based approach for Cn symmetric multimer docking. *Bioinformatics*, 21(8):1472–1478, 2005.

[102] Brian Pierce and Zhiping Weng. ZRANK: reranking protein docking predictions with an optimized energy function. *Proteins: Structure, Function, and Bioinformatics*, 67(4):1078–1086, 2007.

[103] Brian G Pierce, Yuichiro Hourai, and Zhiping Weng. Accelerating protein docking in ZDOCK using an advanced 3D convolution library. *PloS One*, 6(9):e24657, 2011.

[104] J. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. 1998.

[105] Massimiliano Pontil and Alessandro Verri. Properties of support vector machines. *Neural Computation*, 10(4):955–974, 1998.

[106] Petr Popov and Sergei Grudinin. Rapid determination of RMSDs corresponding to macromolecular rigid body motions. *Journal of Computational Chemistry*, 35(12):950–956, 2014.

[107] Petr Popov, David W Ritchie, and Sergei Grudinin. DockTrina: Docking triangular protein trimers. *Proteins: Structure, Function, and Bioinformatics*, 82(1):34–44, 2014.

[108] Oscar Puig, Friederike Caspary, Guillaume Rigaut, Berthold Rutz, Emmanuelle Bouveret, Elisabeth Bragado-Nilsson, Matthias Wilm, and Bertrand Séraphin. The tandem affinity purification (TAP) method: a general procedure of protein complex purification. *Methods*, 24(3):218–229, 2001.

[109] P.E. Ricci. Improving the asymptotics for the greatest zeros of Hermite polynomials. *Computers & Mathematics with Applications*, 30(3):409–416, 1995.

[110] David W Ritchie. Recent progress and future directions in protein-protein docking. *Current Protein and Peptide Science*, 9(1):1–15, 2008.

[111] David W Ritchie and Graham JL Kemp. Protein docking using spherical polar Fourier correlations. *Proteins: Structure, Function, and Bioinformatics*, 39(2):178–194, 2000.

[112] David W Ritchie, Dima Kozakov, and Sandor Vajda. Accelerating and focusing protein–protein docking correlations using multi-dimensional rotational FFT generating functions. *Bioinformatics*, 24(17):1865–1873, 2008.

[113] D.W. Ritchie and G.J.L. Kemp. Protein docking using spherical polar Fourier correlations. *Proteins: Structure, Function, and Bioinformatics*, 39(2):178–194, 2000.

[114] Gregory A Ross, Garrett M Morris, and Philip C Biggin. Rapid and accurate prediction and scoring of water molecules in protein binding sites. *PloS One*, 7(3):e32036, 2012.

[115] M.G. Rossmann, R. Bernal, and S.V. Pletnev. Combining electron microscopic with X-ray crystallographic structures. *Journal of Structural Biology*, 136(3):190–200, 2001.

[116] Dmitry Rykunov and András Fiser. Effects of amino acid composition, finite size of proteins, and sparse statistics on distance-dependent statistical pair potentials. *Proteins: Structure, Function, and Bioinformatics*, 67(3):559–568, 2007.

[117] Edward B Saff and A BJ Kuijlaars. Distributing many points on a sphere. *The Mathematical Intelligencer*, 19(1):5–11, 1997.

[118] D Sayre. The calculation of structure factors by Fourier summation. *Acta Crystallographica*, 4(4):362–367, 1951.

[119] M. Scarsi, N. Majeux, and A. Caflisch. Hydrophobicity at the surface of proteins. *Proteins: Structure, Function, and Bioinformatics*, 37(4):565–575, 1999.

[120] Dina Schneidman-Duhovny, Yuval Inbar, Ruth Nussinov, and Haim J Wolfson. Geometry-based flexible and symmetric protein docking. *Proteins: Structure, Function, and Bioinformatics*, 60(2):224–231, 2005.

[121] Tanvir R Shaikh, Haixiao Gao, William T Baxter, Francisco J Asturias, Nicolas Boisset, Ardean Leith, and Joachim Frank. SPIDER image processing for single-particle reconstruction of biological macromolecules from electron micrographs. *Nature Protocols*, 3(12):1941–1974, 2008.

[122] F.B. Sheinerman and B. Honig. On the role of electrostatic interactions in the design of protein–protein interfaces. *Journal of Molecular Biology*, 318(1):161–177, 2002.

[123] Min-Yi Shen and Andrej Sali. Statistical potential for assessment and prediction of protein structures. *Protein Science*, 15(11):2507–2524, 2009.

[124] Jian Shi, Yifan Wang, Lei Zeng, Yadi Wu, Jiong Deng, Qiang Zhang, Yiwei Lin, Junlin Li, Tiebang Kang, Min Tao, et al. Disrupting the Interaction of BRD4 with diacetylated twist suppresses tumorigenesis in basal-like breast cancer. *Cancer Cell*, 25(2):210–225, 2014.

[125] X. Siebert and J. Navaza. UROX 2.0: an interactive tool for fitting atomic models into electron-microscopy reconstructions. *Acta Crystallographica Section D: Biological Crystallography*, 65(7):651–658, 2009.

[126] M.J. Sippl. Calculation of conformational ensembles from potentials of mena force: an approach to the knowledge-based prediction of local structures in globular proteins. *Journal of Molecular Biology*, 213(4):859–883, 1990.

[127] M.J. Sippl, M. Ortner, M. Jaritz, P. Lackner, and H. Flöckner. Helmholtz free energies of atom pair interactions in proteins. *Folding and Design*, 1(4):289–298, 1996.

[128] H.S. Stone. Convolution theorems for linear transforms. *IEEE Transactions on Signal Processing*, 46(10):2819–2821, October 1998.

[129] George H Stout and Lyle H Jensen. *X-ray structure determination: a practical guide*, volume 2. Macmillan New York, 1968.

[130] K. Suhre, J. Navaza, and Y.H. Sanejouand. NORMA: a tool for flexible fitting of high-resolution protein structures into low-resolution electron-microscopy-derived density maps. *Acta Crystallographica Section D: Biological Crystallography*, 62(9):1098–1100, 2006.

[131] S. Tanaka and H.A. Scheraga. Medium-and long-range interaction parameters between amino acids for predicting three-dimensional structures of proteins. *Macromolecules*, 9(6):945–950, 1976.

[132] Kenneth A Taylor and Robert M Glaeser. Electron diffraction of frozen, hydrated protein crystals. *Science*, 186(4168):1036–1037, 1974.

[133] Lynn F Ten Eyck. Efficient structure-factor calculation for large molecules by the fast Fourier transform. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 33(3):486–492, 1977.

[134] Genki Terashi, Mayuko Takeda-Shitaka, Kazuhiko Kanou, Mitsuo Iwadate, Daisuke Takaya, and Hideaki Umeyama. The SKE-DOCK server and human teams based on a combined method of shape complementarity and free energy estimation. *Proteins: Structure, Function, and Bioinformatics*, 69(4):866–872, 2007.

[135] P.D. Thomas and K.A. Dill. Statistical potentials extracted from protein structures: how accurate are they? *Journal of Molecular Biology*, 257(2):457–469, 1996.

[136] D. Tobi and I. Bahar. Optimal design of protein docking potentials: efficiency and limitations. *Proteins: Structure, Function, and Bioinformatics*, 62(4):970–981, 2005.

[137] D. Tobi and R. Elber. Distance-dependent, pair potential for protein folding: Results from linear optimization. *Proteins: Structure, Function, and Bioinformatics*, 41(1):40–46, 2000.

[138] Peter Uetz, Loic Giot, Gerard Cagney, Traci A Mansfield, Richard S Judson, James R Knight, Daniel Lockshon, Vaibhav Narayan, Maithreyan Srinivasan, Pascale Pochart, et al. A comprehensive analysis of protein–protein interactions in Saccharomyces cerevisiae. *Nature*, 403(6770):623–627, 2000.

[139] Ilya A Vakser. Evaluation of GRAMM low-resolution docking methodology on the hemagglutinin-antibody complex. *Proteins: Structure, Function, and Bioinformatics*, 29(S1):226–230, 1997.

[140] Marin van Heel, George Harauz, Elena V Orlova, Ralf Schmidt, and Michael Schatz. A new generation of the IMAGIC image processing system. *Journal of Structural Biology*, 116(1):17–24, 1996.

[141] V. Vapnik. Estimation of dependences based on empirical data. *Nauka*, 1979.

[142] V. Vapnik. *The nature of statistical learning theory*. Springer, 1999.

[143] D. Vasishtan and M. Topf. Scoring functions for cryoEM density fitting. *Journal of Structural Biology*, 174(2):333–343, 2011.

[144] Adrián Velázquez Campoy and Ernesto Freire. ITC in the post-genomic era...? Priceless. *Biophysical Chemistry*, 115(2):115–124, 2005.

[145] M. Vendruscolo, R. Najmanovich, and E. Domany. Can a pairwise contact potential stabilize native protein folds against decoys obtained by threading? *Proteins: Structure, Function, and Bioinformatics*, 38(2):134–148, 2000.

[146] Vishwesh Venkatraman, Yifeng D Yang, Lee Sael, and Daisuke Kihara. Protein-protein docking using region-based 3D Zernike descriptors. *BMC Bioinformatics*, 10(1):407, 2009.

[147] N. Volkmann and D. Hanein. Quantitative fitting of atomic models into observed densities derived by electron microscopy. *Journal of Structural Biology*, 125(2):176–184, 1999.

[148] Lingle Wang, BJ Berne, and RA Friesner. Ligand binding to protein-binding pockets with wet and dry regions. *Proceedings of the National Academy of Sciences*, 108(4):1326–1330, 2011.

[149] T. Wang and R.C. Wade. Implicit solvent models for flexible protein–protein docking by molecular dynamics simulation. *Proteins: Structure, Function, and Bioinformatics*, 50(1):158–169, 2002.

[150] Shoshana J Wodak and Joël Janin. Computer analysis of protein-protein interaction. *Journal of Molecular Biology*, 124(2):323–342, 1978.

[151] W. Wriggers. Using Situs for the integration of multi-resolution structures. *Biophysical Reviews*, 2(1):21–27, 2010.

[152] Yuling Yan and Gerard Marriott. Analysis of protein interactions using fluorescence technologies. *Current Opinion in Chemical Biology*, 7(5):635–640, 2003.

[153] Martin Zacharias. Protein–protein docking with a reduced protein model accounting for side-chain flexibility. *Protein Science*, 12(6):1271–1282, 2003.

[154] Jian Zhang and Yang Zhang. A novel side-chain orientation dependent potential derived from random-walk reference state for protein fold selection and structure prediction. *PloS One*, 5(10):e15386, 2010.

[155] Xing Zhang, Lei Jin, Qin Fang, Wong H Hui, and Z Hong Zhou. 3.3 Å cryo-EM structure of a nonenveloped virus reveals a priming mechanism for cell entry. *Cell*, 141(3):472–482, 2010.

[156] Hongyi Zhou and Yaoqi Zhou. Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction. *Protein Science*, 11(11):2714–2726, 2002.