

2018

Optimization and machine learning methods for Computational Protein Docking

<https://hdl.handle.net/2144/32673>

"Downloaded from OpenBU. Boston University's institutional repository."

BOSTON UNIVERSITY
COLLEGE OF ENGINEERING

Dissertation

**OPTIMIZATION AND MACHINE LEARNING
METHODS FOR COMPUTATIONAL PROTEIN
DOCKING**

by

SHAHROOZ ZARBAFIAN

B.S., University of Tehran, 2013

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

2018

© 2018 by
SHAHROOZ ZARBAFIAN
All rights reserved

Approved by

First Reader

Pirooz Vakili, PhD
Associate Professor of Mechanical Engineering
Associate Professor of Systems Engineering

Second Reader

Ioannis Paschalidis, PhD
Professor of Electrical and Computer Engineering
Professor of Biomedical Engineering
Professor of Systems Engineering

Third Reader

Sandor Vajda, PhD
Professor of Biomedical Engineering
Professor of Systems Engineering
Professor of Chemistry

Fourth Reader

Roberto Tron, PhD
Assistant Professor of Mechanical Engineering
Assistant Professor of Systems Engineering

We must not, in trying to think about how we can make a big difference, ignore the small daily differences we can make which, over time, add up to big differences that we often cannot foresee. Marian Wright Edelman

Acknowledgments

I would like to sincerely thank my academic advisor Pirooz Vakili for being an understanding mentor and a great friend. I substantially benefited from his constructive feedback on my research performance and his relentless effort to promote my critical thinking abilities. Moreover, I greatly enjoyed his companionship outside the academic environment and his dedication to help me through the challenges that come with starting life in a new country. I also would like to thank my co-advisors Ioannis Paschalidis and Sandor Vajda who inspired me through my PhD program, helped me grow as a researcher and shared their perspective on my research efforts. I would like to thank Roberto Tron who graciously accepted to be part of my PhD committee and whose feedback, specifically on the third chapter of this work, was particularly instructive.

I would like to thank all my friends who made my PhD experience very enjoyable. I would like to thank my colleagues at Structural Bioinformatics and Network Optimization lab and Control (NOC) labs with whom I have had many hours of productive discussion as well as leisure time. Also, I would like to thank Athar, Arian, Iman and Sadra with whom I could keep my mother tongue fluent!

I would like to thank Chantal Fujiwara whose encouragements were always there to keep me going and who came to be the closest one can hope to have as a second mother. I would like to thank Holversons for their hospitality and all the enjoyable moments we have shared so far.

Last but not least, I would like to thank my close family without whom this journey would not have been possible. I would like to thank my beloved mother who has supported my effort to study abroad from the beginning, whose comforting words have helped me keep my spirit up in the face of challenges and whose numerous sacrifices I deeply appreciate. I would like to thank my older brother Shizar who has

been a source of encouragement and support for my efforts and ambitions.

Shahrooz Zarbafian

August 2018

Boston

**OPTIMIZATION AND MACHINE LEARNING
METHODS FOR COMPUTATIONAL PROTEIN
DOCKING**

SHAHROOZ ZARBAFIAN

Boston University, College of Engineering, 2018

Major Professor: Pirooz Vakili, PhD

Associate Professor of Mechanical Engineering
Associate Professor of Systems Engineering

ABSTRACT

Computational Protein Docking (CPD) is defined as determining the stable complex of docked proteins given information about two individual partners, called receptor and ligand. The problem is often formulated as an energy/score minimization where the decision variables are the 6 rigid body transformation variables for the ligand in addition to more variables corresponding to flexibilities in the protein structures. The scoring functions used in CPD are highly nonlinear and nonconvex with a very large number of local minima, making the optimization problem particularly challenging. Consequently, most docking procedures employ a multistage strategy of (i) Global Sampling using a coarse scoring function to identify promising areas followed by (ii) a Refinement stage using more accurate scoring functions and possibly allowing more degrees of freedom.

In the first part of this work, the problem of local optimization in the refinement stage is addressed. The goal of local optimization is to remove steric clashes between protein partners and obtain more realistic score values. The problem is formulated

as optimization on the space of rigid motions of the ligand. Employing a recently introduced representation of the space of rigid motions as a manifold, a new Riemannian metric is introduced that is closely related to the Root Mean Square Deviation (RMSD) distance measure widely used in Protein Docking. It is argued that the new metric puts rotational and translational variables on equal footing as far local changes of RMSD is concerned. The implications and modifications for gradient-based local optimization algorithms are discussed.

In the second part, a new methodology for resampling and refinement of ligand conformations is introduced. The algorithm is a refinement method where the inputs to the algorithm are ensembles of ligand conformations and the goal is to generate new ensembles of refined conformations, closer to the native complex. The algorithm builds upon a previous work and introduces multiple new innovations: Clustering the input conformations, performing dimensionality reduction using Principle Component Analysis (PCA), underestimating the scoring function and resampling and refinement of new conformations. The performance of the algorithm on a comprehensive benchmark of protein complexes is reported.

The third part of this work focuses on using machine learning framework for addressing two specific problems in Protein Docking: (i) Constructing a machine learning model in order to predict whether a given receptor and ligand pair interact. This is of significant importance for constructing the so-called protein interaction networks, an critical step in the Drug Discovery process. The success of the algorithm is verified on a benchmark for discrimination between Biological and Crystallographic Dimers. (ii) A ranking scheme for output predictions of a protein docking server is devised. The machine learning model employs the features of the docking server predictions to produce a ranked list with the top ranked predictions having higher probability of being close to the native solution. Two state-of-the-art approaches to

the ranking problem are presented and compared in detail and the implications of using the superior approach for a structural docking server is discussed.

Contents

1	Introduction	1
2	Preliminaries	7
2.1	Native Complex	7
2.2	Root Mean Square Deviation	7
2.3	Cluspro	9
2.3.1	Global Sampling	9
2.3.2	Refinement	13
3	Ligand-based Metric for Manifold Optimization	14
3.1	Introduction	14
3.2	Space of Rigid Transformations	18
3.2.1	Lie Groups	18
3.2.2	Common Formulation of Rigid Transformations	19
3.2.3	New Formulation of Rigid Transformations	23
3.2.4	Comparison of the Representations	27
3.3	RMSD compatible Riemannian metric	30
3.3.1	New Riemannian metric	35
3.3.2	Future Directions	39
4	Semi Definite Subspace Underestimation	41
4.1	Introduction	41
4.2	Methods	47

4.2.1	Clustering and Outlier Elimination	47
4.2.2	Dimensionality Reduction	48
4.2.3	Underestimation	51
4.2.4	Sampling	54
4.2.5	SSDU Algorithm	55
4.2.6	Local Minimization	56
4.2.7	Energy Function	57
4.2.8	Validation dataset and input preparation	58
4.3	Results and Discussion	59
4.3.1	Protein Docking Refinement	59
4.3.2	Post-Processing Ensemble Enrichment	62
4.4	Conclusions	66
5	Machine Learning methods in Protein Docking	70
5.1	Introduction	70
5.2	Classification Models	72
5.2.1	Sparse Linear Support Vector Machine	72
5.2.2	Alternating Clustering and Classification	73
5.3	Performance Measures for Classification	76
5.3.1	Confusion Matrix	76
5.3.2	Accuracy, True Positive Rate and False Positive Rate	77
5.3.3	Receiver Operating Characteristic Curve	78
5.4	Interacting vs Non-interacting complexes	79
5.4.1	Methodology	80
5.4.2	Results	84
5.4.3	Feature Analysis	85
5.5	Ranking of Clusters of Conformations	87

5.5.1	Methodology	91
5.5.2	Results	96
5.6	Conclusion	96
6	Conclusion	98
	References	102
	Curriculum Vitae	111

List of Tables

4.1	Percentage improvement of Acceptable (or better), Medium (or better) and High quality solutions by SSDU versus SDU and ClusPro for a benchmark of 224 complexes. Please note that for each the entries in the table, complexes with zero number of solutions for both Cluspro and SDU/SSDU are removed.	69
5.1	The feature description of the important features presented in section 5.4.3.	87

List of Figures

2.1	RMSD is in fact an <i>average pairwise atom distance</i> between two different poses of the <i>same protein</i> . The protein is at initial position a and is moved to position b and the RMSD between poses a and b is $\sqrt{\frac{d_1+d_2+d_3}{3}}$	8
2.2	Cluspro is a web-server for simulating protein-protein interaction (Kozakov et al., 2017). The input to the server are the protein pairs of the interest, ligand and receptor, and the output is a selected set of ligand and receptor relative orientation and positions having a high probability of being close to the native complex. Visit Cluspro web page for more information.	10
2.3	An illustrative example of the free binding energy landscape for protein docking. The energy landscape is highly non-convex and has numerous local minima and finding the global minimum is fairly challenging. . .	10
2.4	A schematic of the ligand moving grids in 2D space for PIPER global sampling stage. The protein in red is the receptor which is fixed at the origin of the coordinate axis and the ligand as the green protein is moved on the grid points encompassing all translational directions. The binding free energy of the ligand and receptor is evaluated on each of the grid points.	11

2·5	Performing energy filtering on the outputs of PIPER to keep the top 1000 ligand conformations with the lowest energy values. The low energy conformations tend to cluster around the local minima of the free binding energy function.	12
3·1	The geometric approach to protein docking exploits the special structure of the search space, namely the <i>manifold of rigid transformations</i> . In this figure, the ligand as the protein in purple is moved from initial position x_1 at a higher energy value to the closest local minimum of the energy landscape at x_2 . The search space is a curved space, namely the manifold of rigid transformation where the decision variables are the 3 rotation and 3 translation parameters.	16
3·2	A triangle is moved using a rigid transformation in multiple steps from an initial position. On the left, the center of rotation is translated in each step the same amount as the triangle. On the right, the center of rotation is fixed at the origin of the coordinate axis throughout the movement. The steps of movement are shown in blue dashed lines and the moving centers of rotation are shown using the small blue circles.	29
4·1	An illustration of low-energy clusters of complexes and their underestimators that outline the broad local funnel.	43
4·2	The near-native energy landscape of the 2YVJ complex.	45
4·3	One underestimator (denoted by red dashed lines) per cluster is calculated. If the underestimation step succeeds in capturing the shape of the free energy function, then the sampling step will help generate more conformations in the vicinity of the global minimum of the energy function.	54

4.4	The flowchart of the SSDU procedure.	56
4.5	The <i>x</i> -axis represents 156 out of 224 protein complexes that have either Clupro or SSDU non-zero CAPRI Acceptable (or better) quality solutions. The complexes are sorted by the number of ClusPro counts, and the <i>y</i> -axis shows the number of Acceptable quality solutions out of an ensemble of 1000 or 1500 for enzymes/antibodies and other types conformations respectively produced by ClusPro, or refined by SDU and SSDU.	61
4.6	The <i>x</i> -axis represents 110 out of 224 protein complexes that have either Clupro or SSDU non-zero CAPRI Medium (or better) quality solutions. The complexes are sorted by the number of ClusPro counts, and the <i>y</i> -axis shows the number of Medium quality solutions out of an ensemble of 1000 or 1500 for enzymes/antibodies and other types conformations respectively produced by ClusPro, or refined by SDU and SSDU.	62
4.7	The <i>x</i> -axis represents 29 out of 230 protein complexes that have either Clupro or SSDU non-zero CAPRI High quality solutions. The complexes are sorted by the number of ClusPro counts, and the <i>y</i> -axis shows the number of High quality solutions out of an ensemble of 1000 or 1500 for enzymes/antibodies and other types conformations respectively produced by ClusPro, or refined by SDU and SSDU.	63
4.8	The percentage of increase in the number of solutions for SSDU vs Cluspro among top clusters. The <i>x</i> -axis represents the category of quality of solutions according to the CAPRI criteria.	67

5.1	ACC (Dai, 2015) is an algorithm joint clustering and classification of the input data where the positive class possibly consists of multiple hidden subclusters where the basis for clustering the positive data is the similarity between the members of a cluster in terms of discriminating features from the negative class. The goal is to find hidden clusters of positive data while finding the optimal classifier within each cluster. Note that how the calculated classifiers for cluster 1 and cluster 2, denoted by green dashed lines, are different.	74
5.2	A sample Receiver Operating Characteristic curve.	79
5.3	When performing X-ray crystallography on dimers, there is often a need for further analysis to determine whether the observed interaction between partners in the crystal is (denoted by blue dashed line) Biological or a byproduct of the experimental condition, such as the high concentration of the proteins.	81
5.4	ROC curve for Random Forest on Biological vs Crystallographic Dimers dataset. The AUC on the test data is 0.98.	85
5.5	ROC curve for Sparse Linear SVM on Biological vs Crystallographic Dimers dataset. The AUC on the test data is 0.98.	86
5.6	ROC curve for Alternating Clustering and Classification method on Biological vs Crystallographic Dimers dataset. The AUC on the test data is 0.95.	88
5.7	The highest Permuted Predictor Delta Error for the top 20 features using Random Forest on the Biological versus Crystallographic test data.	89

5.8	The normalized feature value difference among 3 clusters of positive class identified using ACC algorithm. The selected features are the important features identified using Random Forest in section 5.4.3 and the values of the features are averaged over the members of each cluster. The feature descriptions are given in table 5.1	90
-----	---	----

Chapter 1

Introduction

Proteins are essential to many processes in living organisms. They regulate hormones, act as antibodies against disease agents and as catalysts for metabolic reactions. Proteins consist of chains of smaller units called amino acids and there are about 20 amino acids found in human body. In fact, the sequence, type and number of the amino acids are what determines the 3D structure of the protein which will determine its specific functionality. Furthermore, proteins rarely act alone; they interact with other proteins to form complexes and build more sophisticated blocks called “molecular machines” which undertake significant biological functions (Rivas and Fontanillo, 2010).

Proteins are the key component of cell and biological systems whereby interacting with other proteins, they carry out major biological functions such as cell growth, gene regulation, signal transduction, etc. Protein-Protein interactions (PPI) are essential to many biological processes and the study of PPIs spans many fields. For instance, study of protein-protein or protein-small molecule interactions play a major role in Drug Discovery process where to develop a marketable drug, it usually takes 10 years and upto \$1 billion dollars of resources (Hughes et al., 2011). Specifically, in the first stage of Drug Development, namely Drug Discovery, one of the main approaches is to design/identify small molecule drugs that specifically inhibit a target pathogen (disease agent). One way to do so is to examine a large number of drug candidates to identify the most promising ones. However, High throughput experimental procedures

are often time consuming and expensive. For instance, using wet-lab High-throughput Screening (HTS) a scientist can quickly conduct a large number (upto a million) of lab experiments to validate the efficacy of the drug candidates but the high cost and low accuracy of these procedures have led to development of innovative computational methods (Cheng et al., 2012).

Virtual Screening (VS) is a computational technique where a relatively large library of small molecules is examined to identify the most likely candidates that can bind to a target protein. In fact, Docking Based Virtual Screening is one of the most prevalent structure based VS. In this case, a docking program is used to virtually dock a library of candidate proteins onto a target protein where using a scoring function, the docked solutions are ranked and a small portion of the top ranked proteins are retained for further experimentation.

This thesis is focused on the problem of Computational Protein Docking. Formally, Protein Docking is defined as finding the stable complex from two individual protein partners. Protein Docking is used to identify the interface of interaction and measure the docking affinity (strength) of a protein pair termed ligand and receptor. Ligand can be a small molecule or a protein with therapeutic effect and receptor a larger protein considered to be the target. In the simplest model where proteins are considered rigid bodies, this problem can be formulated as a “lock and key” model where the goal is to determine how ligand as a “key” can correctly be inserted into the receptor as the “lock” (Jorgensen, 1991). As a mathematical optimization problem, the receptor is kept fixed in the space and ligand is moved relative to the position of the receptor hence the decision variables are the 6 rigid body transformation variables for ligand in addition to more variables corresponding to flexibilities in the protein structures. Furthermore, the objective is to find the minimum of some scoring functions that represents the interaction between the proteins. This optimization

procedure corresponds to the Minimum Energy Principle where according to the second law of thermodynamics, a closed system is driven to the stable equilibrium with minimum energy.

Two of the main challenges in Protein Docking are conformational changes in the protein structures upon binding and the highly nonlinear scoring functions used. To address the first challenge, some algorithms use information about the protein pairs to identify rigid domains and flexible joints in protein structures when solving the optimization problem (Mirzaei et al., 2015). However, modeling flexibilities in the protein backbone has been a major challenge (Andrusier et al., 2008) so some algorithms only focus on adjusting the side-chains of the protein interfaces (Moghadasi et al., 2015).

Considering the second issue, the scoring functions used in Computational Protein Docking are highly nonlinear and nonconvex and have a considerably large number of local minima. Consequently, most docking procedures employ a multistage strategy where in the first step, using global sampling schemes, the areas of the conformational space that seem promising are identified for further exploration. In this stage, a fairly large number of relative positions of the ligand are evaluated and to manage the computation burden, most algorithms make simplifying assumptions such as using approximate scoring functions. In the second stage called refinement, the highly ranked positions and orientations of the ligand from the global sampling stage are further refined by using more sophisticated scoring functions and possibly allowing more degrees of freedom leading to a more accurate minimization of the scoring functions.

Computational Protein Docking has been an active research area and a number of different research groups have been developing and maintaining docking software and servers. The servers can be categorized into (i) rigid and (ii) flexible docking. Many

rigid docking servers incorporate Fast Fourier Transform (FFT) to perform global sampling. ZDOCK uses FFT to globally optimize a combination of scoring functions including shape complementarity, electrostatics and statistical potential (Pierce et al., 2014). Cluspro performs FFT as first stage to globally sample the conformational space followed by pairwise Root Mean Square Deviation (RMSD) clustering where the selected structures are further refined in the final stage (Kozakov et al., 2013). GRAMM-X incorporates empirically smoothed potentials to evaluate millions of conformations using FFT followed by local minimization refinement (Tovchigrechko and Vakser, 2006). The challenge, however, for rigid docking are the protein pairs that undergo significant conformational change upon binding.

Motivated by induced fit theorem, flexible docking programs have introduced flexibilities in the protein structures, allowing for more accurate minimization of docking poses. While divided on how to place the ligand, flexible docking servers can be broadly classified into shape-based, genetics algorithm, global searches and Monte Carlo Simulation algorithms (Kuntz et al., 1982), (Jones et al., 1997), (Friesner et al., 2004), (Venkatachalam et al., 2002). These docking methods mostly introduce the flexibility in the ligand structure whereas flexible docking of receptor still remains a challenging problem (Pagadala et al., 2017).

In the first part of this thesis, presented in chapter 3, the problem of local optimization in the refinement stage is addressed. After obtaining close to the native solutions, many docking algorithms incorporate a refinement stage to remove steric clashes between the protein partners and obtain more realistic score values (Vajda and Kozakov, 2009). Local optimization is one of the main components of the refinement stage where one can exploit the special structure of the search space and represent it as the product two individual manifolds (Mirzaei et al., 2012). In fact, it has been shown that by representing the space of rigid body movement as a direct product

manifold, one can generalize the efficient optimization algorithms developed for Euclidean space to the direct product manifold (Vakili et al., 2014). In this part, first a general overview of differential geometry and Riemannian Manifold is presented. Secondly, two different representations of the manifold of rigid transformations and their characteristics are discussed. The chapter concludes with the main contribution of this part which is to define a new metric closely related to the widely used metric in Protein Docking, namely Root Mean Square Deviation (RMSD) on the corresponding manifold.

In the second part, presented in chapter 4, a new methodology for resampling and refinement of protein conformations is introduced. The algorithm is a refinement method where the inputs to the algorithm are the conformations from the global sampling stage and the goal is to generate an ensemble of refined conformations closer to the native complex. The algorithm builds upon a previous project (Shen et al., 2008) and introduces multiple new innovations: Clustering the input conformations, performing Principle Component Analysis (PCA), underestimating the scoring function and resampling and refinement of new conformations. Clustering the input conformations distributes the input conformations into different groups and enables finding separate underestimator for each group in the following stages. By performing PCA, the dimension of the search space is decreased and the principle directions of preferred association of protein partners are identified. Finally, the challenging problem of finding the global minimum of a highly non-convex function is alleviated by underestimating the scoring function by a general class of convex polynomials where new samples are generated and refined in the vicinity of the underestimator's global minimum. The performance of the algorithm on a comprehensive benchmark of 224 protein complexes is reported.

The third part of the work, presented in chapter 5, focuses on using a machine

learning framework for addressing two specific problems in Protein Docking: (i) Constructing a machine learning model in order to predict whether a given receptor and ligand pair interact. This is of significant importance for constructing the so-called protein interaction networks, a critical step in the Drug Discovery process. The success of the algorithm is verified on a benchmark for discrimination between Biological and Crystallographic Dimers. (ii) A ranking scheme for output predictions of protein docking servers is devised. In fact, currently no clear preference is given to any docking prediction and all the predictions are presented as having the same likelihood of being the correct solution. This strategy has been shown to result in performance degradation. The machine learning model employs the features of the docking server predictions to produce a ranked list with the top ranked predictions having higher probability of being close to the native solution. Two state-of-the-art approaches to the ranking problem are presented and compared in detail and the implications of using the superior approach for a structural docking server are discussed.

Chapter 2

Preliminaries

This chapter is intended to introduce three notions that are used throughout this thesis, namely (i) the native complex of a protein pair (ii) Root Mean Square deviation as a distance measure between different poses of the same protein and (iii) Cluspro protein docking webserver which generates the input data to most of the projects discussed in this thesis.

2.1 Native Complex

As mentioned before, protein docking strives to find the most likely complex formed by two protein partners. The actual solution to the docking problem, which is determined through experiments such as Nuclear Magnetic Resonance (NMR) (Johnson, 1999) or X-ray Crystallography (Smyth and Martin, 2000), is called the *native complex*. Specifically, the native complex is the docking pose of the two protein partners that is found in nature where the native complex is used as the *ground truth* and the quality of the docking program outputs are measured against the native complex.

2.2 Root Mean Square Deviation

Root mean square deviation or RMSD is a widely used in protein docking to measure the distance between two different orientation and position of the *same protein*. One of the main applications of RMSD in protein docking is to measure how close a

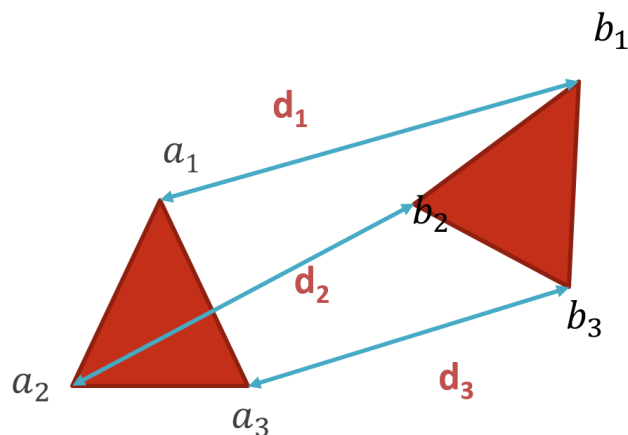


Figure 2.1: RMSD is in fact an *average pairwise atom distance* between two different poses of the *same protein*. The protein is at initial position a and is moved to position b and the RMSD between poses a and b is $\sqrt{\frac{d_1+d_2+d_3}{3}}$

proposed solution of a docking software is to the native ligand where lower RMSD values correspond to higher quality solutions. RMSD is in fact an *average pairwise atom distance* between two different poses of the *same protein* where RMSD of zero corresponds to two identical poses and higher RMSDs imply the poses being further apart (see figure 2.1). Specifically, if a protein has k atoms and the atom coordinates are initially a_i , $i = 1, \dots, k$ and the protein is moved to the final position where atom coordinates are b_i , $i = 1, \dots, k$ then the RMSD between the two poses is calculated as follows:

$$RMSD = \sqrt{\frac{1}{k} \sum_{i=1}^k \|a_i - b_i\|^2} \quad (2.1)$$

where $\|\cdot\|$ denotes the Euclidean norm.

As receptor is usually stationary and the goal is to find the optimal coordinates of the the ligand in a docking problem, RMSD is frequently calculated between different poses of the moving ligand and the native one. RMSD can be defined over a subset of atoms: (i) Interface RMSD or iRMSD is defined over the subset of the ligand atoms

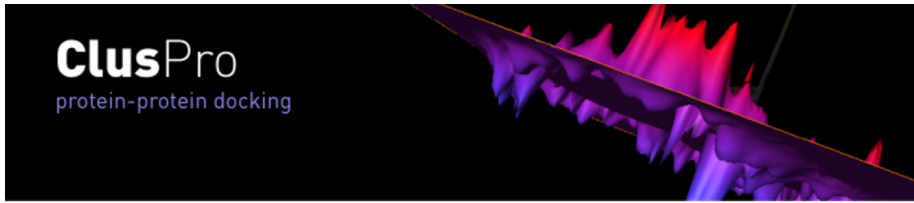
that are present in the surface of interaction between the receptor and the ligand in the native complex. (ii) back-bone RMSD or LRMSD is defined over the carbon alpha atoms of the ligand. The choice between iRMSD and LRMSD as a metric depends on the application. For instance, iRMSD is relevant when one is primarily interested in the interface of interaction of the ligand and the receptor.

2.3 Cluspro

Cluspro is a web-server for simulating protein-protein interaction (Kozakov et al., 2017). The inputs to the server are the protein pairs of interest, ligand and receptor, and the output is a selected set of ligand and receptor relative orientation and positions having a high probability of being close to the native complex (See figure 2.2). As mentioned before, one of the main challenges for protein docking is the free binding-energy landscape being highly non-convex and having numerous local minima (see figure 2.3) where finding the global minimum of a non-convex function is a highly challenging problem. Consequently, Cluspro, similar to many other docking servers (Pierce et al., 2014), (Tovchigrechko and Vakser, 2006), employs a multi-stage docking protocol where initially the energy landscape is *globally* sampled to identify the areas of the search space that have relatively lower energy values and in the following steps representatives from these promising areas are further *refined* using energy minimization through local optimization routines. In the following, the two main stages of the Cluspro algorithm, namely (i) global sampling and (ii) refinement are discussed.

2.3.1 Global Sampling

In the global sampling stage, billions of samples are evaluated where for each relative rotation of the ligand, the receptor is fixed at origin of the coordinate axis and the



Welcome to Cluspro 2.0

Recent news: [ClusPro server tops the competition in the latest rounds of CAPRI experiment](#)

Use Without an Account

[Use the server without the benefits of your own account](#)

--or--

Login

Username:

Password:

Figure 2·2: Cluspro is a web-server for simulating protein-protein interaction (Kozakov et al., 2017). The input to the server are the protein pairs of the interest, ligand and receptor, and the output is a selected set of ligand and receptor relative orientation and positions having a high probability of being close to the native complex. Visit Cluspro web page for more information.

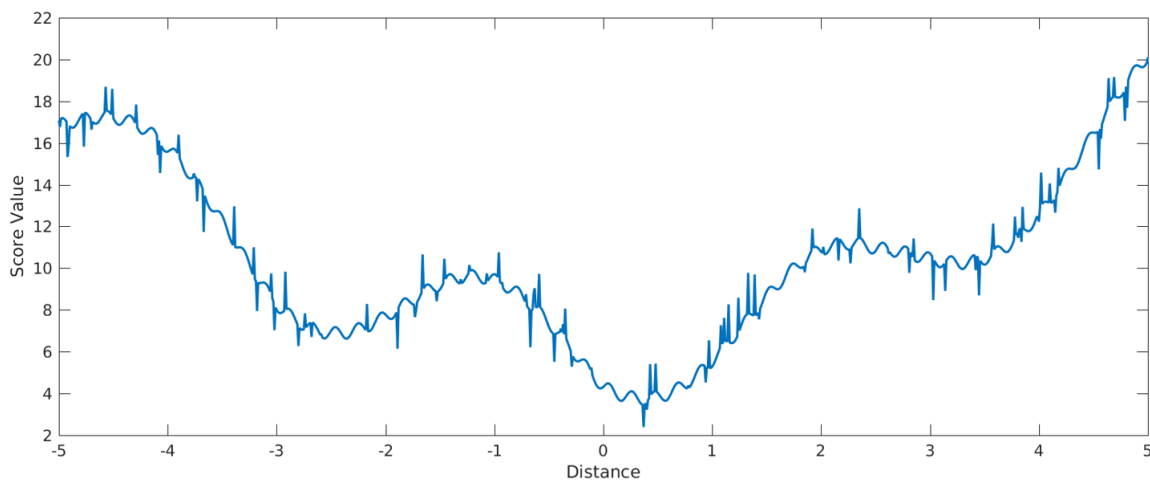


Figure 2·3: An illustrative example of the free binding energy landscape for protein docking. The energy landscape is highly non-convex and has numerous local minima and finding the global minimum is fairly challenging.

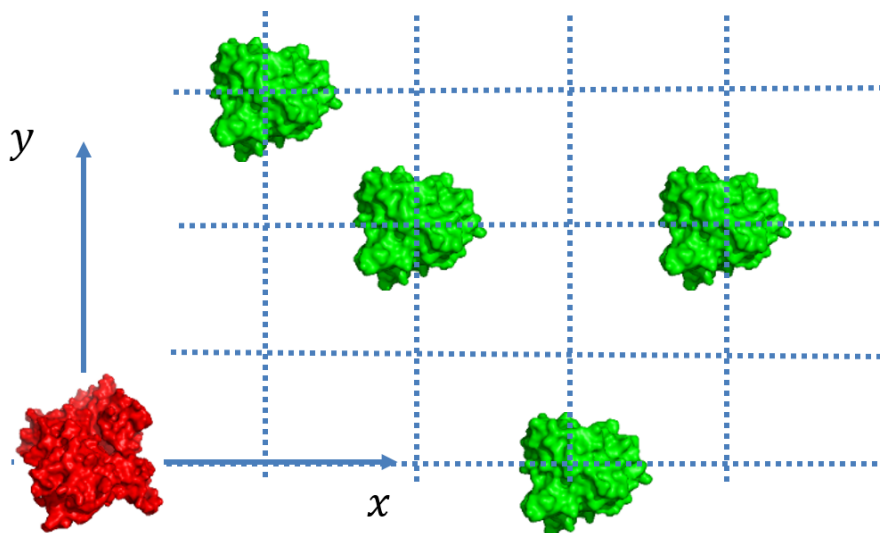


Figure 2.4: A schematic of the ligand moving grids in 2D space for PIPER global sampling stage. The protein in red is the receptor which is fixed at the origin of the coordinate axis and the ligand as the green protein is moved on the grid points encompassing all translational directions. The binding free energy of the ligand and receptor is evaluated on each of the grid points.

binding free energy of the protein pair is evaluated while ligand is moved on grid points encompassing all three translational directions (see figure 2.4). The docking algorithm used within Cluspro webserver to perform the global sampling is called PIPER (Kozakov et al., 2006). To allow for such refined sampling, special form energy functions in conjunction with Fast Fourier Transform (FFT) technique (Katchalski-Katzir et al., 1992) is employed.

The total free binding energy in PIPER is calculated as a linear combination of four energy components: (Kozakov et al., 2006)

$$E_{\text{Free Binding Energy}} = w_1 E_{\text{rep}} + w_2 E_{\text{attr}} + w_3 E_{\text{elec}} + w_4 E_{\text{DARS}} \quad (2.2)$$

The first two terms E_{rep} and E_{attr} represent repulsive and attractive Vander Waals energies respectively, scoring how well the current conformation of the ligand matches

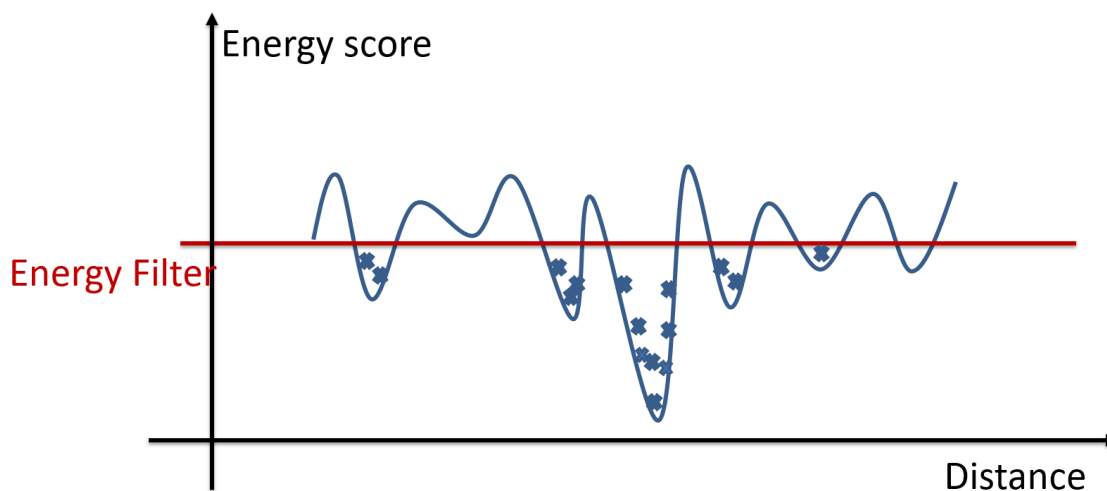


Figure 2.5: Performing energy filtering on the outputs of PIPER to keep the top 1000 ligand conformations with the lowest energy values. The low energy conformations tend to cluster around the local minima of the free binding energy function.

that of the receptor, similar to evaluating how well two pieces of puzzles match one another. The third term E_{elec} corresponds to electrostatic energy function which models the interaction of the atom charges of the receptor and the ligand. The fourth term E_{DARS} is a statistical atomic contact energy term called *Decoys as the Reference State* which prioritizes the atom contacts that appear more frequently in the interface of interaction of different classes of protein-protein interactions (Chuang et al., 2008).

Additional post-processing steps are performed to account for the approximations made in the global sampling stage. Specifically, two additional steps of *energy filtering* and *clustering* are carried out. By performing energy filtering, the top 1000 conformations of the ligand that have the best PIPER energy scores are retained. These top conformations tend to cluster around the local minima of the free binding energy landscape as seen in figure 2.5. By using clustering, one can identify these local minima. In the clustering stage ClusPro employs a greedy algorithm (Kozakov

et al., 2017) where at each iteration, the ligand conformation with the largest number of neighbors is identified (two conformations are considered neighbors if their pairwise iRMSD is less than a 9 Å threshold). Then, the conformation with the highest number of neighbors is labeled a *cluster center* and along with its neighbors form a cluster and removed from the ensemble. The procedure is repeated for the remaining conformations. Overall, a maximum of 30 clusters are formed where each cluster contains at least 10 members. As mentioned before, the clusters of low energy conformations are generally formed close to the local minima of the energy landscape and are considered “promising” for further exploration. The challenge, however, is to pick the best cluster corresponding to the global minimum. Accordingly, Cluspro uses the size of a cluster as an indication of the width of the *energy funnel*, providing information on the Entropic contribution to the free binding energy (Kozakov et al., 2017) where the clusters with higher number of members are considered more likely to be close to the native complex. Hence, the 30 clusters are ranked according to the size and only the 10 largest clusters are retained where the cluster centers of each cluster are chosen as the representatives of the clusters. These centers are further processed in the refinement stage.

2.3.2 Refinement

In the refinement stage, The centers of the top 10 largest clusters are locally minimized using the Vander Waals contribution of CHARMM potential (Brooks et al., 1983) to remove the steric clashes from the interfaces of interaction of the protein partners where the adjustment in the protein partners backbone in this step is usually small. In the end, the top 10 refined clusters centers are presented as the output of the Cluspro protein docking webserver.

Chapter 3

Ligand-based Metric for Manifold Optimization

3.1 Introduction

As mentioned before, protein docking can be formulated as an optimization problem where the goal is to minimize the binding free energy of the protein complex through finding the optimal relative poses of the proteins. Due to the highly rugged landscape of the binding energy landscape, docking protocols use a multi-stage approach where at the initial stage the search space is globally sampled on a grid using a simplified scoring function. The objective is to identify low energy funnels and narrowing down the search to promising regions. However, using grid-based sampling and simplified scoring functions promotes unrealistic poses of the ligand in the final solution. For instance, the scoring function might assign a fairly low energy value to a docking prediction with *steric clashes*, i.e., a prediction where the protein partners have overlapping subunits in the space. Moreover, many docking protocols assume fully rigid protein structures, finding appropriate rigid transformation coordinates but failing to find an accurate conformational change of the proteins backbone (Heo et al., 2016). Consequently, many docking protocols incorporate a *refinement* stage to address these issues.

Well-known refinement procedures such as Monte Carlo stochastic minimization, backbone adjustment routines (Gray et al., 2003), resampling techniques (Zarbfian

et al., 2018) and full atomic minimization (Kozakov et al., 2017) incorporate *local optimization* as one of their main components where the success of the local optimization is pivotal for the success of the refinement protocol. The input to the local optimization are poses of the ligand generated in the global sampling stage. In this stage, *off-grid* minimization is performed where the previous grid constraints of ligand movement are removed. Moreover, as there are far fewer predictions to process, (i) the use of more accurate and sophisticated models for free binding energy is justified and (ii) the rigid assumption of protein structures might be relaxed. All in all, local optimization focuses on refining the selected conformations using more accurate models and computation power.

There have been different approaches to performing local optimization in Protein Docking. The differences originate from how the decision variables and the constraints for preserving atomic bond properties are defined. In *full atomic* minimization, decision variables are the translation parameters along all three directions for every atom in the ligand, resulting in prohibitively large number of decision variables, about 6000, for a medium sized protein. The constraints for full atomic minimization are typically enforced by including scoring functions that penalize deviation of atomic bond properties from their nominal values. On the other hand, one can assume a rigid body model for the proteins' structures and consider the decision variables as a 3×3 rotation matrix R and a 3 dimensional translation vector. Moreover, one should ensure that the transformation matrix R is an *orientation-preserving rotation matrix*:

$$R \in SO(3) = \{R \in \mathbb{R}^{3 \times 3} | \det(R) = 1, R \times R^T = 1\} \quad (3.1)$$

The issue, however, is that above constraint for matrix R is *non-convex* and naive adaptations of Euclidean optimization routines will result in severely inefficient local

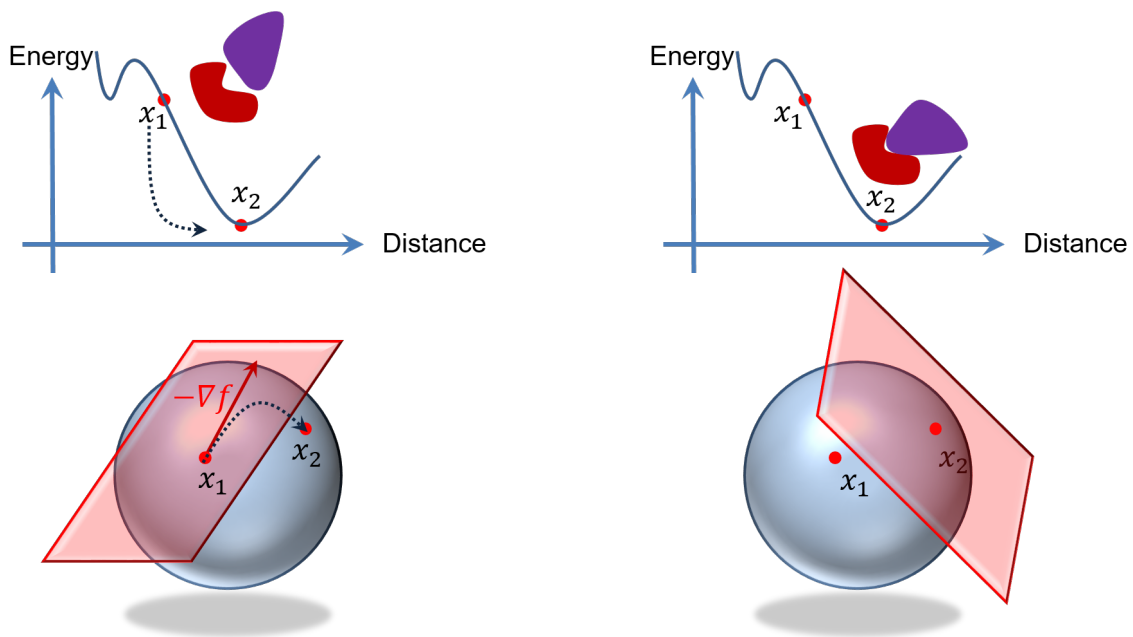


Figure 3.1: The geometric approach to protein docking exploits the special structure of the search space, namely the *manifold of rigid transformations*. In this figure, the ligand as the protein in purple is moved from initial position x_1 at a higher energy value to the closest local minimum of the energy landscape at x_2 . The search space is a curved space, namely the manifold of rigid transformation where the decision variables are the 3 rotation and 3 translation parameters.

optimization protocols requiring an *exponential number of steps* in the worst case to find the global minimum of the optimization problem.

One can avoid the non-convexity of the constraint for the rotation matrix by adopting a *geometric* approach to convert the problem to an unconstrained optimization. The geometric approach exploits the special structure of the search space, namely the *manifold of rigid transformations* where the decision variables are the 3 rotation and 3 translation parameters (see figure 3.1). The commonly used manifold representation of the rigid transformations, namely Special Euclidean group $SE(3)$, has been used extensively in the literature to address problems such as consensus algorithms

for camera sensors (Tron et al., 2011), camera calibration (Gwak et al., 2003) and attitude determination (Park et al., 2000). As $SE(3)$ is a *semi-direct* product of its constituent manifolds, optimization routines on $SE(3)$ face challenges due to the lack of *bi-invariant* Riemannian metrics on this product manifold and there have been efforts to address the issue (Tron and Vidal, 2014). Recently, our group has introduced a new representation for the space of rigid transformation as a *direct product* of the component manifolds where the group structures and the natural Riemannian metrics are compatible hence avoiding the aforementioned issues for optimization routines on $SE(3)$ (Vakili et al., 2014). Consequently, manifold optimization using direct product representation of the rigid transformation space is used as the problem formulation of the local optimization problem for the remainder of this chapter.

The *intrinsic differences* between constituent manifolds of local optimization necessitate special care for local optimization procedures. For instance, angles of rotation are periodic with period of 2π whereas translation quantities are aperiodic. Moreover, similar changes in the translation and rotation quantities can have significantly different implication on the orientation of a ligand. Specifically, one unit of change in translation along any direction results in exactly one unit of change of RMSD whereas the change in RMSD for one unit of change in rotation coordinates are dictated by *tensor of inertia* of the ligand and strongly depends on the shape of the ligand. Consequently, black box optimization algorithms may exhibit unexpected and invalid behavior such as sudden movements of the ligand in the presence of large gradient values.

Defining an *appropriate metric* on the product manifold can make rotation and translation parameters “similar”. Specifically, the metric can be used to appropriately scale rotation and translation coordinates for distance and gradient calculations.

In fact, finding a suitable metric on the space of rigid transformations has been recognized as a challenging and critical problem in the literature (Zefran et al., 1996) where the appropriate scaling of metric can lead to superlinear convergence for gradient based optimization algorithms (Mishra and Sepulchre, 2016). This chapter focuses on defining a metric that is closely related to RMSD where the metric provides a clear basis for scaling the rotational and translational coordinates. This natural scaling helps avoid pitfalls such as jerky movements of the ligand in gradient based local optimization when the gradient is excessively large or corrupted with noise. Moreover, it has been shown that optimization step sizes having bounded RMSD changes can lead to performance improvement for the local optimization procedure (Popov, 2015).

While the motivation for the work in this chapter is optimization with respect to rigid motions of an object in three-dimensional Euclidean space, the results are expressed more generally in terms of rigid motions of an object in n -dimensional Euclidean space.

3.2 Space of Rigid Transformations

In this section, two representations of the space of rigid body movement as Lie Groups are discussed and the differences between them are explained.

3.2.1 Lie Groups

A Lie group, G , is a differentiable manifold as well as a group such that the group product, $(g, g') \rightarrow gg'$, and group inverse $g \rightarrow g^{-1}$ operations are smooth mappings. For example, \mathbb{R}^n together with vector addition as the group operation is a (very simple) Lie group. The set of non-singular $n \times n$ real-valued matrices, denoted by

$GL(n; \mathbb{R})$, simply $GL(n)$ from now on, is a Lie group with matrix product as the group operation. The set of $n \times n$ real-valued orthonormal matrices with unit determinant, denoted by $SO(n)$, is a subgroup of $GL(n)$, and a Lie group.

As will be shown in this chapter, the Lie groups \mathbb{R}^n , associated with translations, and $SO(n)$, associated with rotations, form the main building blocks in describing rigid transformations. In the commonly used formulation, the so-called Special Euclidean Lie group $SE(n)$, they are used to define one type of rigid transformation. In another formulation they are used differently to present another type of rigid transformation.

3.2.2 Common Formulation of Rigid Transformations

The set of all $n \times n$ matrices that are orthonormal and have determinant of +1 is called the Special Orthogonal group:

$$SO(n) = \{R | R \in GL(n), RR^T = I, \det(R) = 1\} \quad (3.2)$$

$SO(n)$ represents the space of all rotations matrices in dimension n .

The Special Euclidean group or $SE(n)$ is the most commonly used representation for the space of rigid body movements. As a manifold, $SE(n)$ is defined as the direct product of $SO(n)$ and \mathbb{R}^n (namely $SO(n) \times \mathbb{R}^n$). On the other hand, as a group $SE(n)$ is a semi-direct product of $SO(n)$ and \mathbb{R}^n :

Let $g = (R, t)$ and $g' = (R', t')$, be two elements of $SO(n) \times \mathbb{R}^n$ then the group operation of $SE(n)$ is defined by :

$$g'g = (R'R, R't + t').$$

Making $SE(n)$ a semi-direct product group.

Here, we briefly review the notion of semi-direct product.

Semi-direct Product of Lie Groups

Let G and G' be two Lie groups with group operations denoted by $*$ and $'$ respectively. Then, the *direct product* of G and G' , denoted by $G \times G'$, is a group where the group operation is defined component-wise by:

$$(g_1, g'_1) \diamond (g_2, g'_2) = (g_1 * g_2, g'_1 *' g'_2) \quad (3.3)$$

on the product space $\{(g_1, g'_1); g_1 \in G, g'_1 \in G'\}$.

It can be easily verified that with the operation \diamond , $G \times G'$ is a group. It is also a product manifold, therefore, it is a Lie group. This Lie group is called the direct product of the component Lie groups.

To define a *semi-direct product* of G and G' , assume that a smooth action

$$h : G \times G' \rightarrow G', \quad (3.4)$$

is given. Define the operation \diamond' on $G \times G'$ by

$$(g_1, g'_1) \diamond' (g_2, g'_2) = (g_1 * g_2, g'_1 *' h(g_1, g'_2)) \quad (3.5)$$

Again, it can be verified that with the operation \diamond' , the product manifold $G \times G'$ is a group and therefore a Lie group. This Lie group is called the semi-direct product of the component Lie groups G and G' and denoted by

$$G \rtimes_h G' \quad (3.6)$$

By contrast to the direct product of G and G' where the group action is performed component-wise, in the case of the semi-direct product, a coupling between the com-

ponents of G and G' is created through the function h . This coupling can be a source of complications when dealing with semi-product groups.

Let $h : SO(n) \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ be defined by $h(R, t) = Rt$. Then we have: $SE(n) = SO(n) \rtimes_h \mathbb{R}^n$. In other words, $SE(n)$ is the semi-product of $SO(n)$ and \mathbb{R}^n defined by using the function h .

Action of $SE(n)$ on \mathbb{R}^n

Each member of $SE(n)$ defines an action on \mathbb{R}^n as follows (see, e.g., (Selig, 2005), Section 2.4). For $g \in SE(n)$, let $g : \mathbb{R}^n \rightarrow \mathbb{R}^n$ be defined by:

$$g(q) = Rq + t.$$

This mapping defines an action of the $SE(n)$ group on \mathbb{R}^n since we have:

$$g' \circ g(q) = g'(g(q)) = R'(Rq + t) + t' = R'Rq + R't + t' = g'g(q),$$

where \circ denotes composition of functions. Therefore,

$$g' \circ g = g'g.$$

As mentioned before, this action corresponds to a rigid body transformation of \mathbb{R}^n (in the stricter sense defined in (Muray et al., 1994), Chapter 2).

Homogeneous Representation of $SE(n)$

One representation of the Special Euclidean group, known as the *homogeneous* representation, is given by:

$$SE(n) \leftrightarrow \left\{ \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}, R \in SO(n), t \in \mathbb{R}^n \right\}, (R, t) \leftrightarrow \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \quad (3.7)$$

With this representation, the group operation of $SE(n)$ corresponds to the product of the matrices associated with elements of $SE(n)$. Furthermore, the homogeneous representation of an element of \mathbb{R}^n , say q , is given by:

$$\begin{bmatrix} q \\ 1 \end{bmatrix} \quad (3.8)$$

With this convention, the action of $(R, t) \in SE(n)$ on $q \in \mathbb{R}^n$ is simply the product of homogeneous representations of (R, t) and q , namely:

$$\begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} q \\ 1 \end{bmatrix} = \begin{bmatrix} Rq + t \\ 1 \end{bmatrix} \quad (3.9)$$

The Lie Algebras of $SO(n)$ and $SE(n)$, denoted by $so(n)$ and $se(n)$ respectively, are defined as follows in the homogeneous representation:

$$so(n) = \{\hat{\omega} | \hat{\omega} \in \mathbb{R}^{n \times n}, \hat{\omega}^T = -\hat{\omega}\},$$

$$se(n) = \{S | S = \begin{bmatrix} \hat{\omega} & v \\ 0 & 0 \end{bmatrix}, \omega \in so(n), v \in \mathbb{R}^n\} \quad (3.10)$$

3.2.3 New Formulation of Rigid Transformations

The semi-direct product structure of rigid motions presents challenges for generalizing optimization algorithms from individual manifolds $SO(n)$ and \mathbb{R}^n to $SE(n)$ (Gwak et al., 2003), (Tron and Vidal, 2014). This fact has motivated introducing a new representation for this space (Vakili et al., 2014).

In this representation, $SO(n) \times \mathbb{R}^n$, is considered both as a direct product manifold and a direct product group. Specifically, let $g = (R, t)$ and $g' = (R', t')$, be two elements of $SO(n) \times \mathbb{R}^n$ then the group operation of $SO(n) \times \mathbb{R}^n$ is defined by :

$$g'g = (R'R, t' + t).$$

As expected this more simple structure leads to significant simplifications:

Direct Product of Lie Groups

Let G_1, \dots, G_k be k Lie groups and let $G = G_1 \times \dots \times G_k$ be the direct product of the component groups. In this case the product group “inherits” many of the relevant structures from its component manifolds (for the simplest case, consider the n dimensional Euclidean space \mathbb{R}^n and its one dimensional components \mathbb{R}):

- G is a Lie group;
- Let $g_i \in G_i$ and let $T_{g_i}G_i$ be the tangent space to G_i at g_i , $i = 1, \dots, k$ and $g = (g_1, \dots, g_k) \in G$. Then,

$$T_gG = T_{g_1}G_1 \times \dots \times T_{g_k}G_k, \tag{3.11}$$

i.e., the tangent space to G at g is simply the direct product of the tangent spaces to the component groups;

- Consider the component exponential maps

$$\Phi_{v_i}(t, g_i) = g_i * \exp_i(tv_i), g_i \in G_i, v_i \in T_{g_i}G_i, i = 1, \dots, k \quad (3.12)$$

then, the exponential map of the product manifold, G , is simply evaluated component-wise, i.e.,

$$\Phi_v(t, g) = (g_1 \exp_1(tv_1), \dots, g_k \exp_k(tv_k)) \quad (3.13)$$

where $g = (g_1, \dots, g_k) \in G$ and $v = (v_1, \dots, v_k) \in T_g G$.

- The gradient of a function f on G can be computed “component-wise“ and the steepest descent algorithm can also be implemented using information on components.

Action of $SO(n) \times \mathbb{R}^n$ on $\mathbb{R}^n \times \mathbb{R}^n$

The novel element of the new representation is the action associated with this group. The action of $SO(n) \times \mathbb{R}^n$ is defined on $\mathbb{R}^n \times \mathbb{R}^n$ as follows. For $g \in SO(n) \times \mathbb{R}^n$, let $g : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n \times \mathbb{R}^n$ be defined by:

$$g(q, p) = (R(q - p) + p + t, p + t),$$

$(q, p \in \mathbb{R}^n)$.

In words, the action of g on the first component $q \in \mathbb{R}^n$ is to rotate q according to the rotation matrix R but with the “center of rotation” (i.e., the origin of the coordinate system) moved to p , and translate it by t . The action of g on the second component simply translates the point p by t . Equivalently, one can think that the action on the second component is of the same type as the action on the first component since $R(p - p) + p + t = p + t$. The following is an immediate result.

Proposition 1. *The above transformation defines an action of the group $SO(n) \times \mathbb{R}^n$ on $\mathbb{R}^n \times \mathbb{R}^n$.*

Proof. Let $g = (R, t)$ and $g' = (R', t')$. Then,

$$\begin{aligned}
 g'(g(q, p)) &= g'(R(q - p) + p + t, p + t) \\
 &= (R'(R(q - p) + p + t - (p + t)) \\
 &\quad + p + t + t', p + t + t') \\
 &= (R'R(q - p) + p + t + t', p + t + t') \\
 &= (g' * g)(q, p).
 \end{aligned}$$

□

Furthermore, for any $p \in \mathbb{R}^n$, the action of $SO(n) \times \mathbb{R}^n$ on $\mathbb{R}^n \times \mathbb{R}^n$ is a rigid body transformation of the first component \mathbb{R}^n .

Let $\pi_i : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ ($i = 1, 2$) be projections on the first and second coordinate ($\pi_1(q, p) = q$, $\pi_2(q, p) = p$). For any fixed $p \in \mathbb{R}^n$, let

$$g_p : \mathbb{R}^n \rightarrow \mathbb{R}^n \times \mathbb{R}^n,$$

be defined by $g_p(q) = g(q, p)$. Then, we have the following proposition:

Proposition 2. *For any p ,*

$$\pi_1 \circ g_p : \mathbb{R}^n \rightarrow \mathbb{R}^n,$$

is a rigid body transformation of \mathbb{R}^n .

Proof. Fix $p \in \mathbb{R}^n$. Let $q, q' \in \mathbb{R}^n$, then

$$\|\pi_1 \circ g_p(q) - \pi_1 \circ g_p(q')\| =$$

$$\begin{aligned} \|R(q - p) + p + t - (R(q - p) + p + t)\| &= \\ \|R(q - q')\| &= \|q - q'\|. \end{aligned}$$

The last equality is due to the fact that R is a rotation matrix. Following the definition in (Muray et al., 1994), we also need to show that $\pi_1 \circ g_p$ is orientation-preserving. In other words, it sends right-handed coordinate frames to right-handed coordinate frames. We show that the action of $\pi_1 \circ g_p$ on vectors in \mathbb{R}^n is the same as the action of $SE(n)$ on such vectors. Therefore, the result follows from the fact that $SE(n)$ is orientation-preserving ((Muray et al., 1994), Proposition 2.7).

Let $q, q' \in \mathbb{R}^n$ as above. Then, under the $\pi_1 \circ g_p$ transformation, the vector $q' - q$ is transformed into the vector $\pi_1 \circ g_p(q') - \pi_1 \circ g_p(q)$. We showed above that the latter is equal to $R(q - q')$. Under $SE(n)$ transformation and $g = (R, t)$ $q' - q$ is transformed into the vector $Rq' + t - (Rq + t) = R(q' - q)$. Hence the proof that $\pi_1 \circ g_p$ is orientation-preserving. \square

Homogeneous Representation of $SO(n) \times \mathbb{R}^n$

The *homogeneous* representation of a group element corresponding to a rotation and translation pair (R, t) is defined as follows (Mirzaei et al., 2014):

$$\begin{bmatrix} R & I - R & t \\ 0 & I & t \\ 0 & 0 & 1 \end{bmatrix} \tag{3.14}$$

One can verify that by combining two elements g_1, g_2 , the new element g_3 is :

$$\begin{aligned}
 g_1 * g_2 &= \begin{bmatrix} R_1 & I - R_1 & t_1 \\ 0 & I & t_1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_2 & I - R_2 & t_2 \\ 0 & I & t_2 \\ 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} R_1 R_2 & I - R_1 R_2 & t_1 + t_2 \\ 0 & I & t_1 + t_2 \\ 0 & 0 & 1 \end{bmatrix} = g_3
 \end{aligned} \tag{3.15}$$

The above implies that the equivalent rotation and translation elements are:

$$R_3 = R_1 R_2, \quad t_3 = t_1 + t_2 \tag{3.16}$$

Similar to $SE(n)$, the action of an element of the new representation (R, t) on a point $(q, p) \in \mathbb{R}^{2n}$ can be expressed as a matrix multiplication where the point (q, p) is appended with 1:

$$\begin{bmatrix} R & I - R & t \\ 0 & I & t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} q \\ p \\ 1 \end{bmatrix} = \begin{bmatrix} R(q - p) + p + t \\ p + t \\ 1 \end{bmatrix} \tag{3.17}$$

3.2.4 Comparison of the Representations

When using the action of the $SE(n)$ group, the center of rotation is fixed at the origin of the coordinate system whereas in the case of $SO(n) \times \mathbb{R}^n$ group, one has an arbitrary

choice of the center. Furthermore, using $SO(n) \times \mathbb{R}^n$ after each transformation of an object, the center of the rotation is translated the same amount as the object so that the relative distance between the object and the center of rotation is preserved. Consequently, no matter where the object lies in the space, the effect of a rotation matrix will be the same. This is not the case for the $SE(n)$ group where a coupling is formed between successive rotations and translations. For an illustration of the difference between the representations in two dimensions, see figure 3.2. In this figure, a triangle is moved from an initial positions with same amount of rotation and translations using the two representations. Note how moving center of rotation in each step direct product representation makes a difference in the final position of the triangle.

To gain more insight, consider an object undergoing rigid transformations. Depending on where the object is located relative to the coordinate axis origin, the effect of the same rotation matrix will be different. Specifically, the amount of orientation change will be the same but the distance the object travels depends on how far the object is from the origin. Moreover, consider the object going through two successive transformations g_1 and g_2 in order. After performing g_1 , the object distance from the origin of the coordinate axis is changed and hence the second rotation R_2 effect on the object is different compared to when the object was at the initial position.

A critical feature of the new representation of rigid body transformations is that, by contrast to the $SE(n)$ formulation, translational moves and rotational moves are decoupled. For example, let $g = (I, t)$, i.e., translation only by t and $g' = (R, 0)$, i.e., rotation only by R . Then it can be easily seen that in $SE(n)$,

$$gg' \neq g'g,$$

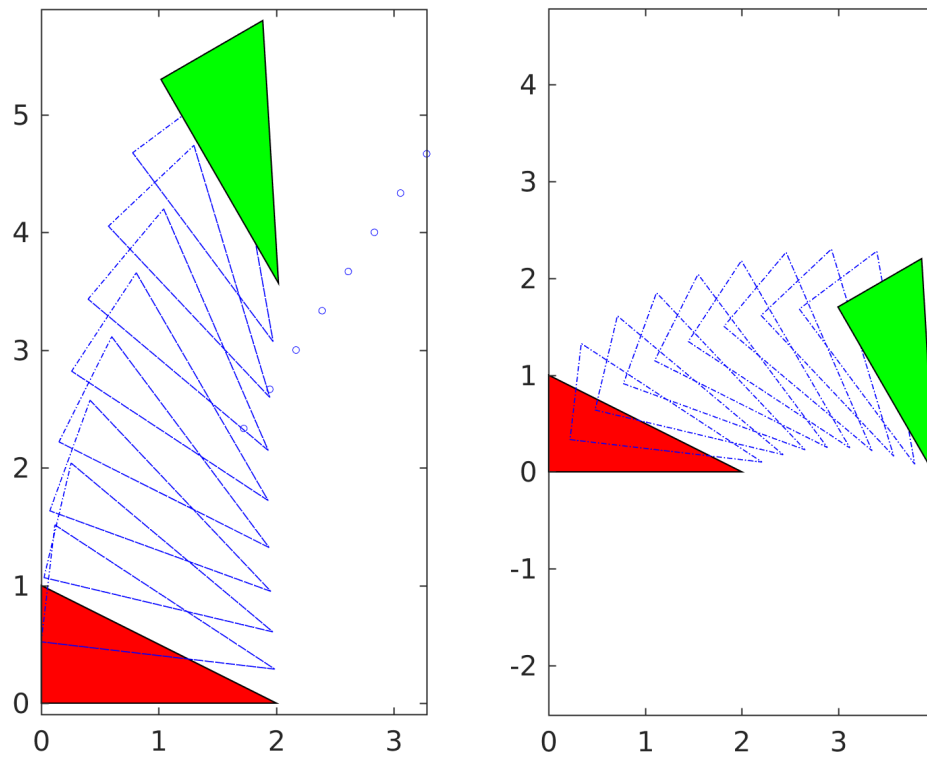


Figure 3.2: A triangle is moved using a rigid transformation in multiple steps from an initial position. On the left, the center of rotation is translated in each step the same amount as the triangle. On the right, the center of rotation is fixed at the origin of the coordinate axis throughout the movement. The steps of movement are shown in blue dashed lines and the moving centers of rotation are shown using the small blue circles.

where as in $SO(n) \times \mathbb{R}^n$

$$g * g' = g' * g.$$

The above can be verified by considering the homogeneous representation of g and g' in $SE(n)$ and $SO(n) \times \mathbb{R}^n$, respectively.

Another interesting note for the new representation is an interpretation of the action of an element of the group on $\mathbb{R}^n \times \mathbb{R}^n$. The space $\mathbb{R}^n \times \mathbb{R}^n$ can be thought of the space of end points of vectors $(x_s, x_e) \in \mathbb{R}^n \times \mathbb{R}^n$ starting at x_s and ending at x_e . The action of an element of the new representation (R, t) on a point (x_s, x_e) is to rotate the whole vector around x_s with R and translate the vector with t :

$$\begin{bmatrix} x_s \\ x_e \end{bmatrix} \rightarrow \begin{bmatrix} x_s + t \\ R(x_e - x_s) + x_s + t \end{bmatrix} \quad (3.18)$$

3.3 RMSD compatible Riemannian metric

As pointed out earlier, the motivation for the introduction of a new Riemannian metric on the Lie group of rotations is to take the differences between the effect of translation and rotation on displacing a ligand into account and to make them similar. Again, as mentioned earlier, the new metric will be introduced in the more general setting of rigid displacement of an object in n -dimensional \mathbb{R}^n .

Consider an object consisting of k distinct points in \mathbb{R}^n (e.g., a molecule consisting of k atoms in \mathbb{R}^3). Let $\mathbf{q}_1, \dots, \mathbf{q}_k$ denote the locations of the k points in \mathbb{R}^n . Assume that as a result of a *rigid displacement* of the object the points are moved to $\mathbf{q}'_1, \dots, \mathbf{q}'_k$. As mentioned before, the RMSD in this case is given by:

$$d^2 = \frac{1}{k} \sum_{i=1}^k \|\mathbf{q}'_i - \mathbf{q}_i\|^2.$$

Here, rigid motions given by translations and rotations about a chosen “center of rotation” $\mathbf{p} \in \mathbb{R}^n$ are considered. Let $\mathbf{t} \in \mathbb{R}^n$ be the translation vector and $R \in SO(n)$ the rotation matrix. Then, we have

$$\mathbf{q}'_i = R(\mathbf{q}_i - \mathbf{p}) + \mathbf{p} + \mathbf{t}.$$

In this case, the RMSD can be expressed in terms of R , \mathbf{t} , and \mathbf{p} as follows (transpose is denoted by T):

$$\begin{aligned} d^2 &= \frac{1}{k} \sum_{i=1}^k \|\mathbf{q}'_i - \mathbf{q}_i\|^2 \\ &= \frac{1}{k} \sum_{i=1}^k \|R(\mathbf{q}_i - \mathbf{p}) + \mathbf{p} + \mathbf{t} - \mathbf{q}_i\|^2 \\ &= \frac{1}{k} \sum_{i=1}^k ((R - I)(\mathbf{q}_i - \mathbf{p}) + \mathbf{t})^T ((R - I)(\mathbf{q}_i - \mathbf{p}) + \mathbf{t}) \\ &= \frac{1}{k} \sum_{i=1}^k (\mathbf{q}_i - \mathbf{p})^T (R - I)^T (R - I) (\mathbf{q}_i - \mathbf{p}) + 2\mathbf{t}^T (R - I) \left(\frac{1}{k} \sum_{i=1}^k (\mathbf{q}_i - \mathbf{p}) \right) + \|\mathbf{t}\|^2. \end{aligned} \tag{3.19}$$

Let $\mathbf{c} = \frac{1}{k} \sum_{i=1}^k \mathbf{q}_i$ (\mathbf{c} is the center of geometry of the object), and $\mathbf{v}_i = \mathbf{q}_i - \mathbf{p}$. Then, $RMSD^2$ can be written as

$$\begin{aligned} d^2 &= \frac{1}{k} \sum_{i=1}^k \mathbf{v}_i^T (R - I)^T (R - I) \mathbf{v}_i + 2\mathbf{t}^T (R - I) (\mathbf{c} - \mathbf{p}) + \|\mathbf{t}\|^2 \\ &= \frac{1}{k} \sum_{i=1}^k ((R - I)\mathbf{v}_i)^T (R - I)\mathbf{v}_i + 2\mathbf{t}^T (R - I) (\mathbf{c} - \mathbf{p}) + \|\mathbf{t}\|^2 \\ &= \frac{1}{k} \sum_{i=1}^k \|(R - I)\mathbf{v}_i\|^2 + 2\mathbf{t}^T (R - I) (\mathbf{c} - \mathbf{p}) + \|\mathbf{t}\|^2. \end{aligned} \tag{3.20}$$

It is clear from the above expression for RMSD that if the center of rotation is selected to be the center of geometry of the object, then RMSD decomposes into two

distinct parts, one depending only on the rotation matrix R , and the other only on the translation vector \mathbf{t} . In other words, if $\mathbf{p} = \mathbf{c}$, then

$$d^2 = \frac{1}{k} \sum_{i=1}^k \|(R - I)\mathbf{v}_i\|^2 + \|\mathbf{t}\|^2 = d_1 + d_2.$$

Where

$$d_1 = \frac{1}{k} \sum_{i=1}^k \|(R - I)\mathbf{v}_i\|^2, \quad d_2 = \|\mathbf{t}\|^2 \quad (3.21)$$

A closer look at d_1 results in the following:

Proposition 3. *Let $Q = \frac{1}{k} \sum_{i=1}^k \mathbf{v}_i \mathbf{v}_i^T$, and $B = (R - I)^T(R - I)$. Then, we have*

$$d_1^2 = \frac{1}{k} \sum_{i=1}^k \|(R - I)\mathbf{v}_i\|^2 = \text{Trace}(BQ).$$

Proof. Let $\mathbf{v} \in \mathbb{R}^n$ and $R \in SO(n)$. Then, given that for any vector $\mathbf{w} \in \mathbb{R}^n$ $\|\mathbf{w}\|^2 = \text{Trace}(\mathbf{w}\mathbf{w}^T)$, we have

$$\begin{aligned} \|(R - I)\mathbf{v}\|^2 &= \text{Trace}((R - I)\mathbf{v}((R - I)\mathbf{v})^T) \\ &= \text{Trace}((R - I)\mathbf{v}\mathbf{v}^T(R - I)^T) \\ &= \text{Trace}((R - I)^T(R - I)\mathbf{v}\mathbf{v}^T) \\ &= \text{Trace}(B\mathbf{v}\mathbf{v}^T). \end{aligned} \quad (3.22)$$

In the second to the last identity above the invariance property of the trace under circular permutation is used, namely the fact that $\text{Trace}(RST) = \text{Trace}(TRS)$.

Therefore,

$$\begin{aligned}
 d_1^2 &= \frac{1}{k} \sum_{i=1}^k \|(R - I)\mathbf{v}_i\|^2 \\
 &= \frac{1}{k} \sum_{i=1}^k \text{Trace}(B\mathbf{v}_i\mathbf{v}_i^T) \\
 &= \text{Trace}\left(B \left(\frac{1}{k} \sum_{i=1}^k \mathbf{v}_i\mathbf{v}_i^T\right)\right) \\
 &= \text{Trace}(BQ)
 \end{aligned} \tag{3.23}$$

Efficient computation of RMSD. Sidestepping the main concern of this section, a corollary of the above derivation is highlighted.

In light of the above derivations, we can write

$$d^2 = \text{Trace}(BQ) + 2\mathbf{t}^T(R - I)(\mathbf{c} - \mathbf{p}) + \|\mathbf{t}\|^2.$$

There are practical instances when one is interested in computing RMSD values for multiple (and possibly many) rigid transformations specified by different rotation matrices and translation vectors. Using the expression above for RMSD, the terms Q and \mathbf{c} can be computed *only once* at the beginning of the process. Subsequent computations of RMSD values will be of order n^3 as opposed to kn^2 where n and k are the dimension and the number of points in the object, respectively. This is of significant importance where the number of points is relatively large. For instance in 3D for a protein with 3000 atoms, $n = 3$ and $k = 3000$ leading to a speed up of the order 1000.

Returning to the main concern of this section, the first step towards defining a new metric is the following:

Proposition 4. Let $Q = \frac{1}{k} \sum_{i=1}^k \mathbf{v}_i \mathbf{v}_i^T$, be as defined above. Then, (i) Q is always a positive semi-definite matrix; and (ii) If $\mathbf{v}_1, \dots, \mathbf{v}_k$ span \mathbb{R}^n , then Q is a positive definite matrix.

Proof.

(i) Note that

$$Q^T = \left(\frac{1}{k} \sum_{i=1}^k \mathbf{v}_i \mathbf{v}_i^T \right)^T = \frac{1}{k} \sum_{i=1}^k \mathbf{v}_i \mathbf{v}_i^T = Q.$$

Therefore, Q is a symmetric matrix. Now, let $\mathbf{q} \in \mathbb{R}^n$; then

$$\begin{aligned} \mathbf{q}^T Q \mathbf{q} &= \mathbf{q}^T \left(\frac{1}{k} \sum_{i=1}^k \mathbf{v}_i \mathbf{v}_i^T \right) \mathbf{q} \\ &= \frac{1}{k} \sum_{i=1}^k (\mathbf{q}^T \mathbf{v}_i) (\mathbf{v}_i^T \mathbf{q}) \\ &= \frac{1}{k} \sum_{i=1}^k \langle \mathbf{q}, \mathbf{v}_i \rangle \langle \mathbf{v}_i, \mathbf{q} \rangle \\ &= \frac{1}{k} \sum_{i=1}^k \langle \mathbf{q}, \mathbf{v}_i \rangle^2 \geq 0 \end{aligned} \tag{3.24}$$

$\langle \mathbf{a}, \mathbf{b} \rangle$ denotes the usual inner product of vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$. Therefore, it is shown that in general Q is positive semi-definite.

(ii) Now assume $\mathbf{v}_1, \dots, \mathbf{v}_k$ span \mathbb{R}^n and let $\mathbf{q} \in \mathbb{R}^n$ be a non-zero vector. Then, we have

$$\mathbf{q}^T Q \mathbf{q} = \frac{1}{k} \sum_{i=1}^k \langle \mathbf{q}, \mathbf{v}_i \rangle^2$$

Given that $\mathbf{q} \neq \mathbf{0}$, not all inner products $\langle \mathbf{q}, \mathbf{v}_i \rangle$ can be equal to zero, otherwise $\mathbf{v}_1, \dots, \mathbf{v}_k$ will not span \mathbb{R}^n . Therefore,

$$\mathbf{q}^T Q \mathbf{q} = \frac{1}{k} \sum_{i=1}^k \langle \mathbf{q}, \mathbf{v}_i \rangle^2 > 0,$$

and Q is positive definite.

3.3.1 New Riemannian metric

Let $\mathbf{v} \in \mathbb{R}^n$ be a vector corresponding to a direction of translation and let $g(t) = \mathbf{v}t$, $t \geq 0$ denote the half-line in the direction of \mathbf{v} corresponding to different magnitudes of possible translations in the direction of \mathbf{v} . Note that \mathbf{v} can also be thought of as an element of the tangent space to the Lie group \mathbb{R}^n and $g(t) = \mathbf{v}t$ $t \geq 0$ as a half-line in the direction of \mathbf{v} on this tangent space. The projection of \mathbf{v} and $g(t)$ for any t on the Lie group \mathbb{R}^n are simply the same quantities.

For any translation vector $g(t) = \mathbf{v}t$, the RMSD of the displaced object relative to its original location, denoted by $f(t)$, is given by $\|t\mathbf{v}\|$. It is worth noting that $f(t)$ is *completely independent* of the shape of the object (ligand). Therefore, for "small" t , the RMSD of the displacement is given by

$$f(t) \approx t\|\mathbf{v}\|.$$

Turning to rotations, recall that the tangent space of $SO(n)$ at identity consisting of skew symmetric matrices $[\omega]$ is isomorphic to the n -dimensional Euclidean space \mathbb{R}^n (the same is true at any other point of the group). In what follows, let $\omega \in \mathbb{R}^n$ be the vector associated with $[\omega] \in so(n)$. Let $t\omega$ $t \geq 0$ denote the half-line in the direction of ω . Then, $g(t) = \exp^{t[\omega]}$ represent rotations in $SO(n)$ for different values of t . Let $f(t)$, as above, denote the RMSD of the displaced object relative to its original location when rotated by rotation $g(t)$. Then, equation (3.20) implies

$$f(t) = \sqrt{\frac{1}{k} \sum_{i=1}^k \mathbf{v}_i^T (R - I)^T (R - I) \mathbf{v}_i} = \sqrt{\frac{1}{k} \sum_{i=1}^k \mathbf{v}_i^T (2I - (R + R^T)) \mathbf{v}_i} \quad (3.25)$$

where R is the rotation corresponding to $\exp^{t[\omega]}$. The Taylor expansion of the rotation at $t = 0$ is given by

$$R = I + t[\omega] + \frac{(t[\omega])^2}{2!} + \frac{(t[\omega])^3}{3!} + o(t^3) \quad (3.26)$$

Therefore:

$$\begin{aligned} 2I - (R + R^T) &= -2 \left(\frac{(t[\omega])^2}{2!} + \frac{(t[\omega])^4}{4!} + \frac{(t[\omega])^6}{6!} + o(t^6) \right) \\ &= - (t[\omega])^2 + o(t^2) \end{aligned} \quad (3.27)$$

The derivative of $f(t)$ at $t = 0$ can be computed as:

$$\begin{aligned} \left. \frac{df(t)}{dt} \right|_{t=0} &= \lim_{h \rightarrow 0} \frac{f(h) - f(0)}{h} = \lim_{h \rightarrow 0} \frac{f(h)}{h} = \sqrt{\frac{1}{k} \sum_{i=1}^k \frac{\mathbf{v}_i^T (- (h[\omega])^2 + o(h^2)) \mathbf{v}_i}{h}} \\ &= \sqrt{\frac{1}{k} \sum_{i=1}^k \mathbf{v}_i^T (- ([\omega])^2) \mathbf{v}_i} = \sqrt{\frac{1}{k} \sum_{i=1}^k \mathbf{v}_i^T ([\omega]^T [\omega]) \mathbf{v}_i} \\ &= \sqrt{\frac{1}{k} \sum_{i=1}^k \text{Trace}(\mathbf{v}_i^T [\omega]^T [\omega] \mathbf{v}_i)} = \sqrt{\frac{1}{k} \sum_{i=1}^k \text{Trace}([\omega] \mathbf{v}_i \mathbf{v}_i^T [\omega]^T)} \\ &= \sqrt{\text{Trace} \left([\omega] \frac{1}{k} \sum_{i=1}^k \mathbf{v}_i \mathbf{v}_i^T [\omega]^T \right)} = \sqrt{\text{Trace}([\omega] Q [\omega]^T)} \\ &= \sqrt{\text{Trace}([\omega]^T Q [\omega])} \end{aligned} \quad (3.28)$$

It can easily be verified that

$$\begin{aligned} \text{Trace}([\omega]^T Q [\omega]) &= \|\omega\|^2 \text{Trace}(Q) - \omega^T Q \omega = \omega^T (\text{Trace}(Q) I - Q) \omega \\ &= \omega^T J \omega \end{aligned} \quad (3.29)$$

Where $J = (\text{Trace}(Q) I - Q)$ and I denotes the identity matrix. Furthermore, it can also easily be verified that

$$\text{Trace}(Q) = \frac{1}{k} \sum_{i=1}^k \|\mathbf{v}_i\|^2.$$

In the following, it is shown that under very mild assumptions J is a positive definite matrix and can be used to define a new metric on $so(n)$:

Proposition 5. *Let $J = \frac{1}{k} \sum_{i=1}^k \|\mathbf{v}_i\|^2 I - Q$ where $Q = \frac{1}{k} \sum_{i=1}^k \mathbf{v}_i \mathbf{v}_i^T$, as defined above. Then, (i) J is always a positive semi-definite matrix; (ii) If not all $\mathbf{v}_1, \dots, \mathbf{v}_k$ are collinear, then J is a positive definite matrix.*

Proof.

(i) We have

$$J^T = \frac{1}{k} \sum_{i=1}^k \|\mathbf{v}_i\|^2 I - Q^T = \frac{1}{k} \sum_{i=1}^k \|\mathbf{v}_i\|^2 I - Q = J$$

Where the second equality is using the fact that Q is symmetric according to Proposition 4. Therefore, J is symmetric. Now, let $\mathbf{q} \in \mathbb{R}^n$; we have

$$\begin{aligned} \mathbf{q}^T J \mathbf{q} &= \left(\frac{1}{k} \sum_{i=1}^k \|\mathbf{v}_i\|^2 \|\mathbf{q}\|^2 - \mathbf{q}^T Q \mathbf{q} \right) = \left(\frac{1}{k} \sum_{i=1}^k \|\mathbf{v}_i\|^2 \|\mathbf{q}\|^2 - \frac{1}{k} \sum_{i=1}^k \mathbf{q}^T \mathbf{v}_i \mathbf{v}_i^T \mathbf{q} \right) \\ &= \frac{1}{k} \sum_{i=1}^k (\|\mathbf{v}_i\|^2 \|\mathbf{q}\|^2 - \langle \mathbf{v}_i, \mathbf{q} \rangle^2) \end{aligned} \quad (3.30)$$

Where all the summation terms are nonnegative according to Cauchy-Schwarz inequality. Hence, J is positive semi-definite.

(ii) As mentioned before, all the summation terms are nonnegative and they can be equal to zero if \mathbf{q} and \mathbf{v}_i are collinear. Consequently, if there exist at least two \mathbf{v}_i not parallel to each other then \mathbf{q} cannot be collinear with all \mathbf{v}_i and the summation will be positive and J positive definite.

In what follows it is assumed that $\mathbf{v}_1, \dots, \mathbf{v}_k$ are not all collinear, then according to Proposition 5 J is a positive definite matrix and we can define a new metric on

$so(3) \sim \mathbb{R}^3$ by

$$\|\!\| \!\|^2 = \!\!^T J \!\!$$

So far, the new metric is defined on $so(n)$, the Lie algebra of $SO(n)$. Following the common approach in Lie groups, the metric is extended to the tangent spaces of all elements of $SO(n)$ via left translation as follows: Let $R \in SO(n)$ be a rotation matrix. R defines a left translation on $SO(n)$, denoted by $L_R : SO(n) \rightarrow SO(n)$ where $L_R(R') = RR'$. Note that the identity matrix is mapped to the rotation matrix R , i.e., $L_R(I) = R$. This mapping induces a linear mapping from the tangent space to $SO(n)$ at the identity $T_I SO(n)$ to the tangent space at R , $T_R SO(n)$, given by $[\omega] \rightarrow R[\omega]$. Furthermore, the inner product defined on $T_I SO(n)$ via the positive definite matrix J is extended to an inner product on $T_R SO(n)$ as follows. Let ω_1 and ω_2 be two vectors on the Euclidean space associated with $T_R SO(n)$, then

$$\begin{aligned} \langle [\omega_1], [\omega_2] \rangle_R &= \langle R^{-1}[\omega_1], R^{-1}[\omega_2] \rangle_I \\ &= (R^{-1}[\omega_1])^T J R^{-1}[\omega_2] \\ &= [\omega_1]^T R J R^T [\omega_2] \end{aligned} \tag{3.31}$$

Therefore, the inner product on $T_R SO(n)$ is defined by the matrix $R J R^T$. This inner product is consistent with RMSD changes after a rotation of the object by the rotation matrix R .

As already mentioned, for small t the RMSD change of the object due to translation in the direction of vector \mathbf{v} is approximately equal to $t\|\mathbf{v}\|$. With the new metric on the space of rotations, for small t the RMSD change of the object due to rotation in the direction of vector $\!\!$ is approximately equal to $t\|\!\!\|_J$.

This signifies that infinitesimally at $t = 0$ the rate of RMSD change can be rendered analogous to that of translation by changing the metric to J on $SO(n)$ tangent space.

Assuming $n = 3$, one can expand J to get:

$$\begin{bmatrix} \sum_{i=1}^k (y_k^2 + z_k^2) & -\sum_{i=1}^k (x_k y_k) & -\sum_{i=1}^k (x_k z_k) \\ -\sum_{i=1}^k (x_k y_k) & \sum_{i=1}^k (x_k^2 + z_k^2) & -\sum_{i=1}^k (y_k z_k) \\ -\sum_{i=1}^k (x_k z_k) & -\sum_{i=1}^k (y_k z_k) & \sum_{i=1}^k (y_k^2 + z_k^2) \end{bmatrix} \quad (3.32)$$

It is worth mentioning there is a close relation between J defined above and the *tensor of inertia* in Kinematics field. Tensor of inertia determines the amount of torque needed to initiate angular acceleration along a given axis, similar to how mass dictates the amount of force needed to create linear acceleration. The tensor of inertia depends how the body mass is distributed along principle axes. A similar metric in chapter 4 of (Bullo and Lewis, 2005) is defined for calculating kinetic energy of the mechanical systems.

3.3.2 Future Directions

In the previous section, a ligand structure dependent metric has been introduced and it has been shown to be positive semi-definite and how to perform fast RMSD calculations using this metric. Furthermore, This metric defines a “proper” scaling of rotation and translation parameters, making them infinitesimally “similar”. Consequently, The RMSD compatible measure leads to defining a consistent metric over manifold where the norm on the tangent space changes smoothly. There are possible implication and direction for these derivations:

- Gradient scaling: It can be shown that by changing the norm on a manifold, the gradient should be adjusted accordingly. It is then a question whether the new metric, and hence the new gradient calculation results in more appropriate directions of descent for gradient-based optimization algorithms. Therefore, it

seems appropriate to have practical testing cases, especially in terms of protein docking applications, for evaluating the effectiveness of algorithms with the new gradient calculations.

Specifically, It would be very interesting to see whether the new natural metric will also enhance the performance of the current state of the art optimization algorithms in Protein Docking. As mentioned before, the scoring functions used in docking are highly nonlinear and noisy where the value depends on pairwise interaction receptor and ligand atoms. Therefore, the gradient information is local in terms of RMSD change and by preventing the algorithm from taking large RMSD steps, one can avoid unbounded changes in the scoring values and gradients. Moreover, one can set lower bound on the steps sizes of the gradient-based optimization algorithms in terms of RMSD, to avoid excessively slow convergence rate during the final steps of the algorithms.

Chapter 4

Semi Definite Subspace Underestimation

4.1 Introduction

As mentioned in the introduction, protein docking is regarded as a very challenging problem in structural biology due to the complexity of the energy landscape of protein-protein or small-protein interactions (Huang et al., 2013). This complexity stems from the fact that the energy function is highly non-convex and composed of multiple force-field energy terms (such as the Lennard-Jones potential, solvation, hydrogen bonding, electrostatics, etc.) acting in different space scales and resulting in a multi-frequency behavior of the various energy terms. Therefore, the energy function exhibits multiple deep funnels and extremely many local minima over its multidimensional domain.

To solve this challenging optimization problem, the state-of-the-art docking protocols employ a two-stage approach. At the first stage typically a simplified energy function is used and an enormous number of samples is generated on a grid in the conformational space corresponding to docked receptor-ligand conformations and evaluated efficiently using specialized methods such as Fast Fourier Transforms (FFT). These conformations are then sorted by their scores (energy values), and the top few thousands with the lowest energy are retained for further processing. At the second stage of docking protocols, low energy conformations are *refined* by moving off-grid and utilizing more elaborate energy functions. The work in this chapter focuses on this *refinement stage*, see, e.g., (Heo et al., 2016). One of the distinguishing features of

this work is that it *does not* assume any prior knowledge about the native structure. In fact, the inputs to the algorithm are the outputs of the PIPER docking software, which are the top globally sampled conformations in terms of energy. In this work, the performance of the current refinement protocol is assessed by considering the number of good quality solutions in the refined ensemble.

The *refinement problem* outlined, inherits the complex structure of the the binding energy landscape. Approaches that have been considered almost invariably involve efficient sampling and methods that attempt to “smooth” the energy function. A successful strategy is to use *Monte Carlo*-based sampling (Gray et al., 2003). An alternative method that resamples around low-energy PIPER structures has also been proposed (Mamonov et al., 2016). A host of methods seek to leverage the *funnel-like* shape of the energy function (McCammon, 1998),(Zhang et al., 1999),(Tovchigrechko and Vakser, 2001). In fact, similar strategies have been used in protein folding (Leopold et al., 1992), (Bryngelson et al., 1995),(Dill, 1999),(Tsai et al., 1999). The binding energy funnel is restricted to a neighborhood of the native complex (Selzer et al., 2001) and there is a free energy gradient toward the native state. However, the funnel is rough, giving rise to many local minima (Trosset and Scheraga, 1998) that correspond to encounter complexes, some of which may be visited along a particular association pathway (Camacho et al., 1999),(Camacho et al., 2000b). Fig. 4-1 sketches the funnel-like structure of the energy landscape, allowing for the possibility of multiple local funnels.

Underestimation. An early algorithm designed for protein folding, the *Convex Global Underestimator (CGU)* method (Phillips et al., 2001), introduced the idea of using an approximation of the envelope spanned by the local minima of the energy function in the form of *convex canonical quadratic* underestimators. CGU, however, used a restricted class of underestimators (Paschalidis et al., 2007), limiting its ef-

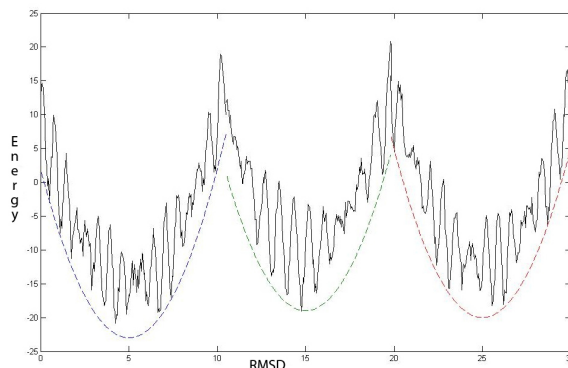


Figure 4.1: An illustration of low-energy clusters of complexes and their underestimators that outline the broad local funnel.

fectiveness. The *Semi-Definite programming-based Underestimation (SDU)* method (Paschalidis et al., 2007), (Shen et al., 2008) uses the same approach as CGU but it considers the class of “general” convex quadratic functions to underestimate, in addition to introducing an exploration strategy biased by the underestimator.

This chapter is built upon SDU algorithm (Paschalidis et al., 2007), (Shen et al., 2008) and a number of generalizations are proposed. First, and following the earlier preliminary work (Nan et al., 2014), the more general class of *SOS-convex polynomial functions* for underestimation is considered. Polynomial functions are more flexible than quadratic functions used in the aforementioned methods (Phillips et al., 2001), (Paschalidis et al., 2007), (Shen et al., 2008) and can more tightly approximate a funnel.

A second generalization is the ability to handle multiple local funnels in the original cluster presented for refinement (e.g., as in Fig. 4.1). This is important because by deriving a single underestimator (as in (Nan et al., 2014)), one will tend to “average” a complex energy landscape and produce a minimum of the underestimator that may not correspond to a low-energy funnel basin. In this work, this issue is resolved by establishing an effective exploration procedure using density-based clustering as follows. First, a density-based clustering algorithm is run on the set of

(PIPER) structures which are the inputs to the refinement protocol. This phase eliminates outliers and low-density regions of the conformational space, resulting in multiple sub-clusters whose size is greater than a pre-specified threshold. Then, one underestimator per sub-cluster is constructed which enables approximating and exploring each sub-cluster separately. Finally, all the sampled conformations from all clusters are combined, and the low-energy conformations are picked as the output of the refinement protocol.

Dimensionality reduction. An important question in underestimation is to determine the right multi-dimensional space in which underestimation takes place. According to prior experience for many complexes, underestimation in the entire 6D space of conformational variables (translations and rotations of the ligand with respect to the receptor) may not be effective and produce underestimators whose minimum is outside the range of the cluster. This is due to “singularities” of the energy landscape resulting in energy being very steep along some directions and flat along others.

Realizing this, in the original SDU (Paschalidis et al., 2007), (Shen et al., 2008) the center-to-center distance of receptor and ligand from the 6D parameterization of the space is removed as this dimension does not exhibit any significant variation over the ensemble of input samples, thus, suggesting a very narrow energy funnel along this dimension. These initial attempts led to a more fundamental re-assessment of the space in which underestimation must take place. In a previous work (Kozakov et al., 2014), it was discovered that the near-native cluster in protein-protein complexes exhibits reduced dimensionality, suggesting that *proteins associate along preferred pathways*, similar to sliding of a protein along DNA in the process of protein-DNA recognition. The landscape features were extracted via *Principal Component Analysis (PCA)* using two distinct energy functions, one derived from PIPER sampling (Kozakov et al., 2006) and the other using RosettaDock (Gray et al., 2003). In both cases, it was

found that most of the variability (more than 75%) in the cluster can be explained by 3 (and sometimes 2) eigenvectors, suggesting that the energy landscape consists of a *permissive subspace* spanned by the 2 or 3 eigenvectors with the largest eigenvalues and a *restrictive landscape* spanned by the remaining eigenvectors, respectively. Fig. 4.2 illustrates the landscape of the 2YVJ complex. It plots the distributions of Interface RMSD (root mean square deviation of interface atoms from the native) in Å and energy values based on structures generated by PIPER along the 5 eigenvectors produced by PCA, plotted from top to bottom in decreasing corresponding eigenvalue. The analysis is performed in the space of rigid transformations where the center-to-center coordinate is dropped. Dark blue diamonds indicate low energy data points used for the PCA. Notice how the variability of the data points decreases from top (very wide) to bottom (very narrow).

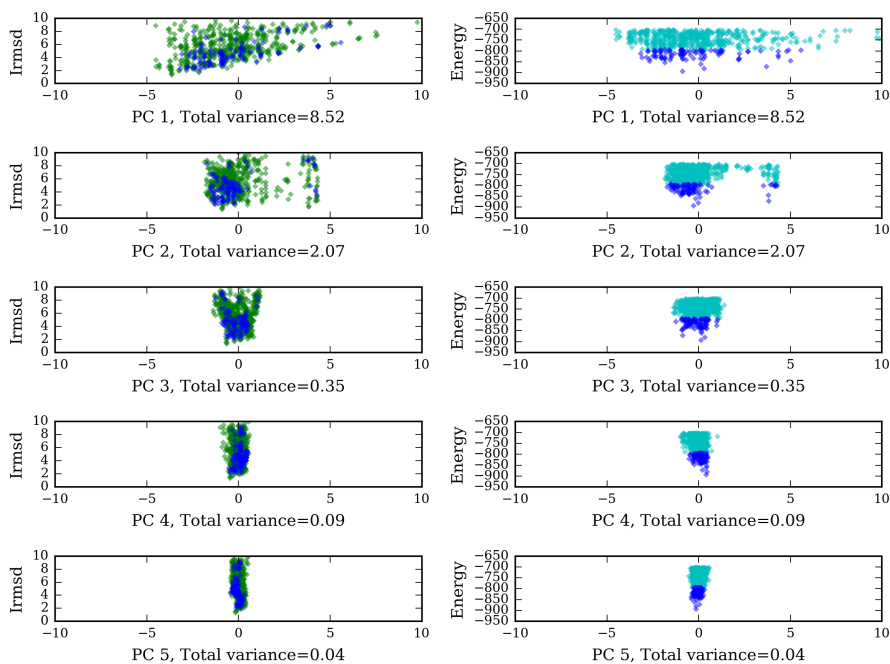


Figure 4.2: The near-native energy landscape of the 2YVJ complex.

This behavior has a deep biophysical explanation. Docking is initially driven

by a diffusive search governed by Brownian motion which brings the two molecules close. The encounter complex can be thought of as an ensemble of conformations in which the two molecules can rotationally diffuse along each other, or participate in a series of “microcollisions” that properly align the reactive groups. The second step of association consists of conformational rearrangements leading to the native complex. While it has been generally recognized that association proceeds through a transition state, little was known of the encounter complex structures and configurations as their populations are low, their lifetimes are short, and they are difficult to trap. In the earlier work (Kozakov et al., 2014), results from the application of Nuclear Magnetic Resonance (NMR) Paramagnetic Relaxation Enhancement (PRE) has been used, a technique that is extremely sensitive to the presence of lowly populated states in the fast exchange regime (Iwahara and Clore, 2006),(Clore, 2008),(Fawzi et al., 2010). According to the results the PRE profiles obtained experimentally are consistent with the presence of the encounter complexes that the landscape dimensionality analysis revealed.

Using this insight a new *stochastic global optimization* algorithm called *Subspace Semi-Definite programming-based Underestimation (SSDU)* is proposed. SSDU is based on SDU with all the generalizations that was introduced earlier. The most fundamental difference however, is that underestimation takes place only in the permissive conformational subspace found by PCA. This has the effect of avoiding high-energy barriers and evaluating the energy function only at non-singular points. Since the (typically) 3D permissive subspace contains encounter complexes, the *sequence of permissive subspaces* that SSDU’s PCA routine generates amounts to a characterization of a *smooth preferred association pathway*. Put differently, these subspaces correspond to a decreasing sequence of *energy plateaux* paving a smoother way of descending to the native state.

The remainder of the chapter is organized as follows. the SSDU algorithm is presented (Methods). The computational results on a benchmark set of protein structures are presented and discussed in the “Results and Discussion” Section. The chapter concludes with some final remarks.

Notation: Vectors will be denoted using lower case bold letters and matrices by upper case bold letters. For economy of space $\mathbf{v} = (v_1, \dots, v_n)$ is written as $\mathbf{v} \in \mathbb{R}^n$. Prime denotes transpose. For a matrix \mathbf{P} , $\mathbf{P} \succeq 0$ indicates positive semi-definiteness.

4.2 Methods

In this section, the four main steps of the SSDU algorithm are discussed. (i) Clustering using a density-based clustering algorithm is performed to remove noise and low density regions. (ii) The dimension of the search space is reduced from 5 to 3 using Principle Component Analysis. (iii) The energy landscape is under-estimated through a Sum-of-Squares (SOS) convex polynomial formulation and (iv) new samples are generated close to the global minimum of the under-estimator. These steps are iteratively performed until a termination criterion is reached. Moreover an additional step of cluster enrichment is performed on the top clusters generated by SSDU using a machine learning framework to increase the chance of picking a high quality representative from these clusters. Detailed descriptions of the SSDU four main steps are presented in the following and the cluster enrichment step is further discussed in the Results and Discussion section.

4.2.1 Clustering and Outlier Elimination

As was discussed in introduction, the input conformations may span several energy funnels (as in Fig. 4.1). To separate these funnels before underestimation, *clustering and outlier elimination* is performed. The idea is simply to cluster the input

conformations with respect to a distance measure (Euclidean distance is used). To that end, a *density-based* clustering method called *Density-Based Spatial Clustering of Applications with Noise (DBSCAN)* (Ester et al., 1996) is employed. Given a set of sample points in the conformational space, DBSCAN groups the points which are closely packed together in a dense region and eliminates the outlier points sitting in the low-density regions. In this scheme, the dense regions are defined as the *clusters*, which are separated by the low-density regions. DBSCAN requires two input parameters: (i) ϵ , the distance threshold which is defined as the maximum distance of two sample points to be considered as neighbors, and (ii) N_{min} , the minimum number of points required to form a cluster. The second parameter N_{min} ensures that all clusters found by DBSCAN will contain at least N_{min} points, and the algorithm will automatically eliminate outliers located in low-density regions.

In case of having multiple local minima in the neighborhood of the native structure, the clustering phase will tend to group the conformations around each local minimum in a separate cluster. In the sequel, it is explained how these clusters are used to handle situations in which most of the underestimation-based refinement methods with a single underestimator (Paschalidis et al., 2007),(Shen et al., 2008),(Nan et al., 2014) may fail to locate the global minimum of the energy function in the near-native region.

4.2.2 Dimensionality Reduction

A receptor-ligand conformation can be parameterized by a 6D vector $\psi = (\boldsymbol{\rho}, \mathbf{W}) \in SE(3)$, where $\boldsymbol{\rho} = (r, a, b) \in \mathbb{R}^3$ represents the translation vector from ligand center to receptor center and $\mathbf{W} = (w_1, w_2, w_3) \in \mathbb{R}^3$ specifies the rotation of the ligand using the exponential map from \mathbb{R}^3 to the *Special Orthogonal group* $SO(3)$ containing all rotation matrices. Here, $SE(3)$ denotes the *Special Euclidean group*, which is the

space of rigid-body motions and can be expressed as the semi-direct product of \mathbb{R}^3 (translations) and $SO(3)$ (rotations). $SE(3)$ is a nonlinear manifold and the exponential map is simply a projection from a (flat) tangent space to the manifold itself, projecting straight lines on the tangent space map onto geodesics of the manifold. Note that only relative orientation of the receptor and the ligand is important and one can assume the origin of the coordinate axis is at the receptor's center.

In the translation vector $\boldsymbol{\rho}$, r is the length of the vector and a, b indicate the spherical coordinates of the azimuth and zenith angles of $\boldsymbol{\rho}$, where the azimuth angle θ is the angle between the projection of $\boldsymbol{\rho} = (\rho_1, \rho_2, \rho_3)$ on the $\rho_1\rho_2$ plane and the ρ_1 axis, and the zenith angle ϕ is the one between $\boldsymbol{\rho}$ and the ρ_3 axis. The associated exponential coordinates are $(a, b) = (-\phi \sin \theta, \phi \cos \theta)$. $f : \mathbb{R}^6 \rightarrow \mathbb{R}$ is denoted as the energy function of a conformation parameterized by $\boldsymbol{\psi} \in \mathbb{R}^6$ as follows:

$$\boldsymbol{\psi} = (r, a, b, w_1, w_2, w_3). \quad (4.1)$$

As mentioned earlier, in low-energy clusters where conformations are well-packed, there is no significant variation in the center-to-center distance r between a ligand and the receptor, and this variable can be easily optimized separately once all other variables are determined. Thus, r is removed from $\boldsymbol{\psi}$ and f is minimized with respect to the remaining variables $\mathbf{x} \in \mathbb{R}^5$ which are:

$$\mathbf{x} = (a, b, w_1, w_2, w_3) \in \mathbb{R}^5. \quad (4.2)$$

It was discussed previously that the region of the space in the neighborhood of the native state is composed of high energy barriers that prevent the ligand to move in one or two directions (Kozakov et al., 2014), giving rise to a *restrictive sub-manifold* spanned by these directions. Orthogonal to the restrictive subspace there is a *permissive subspace* where the energy is much smoother. To identify the restrictive and

permissive subspaces, PCA is applied and the 5D parameterization of the conformational space (\mathbf{x}) is converted into linearly uncorrelated variables called *principal components* using an orthogonal transformation. This transformation seeks to find a set of principal components with the following property: the first principal component accounts for the largest possible variability in the data, and each succeeding component has the highest variance amongst all possible components which are orthogonal to the preceding components.

To describe the PCA procedure, assume a sample of K local minima of f in the \mathbf{x} -space has been obtained together with their corresponding energy values: $(\mathbf{x}^{(i)}, f^{(i)} = f(\mathbf{x}^{(i)}))$, $i = 1, \dots, K$. Let $\mathbf{X} \in \mathbb{R}^{5 \times K}$ be a matrix whose columns are of the form $\mathbf{x}^{(i)} - \bar{\mathbf{x}}$, $i = 1, \dots, K$, where $\bar{\mathbf{x}}$ is the mean of the K local minima. Then, the eigen-decomposition of $\mathbf{X}\mathbf{X}'$ is calculated as:

$$\mathbf{X}\mathbf{X}' = \mathbf{W}\mathbf{\Sigma}\mathbf{W}', \quad (4.3)$$

where \mathbf{W} is a 5×5 square matrix whose i th column is the i th eigenvector of $\mathbf{X}\mathbf{X}'$ and $\mathbf{\Sigma}$ is a diagonal matrix whose i th diagonal element is the i th corresponding eigenvalue. Let $\mathbf{z}^{(i)} = \mathbf{W}'(\mathbf{x}^{(i)} - \bar{\mathbf{x}})$ be the i th sample point transformed into the principal coordinates. It was shown in an earlier work (Kozakov et al., 2014) that in most protein-protein complexes, the first 3 eigenvalues of $\mathbf{X}\mathbf{X}'$ are significantly larger than the other 2 eigenvalues. Thus, only the first 3 principal components $\{z_1, z_2, z_3\}$ are taken to form the permissive subspace, while the remaining 2 components $\{z_4, z_5\}$ form the restrictive subspace which are eliminated. Let the new coordinates of the i th sample point in the 3D permissive subspace denoted by

$$\boldsymbol{\phi}^{(i)} = (z_1^{(i)}, z_2^{(i)}, z_3^{(i)}) \in \mathbb{R}^3. \quad (4.4)$$

Next, the goal is to minimize the energy function f by constructing a semidefinite

underestimator over the samples $\phi^{(i)}$, $i = 1, \dots, K$, in the permissive landscape.

4.2.3 Underestimation

As discussed in the previous section, this algorithm is based on finding convex underestimators which can be regarded as an approximation of the envelope spanned by the local minima of the binding energy function. In an effective underestimation, the minimum of the convex underestimator will be an approximation of the global minimum of the funnel-like binding energy function. Therefore sampling can be further biased toward the underestimator’s minimum. Below, it is first discussed how the convex underestimator can be calculated, then in the next subsection, it is further explained in detail how to bias sampling towards to the underestimator’s minimum point.

Following an earlier work (Nan et al., 2014), the class of general convex polynomial underestimators is considered. Let $U(\phi)$ be a degree $2d$ polynomial and $\phi \in \mathbb{R}^n$, where $n = 3$ in the case of seeking an underestimation in the 3D permissive subspace. Let $H = \nabla^2 U(\cdot)$ be the Hessian matrix of $U(\cdot)$. The convexity of a continuous, twice differentiable function $U(\cdot)$ on a convex set is guaranteed if and only if its Hessian matrix $H(\cdot)$ is positive semidefinite on the interior of the convex set. However, for the current application, since each entry of H is a polynomial term, the positive semidefiniteness of $H(\cdot)$ is difficult to establish analytically (except for the special case of quadratic underestimators where $2d = 2$). It is shown that even verifying the convexity of a degree-4 polynomial is an intractable problem (strongly NP hard) (Ahmadi et al., 2010).

Instead, a computationally tractable relaxation for convexity, called *SOS-convexity* (Ahmadi and Parrilo, 2013) is used. Let $\xi \in \mathbb{R}^n$ be a vector of variables. It is shown below that if $\xi' H(\phi) \xi$ is a Sum-of-Squares (SOS) in (ϕ, ξ) , then the convexity of $U(\cdot)$

is guaranteed in ϕ .

Let $\xi \in \mathbb{R}^n$ be a vector of variables, and consider $p(\phi, \xi) = \xi' H(\phi) \xi$ to be a scalar polynomial of degree $2d$ with $2n$ variables (ϕ, ξ) . Also, let

$$\mathbf{v} = (\xi_1, \dots, \xi_n, \xi_1 \phi_1, \dots, \xi_n \phi_n^{(d-1)}) \quad (4.5)$$

be a vector with length $\binom{d-1+n}{n} \times n$. The following theorem (Nan et al., 2014) uses SOS-convexity as a sufficient condition for convexity.

Theorem 1. *If there exists a matrix $\mathbf{P} \succeq 0$ such that $\mathbf{v}' \mathbf{P} \mathbf{v} = p(\phi, \xi) = \xi' H(\phi) \xi$, then the polynomial $U(\cdot)$ is convex.*

The condition in Theorem 1 is equivalent to saying that $\xi' H(\phi) \xi$ is SOS (a sum of squares) in (ϕ, ξ) , which suffices to ensure the convexity of $U(\cdot)$.

Therefore, one can formulate the problem of finding a convex polynomial underestimator of the sample points $(\phi^{(i)}, i = 1, \dots, K)$ as the following problem:

$$\begin{aligned} \min_{U(\cdot)} \quad & \sum_{i=1}^K [f^{(i)} - U(\phi^{(i)})] \\ \text{s.t.} \quad & f^{(i)} \geq U(\phi^{(i)}), \quad \forall i, \\ & \xi' H(\phi) \xi \text{ is SOS in } (\phi, \xi), \end{aligned} \quad (4.6)$$

where the optimization is over the coefficients of the polynomial $U(\cdot)$.

Let's consider the following example to show how one can formulate the optimization problem (4.6) as a tractable semi-definite program. Consider the special case of a degree-4 polynomial underestimator, i.e., $2d = 4$, and set $n = 3$ since the goal is to underestimate in the 3D permissive subspace. In this setting the underestimator has

the following form:

$$\begin{aligned}
U(\phi) = & a_1 + a_2\phi_1 + a_3\phi_1^2 + a_4\phi_1^3 + a_5\phi_1^4 + a_6\phi_2 + a_7\phi_1\phi_2 + a_8\phi_1^2\phi_2 \\
& + a_9\phi_1^3\phi_2 + a_{10}\phi_2^2 + a_{11}\phi_1\phi_2^2 + a_{12}\phi_1^2\phi_2^2 + a_{13}\phi_2^3 + a_{14}\phi_1\phi_2^3 \\
& + a_{15}\phi_2^4 + a_{16}\phi_3 + a_{17}\phi_1\phi_3 + a_{18}\phi_1^2\phi_3 + a_{19}\phi_1^3\phi_3 + a_{20}\phi_2\phi_3 \\
& + a_{21}\phi_1\phi_2\phi_3 + a_{22}\phi_1^2\phi_2\phi_3 + a_{23}\phi_2^2\phi_3 + a_{24}\phi_1\phi_2^2\phi_3 + a_{25}\phi_2^3\phi_3 \\
& + a_{26}\phi_3^2 + a_{27}\phi_1\phi_3^2 + a_{28}\phi_1^2\phi_3^2 + a_{29}\phi_2\phi_3^2 + a_{30}\phi_1\phi_2\phi_3^2 \\
& + a_{31}\phi_2^2\phi_3^2 + a_{32}\phi_3^3 + a_{33}\phi_1\phi_3^3 + a_{34}\phi_2\phi_3^3 + a_{35}\phi_3^4.
\end{aligned} \tag{4.7}$$

Based on Theorem 1, $\xi'H(\phi)\xi$ is SOS in (ϕ, ξ) is equivalent to $\mathbf{P} \succeq 0$ where $\mathbf{v}'\mathbf{P}\mathbf{v} = \xi'H(\phi)\xi$. Therefore, by relating the elements of \mathbf{P} with coefficients of $U(\phi)$, one can reformulate (4.6) as the following semi-definite problem (SDP):

$$\begin{aligned}
& \min_{a_1, \dots, a_{35}, \mathbf{P}} \sum_{i=1}^K s^{(i)} \\
& \text{s.t. } f^{(i)} - (a_1 + a_2\phi_1 + \dots + a_{35}\phi_3^4) = s^{(i)}, \quad i = 1, \dots, K, \\
& P_{1,1} = 12a_5, \quad P_{4,4} = 2a_{12}, \quad 2P_{1,4} = 6a_9, \\
& \vdots \\
& 2P_{10,12} = 2a_{17}, \quad 2P_{11,12} = 2a_{20}, \quad P_{12,12} = 2a_{26}, \\
& \mathbf{P} \succeq 0, \quad s^{(i)} \geq 0, \quad i = 1, \dots, K.
\end{aligned} \tag{4.8}$$

To solve this SDP, the CSDP solver (Borchers, 1999) is used. Solving (4.8) outputs the optimal coefficients (a_1^*, \dots, a_{35}^*) of the polynomial convex function $U(\phi)$ that can be regarded as a tight underestimator of the K local minima $(\phi^{(i)}, i = 1, \dots, K)$.

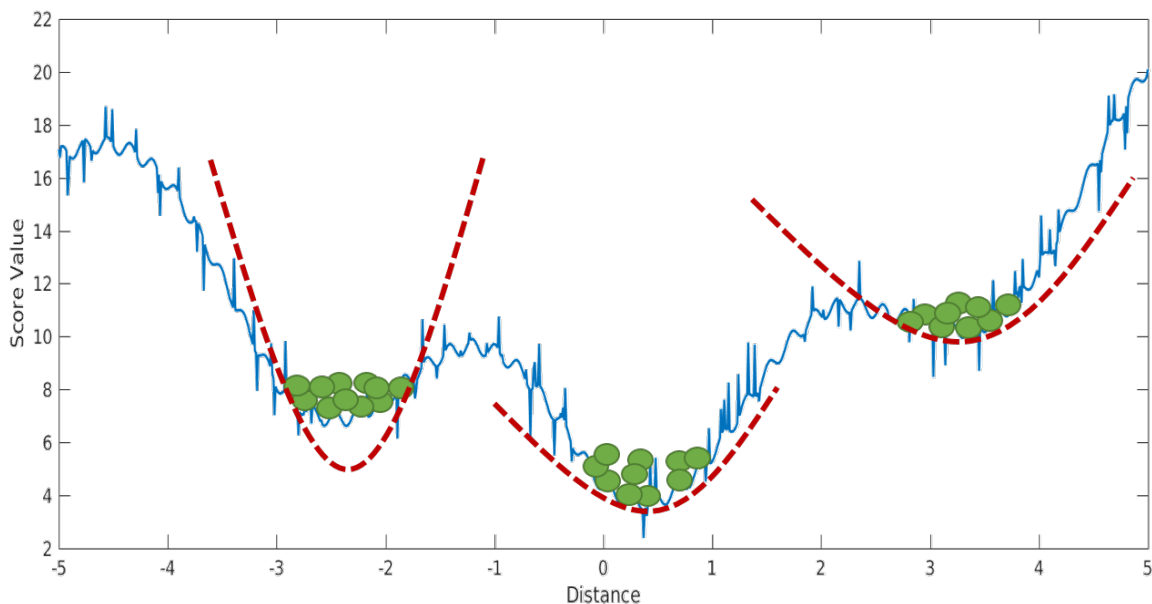


Figure 4-3: One underestimator (denoted by red dashed lines) per cluster is calculated. If the underestimation step succeeds in capturing the shape of the free energy function, then the sampling step will help generate more conformations in the vicinity of the global minimum of the energy function.

4.2.4 Sampling

Let $\phi^* \in \mathbb{R}^3$ be the global minimum of the convex underestimator obtained from the solution of (4.6). More conformations are generated in the vicinity of the global minimum ϕ^* . If the underestimation step succeeds in capturing the shape of the free energy function, then the sampling step will help generate more conformations in the vicinity of the global minimum of the energy function (see figure 4-3).

First, \bar{K} random samples are generated $\mathbf{s}^{(l)} \in \mathbb{R}^5$, $l = 1, \dots, \bar{K}$, where each random dimension $s_i^{(l)}$ has a uniform distribution in the range of $(-0.5\beta\sigma_i, 0.5\beta\sigma_i)$, $i = 1, \dots, 5$, where β is a constant and σ_i is the i th diagonal element of Σ in (4.3), hence, $\sigma_1 \geq \dots \geq \sigma_5$. Then, one can construct the 5D global minimum \mathbf{z}^* by appending an approximation of z_4^* , z_5^* to ϕ^* as in (4.10). As discussed earlier, the last two principal coordinates z_4 , z_5 have small variation over the samples; therefore one

can consider their sample mean as a good approximation:

$$z_i^* = \frac{1}{K} \sum_{j=1}^K z_i^{(j)}, \quad i = 4, 5 \quad (4.9)$$

And set

$$\mathbf{z}^* = (\phi^*, z_4^*, z_5^*). \quad (4.10)$$

Next, the new sample points are generated in the vicinity of the underestimator's global minimum and are transformed from the principal coordinates to the original coordinates as follows:

$$\tilde{\mathbf{x}}^{(l)} = \mathbf{W}(\mathbf{z}^* + \mathbf{s}^{(l)}) + \bar{\mathbf{x}}. \quad (4.11)$$

The sampling range of random samples $\mathbf{s}^{(l)}$ at each dimension i is proportional to the variance σ_i to guarantee an effective coverage of the conformational space which preserves the sample distribution. Furthermore, in order to construct the 6D conformational parameterization of these generated sample points, one needs to append the sample mean of the center-to-center distance r in (4.1), i.e., $\bar{r} = \frac{1}{K} \sum_{i=1}^K r^{(i)}$, which results in the new sample conformation in \mathbb{R}^6 :

$$\tilde{\boldsymbol{\psi}}^{(l)} = (\bar{r}, \mathbf{x}^{(l)}). \quad (4.12)$$

4.2.5 SSDU Algorithm

All key steps of the SSDU algorithm has been discussed so far. The entire algorithm is outlined below in Algorithm 1. Note that the algorithm explores separately the potential multiple sub-clusters discovered by DBSCAN. Using the sampling approach outlined, K conformations are sampled in each sub-cluster. Afterwards, all these conformations are merged and the top conformations based on energy are picked. One can iterate over the steps of SSDU until meeting the stopping criteria. The retained conformations can be regarded as the SSDU outputs. Figure 4-4 shows a

flowchart of the SSDU procedure demonstrating the process of refining the initial PIPER sample conformations to produce the ensemble of refined structures.

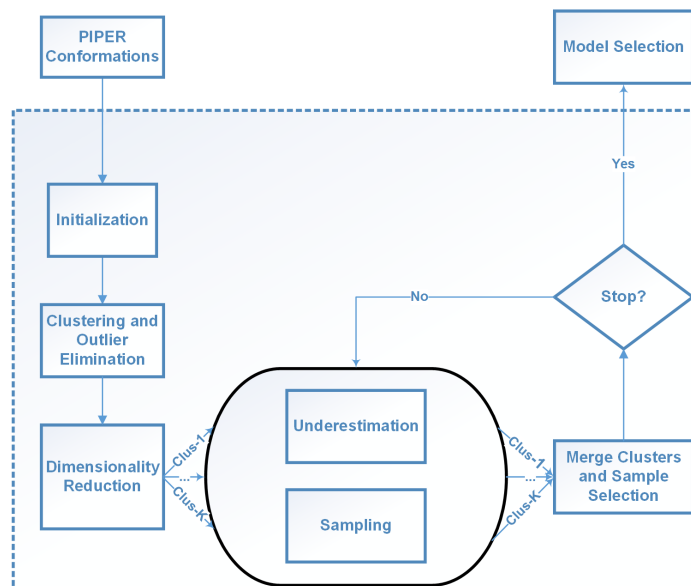


Figure 4-4: The flowchart of the SSDU procedure.

4.2.6 Local Minimization

All the presented sampling approaches use a common local minimization subroutine. Its main role is to account for flexibility of side chains during the search. This protocol has been explored and optimized in the work (Moghadası et al., 2015). It consists of the following steps. Initially the *side-chain positioning (SCP)* algorithm (Moghadası et al., 2013),(Moghadası et al., 2015) is run that solves a relaxed formulation of a combinatorial optimization problem in order to repack the amino acid residues at the interface of the receptor-ligand complex. Then *rigid-body energy minimization* algorithm is run (Mirzaei et al., 2012) which locally minimizes the position and orientation of the ligand with respect to the receptor.

Algorithm 1 SSDU Algorithm

- 1: **Initialization:** Starting from K sample points in conformational space \mathcal{S} , perform local minimization to obtain K distinct local minima $\boldsymbol{\psi}^{(1)}, \dots, \boldsymbol{\psi}^{(K)}$ of $f(\cdot)$.
 - 2: **Clustering and Outlier Elimination:** Run DBSCAN over the input sample points to split the dataset into several clusters. Let n be the number of clusters the algorithm finds and $\{\mathcal{C}_1, \dots, \mathcal{C}_n\}$ be the corresponding clusters.
 - 3: **Dimensionality Reduction:** For each sample point i reduce $\boldsymbol{\psi}^{(i)} \in \mathbb{R}^6$ to $\mathbf{x}^{(i)} \in \mathbb{R}^5$ in (4.2), then transform $\mathbf{x}^{(i)}$ to $\boldsymbol{\phi}^{(i)} \in \mathbb{R}^3$ in (4.4) using PCA.
 - 4: **Exploration:** For each cluster \mathcal{C}_i , $i = 1, \dots, n$,
 - **Underestimation:** Solve the SDP in (4.6) to obtain the convex polynomial underestimator $U_i(\boldsymbol{\phi})$. Set the predictive point $\boldsymbol{\phi}_i^*$ to be the minimizer of $U_i(\boldsymbol{\phi})$. Transform $\boldsymbol{\phi}_i^*$ to \mathbf{z}_i^* in the 5D conformational space as in (4.10).
 - **Sampling:** Transfer \mathbf{z}_i^* from the principal coordinates into the original coordinates and generate random samples $\tilde{\mathbf{x}}_i^{(l)}$, $l = 1, \dots, K$, as in (4.11) for each cluster \mathcal{C}_i . Construct $\tilde{\boldsymbol{\psi}}_i^{(l)}$ in the 6D conformational space from $\tilde{\mathbf{x}}_i^{(l)}$ as in (4.12).
 - 5: **Sample Selection:** Merge the output sampled conformations of all clusters and the inputs to the algorithm and select K top conformations with the lowest energy value. Let $\boldsymbol{\psi}^G$ be the conformation with minimum energy value amongst the K retained conformations.
 - 6: **Termination:** Let $\boldsymbol{\psi}^*$ be the global minimum of the underestimator in the original 6D space. If $\|\boldsymbol{\psi}^G - \boldsymbol{\psi}^*\| < \eta$ or there is no progress in reducing $f(\boldsymbol{\psi})$ or the maximum number of iterations is reached then stop; otherwise go to step 4.
-

4.2.7 Energy Function

The choice of energy function is a high-accuracy docking energy potential that can be calculated as a weighted sum of a number of force-field and knowledge-based energy terms (Gray et al., 2003),(Andrusier et al., 2007),(Pierce and Weng, 2007). The following energy terms to find the interaction free energy value were used:

$$E = w_{VDW}E_{VDW} + w_{SOL}E_{SOL} + w_{COUL}E_{COUL} + w_{HB}E_{HB} + w_{DARS}E_{DARS} + w_{RP}E_{RP},$$

where E_{VDW} is the Lennard-Jones potential, E_{SOL} is an implicit solvation term (Schaefer and Karplus, 1996), E_{COUL} is the Coulomb potential, E_{HB} is a knowledge-

based hydrogen bonding term (Kortemme et al., 2003), and E_{DARS} is a structure-based intermolecular potential that is derived from the non-redundant database of native protein-protein complexes which uses a novel *DARS (Decoys as Reference State)* (Chuang et al., 2008) reference set. The last term, E_{RP} , is a statistical energy term associated with a set of rotamers selected from the backbone-dependent rotamer library (Shapovalov and Dunbrack Jr, 2011). The weight set of the energy function is adopted according to the selections in Gray et al. (Gray et al., 2003).

4.2.8 Validation dataset and input preparation

The algorithm was validated on a comprehensive benchmark of protein complexes comprising *Enzymes, Antibodies and Other types* where this benchmark includes 230 complexes in total (Vreven et al., 2015). The results for 6 complexes are not reported due to technical difficulties discussed in Protein Docking Refinement section.

Other types of complexes exhibit multiple deep funnels in the vicinity of the native structure which makes them particularly difficult cases for protein docking refinement, whereas enzyme interactions are usually driven by shape complementarity, making them relatively easier cases. In fact, considering a wide spectrum of docking test cases in terms of difficulty, make it possible to examine the performance gain compared to the ClusPro web-server, as outlined in chapter 2, in different scenarios. Moreover, other types of complexes present an opportunity to evaluate the effect of the density based clustering component built into SSDU where fitting multiple underestimators seems inevitable. Input preparation consists of two steps: (1) running global FFT sampling using PIPER; and (2) filtering the conformations to retain the top 1000 and 1500 for enzymes/antibodies and other types respectively. These top energy conformations are supplied as the input to the SSDU algorithm.

4.3 Results and Discussion

In this section, the SSDU-produced ensemble are compared against the corresponding input ensemble produced by ClusPro. ClusPro is used as a baseline for comparison because it has been established to perform comparably well to other methods (Kozakov et al., 2017). In fact, ClusPro has ranked first multiple times among automated servers in the rounds of the Critical Assessment of Prediction of Interactions (CAPRI) community-wide experiment in the years 2009, 2013 and 2016. Furthermore access to the ClusPro source code has made it convenient to appropriately adjust its output for the purposes of the refinement experiments. In what follows, both the number of near-native conformations in each ensemble and the implications in selecting a near-native conformation out of the refined ensemble without knowing the native structure are considered.

The results are based on the following parameter settings: $K = 1000$ indicates the number of conformations for enzymes and antibodies and $K = 1500$ for other types of complexes, provided as the input to SSDU, $\epsilon = 1.0$ and $N_{min} = 100$ are the parameters used in DBSCAN (Step 2 of Alg. 1), $\eta = 0.3$ (Step 6 of Alg. 1), and a maximum number of iterations equal to 3 is used for SSDU termination.

4.3.1 Protein Docking Refinement

To show the impact of SSDU algorithm, three different plots (4-5, 4-6, 4-7) are presented for the number of counts of quality solutions before and after SSDU. The quality of solutions are assessed based on a community-wide protein-docking competition called *Critical Assessment of Prediction of Interactions* (CAPRI) (Janin, 2005). According to CAPRI, the quality of a solution is determined based on interface RMS(iRMSD), backbone RMS(LRMSD) and the number of native contacts preserved(Fnat). To determine the CAPRI classification of a conformation, the pro-

gram DockQ was used (Basu and Wallner, 2016). DockQ combines normalized values of iRMSD, LRMSD and Fnat to generate a continuous score in the range [0,1] where the higher the score, the better the quality of a solution. Specifically, a conformation of a protein complex is classified into one of four categories: Incorrect, Acceptable, Medium or High based on its dockQ score. Moreover, the quality of the solutions produced by the SDU algorithm (Shen et al., 2008) is presented as well to measure the performance boost from the innovations we introduce in this chapter.

Note that the *unbound protein structures*, protein structures before binding, were used to generate the input to the SSDU/SDU algorithms. The use of unbound structures is important to assess docking performance in the absence of any knowledge about the native conformation. As mentioned earlier, the inputs to SSDU/SDU are the top 1000 and 1500 energy conformations from ClusPro for enzymes/anibodies and other types, respectively. The output has the same number of conformations as the input and contains a mixture of conformations from the input and SSDU/SDU re-sampled conformations. Specifically, the re-sampled conformations from SSDU/SDU are merged with the input conformations and then subjected to energy filtering to retain the same number of top energy conformations as the input. For example, if the input has 1000 conformations and SSDU density-based clustering discovers three clusters, the number of conformations after the merge will be 4000 from which the 1000 top energy conformations are reported as the SSDU output.

Note that the results reported in this work are on 224 out of 300 complexes of the benchmark (Vreven et al., 2015). The 6 removed complexes are 4GAM, 4GXU, 2H7V, 4FQI, 1DE4, 1N2C. These complexes were removed because one of the programs we use failed to produce a score/solution for many conformations due to technical issues (DockQ for the first three, SSDU for the fourth, and SDU for the last two).

As it is apparent from figures 4·5, 4·6 and 4·7, SSDU substantially increases the

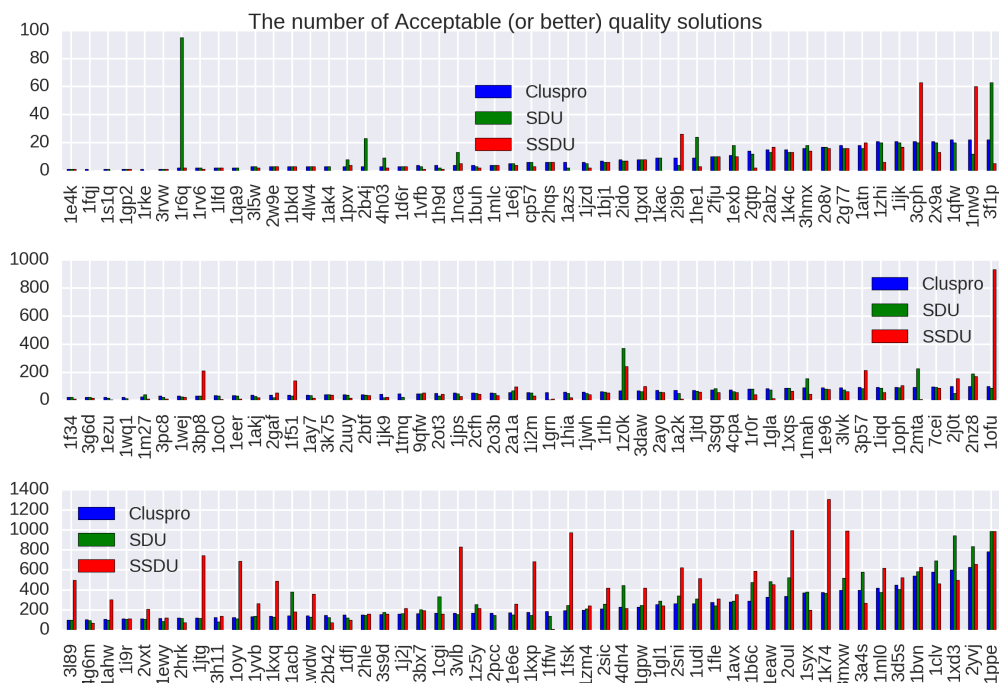


Figure 4-5: The x -axis represents 156 out of 224 protein complexes that have either Clupro or SSDU non-zero CAPRI Acceptable (or better) quality solutions. The complexes are sorted by the number of ClusPro counts, and the y -axis shows the number of Acceptable quality solutions out of an ensemble of 1000 or 1500 for enzymes/antibodies and other types conformations respectively produced by ClusPro, or refined by SDU and SSDU.

number of quality solutions in different categories. Note that there are cases where SDU or Cluspro perform better than SSDU, especially where the number of *input quality solutions* to the algorithms are lower. As SSDU/SDU are both refinement protocols, the assumption is that the input contains reasonable number of good quality conformations and SSDU is not expected to perform optimally in the case of a protein complex with relatively poor input quality solutions.

The amount of improvement by SSDU compared to SDU and ClusPro is reported in Table 4.1. The average improvement is determined by calculating the percentage improvement for each protein complex and averaging over different complexes in the

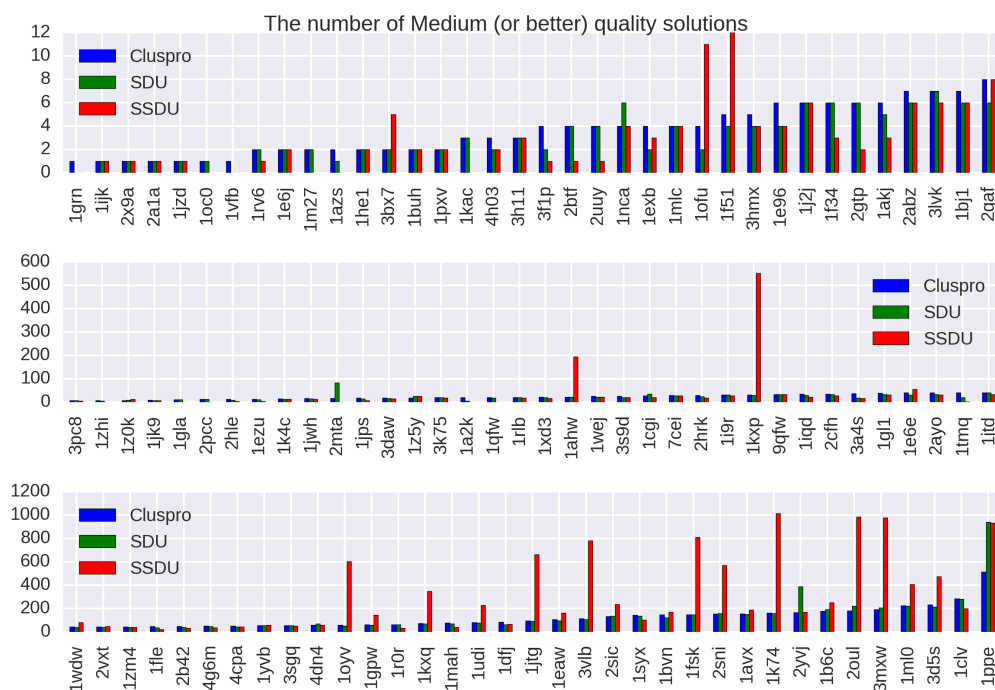


Figure 4-6: The x -axis represents 110 out of 224 protein complexes that have either Clupro or SSDU non-zero CAPRI Medium (or better) quality solutions. The complexes are sorted by the number of ClusPro counts, and the y -axis shows the number of Medium quality solutions out of an ensemble of 1000 or 1500 for enzymes/antibodies and other types conformations respectively produced by ClusPro, or refined by SDU and SSDU.

benchmark, whereas the total improvement is the percentage improvement when the number of near-native hits are aggregated over all the complexes in the benchmark.

4.3.2 Post-Processing Ensemble Enrichment

It has been established that SSDU generates outputs with significantly higher quality compared to the input ClusPro conformations. Next, it is examined whether one can select a small number (specifically, 10) of enriched clusters from this SSDU ensemble which maintain a significant portion of the high quality conformations.

Selecting a high quality conformation remains a very challenging problem in the

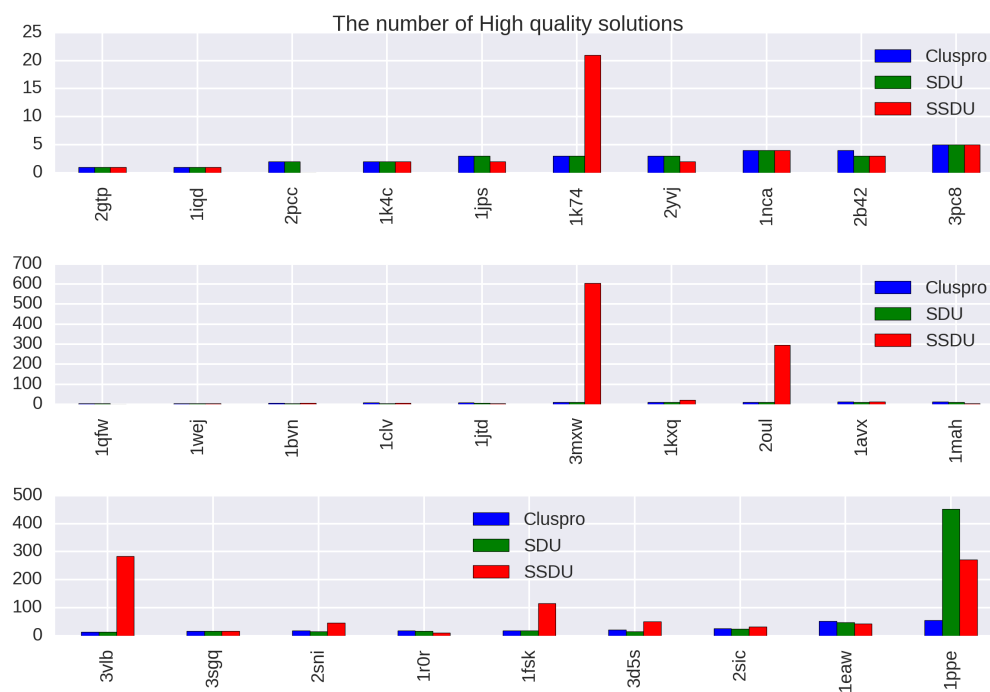


Figure 4-7: The x -axis represents 29 out of 230 protein complexes that have either Clupro or SSDU non-zero CAPRI High quality solutions. The complexes are sorted by the number of ClusPro counts, and the y -axis shows the number of High quality solutions out of an ensemble of 1000 or 1500 for enzymes/antibodies and other types conformations respectively produced by ClusPro, or refined by SDU and SSDU.

protein docking community. In the CAPRI, participating groups test their methods in blind predictions of given target protein complexes. As mentioned before iRMSD, LRMSD and Fnat are used to categorize the predictions into Incorrect, Acceptable, Medium, and High quality. Reflecting how challenging the problem is, CAPRI allows for 10 submissions from each participating group.

ClusPro, against which SSDU results are compared, uses clustering as a way of taking into account entropic metrics that were not included in the energy function we described earlier. The description of the clustering algorithm can be found in chapter 2. ClusPro then selects the centers of the 10 largest clusters as its submissions to CAPRI. Note that center of a cluster is defined as the member of the cluster with the

highest number of neighbors.

It is examined whether replacing the ClusPro ensemble with the SSDU ensemble also enriches the top 10 selected clusters. In this work, and because SSDU is an improved sampling method, the focus is solely on the question of cluster discrimination, that is, selecting 10 enriched clusters. The question of conformation discrimination, which amounts to selecting a single representative conformation from each top cluster, is outside the scope of this work and is left open to future work.

The SSDU clusters are formed by clustering the conformations in the SSDU ensemble in exactly the same way as ClusPro. These clusters are ranked using a ranking method described in the sequel. For each complex two sets of clusters are compared. The first (ClusPro) set is formed by clustering the ClusPro produced structures and ranking the clusters in decreasing cluster size. The second (SSDU) set is formed by first refining with SSDU the ClusPro ensemble, then generating (typically 30) clusters using the ClusPro clustering algorithm, and finally ranking these clusters using the method described next. In each case, the number of Acceptable/Medium/High quality solutions among the top 3, 5 and 10 clusters are computed.

Ranking the SSDU ensemble. A machine learning approach is employed for ranking the 30 clusters of the SSDU set. Different classification algorithms on this dataset are employed. Namely random forest, support vector machine with linear and radial kernels and logistic regression were used. Specifically, *random forest* achieved the highest performance of all which would be the focus of the remainder of this section (Breiman, 2001). Some related work on using machine learning approaches, different than this work, for ranking has appeared in (Moal et al., 2017),(Pfeifferberger et al., 2016). To perform the classification each cluster is characterized with a set of 9 features described below:

1. The first four consist of the average energy value of the top 25%, 50%, 75% and

100% lowest energy conformations in the cluster, respectively.

2. The 5th feature is the number of conformations (size) of the cluster.
3. The last four features consist of the average RMSD between the cluster center and the top 25%, 50%, 75% and 100% conformations, respectively, in an ordered list of cluster conformations ranked in increasing RMSD from the cluster center.

Each cluster is labeled by evaluating the dockQ score of the cluster center: if it has Acceptable quality score (or better) it is given a label of +1 (positive class); otherwise a label of -1 (negative class).

The Random Forest classification algorithm trains a set of unpruned de-correlated classification trees using random selection of training data and random selection of variables. It classifies a new sample by taking a majority vote of all trees, which reduces through averaging the variance of the decision. To each new sample a probability is associated of the sample belonging to the positive class as follows. The new sample is classified by each tree in the random forest and ends up in some leaf node of the tree. The fraction of training samples assigned to that leaf node is used as a surrogate of the probability that the new sample belongs to the same class as the training samples in the leaf node. These probabilities are then averaged over all trees to compute an overall probability that the sample belongs to the positive class. A classification decision can then be made by comparing that probability to a given threshold. Moreover, samples can be ranked using this probability.

A random forest classifier is trained by randomly dividing the whole dataset into *non-overlapping* training and testing datasets having 60% of the complexes as the training set and leaving the remaining 40% as a test dataset. The classification performance is evaluated through the Receiver Operating Characteristic (ROC) curve computed on the test set. The ROC plots the true positive rate (fraction of posi-

tive test samples correctly identified as positive) vs. the false positive rate (fraction of negative samples incorrectly identified as positive) as the threshold used for the classification decision changes. The Area Under the ROC Curve (AUC) is used as a prediction performance metric. An AUC of 1 represents perfect classification accuracy, whereas an AUC of 0.5 represents a naive random classifier which assigns samples to a class by flipping a coin.

The probability of a sample belonging to the positive class is used in order to rank (in decreasing order of the probability) the SSDU set of clusters. Similar to Cluspro results, the number of Acceptable/Medium/High quality solutions is counted among top 3, 5 and 10 clusters. Finally, the improvement is measured in the number of quality solutions in each of the categories.

As described, the SSDU cluster set are processed using *non-overlapping datasets* for training and testing. The training and testing is repeated 15 times, each time with a different random split of the data-set, and averaged the AUC computed on the test set over the 15 runs. This yielded an average AUC for other complexes of 0.62 . This value indicates adequate classification accuracy, significantly better than random selection. Figure 4-8 shows the amount of the improvement of SSDU over Cluspro for different quality categories of Acceptable/Medium/High among top 3, 5 and 10 clusters. It is apparent from these results that SSDU can noticeably enrich the top clusters among different categories of solutions quality. For instance, SSDU can improve the density of Acceptable, Medium and High quality solutions among the top 10 clusters by 61%, 20% and 38% respectively.

4.4 Conclusions

A new protein docking refinement protocol was presented which is shown to effectively refine the quality of the solutions produced by first-stage global search methods like

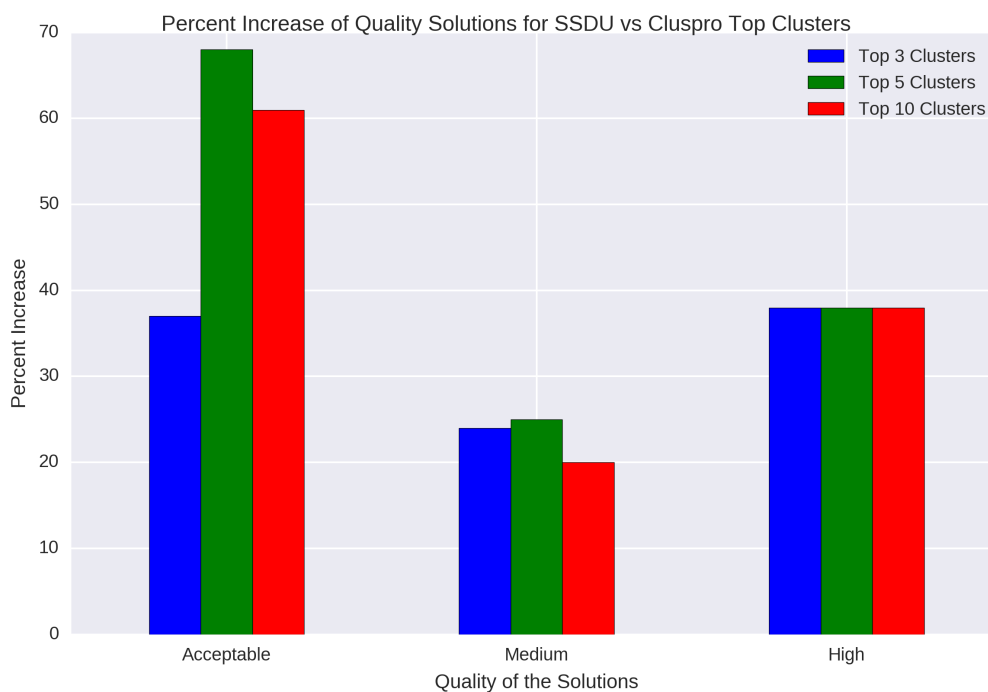


Figure 4·8: The percentage of increase in the number of solutions for SSDU vs Cluspro among top clusters. The x -axis represents the category of quality of solutions according to the CAPRI criteria.

PIPER which is implemented in the protein docking server ClusPro 2.0.

The SSDU algorithm developed builds on an earlier SDU method (Paschalidis et al., 2007),(Shen et al., 2008) and works by underestimating the energy function in a set of local minima generated by local minimization methods. SSDU uses the minimum of the convex underestimator it generates to concentrate further sampling in its vicinity, assuming that this minimum resides close to the basin of the energy funnel spanned by the local minima. Four innovations introduced in this work are: (i) the use of the landscape analysis in (Kozakov et al., 2014) to restrict underestimation in a lower-dimensional (typically 3D) permissive conformational subspace that avoids high-energy barriers; (ii) the use of density-based clustering to eliminate low-density regions and identify potential multiple high-density sub-clusters that are

then separately refined by SSDU; *(iii)* the use of more flexible convex polynomial underestimators, and *(iv)* the use of a machine learning approach to effectively increase the number of Acceptable/Medium/High CAPRI quality solutions among the top clusters.

The effectiveness of SSDU is demonstrated on a comprehensive benchmark of 224 complexes containing Enzymes, Antibodies and Other Types complexes. It is shown that SSDU is capable of increasing the number of quality solutions on a spectrum of different complexes in different quality categories defined by CAPRI competition. It was also shown that novelties introduced in this work make SSDU superior to its predecessor SDU algorithm. Furthermore, it was shown that one can further process the outputs to refine the quality of the solutions among the top clusters generated by SSDU, whereby increasing the chance of picking a high quality representative from these clusters by other algorithms.

Table 4.1: Percentage improvement of Acceptable (or better), Medium (or better) and High quality solutions by SSDU versus SDU and ClusPro for a benchmark of 224 complexes. Please note that for each the entries in the table, complexes with zero number of solutions for both Cluspro and SDU/SSDU are removed.

Benchmark		SSDU vs. ClusPro	SSDU vs. SDU
Acceptable (or better)	Average	24.62%	21.31%
	Total	53.14%	30.37%
Medium (or better)	Average	53.26%	58.25%
	Total	132.69%	112.43%
High	Average	410.71%	405.88%
	Total	424.93%	157.06%

Chapter 5

Machine Learning methods in Protein Docking

5.1 Introduction

Machine Learning (ML) is a framework for designing algorithms that capture a pattern within a *training dataset* and are capable of making informed decisions or exhibiting desired action for an unseen *testing dataset*.

Two of the main branches of ML are (i) Supervised Learning and (ii) Unsupervised Learning. In Supervised Learning, each data point is represented by a pair (x, y) where x is a *feature vector* containing information about the data point and y as the *label* which either specifies the category of the data point or is a numeric value measuring a quantity of interest. The goal of Supervised Learning algorithms is to obtain a mathematical model that can predict the label of unseen data points using their feature vectors. For instance, one might be interested to develop a model for discriminating between Soccer and Tennis balls. In this manner one can construct feature vectors from the diameter, weight and color of each ball where the labels would indicate either “Soccer ball” or “Tennis ball”.

Unsupervised Learning methods focus on understanding hidden structure of the data in the absence of labels. In fact, these methods do not require “supervision” for labeling each data point. To give an example, it is theorized that earthquakes happen as a result of tectonic plates movement. One can examine this hypothesis by

analyzing the occurrences of earthquakes over the globe and observe how earthquakes are related. To do so, one can look at the coordinates of occurrences of earthquakes and observe whether there are “clusters” of these earthquakes that pack closely. Afterwards, it can be verified whether these packed sites overlap with any boundaries of tectonic plates.

A plethora of biological datasets has become available recently and is expanding exponentially thanks to breakthroughs in High-Throughput Screening. The analysis of these large datasets is beyond manual human work and there has been a need for theoretically sound and practically accurate and efficient computer algorithms to gain knowledge from these untapped datasets (Chicco, 2017).

Due to its scalability and automation, Machine Learning provides the perfect framework for analyzing these datasets. Machine Learning has been applied to different problems in this field (Baldi, 2001), (Tarca et al., 2007), (Schlkopf et al., 2004). For instance, Genomics, Systems Biology and Proteomics are some of the major areas within this area of study (Larraaga et al., 2006).

Following other sections of this thesis, Protein-Protein interaction (PPI) is the focus of this part. The investigation of PPIs is still a challenging problem and the availability of biological datasets has given rise to data-driven approaches to this problem (Tahir and Haya, 2017). These models take as the input the structural information and/or sequence based features to predict binding between different proteins (Sudhaa et al., 2014), (Agrawal et al., 2014). Furthermore, these PPI predictions have been exploited for constructing *Protein Interaction Networks*, studying complex biological systems such as human diseases (Safari-Alighiarloo et al., 2014), (Sevimoglu and Arga, 2014), (Malod-Dognin et al., 2017) for the development of new medications.

In this part, first an introduction to two variants of Support Vector Machine binary classifier that are used throughout this chapter is presented. Moreover, multiple

fundamental metrics for assessing binary classifiers performance are discussed which are used to assess the performance of the aforementioned classifiers in the following sections. Furthermore, the application of Machine Learning to two different projects are presented. The first problem focuses on discrimination between interacting and non-interacting protein pairs and a dataset of discrimination between Biological versus Crystallographic Dimers is used for evaluation of the methodology. The second project focuses on developing an algorithm for ranking the output predictions of docking servers where two current state-of-the-art works are discussed and compared and a framework for deriving the ranking model for Cluspro 2.0 web server is proposed.

5.2 Classification Models

5.2.1 Sparse Linear Support Vector Machine

In this section, a brief description of a variant of Support Vector Machine (SVM) classifier called Sparse Linear SVM (SLSVM) is presented (Dai, 2015). Assume that the dimension of input data is D and the goal is to find the hyperplane $\beta = (\beta_0, \beta_1, \dots, \beta_D)$ which has the maximal margin from the training data points, where the margin is defined as the smallest distance between any data point and the hyperplane, and β has a sparse structure. Intuitively, sparsity structure leads to a simpler model, potentially reducing overfitting of the model to the training data. Furthermore, one can interpret the regularization used as a factor for choosing important features.

Assume that the number of positive, negative and total number of samples are denoted by N^+ , N^- and N respectively where $N^+ + N^- = N$. Also, let (x_i^+, y_i^+) , $i \in \{1, \dots, N^+\}$ and (x_j^-, y_j^-) , $j \in \{1, \dots, N^-\}$ denote positive and negative training samples respectively where x and y are the feature vector and the corresponding label respectively.

The optimization problem to find the Sparse Linear SVM (SLSVM) is as follows(Dai, 2015) ($|\cdot|$ and $\|\cdot\|$ denote absolute value and l_2 norm respectively.):

$$\begin{aligned}
& \min_{\beta, \beta_0, \xi_i, \zeta_j} \frac{1}{2} \|\beta\|^2 + \lambda^+ \sum_{i=1}^{N^+} \xi_i + \sum_{j=1}^{N^-} \zeta_j \\
& \text{s.t.} \sum_{d=1}^D |\beta_d| \leq K, \\
& \xi_i, \zeta_j \geq 0 \\
& \xi_i \geq 1 - y_i^+ \beta_0 - \sum_{d=1}^D y_i^+ \beta_d x_{i,d}^+, \quad \forall i \in \{1, \dots, N^+\} \\
& \zeta_j \geq 1 - y_j^- \beta_0 - \sum_{d=1}^D y_j^- \beta_d x_{j,d}^-, \quad \forall j \in \{1, \dots, N^-\}
\end{aligned} \tag{5.1}$$

Where $y_i^+ = 1$ and $y_j^- = -1 \forall i \in \{1, \dots, N^+\}, \forall j \in \{1, \dots, N^-\}$

Where slack variables ξ and ζ correspond to positive and negative training points respectively and parameter K controls sparsity of the linear coefficient and λ^+ and λ^- are the corresponding penalty coefficients for the slack variables. Note that formulation 5.1 corresponds to the so called Support Vector Machine where the additional constraint $\sum_{d=1}^D |\beta_d| \leq K$ enforces the sparsity structure for the hyperplane.

5.2.2 Alternating Clustering and Classification

The problem of discriminating between interacting and non-interacting protein partners has a special structure. Specifically, interacting pairs can belong to different categories of complexes such as Enzymes or Antibodies where the driving force and features of interaction are different within each category.

Therefore, as a machine learning framework one can assume the positive class, interacting pairs, consists of multiple clusters where the discriminating features can be

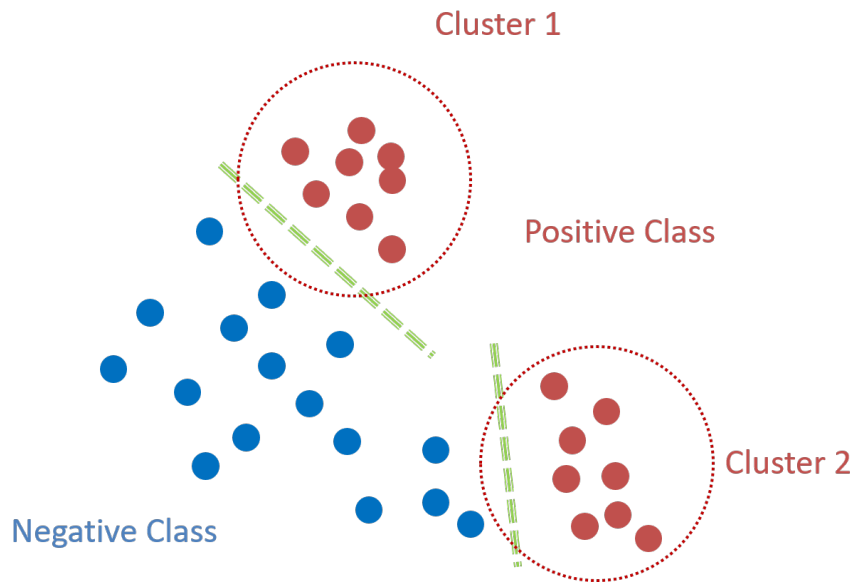


Figure 5-1: ACC (Dai, 2015) is an algorithm joint clustering and classification of the input data where the positive class possibly consists of multiple hidden subclusters where the basis for clustering the positive data is the similarity between the members of a cluster in terms of discriminating features from the negative class. The goal is to find hidden clusters of positive data while finding the optimal classifier within each cluster. Note that how the calculated classifiers for cluster 1 and cluster 2, denoted by green dashed lines, are different.

substantially different for different clusters while the negative class consist of uniform data points, belonging to a single cluster.

Specifically, one can formulate the problem as that of joint clustering and classification of the input data where the positive class possibly consists of multiple hidden subclusters whereas the negative class is assumed to be drawn from only one distribution. The basis for clustering the positive data is the similarity between the members of a cluster in terms of discriminating features from the negative class. The goal, therefore, is to find hidden clusters of positive data while finding the optimal classifier within each cluster (see figure 5-1).

The framework for joint clustering and classification introduced in the work (Dai,

2015) is adopted in this chapter and a brief description is provided in the following.

The overall procedure consists of two main modules, namely a clustering module and a classification module. The classifier used is SLSVM and the number of clusters is an input parameter of the procedure.

Initially, the positive data points are randomly divided into multiple clusters. As the clustering of data is only assumed for the positive data points, all the negative data points are copied into every cluster and the optimal SLSVM for each cluster is calculated.

For each of the following iterations, first the positive data points are re-assigned to new clusters according to the following two conditions. Let there be L clusters and the current positive point to be x^+ where the current cluster assignment of the data point is l . The goal is to find the new cluster assignment l^* where:

1. $l^* = \underset{l}{argmax} \langle x_c^+, \beta_c^l \rangle$
2. $\langle x^+, \beta^{l^*} \rangle \geq \langle x^+, \beta^{l_i} \rangle \quad \forall l_i \in \{1, \dots, L\}$

Where x_c^+ denotes the subset of features of a data point x which are used for clustering. Therefore, each positive data point x^+ is assigned to the cluster where x_c^+ has the largest projection onto the corresponding classifier. Furthermore, the second condition is to ensure that the objective value of the overall optimization problem is monotonically non-increasing, guaranteeing convergence of the algorithm.

After re-assignment of all the positive data points, the SLSVM coefficients are updated using the new data points. The algorithm terminates if the assignment of the positive data points is not changed or the change in the total objective value is less than a threshold.

5.3 Performance Measures for Classification

The choice of the performance measure, dictated by the application, provides a basis for comparing different classifiers and has a pivotal role in the success of the classification on unseen test data. Each binary classification model predicts either of two classes for a data point based on its features where the correctness of the prediction can be assessed using the ground truth of the data point. In this section, a brief introduction to quantities that are commonly used in machine learning literature for assessing the performance of a model is presented.

5.3.1 Confusion Matrix

The four fundamental quantities that capture all aspects of the performance of a binary classification model on a dataset are discussed in the following. These quantities are used for deriving other performance measures:

- True Positive (TP): The number of positive data points that are correctly identified as positive.
- False Positive (FP): The number of negative data points that are incorrectly identified as positive.
- True Negative (TN): The number of negative data points that are correctly identified as negative.
- False Negative (FN): The number of positive data points that are incorrectly identified as negative.

The confusion matrix contains all four quantities TP, FP, TN and FN as a 2×2 matrix with the following format:

$$\text{Confusion Matrix} = \begin{array}{|c|c|} \hline \text{True Positive} & \text{False Positive} \\ \hline \text{False Negative} & \text{True Negative} \\ \hline \end{array}$$

The diagonal terms in the confusion matrix contain the number of correctly identified samples while the off-diagonal terms correspond to the error count of the classifier. In general, there is always a compromise between the number of positive samples correctly identified (TP) and the number of negative samples incorrectly labeled as positive (FP). Moreover, solely considering the total number of samples correctly identified may not be the optimal strategy in all applications. In this manner, the confusion matrix conveys an informative picture about different performance aspects of a classification model.

5.3.2 Accuracy, True Positive Rate and False Positive Rate

The accuracy is the proportion of total samples correctly identified :

$$\text{Acc} = \frac{TP + TN}{\text{Total}} \quad (5.2)$$

Where Total is the total number of samples.

Accuracy is one of the traditional measures to assess the overall performance of a classifier but can be misleading for the datasets with an unbalanced proportion of positive and negative samples. For instance, a classifier that predicts only positive label regardless of the input features will achieve 99% percent accuracy on a dataset of 99 positive and 1 negative samples.

True Positive Rate, or Sensitivity, is the proportion of the positive samples correctly identified:

$$\text{TPR} = \frac{TP}{TP + FN} \quad (5.3)$$

False Positive Rate, or False Alarm, is the proportion of negative samples incorrectly

identified:

$$\text{FPR} = \frac{FP}{FP + TN} \quad (5.4)$$

As mentioned before, there is a compromise between TPR and FPR where an ideal classifier will have a fairly high TPR while keeping the FPR at a minimum.

There are multiple other quantities such as Specificity or F1 score for measuring the performance of a classifier but in the remainder of this work the main focus will be on Receiver Operating Characteristic curve.

5.3.3 Receiver Operating Characteristic Curve

Receiver Operating Characteristic or ROC curve is a visual aid for showing how TPR and FPR change as one tunes the parameters of a classifier. Conventionally, the vertical and horizontal axes denote True positive and False positive rates respectively.

Many classifiers report a probability or a score for each test data point where higher scores signify higher confidence of the data point belonging to the positive class. By varying the threshold for which a classifier declares a data point positive, one gets a spectrum of (TPR, FPR) pairs that are plotted as the ROC curve.

Specifically, consider a classifier that randomly assigns positive label with probability p and negative label with probability $1 - p$ to any test sample. When $p = 0$ all the samples are given negative label leading to (TPR, FPR)=(0,0) while for $p = 1$ all the test samples are labeled as positive resulting in (TPR, FPR)=(1,1). It can be easily shown that for the values $p \in [0, 1]$ the (TPR, FPR) lie on a line between (0,0) and (1,1) (see figure 5.2).

The random classifier mentioned above provides the baseline for the performance and it is expected that the ROC curve of any trained model lie above the random classification line in ROC curve. Specifically, a superior classifier will yield much higher TPR for similar amount of FPR when compared to other classifiers.

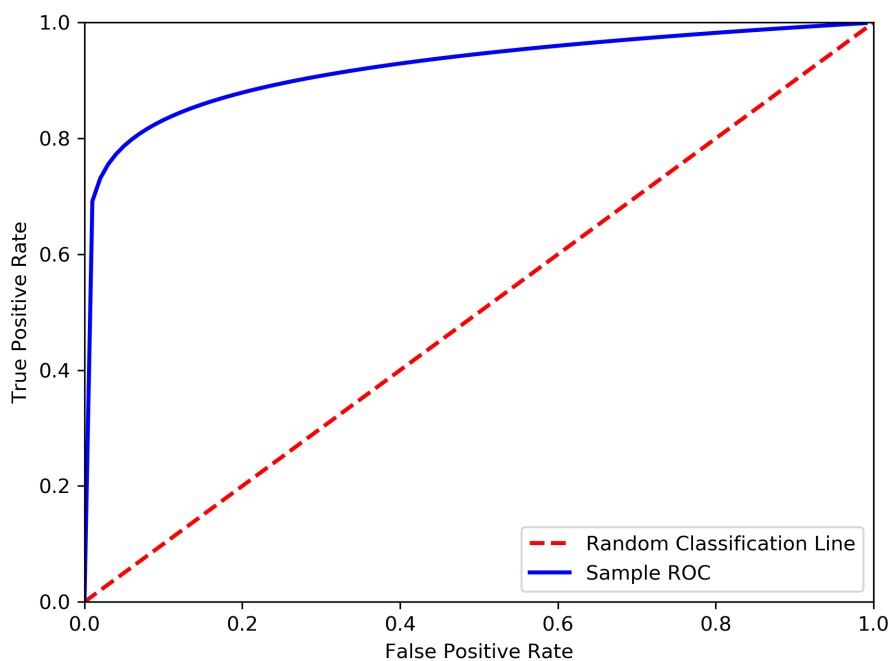


Figure 5.2: A sample Receiver Operating Characteristic curve.

One way to summarize the ROC curve is to calculate the Area Under the Curve (AUC) for the ROC plot. The perfect classification corresponds to AUC of 1 whereas the base line random classification corresponds to AUC of 0.5 and higher AUC values correspond to better overall performance of the classification model.

5.4 Interacting vs Non-interacting complexes

The goal of this part is to devise a docking-based approach for predicting whether a given pair of proteins interact or not. One of the main contributions of the present methodology is to predict the interaction in cases where there is no prior information about the potential surface of interaction between the protein partners. This is of significant importance as usually the crystal structures of individual protein partners are available but no co-crystallized structure of the partners together is available.

As mentioned, the present methodology is docking based; the input to the algo-

rithm are the docking results from PIPER program. The goal is to train a classification model predicting the likelihood that a protein pair interact.

5.4.1 Methodology

Crystallographic vs Biological Dimers Dataset

The dataset for validating our methodology is similar to the work in (Yueh et al., 2017). In this work, the goal is to discriminate between *Biological (real)* vs *Crystallographic (false)* Dimers.

One of the most prevalent and reliable means to determine atomic level structural data for protein complexes is X-ray Crystallography. In this experiment, a crystal from a solution of the proteins is synthesized where by illuminating x-ray through the crystal, one can calculate the 3D electron density map of the proteins in the crystal by analyzing the diffraction pattern of the x-ray. The electron density map can be further analyzed to construct a 3D model of the protein structure at atomic level.

Many proteins partners have similar or homologous structures where the sequence similarity between the ligand and receptor is relatively high to the point where the partners have identical structure. In fact, complexes where the two interacting units are identical are called *Dimers*. When performing X-ray crystallography on dimers, there is often a need for further analysis to determine whether the observed interaction between partners in the crystal is Biological or a byproduct of the experimental condition, such as the high concentration of the proteins (see figure 5-3). As further experimental validation of these interactions is often unavailable, distinguishing a biological interaction from one induced by the crystal structure for dimers has become a recognized problem in Bionformatics community for which several computational tools has been developed (Yueh et al., 2017).

The datasets for training and testing are those used by (Yueh et al., 2017). Specif-

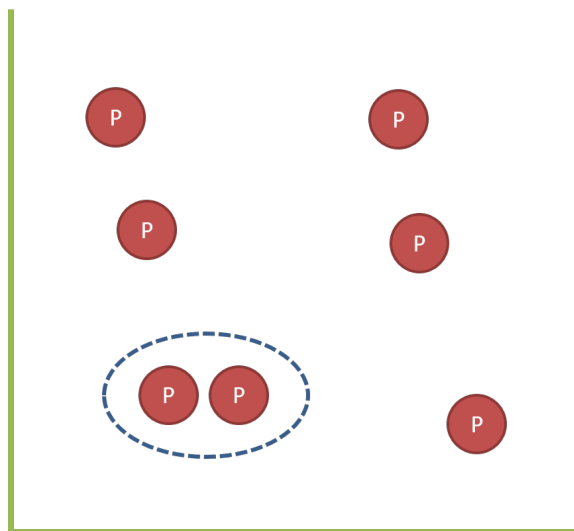


Figure 5-3: When performing X-ray crystallography on dimers, there is often a need for further analysis to determine whether the observed interaction between partners in the crystal is (denoted by blue dashed line) Biological or a byproduct of the experimental condition, such as the high concentration of the proteins.

ically, there are 120 Dimers as the positive data and 109 large interface Monomers as the negative data. The positive data is similar to the homodimers from the work (Ponstingl et al., 2003) whereas crystal dimers are mostly taken from (Bahadur et al., 2004). For testing, there are 293 biological dimers and 490 monomers which are manually taken from Protein Data Bank (Berman et al., 2000) which only includes a single type of protein structure, also known as Homodimers, with additional conditions outlined in (Yueh et al., 2017).

For both datasets, each data point includes the protein files corresponding to the 3D structures of receptor and ligand and the docking results from PIPER program which corresponds to the top energy conformations of rigid docking of the protein partners. The docking files are further processed to extract the features discussed in the following section.

Feature extraction

To extract the features for each protein pair, the outputs of docking from PIPER docking software are analyzed as follows. First, the top energy conformations of ligand from the docking stage are clustered using the Cluspro greedy clustering algorithm (Kozakov et al., 2017). This algorithm uses interface RMSD as the metric of distance where the members in a cluster are all close to a common center. Afterwards, the N largest clusters are retained and two sets of features are extracted:

- The cluster features that only depend on the members of each cluster.
- The global features that depend on all the top energy ligand conformations.

The cluster features are calculated for each of the N clusters individually as follows:

- Cluster size: The number of members within a cluster.
- The five energy components of PIPER program for the cluster center. The cluster center is defined as the member with the highest number of neighbors.
- The total energy of PIPER program for the cluster center. The total energy is a linear combination of the five energy components where the linear coefficients are parameters of PIPER program.
- The five energy components of PIPER program averaged over all cluster members.
- The total energy of PIPER program averaged over all cluster members.
- The average distance of cluster members from the cluster center.
- The variance of distances of cluster members from the center.

- The interface area of interaction for the cluster center. The interface area is defined as half the difference between the surface areas of (i) the ligand-receptor pair as a complex and (ii) the sum of individual surface areas of receptor and ligand. This quantity is calculated using Pymol molecular visualization (WL, 2002) program.

Additionally, SSDU program as described in chapter 4 was run on PIPER outputs to generate *refined* conformations. Note that SSDU was run only one step to reduce computation cost. The following features for each of the N clusters were generated:

- Principle Component Analysis (PCA) five eigenvalues which correspond to the highest variations in the data along the principle directions. (The center-to-center coordinate from the 6 rigid transformation coordinates is dropped due to low variation among different data points.)
- The three eigenvalues of the underestimator hessian matrix.
- The average value of 11 energy components over cluster members calculated in SSDU routine. These energy components correspond to more accurate and off-grid energy calculations as opposed to PIPER program.
- The average value of total energy of SSDU program over cluster members. The total energy is a linear combination of the 11 energy components where the linear coefficients are parameters of SSDU program (Zarbfian et al., 2018).

The global features are calculated using all the top energy conformations as follows:

- The average distance of the all the top energy conformations from the cluster center of the largest cluster.

- The variance of distances of the all the top energy conformations from the cluster center of the largest cluster.

5.4.2 Results

The top number of clusters to retain was set $N = 5$ as a compromise between including more information and avoiding over-fitting. As mentioned before, the dataset is the same as the work (Yueh et al., 2017) where the goal is to differentiate between Biological Dimers as positive data points versus Crystallographic ones as negative points.

ROC Curves

The performance of three classifiers SLSVM, Random Forest (Breiman, 2001) and ACC introduced earlier are reported in this section. The parameters of each classifier were tuned using cross-validation. Specifically, SLSVM “penalty for miss-classification” and “penalty for l_1 norm Sparsity” parameters were cross-validated on the sets $\{0.01, 0.1, 1, 10, 100, 1000\}$ and $\{1, 3, 5, 7, 9, 10\}$ respectively and Random Forest parameters “number of trees” and “minimum number of samples per leaf” were cross-validated on the sets $\{100, 500, 1000\}$ and $\{1, 10, 100\}$ respectively. The number of clusters to find for ACC algorithm was chosen to be 3 and the parameters corresponding to the SLSVMs “penalty for miss-classification” and “penalty for l_1 norm Sparsity” were cross-validated on the sets $\{0.01, 0.1, 1, 10, 100, 1000\}$ and $\{0.01, 0.1, 1, 10, 100, 1000\}$ respectively.

Random Forest and SLSVM classifiers attained AUC value of 0.98 whereas ACC classifier attained AUC value of 0.95. Overall, the relatively high AUC values of three different classifiers signifies that the features considered are highly predictive for discriminating Biological Dimers from Crystallographic ones.

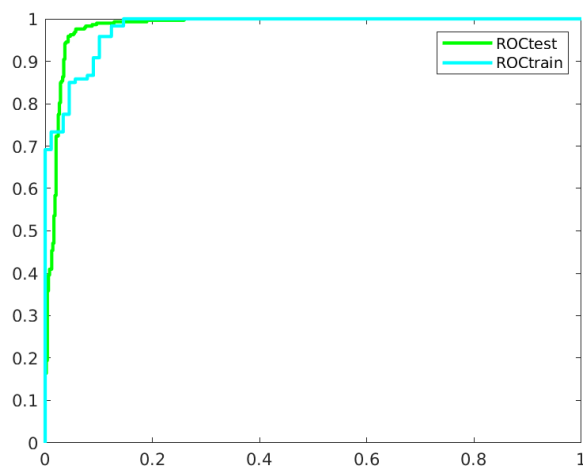


Figure 5·4: ROC curve for Random Forest on Biological vs Crystallographic Dimers dataset. The AUC on the test data is 0.98.

5.4.3 Feature Analysis

One pivotal question to answer is which features are the most informative for discriminating real Dimer interactions from fabricated ones. In this section, the results for three different approaches are presented:

Permuted Predictor Delta Error using Random Forest

The Random Forest was trained using the MATLAB function `TreeBagger`. This function provides a quantitative measure called Permuted Predictor Delta Error (PPDE) for signifying the importance of different features. Specifically, the trained model outputs a vector PPDE having the same size as the number of features where each element denotes the prediction error increase on testing data when the values of the feature are randomly permuted across different samples. Intuitively, this is an efficient manner to measure the effect of removing a feature while avoiding training a new model (see figure 5·7 and find features description in table 5.1). It is worth noting that out of 20 top features recognized using PPDE, only 4 features are not

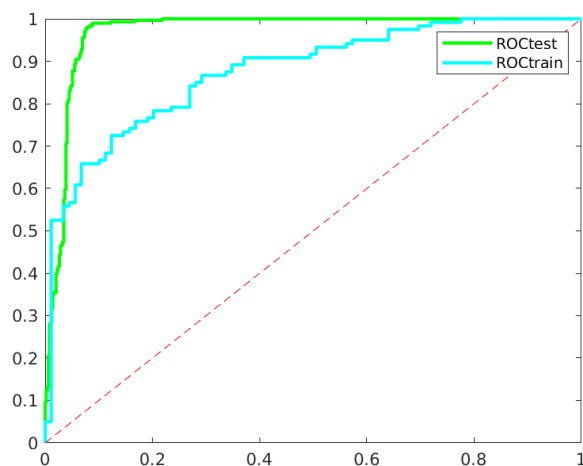


Figure 5.5: ROC curve for Sparse Linear SVM on Biological vs Crystallographic Dimers dataset. The AUC on the test data is 0.98.

energy related.

TE-average-C(num)	Average of total energy over the cluster members of the (num)th largest cluster of a complex.
TE-center-C(num)	Total energy of the cluster center of the (num)th largest cluster of a complex.
E5	DARS statistical potential energy (Chuang et al., 2008).
E3	Coulombic electrostatic potential.
E2	Attractive Vander Waals energy potential.
SDUEig(num)	The (num)th largest eigenvalue of the underestimator hessian matrix calculated using SSDU algorithm.

InterArea(<i>num</i>)	The interface area of the cluster center of the (<i>num</i>)th largest cluster of a complex.
Global-VarDist	The variance of distances of all the top energy conformations from the cluster center of the largest cluster of a complex.

Table 5.1: The feature description of the important features presented in section 5.4.3.

Feature Variation Among Clusters Using ACC

As mentioned before, one of the novelties of the ACC algorithm is that it finds hidden clusters among the positive data points. The conjecture is, each cluster of positive points is different from other positive clusters in terms of its characteristics, requiring a different classifier for discrimination from negative dataset. Consequently, it is insightful to examine the feature values of different clusters of positive data points.

Note that in figure 5-8 features of cluster 1, especially the energy values, are significantly different than that of cluster 2 and 3. This might imply that complexes of cluster 1 belong to a different category than the ones in the other two clusters.

5.5 Ranking of Clusters of Conformations

This part focuses on formulating a ranking scheme for the output models of protein docking servers. This a more generalized but similar approach to the cluster ranking scheme that was presented in chapter 4. The novelty of the current chapter lies in using additional features alongside other binary classification models to improve prediction accuracy of the machine learning models. Moreover, a feature analysis study is presented at the end.

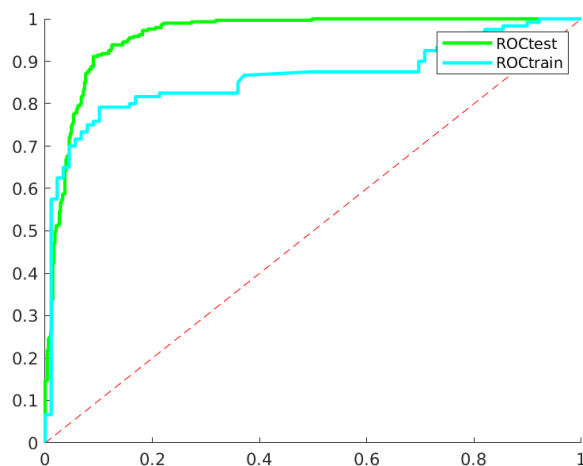


Figure 5-6: ROC curve for Alternating Clustering and Classification method on Biological vs Crystallographic Dimers dataset. The AUC on the test data is 0.95.

As Protein Docking is a considerably challenging problem, protein docking servers conventionally produce multiple, often 10, top docking predictions instead of a single solution for a given protein pair. Moreover, no clear preference is given for any of the predictions and all the predictions are presented as having the same likelihood of being the correct solution. This is of significant concern for the automated docking servers that generate ensembles of conformations that often contain a near-native prediction but the near-native prediction is not present in the final output. This is mainly due to the fact that these servers are currently not equipped with an accurate model for ranking and discriminating near-native solutions from non near-native ones. Specifically, Cluspro 2.0 generate 30 clusters of ligand conformation for a protein pair where output *predictions* are the *centers* of these clusters, i.e. members with the highest number of neighbors. Typically, the requirements dictate to limit the outputs to less than 30, usually 10, centers of top clusters when ranked according to size in a descending order. However, using only cluster size as the ranking criterion results

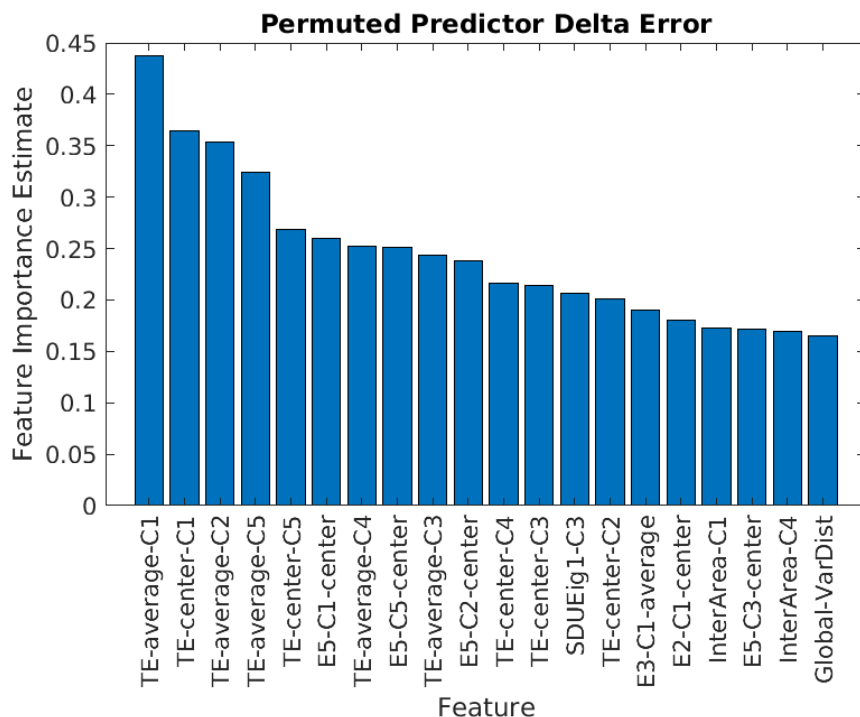


Figure 5-7: The highest Permuted Predictor Delta Error for the top 20 features using Random Forest on the Biological versus Crystallographic test data.

in significant performance deterioration. Specifically, the number of successful cases¹ on a benchmark of 230 protein-protein interactions drop 20% (Vreven et al., 2015) when reducing the number of predictions from 30 to 10. Consequently, there is a need for constructing more accurate ranking models that use more informative features to minimize the performance loss of the docking servers

Multiple approaches to this ranking problem have been devised but many share the same methodology. Specifically, the goal is to construct a custom potential or scoring function that produces scores for different predictions of a protein pair where the conformations with higher scores are more likely to have higher quality. A training set is constructed from a set of protein pairs where the quality of the docking server

¹A successful case is defined as a complex having an Acceptable or better CAPRI quality solution among the docking server output models.

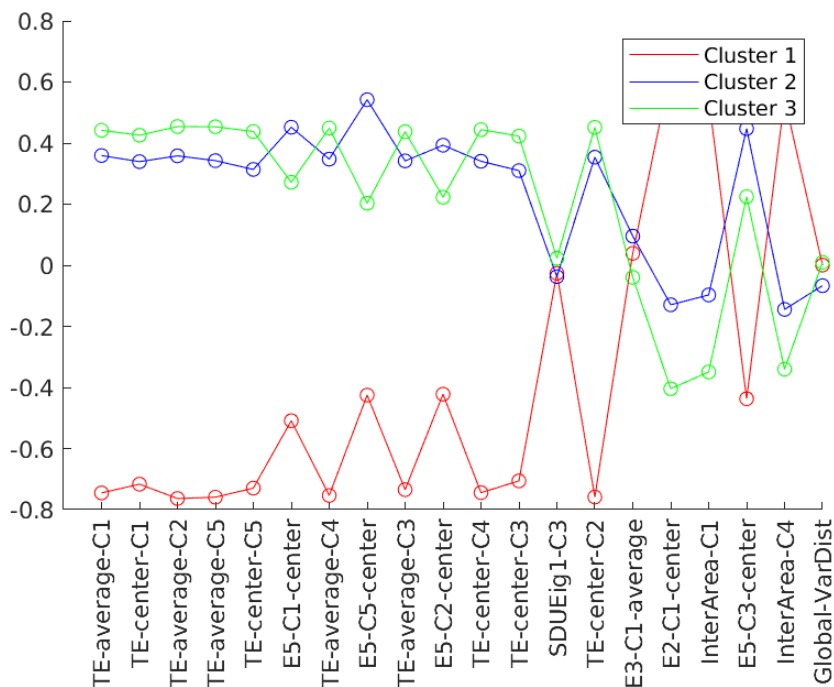


Figure 5-8: The normalized feature value difference among 3 clusters of positive class identified using ACC algorithm. The selected features are the important features identified using Random Forest in section 5.4.3 and the values of the features are averaged over the members of each cluster. The feature descriptions are given in table 5.1

predictions are known and the parameters of the scoring function are tuned through different optimization methods such as least-squares fitting (Moal and Fernandez-Recio, 2013) or Linear Programming (Pierce and Weng, 2007). The scoring function itself can be a linear combination of multiple energy potential functions (Camacho et al., 2000a) or a loss function of incorrect ranking over training data (Cheng et al., 2007). As mentioned in (Moal et al., 2017), the ranking scores of the predictions have relevance only when considering a specific protein pair. For instance the ranking score of a prediction for a protein pair is only comparable to the other predictions of the same protein pair whereas the comparison of the scores of predictions over different protein pairs is meaningless. Moreover, when training the ranking model,

improvement among the higher ranks should be prioritized; i.e. improvement of rank from 10 to 1 should be prioritized over going from 40 to 31.

In this work, the goal is to construct a methodology for ranking docking predictions of protein docking servers similar to (Moal et al., 2017). The ranking problem is converted into a classification problem where the data points for the classification are considered as the comparison of different prediction pairs. Similar to the previous chapters of this thesis, the application will be focused on Cluspro 2.0 (Kozakov et al., 2017) server and the train-test datasets are produced on benchmark of complexes for which the quality of predictions and hence the correct order of ranking are known a priori.

5.5.1 Methodology

Train and Test Data

To make a formal comparison between the results between the present work and those reported in the literature, the training and testing sets were defined the same as the work (Moal et al., 2017). The training set are all the 176 protein pairs from PPI benchmark 4.0 (Howook Hwang and Weng, 2010) and the test set are the 54 additional protein pairs in PPI benchmark 5.0 (Vreven et al., 2015). As mentioned before, the raw features are generated using Cluspro 2.0 docking server where each protein pair is docked and the top 30 largest clusters are retained.

Feature Extraction

The feature extraction is performed in two stages. The first stage is to generate features for top clusters of each complex. In the second stage these features are combined to generate pairwise comparison features.

To generate cluster features, each protein complex is docked and the top energy conformations are retained and clustered. Moreover, the top 30 largest clusters are

retained and the following features, similar to section 5.4.1, are extracted:

- Cluster size
- The five energy component of PIPER program for the cluster center.
- The total energy of PIPER program for the cluster center.
- The five energy component of PIPER program averaged over all cluster members.
- The total energy of PIPER program averaged over all cluster members.
- The average distance of the cluster members from the cluster center.
- The variance of distances of the cluster members from the center.
- The interface area of interaction for the cluster center.

Additionally, SSDU features are calculated as before:

- PCA five eigenvalues.
- The three eigenvalues of the underestimator hessian matrix.
- The average value of 11 energy components over cluster members calculated in SSDU routine.
- The average value of total energy of SSDU program over cluster members.

In the second stage, clusters of the same complex are classified into different CAPRI quality categories (Basu and Wallner, 2016) and all pairs of clusters (i, j) for i and j having different quality categories are considered. Next, a new data point ij is constructed as the comparison of clusters i and j ; i.e. the feature is the *combination* of the two cluster features and the label is whether cluster i has a better CAPRI

category than j . To *combine* the features, one can calculate the difference as in (Moal et al., 2017) : $x_i - x_j$ or concatenate the two feature vectors as in (Pfeiffenberger et al., 2016) : (x_i, x_j) . By taking the difference, one can avoid increasing the dimension and hence possibly avoid overfitting and excessive computation time whereas concatenation preserve all the information from the original data hence leading to potentially superior ranking models. Both approaches for combining the features are explored in this work.

Moreover for each pairwise feature the label is randomly generated to be positive or negative with equal probability and the corresponding combined label is adjusted accordingly. This leads to generate a balanced dataset having similar number of positive and negative samples.

Training a Classifier and Generating a Ranked List

As mentioned in the previous section, it was discussed how to convert training a ranking model to a classification one by constructing the pairwise comparisons from a ranked list. The remaining question, however, is how to construct a ranked list from the pairwise comparison predictions on a test set. In this section, two different approaches in the literature are discussed.

In the work (Pfeiffenberger et al., 2016) after constructing the pairwise comparison features, an Extremely Randomized Tree (Pierre Geurts and Wehenkel, 2006) is trained and validated. The trained classification model is tested on the test data to generate pairwise comparison predictions. For each tested cluster, the pairwise predictions are used to calculate the number of times the cluster is predicted to be better than any other cluster, namely the number of *wins*. Finally, the ranked list is the ordered list of testing clusters sorted in descending order according to the count of the wins.

The work (Moal et al., 2017) takes a different approach where the training stage is performed using an ensemble of SVMs where each SVM is trained on a random Bootstrap sampling of the pairwise comparison transformation of the training data, similar to a Random Forest model (Breiman, 2001). The performance of all trained SVMs are evaluated on the validation set using a novel approach as discussed below:

The total score of each SVM S is calculated as a sum of the scores of the SVM on different complexes of the dataset s_i , $\forall i \in \{1, \dots, n_t\}$ compared to the average score over all SVMs where n_t is the number of protein complexes in the validation set:

$$S = \sum_{i=1}^{n_t} (s_i - \bar{s}_i) \quad (5.5)$$

This score favors the SVMs that perform better on difficult complexes and attributes a low score to the SVMs that struggle on complexes that other SVMs perform relatively well.

To calculate each of the s_i , first the ligand conformations are clustered and the resulting clusters are ranked according to their CAPRI quality scores. The score of each cluster is defined as the maximum quality score calculated over their members. The rank r for each complex is then defined as the rank of the first cluster in the ranked list having an acceptable or better CAPRI quality score for $r \in \{1, \dots, n_c\}$ where n_c is the total number of clusters for each complex. The s_i score for the complex is calculate as follows:

$$s_i = \frac{\log_{10}(n_c) - \log_{10}(r)}{\log_{10}(n_c)} \quad (5.6)$$

Note that s_i can range from zero for a complex where only the last cluster contains an acceptable or better quality solution to one for a complex having an acceptable or better quality solution in the first cluster. Additionally, the use of logarithm in calculating of s_i leads to SVMs that prioritize correct rank prediction among the lower ranks; i.e. improvement of rank prediction from 11 to 1 has a higher impact on s_i

compared to improvement from 110 to 100.

After training the classifiers, a limited number of top scoring SVMs are retained. To generate a ranked list for each of the test data protein pairs, first all the retained classifiers are employed to generate pairwise comparison predictions for the clusters of each protein complex separately. These pairwise predictions are combined using the Shulze electoral voting system (Schulze, 2010) as follows: First a weighted directed graph is constructed where the nodes represent different clusters of the protein complex and an edge e_{ij} from cluster i to j has a weight equal to the number of times the top classifiers has declared cluster i to be superior to cluster j . Afterwards, the *strongest path* for every cluster pair (i, j) is calculated where the strongest path is the path, of all the paths, between i and j that has the highest weight. The weight of a path is defined as the minimum of edge weight over constituent edges of a path.

Let the weight of the strongest path between (i, j) be denoted w_{ij}^s . If $w_{ij}^s > w_{ji}^s$ then it is said that cluster i is preferred to j according to the ensemble of classifiers. Furthermore, it can be shown (Schulze, 2010) that preference relation is transitive: if $w_{ij}^s > w_{ji}^s$ and $w_{jk}^s > w_{kj}^s$ then it is implied $w_{ik}^s > w_{ki}^s$. Therefore, a ranked list of the clusters can be constructed from the pairwise comparisons predictions.

Note that the latter methodology (Moal et al., 2017) has two significant advantage: (i) An ensemble of classifiers which *directly contribute to generating the ranked list* reduce the bias toward the training set, potentially alleviating issues regarding overfitting. (ii) The Shulze method in (Moal et al., 2017) is an example of a Condorcet voting system with a mechanism to avoid Condorcet Paradox while the the work in (Pfeifferberger et al., 2016) employs majority voting system that can easily lead to choosing an “unpopular” cluster with fairly low “support” when the preferences are uniformly divided among different clusters.

Consequently, the training and testing in this work is a an adaptation of (Moal

et al., 2017) with the differences as follows:

- The ensemble of pairwise classifier in (Moal et al., 2017) are constructed from SVMs while in this work SLSVM, Random Forest and Logistic Regression are going to be considered for building the ensemble and their relative performance will be reported.
- In the work (Moal et al., 2017), the pairwise feature for a cluster pair i and j is constructed from the difference of the individual feature vectors whereas in this work the concatenation of the feature vectors will also be considered and the performance change will be analyzed.
- In the work (Moal et al., 2017) The quality of a cluster is calculated using the conformation having the highest quality in the cluster whereas in this work the quality of a cluster will be determined by the quality of its cluster center according to the previous conventions of Cluspro 2.0 server.

5.5.2 Results

This project is in the process of being completed and it is therefor left as a future work to implement and test the algorithm on a benchmark.

5.6 Conclusion

In this chapter, the application of machine learning framework for addressing two specific problems in Protein Docking was discussed. Firstly, a machine learning model for discriminating between interacting and non-interacting protein pairs was developed. The algorithm was successfully tested on a benchmark for discrimination between Biological and Crystallographic Dimers and a thorough analysis of descriptors was performed to identify the most informative ones. Secondly, a ranking scheme for

output predictions of a protein docking server is devised. The ranking model was translated into a classification model by constructing *pairwise comparison* features and two relevant state-of-the-art approaches were discussed in detail. Moreover, advantages and disadvantages of each approach were pointed out and the procedure for implementation and adaptation of the aforementioned approaches for Cluspro 2.0 web server was discussed.

Chapter 6

Conclusion

Proteins are essential to many biological process such as gene regulation and metabolism where they act in pairs to carry out these functions in living organisms. Protein Docking is the study of how proteins interact and form complexes. In protein docking, the goal is to find the most likely structure formed by two individual proteins.

While being the gold standard for verifying protein-protein interactions (PPI), experimental methods such as NMR and Xray-Crystallography are time consuming, expensive and not applicable to all classes of the proteins. Computational methods, on the other hand, allow for much larger scale analysis of PPIs while requiring considerably less resources. Moreover, they provide more insight into the mechanism of PPIs leading to innovations such as drug engineering for the Drug Discovery process.

Due to challenges involved in solving protein docking as an optimization problem, docking protocols employ a multi-stage approach where in the initial stage the energy landscape of free binding energy is globally sampled and the in following stages representatives identified from the *global sampling* stage are further *refined*.

The first part of this work, discussed in chapter 3, focused on the local optimization component of the refinement stage. The main contribution of the chapter was to define a new metric, closely related to RMSD, on the space of rigid transformation that makes rotation and translation components *infinitesimally* “similar”. It is left as the future work to examine whether the new manifold leads to computation efficiency in gradient-based optimization for practical applications. For instance, it can be shown

that by defining a new metric on a manifold, one has to accordingly adjust/scale the gradient when performing gradient-based optimization. Moreover, one can control the amount of RMSD change for each step of the optimization and avoid pitfalls such as too large or too small steps sizes. It is therefore specifically interesting to examine the performance of optimization routines with and without the suggested modifications on a challenging benchmark protein complexes where (i) there are initially steric clashes between protein partners or (ii) the change in RMSD is fairly small during the last steps of the traditional optimization algorithms. The results in chapter 3 are applicable to arbitrary dimensions and it might prove useful to explore the application of the work of chapter 3 in fields such as data science where the dimensions of the data are orders of magnitude larger.

The second part of the work, discussed in chapter 4, focused on a resampling technique called Subspace Semi-Definite Underestimation (SSDU) for the refinement stage. The algorithm discussed in the chapter takes as the input the top energy conformations from the global sampling stage and outputs a new ensemble of conformations with higher quality solutions, increasing the chance of choosing a high quality representative from the ensemble. The algorithm builds upon a previous work and introduces four innovations: (i) Clustering of the input conformations using a density-based clustering algorithm to remove noise from the data (ii) Dimensionality reduction using Principle Component Analysis to reduce computation cost and further dampen the noise of input the data (iii) Under-estimation of the energy landscape using a general class of SOS-convex polynomials to estimate the global minimum of the energy landscape and (iv) Resampling in the vicinity of global minimum of the under-estimator to generate near-native conformations. It was shown on a comprehensive benchmark of protein complexes that the new ensemble generated by SSDU, contains considerably more number of quality solutions among different categories of

qualities defined by international CAPRI competition. Furthermore, it was shown that SSDU can enrich the number of quality solutions among the top 10, 5 and 3 largest clusters of the output conformations when compared to Cluspro and SDU thereby increasing the chance of picking a high quality representative using other algorithms.

The third part of this thesis, discussed in chapter 5, focused on two application of machine learning to protein docking. The first part focused on a data-driven framework for discriminating between Biological Dimers and Crystallographic ones in the absence of additional experimental data. In this project, predictive models using (i) Random Forest, (ii) Sparse Linear SVM and (iii) Alternating Clustering and Classification machine learning models were trained on a set of approximately 230 data points and was successfully tested on a set 780 Biological and Crystallographic dimers. Overall, it was shown that all three machine learning models achieved considerably high Area Under the Curve (AUC) above 95% on the testing set, verifying that the extracted feature for discrimination are highly informative. Furthermore, the importance of different features for the success of predictive models were examined using a metric calculated by Random Forest model where it was discovered that majority of the top feature, in terms of contribution to the success of the model, were related to the energy calculations. Furthermore, it was shown that using ACC algorithm there is a distinct clustering of the positive data into two clusters and it might prove informative to further examine the biological interpretations of such division of the positive data. In the second part of the chapter, a ranking procedure based on state-of-the-art works were devised for the outputs of Cluspro web-server. The input to the procedure are the top clusters from Cluspro and the goal is to rank the clusters so that the chance of picking high quality representatives from the top rank clusters are higher than that of the lower ranked clusters. It was proposed that the ranking

problem be converted to a classification problem where an ensemble of classifiers are trained on data points constructed from pairwise comparison of the original cluster ranking data. Furthermore, for ranking the clusters of a given protein complex as the test data, the pairwise comparison predictions out of the trained ensemble of classifiers are converted to a ranked list of clusters through a procedure called Shulze electoral voting system. The implementation and evaluation of the protocol was left as a future work.

References

- Agrawal, N. J., Helk, B., and Trout, B. L. (2014). A computational tool to predict the evolutionarily conserved protein-protein interaction hot-spot residues from the structure of the unbound protein. *FEBS Letters*, 588(2):326–333.
- Ahmadi, A. A., Olshevsky, A., Parrilo, P. A., and Tsitsiklis, J. N. (2010). Np-hardness of deciding convexity of quartic polynomials and related problems. *Mathematical Programming*, 137(1-2):453–476.
- Ahmadi, A. A. and Parrilo, P. A. (2013). A complete characterization of the gap between convexity and sos-convexity. *SIAM Journal on Optimization*, 23(2):811–833.
- Andrusier, N., Mashiach, E., Nussinov, R., and Wolfson, H. J. (2008). Principles of flexible protein-protein docking. *Proteins*, 73(2):271 – 289.
- Andrusier, N., Nussinov, R., and Wolfson, H. (2007). FireDock: Fast interaction refinement in molecular docking. *Proteins*, 69(1):139–59.
- Bahadur, R. P., Chakrabarti, P., Rodier, F., and Janin, J. (2004). A dissection of specific and non-specific proteinprotein interfaces. *Journal of Molecular Biology*, 336(4):943–955.
- Baldi, P. (2001). *The Machine Learning Approach*. MIT press.
- Basu, S. and Wallner, B. (2016). DockQ: A quality measure for protein-protein docking models. *Plos One*, page doi: 10.1371/journal.pone.0161879.
- Berman, H., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. (2000). The protein data bank. *Nucleic Acids Research*, 28:235–242. <http://www.rcsb.org/pdb/home/home.do>.
- Borchers, B. (1999). CSDP, a C library for semidefinite programming. *Optimization Methods and Software*, 11(1-4):613–623.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.
- Brooks, B. R., Bruccoleri, R. E., Olafson, D. J., States, D., Swaminathan, S., and Karplus, M. (1983). CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *Journal of Computational Chemistry*, 4:187–217.

- Bryngelson, J., Onuchic, J. N., Socci, N. D., and Wolynes, P. G. (1995). Funnels, pathways, and the energy landscape of protein-folding - a synthesis. *Proteins*, 21(3):167–195.
- Bullo, F. and Lewis, A. D. (2005). *Geometric Control of Mechanical Systems: Modeling, Analysis, and Design for Simple Mechanical Control Systems*. Springer.
- Camacho, C. J., Gatchell, D. W., Kimura, S. R., and Vajda, S. (2000a). Scoring docked conformations generated by rigidbody proteinprotein docking. *Proteins*, 40(3):525–537.
- Camacho, C. J., Kimura, S. R., DeLisi, C., and Vajda, S. (2000b). Kinetics of desolvation-mediated protein-protein binding. *Biophysical Journal*, 78(3):1094–1105.
- Camacho, C. J., Weng, Z., Vajda, S., and DeLisi, C. (1999). Free energy landscapes of encounter complexes in protein-protein association. *Biophysical Journal*, 76(3):1166–1178.
- Cheng, T., Li, Q., Zhou, Z., Wang, Y., and Bryant, S. H. (2012). Structure-based virtual screening for drug discovery: a problem-centric review. *AAPS Journal*, 14(1):133 – 141.
- Cheng, T. M., Blundell, T. L., and FernandezRecio, J. (2007). pyDock: Electrostatics and desolvation for effective scoring of rigidbody proteinprotein docking. *Proteins*, 68(2):503–515.
- Chicco, D. (2017). Ten quick tips for machine learning in computational biology. *BioData Mining*, 10(35):doi: 10.1186/s13040-017-0155-3.
- Chuang, G. Y., Kozakov, D., Brenke, R., Comeau, S. R., and Vajda, S. (2008). DARS (Decoys As the Reference State) potentials for protein-protein docking. *Biophys. J.*, 95(9):4217–4227.
- Clore, G. M. (2008). Visualizing lowly-populated regions of the free energy landscape of macromolecular complexes by paramagnetic relaxation enhancement. *Molecular BioSystems*, 4(11):1058–1069.
- Dai, W. (2015). *Detection and Prediction Problems with Application in Personalized Health Care*. PhD thesis, Boston University.
- Dill, K. (1999). Polymer principles and protein folding. *Protein Science*, 8(6):1166–1180.

- Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD'96*, pages 226–231. AAAI Press.
- Fawzi, N. L., Doucleff, M., Suh, J.-Y., and Clore, G. M. (2010). Mechanistic details of a protein–protein association pathway revealed by paramagnetic relaxation enhancement titration measurements. *Proceedings of the National Academy of Sciences*, 107(4):1379–1384.
- Friesner, R. A., Banks, J. L., Murphy, R. B., Halgren, T. A., Klicic, J. J., Mainz, D. T., Repasky, M. P., Knoll, E. H., Shelley, M., Perry, J. K., Shaw, D. E., Francis, P., and Shenkin, P. S. (2004). Glide: a new approach for rapid, accurate docking and scoring. 1. method and assessment of docking accuracy. *Journal of Medicinal Chemistry*, 47(7):1739–1749.
- Gray, J. J., Moughon, S., Wang, C., Schueler-Furman, O., Kuhlman, B., Rohl, C. A., and Baker, D. (2003). Protein-protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations. *J. Molecular Biology*, 331(1):281–299.
- Gwak, S., Kim, J., and Park, F. C. (2003). Numerical optimization on the Euclidean group with applications to camera calibration. *IEEE Transactions on Robotics and Automation*, 19(1):65–74.
- Heo, L., Lee, H., and Seok, C. (2016). GalaxyRefineComplex: Refinement of protein-protein complex model structures driven by interface repacking. *Scientific Reports*, 6(32153).
- Howook Hwang, Thom Vreven, J. J. and Weng, Z. (2010). Protein–protein docking benchmark version 4.0. *Proteins*, 78(15):3111–3114.
- Huang, Y., Liu, S., Guo, D., Li, L., and Xiao, Y. (2013). A novel protocol for three-dimensional structure prediction of RNA-protein complexes. *Scientific reports*, 3(1887).
- Hughes, J., Rees, S., Kalindjian, S., and Philpott, K. (2011). Principles of early drug discovery. *British Journal of Pharmacology*, 162(6):1239 – 1249.
- Iwahara, J. and Clore, G. M. (2006). Detecting transient intermediates in macromolecular binding by paramagnetic NMR. *Nature*, 440(7088):1227–1230.
- Janin, J. (2005). Assessing predictions of protein-protein interaction: The capri experiment. *Protein Science*, 14(2):278–283.

- Johnson, C. J. (1999). Diffusion ordered nuclear magnetic resonance spectroscopy: principles and applications. *Progress in nuclear magnetic resonance spectroscopy*, 34(3-4):203–256.
- Jones, G., Willett, P., Glen, R. C., Leach, A. R., and Taylor, R. (1997). Development and validation of a genetic algorithm for flexible docking. *Journal of Molecular Biology*, 267(3):727–748.
- Jorgensen, W. (1991). Rusting of the lock and key model for protein-ligand binding. *Science*, 254(5034):954 – 955.
- Katchalski-Katzir, E., Shariv, I., Eisenstein, M., Friesem, A. A., Aflalo, C., and Vakser, I. A. (1992). 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.
- Kortemme, T., Morozov, A. V., and Baker, D. (2003). An orientation-dependent hydrogen bonding potential improves prediction of specificity and structure for proteins and protein–protein complexes. *Journal of Molecular Biology*, 326(4):1239–1259.
- Kozakov, D., Beglov, D., Bohnuud, T., Mottarella, S. E., Xia, B., Hall, D. R., and Vajda, S. (2013). How good is automated protein docking? *Proteins*, 81(12):2159 – 2166.
- Kozakov, D., Brenke, R., Comeau, S. R., and Vajda, S. (2006). PIPER: an FFT-based protein docking program with pairwise potentials. *Proteins*, 65(2):392–406.
- Kozakov, D., Hall, D. R., Xia, B., Porter, K. A., Padhorny, D., Yueh, C., Beglov, D., and Vajda, S. (2017). The ClusPro web server for protein-protein docking. *Nature Protocols*, 12(2):255–278.
- Kozakov, D., Li, K., Hall, D., Beglov, D., Zheng, J., Vakili, P., Schueler-Furman, O., Paschalidis, I., Clore, G. M., and Vajda, S. (2014). Encounter complexes and dimensionality reduction in protein-protein association. *eLife J*, page DOI: 10.7554/eLife.01370.001.
- Kuntz, I. D., M. Blaney, J., J. Oatley, S., Langridge, R., and E. Ferrin, T. (1982). A geometric approach to macromolecule-ligand interactions. *Journal of Molecular Biology*, 161(2):269–288.
- Larraaga, P., Calvo, B., Santana, R., Bielza, C., Galdiano, J., Inza, I., Lozano, J. A., Armaanzas, R., Guzmán Santaf, A. P., and Robles, V. (2006). Machine learning in bioinformatics. *Briefings in Bioinformatics*, 7(1):86–112.

- Leopold, P. E., Montal, M., and Onuchic, J. N. (1992). Protein folding funnels: a kinetic approach to the sequence-structure relationship. *Proceedings of the National Academy of Sciences*, 89(18):8721–8725.
- Malod-Dognin, N., Ban, K., and Prulj, N. (2017). Unified alignment of protein-protein interaction networks. *Scientific Reports*, 7(953).
- Mamonov, A. B., Moghadasi, M., Mirzaei, H., Zarbafian, S., Grove, L., Bohnuud, T., Vakili, P., Paschalidis, I. C., Vajda, S., and Kozakov, D. (2016). Focused grid-based resampling for protein docking and mapping. *Journal of Computational Chemistry*, 37(11):961–970.
- McCammon, J. (1998). Theory of biomolecular recognition. *Current Opinion in Structural Biology*, 8(2):245–249.
- Mirzaei, H., Beglov, D., Paschalidis, I. C., Vajda, S., Vakili, P., and Kozakov, D. (2012). Rigid body energy minimization on manifolds for molecular docking. *Journal of Chemical Theory and Computation*, 8(11):4374 – 4380.
- Mirzaei, H., Kozakov, D., Beglov, D., Paschalidis, I. C., Vajda, S., and Vakili, P. (2014). A new approach to rigid body minimization with application to molecular docking. In *IEEE Conference on Decision and Control*, pages 2983–2988.
- Mirzaei, H., Zarbafian, S., Villar, E., Mottarella, S., Beglov, D., Vajda, S., Paschalidis, I. C., Vakili, P., and Kozakov, D. (2015). Energy minimization on manifolds for docking flexible molecules. *Journal of Chemical Theory and Computation*, 11(3):1063 – 1076.
- Mishra, B. and Sepulchre, R. (2016). Riemannian preconditioning. *SIAM Journal on Optimization*, 26(1):635–660.
- Moal, I. H., Barradas-Bautista, D., Jimnez-Garca, B., Torchala, M., van der Velde, A., Vreven, T., Weng, Z., Bates, P. A., and Fernandez-Recio, J. (2017). Irappa: Information retrieval based integration of biophysical models for protein assembly selection. *Bioinformatics*, 33(2):1806–1813.
- Moal, I. H. and Fernandez-Recio, J. (2013). Intermolecular contact potentials for protein-protein interactions extracted from binding free energy changes upon mutation. *Journal of Chemical Theory and Computation*, 9(8):3715–3727.
- Moghadasi, M., Kozakov, D., , Vakili, P., Vajda, S., and Paschalidis, I. (2013). A new distributed algorithm for side-chain positioning in the process of protein docking. in *Proceedings of 52nd IEEE Conference on Decision and Control, Firenze, Italy*.

- Moghadasi, M., Mirzaei, H., Mamonov, A., Vakili, P., Vajda, S., Paschalidis, I. C., and Kozakov, D. (2015). The impact of side-chain packing on protein docking refinement. *Journal of Chemical Information and Modeling*, 55(4):872 – 881.
- Murray, R. M., Li, Z., and Sastry, S. S. (1994). *A Mathematical Introduction to Robotic Manipulation*. CRC Press Taylor and Francis Group.
- Nan, F., Moghadasi, M., Vakili, P., Vajda, S., Kozakov, D., and Paschalidis, I. (2014). A subspace semi-definite programming-based underestimation (ssdu) method for stochastic global optimization in protein docking. *in Proceedings of 53rd IEEE Conference on Decision and Control*, page doi: 10.1109/CDC.2014.7040111.
- Pagadala, N. S., Syed, K., and Tuszynski, J. (2017). Software for molecular docking: a review. *Biophysical Reviews*, 9(2):91–102.
- Park, F. C., Kim, J., and Kee, C. (2000). Geometric descent algorithms for attitude determination using the global positioning system. *Journal of Guidance, Control, and Dynamics*, 23(1):26–33.
- Paschalidis, I. C., Shen, Y., Vakili, P., and Vajda, S. (2007). SDU: A semi-definite programming-based underestimation method for stochastic global optimization in protein docking. *IEEE Trans. Automat. Contr.*, 52(4):664–676.
- Pfeiffenberger, E., Chaleil, R. A., Moal, I. H., and Bates, P. A. (2016). A machine learning approach for ranking clusters of docked protein-protein complexes by pairwise cluster comparison. *Proteins*, 85(3):528–543.
- Phillips, A., Rosen, J., and Dill, K. (2001). *From Local to Global Optimization*, chapter Convex Global Underestimation for Molecular Structure Prediction, pages 1–18. Kluwer Academic Publishers.
- Pierce, B. and Weng, Z. (2007). ZRANK: Reranking protein docking predictions with an optimized energy function. *Proteins*, 67(4):1078–86.
- Pierce, B. G., Wiehe, K., Hwang, H., Kim, B.-H., Vreven, T., and Weng, Z. (2014). ZDOCK server: interactive docking prediction of protein-protein complexes and symmetric multimers. *Bioinformatics*, 30(12):1771 – 1773.
- Pierre Geurts, D. E. and Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, 63(1):3–42.
- Ponstingl, H., Kabir, T., and Thornton, J. M. (2003). Automatic inference of protein quaternary structure from crystals. *Journal of Applied Crystallography*, 36(5):1116–1122.

- Popov, P. (2015). *Nouvelles mthodes de calcul pour la prdiction des interactions protine-protine au niveau structural*. PhD thesis, Universite De Grenoble.
- Rivas, J. D. L. and Fontanillo, C. (2010). Protein-protein interactions essentials: Key concepts to building and analyzing interactome networks. *PLoS Computational Biology*, 6(6):1 – 8.
- Safari-Alighiarloo, N., Taghizadeh, M., Rezaei-Tavirani, M., Goliaei, B., and Peyvandi, A. A. (2014). Protein-protein interaction networks (PPI) and complex diseases. *Gastroenterology and Hepatology From Bed to Bench*, 7(1):17–31.
- Schaefer, M. and Karplus, M. (1996). A Comprehensive Analytical Treatment of Continuum Electrostatics. *The Journal of Physical Chemistry*, 100(5):1578–1599.
- Schulze, M. (2010). A new monotonic, clone-independent, reversal symmetric, and condorcet-consistent single-winner election method. *Social Choice and Welfare*, 36(2):267–303.
- Schlkopf, B., Tsuda, K., and Vert, J.-P., editors (2004). *Kernel Methods in Computational Biology*. MIT Press.
- Selig, J. M. (2005). *Geometric Fundamentals of Robotics*. Springer.
- Selzer, T., , and Schreiber, G. (2001). New insights into the mechanism of protein-protein association. *Proteins*, 45(3):190 – 198.
- Sevimoglu, T. and Arga, K. Y. (2014). The role of protein interaction networks in systems biomedicine. *Computational and Structural Biotechnology*, 11(18):22–27.
- Shapovalov, M. and Dunbrack Jr, R. (2011). A smoothed backbone-dependent rotamer library for proteins derived from adaptive kernel density estimates and regressions. *Structure*, 19(6):844–858.
- Shen, Y., Paschalidis, I. C., Vakili, P., and Vajda, S. (2008). Protein Docking by the Underestimation of Free Energy Funnels in the Space of Encounter Complexes. *PLoS Computational Biology*, 4(10).
- Smyth, M. S. and Martin, J. H. J. (2000). x Ray crystallography. *Molecular Pathology*, 53(1):8–14.
- Sudhaa, G., Nussinov, R., and Srinivasan, N. (2014). An overview of recent advances in structural bioinformatics of proteinprotein interactions and a guide to their principles. *Progress in Biophysics and Molecular Biology*, 116(2-3):141–150.
- Tahir, M. and Haya, M. (2017). Machine learning based identification of proteinprotein interactions using derived features of physiochemical properties and evolutionary profiles. *Artificial Intelligence in Medicine*, 78:61–71.

- Tarca, A. L., Carey, V. J., wen Chen, X., Romero, R., and Drghici, S. (2007). Machine learning and its applications to biology. *Plos Computational Biology*, 3(6):e116.
- Tovchigrechko, A. and Vakser, I. (2001). How common is the funnel-like energy landscape in protein-protein interactions? *Protein Science*, 10(8):1572–1583.
- Tovchigrechko, A. and Vakser, I. A. (2006). GRAMM-X public web server for protein-protein docking. *Nucleic Acids Research*, 34:W310 – W314.
- Tron, R., Terzis, A., and Vidal, R. (2011). Distributed consensus algorithms for image-based localization in camera sensor networks. In et al., B. B., editor, *Distributed Video Sensor Networks*, chapter 20, pages 289–302. Springer-Verlag.
- Tron, R. and Vidal, R. (Dec. 2014). Distributed 3-D localization of camera sensor networks from 2-D image measurements. *IEEE Transactions On Automatic Control*, 59(12):3325 – 3340.
- Trosset, J.-Y. and Scheraga, H. A. (1998). Reaching the global minimum in docking simulations: A Monte Carlo energy minimization approach using Bezier splines. *PNAS*, 95(14):8011–8015.
- Tsai, C.-J., Kumar, S., Ma, B., and Nussinov, R. (1999). Folding funnels, binding funnels, and protein function. *Protein Sci.*, 8(6):1981–1990.
- Vajda, S. and Kozakov, D. (2009). Convergence and combination of methods in protein-protein docking. *Current Opinion in Structural Biology*, 19(2):164 – 170.
- Vakili, P., Mirzaei, H., Zarbafian, S., Paschalidis, I. C., Kozakov, D., and Vajda, S. (2014). Optimization on the space of rigid and flexible motions: an alternative manifold optimization approach. In *53rd IEEE Conference on Decision and Control*, pages 5825 – 5830.
- Venkatachalam, C. M., X.Jiang, Oldfield, T., and M.Waldman (2002). LigandFit: a novel method for the shape-directed rapid docking of ligands to protein active sites. *Journal of Molecular graphics and Modeling*, 21(4):289–307.
- Vreven, T., Moal, I. H., Vangone, A., Pierce, B. G., Kastritis, P. L., Torchala, M., Chaleil, R., Jimenez-Garcia, B., Bates, P. A., Fernandez-Recio, J., Bonvin, A. M., and Weng, Z. (2015). Updates to the integrated protein-protein interaction benchmarks: Docking benchmark version 5 and affinity benchmark version 2. *J. Molecular Biol.*, (19):3031 – 3041.
- WL, D. (2002). The pymol molecular graphics system. <http://www.pymol.org>.

- Yueh, C., Hall, D. R., Xia, B., Padhorny, D., Kozakov, D., and Vajda, S. (2017). ClusPro-DC: Dimer classification by the Cluspro server for protein-protein docking. *J Mol Bio*, 492(3):372 – 381.
- Zarbaian, S., Moghadasi, M., Roshandelpoor, A., Nan, F., Li, K., Vakli, P., Vajda, S., Kozakov, D., and Paschalidis, I. C. (2018). Protein docking refinement by convex underestimation in the low-dimensional subspace of encounter complexes. *Scientific Reports*, 8(5896).
- Zefran, M., Kumar, V., and Croke, C. (1996). Choice of Riemannian metrics for rigid body kinematics. In *The 1996 ASME Design Engineering Technical Conference and Computers in Engineering Conference*.
- Zhang, C., Chan, J., and DeLisi, C. (1999). Protein-protein recognition: Exploring the energy funnels near the binding sites. *Proteins*, 34(2):255–267.

CURRICULUM VITAE

