12-31-2018

Application of Graphical Models in Protein-Protein Interactions and Dynamics

Amir Vajdi Hoojghan
University of Massachusetts Boston

APPLICATION OF GRAPHICAL MODELS IN PROTEIN-PROTEIN INTERACTIONS AND DYNAMICS

A Dissertation Presented

by

AMIR VAJDI HOOJGHAN

Submitted to the Office of Graduate Studies, University of Massachusetts Boston, in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

December 2018

Computer Science Program

APPLICATION OF GRAPHICAL MODELS IN PROTEIN-PROTEIN INTERACTIONS AND DYNAMICS

A Dissertation Presented

by

AMIR VAJDI HOOJGHAN

Approved as to style and content by:

Nurit Haspel, Associate Professor
Chairperson of Committee

Kourosh Zarringhalam, Associate Professor
Member

Dan Simovici, Professor
Member

Ming Ouyang, Associate Professor
Member

Dan Simovici, Program Director
Computer Science Program

Peter Fejer, Chairperson
Computer Science Department

ABSTRACT

APPLICATION OF GRAPHICAL MODELS IN PROTEIN-PROTEIN INTERACTIONS AND DYNAMICS

December 2018

Amir Vajdi Hoojghan,
B.S., Amirkabir University of Technology, Tehran, Iran
Ph.D., University of Massachusetts Boston

Directed by Associate Professor Nurit Haspel

Every organism contains a few hundred to thousands of proteins. A protein is made of a sequence of molecular building blocks named amino acids. Amino acids will be referred to as residues. Every protein performs one or more functions in the cell. In order for a protein to do its job, it requires to bind properly to other partner proteins. Many genetic diseases such as cancer are caused by mutations (changes) of specific residues which cause disturbances in the functions of those proteins.

The problem of prediction of protein binding site is a crucial topic in computational biology. A protein is usually made up of 50 to a few thousand residues. A contact site can occur within a protein or with other proteins. By having a robust and accurate model for identifying residues that are involved in the binding site, the scientists can investigate the impact of critical mutations and residues that can cause genetic diseases.

The main focus of this thesis is to propose a machine learning model for predicting the binding site between two proteins. By extracting structural information from a protein, we can have additional knowledge of binding sites. This structural information

can be converted into a penalty matrix for a graphical model to be learned from the protein sequence. The second part of this thesis is mostly focused on motion planning algorithms for proteins and simulation of the protein pathway changes using a Monte Carlo based method. Later, by applying a novel geometry based scoring function, we cluster the intermediate conformations into corresponding subsets that may indicate interesting intermediate states.

ACKNOWLEDGEMENTS

I thank God for bequeathing me the strength to understand some of the beauties of nature and all the people who have had an impact on my way here at the moment especially, my family, professors, and friends. I would like to start my expression to acknowledge all the support that is provided by my advisor, Dr. Nurit Haspel. Her guidelines were highly crucial to me in order to be able to overcome all the technical and even non-technical challenges that occurred during these 4 years. I believe this achievement would have been harder to reach without her help. I am so honored to finish my Ph.D. under her direction.

I am so thankful to my dear committee members Dr. Kourosh Zarringhalam, Dr. Ming Ouyang, and Dr. Dan Simovici for all the advice and guidelines that I received from them during my work completion. Kourosh taught me how to be a real scientist and I cannot even describe it in words, I will not forget all that support. A special vote of thanks for supports as mentors to professor Simovici and Ouyang, directors of the program. My lovely family is a part of this accomplishment and this could not have been done without their support. It was so hard to be away from them during these 4 years. Thanks to my mother Mina, who is the first person who taught me the meaning of love. My father Mohammad, who has done his best for me to be a doctor and support me from the first day of my life. My sister Masoumeh and my brother Reza who are the best siblings that anyone can have. I am so blessed to be a member of this family.

Finally, I would like to thank my amazing friends who helped me a lot in especially with the absence of my family they became my second family. Dr. Todd Riley, Mohammadreza, Saman, Mahdi, Farhad, Arpita, Sawsane, Yasaman, Ehsan, Mohammad Javad, Robin, Kian, Farnoosh, Hamidreza, Ali, Payman, Pejman, Reza, Mohammad Sadegh, Akram, Roshank, and Maryam. I am so lucky to have them as

my close friends.

And I thank everybody else that I may have forgotten!

TABLE OF CONTENTS

CHAPTER Page

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

Machine learning has become a fundamental approach for big data analysis in the recent years. Applications of machine learning can be found everywhere, including biomedical research, text to speech, image processing, and so on. Recently there has been a big development in our understanding of data analysis due to the increasing availability of medical and biological data on one hand, and the development of machine learning methods on the other hand. The focus of this chapter is to give a brief background on protein structure. After that, we provide a review of some machine learning methods related to chapter two and three.

Finally, we present the research problems that we worked on in this dissertation along with a brief background. Lastly, we provide some of the important terminologies used in the dissertation to make the next chapters of the dissertation easier to follow.

1.1 Central Dogma of Molecular Biology

Genetic information of every species is stored and carried in a double helix molecule which is called Deoxyribo**n**ucleic Acid (DNA). The DNA is located inside the nucleus of a cell and is made of four different molecular building block, nucleic acids, named cytosine (C), guanine (G), adenine (A) or thymine (T). The whole DNA of a species is also called genome. On average, a human genome has around 3 billion base pairs. A DNA under a transcription and translation processes tuned into its final product which is a protein. Based on the central dogma of molecular biology, the information

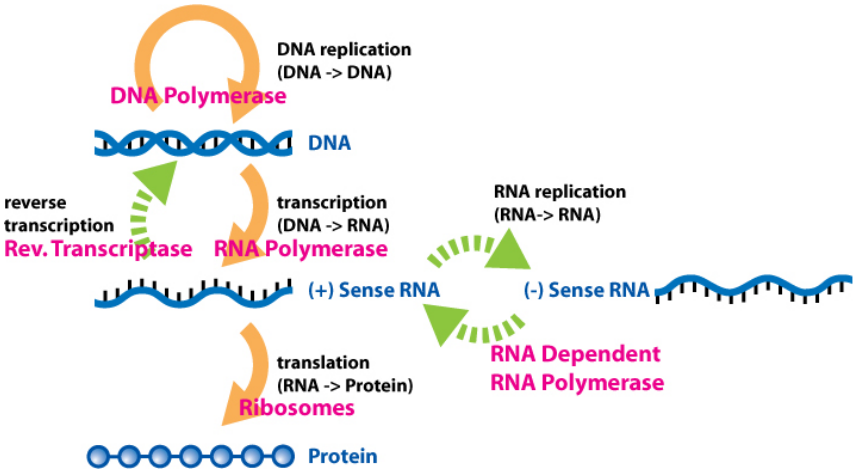


Figure 1.1: An overview of the central dogma of molecular biology. Figure from [18]

of a particular gene in the DNA is being transcribed (copied) to RNA, specifically messenger RNA, and the information in the messenger RNA is translated into producing a protein. In the next two subsections, these two processes are being explained in detail.

One can think of the DNA as a book written in a four-letter alphabet corresponding to each of the four different bases (A, T, C, G). So, any arbitrary combination of the alphabet with any length will give us a new word, but not all words are valid. Only some combinations of words are valid. Finally, a collection of some meaningful words gives sentences.

1.1.1 DNA and Transcription

Every DNA is made up of multiple segments called introns or exons. An exon is also called coding region which, under some regulation and processes, is being transcribed into RNA (messenger RNA). An intron corresponds to the non-coding region and the function of these fragments are to help control the expression of the coding regions. During transcription, a DNA sequence is read by an RNA polymerase, which produces a complementary, antiparallel RNA strand called a primary transcript. RNA is very

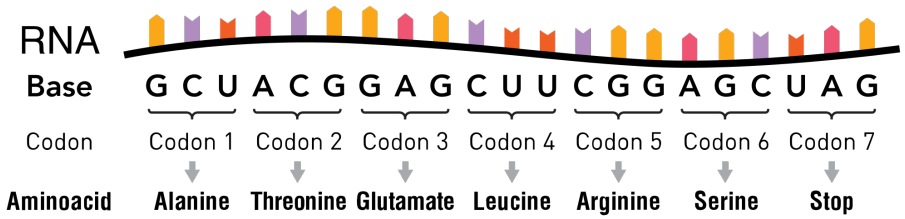


Figure 1.2: RNA decoded into amino acid. Figure from [20]

similar to the DNA and the only difference is about the transcription of Thymine (T) to Uracil (U).

1.1.2 RNA and Translation

During translation, the messenger RNA (mRNA) representing a gene is decoded into amino acids. Every three consecutive mRNAs, which are also called codons, are being decoded to one amino acid. There are start and end codons which correspond to the starting site of translation and end site of translation, respectively. There are 20 different amino acids. Table 1 represents the RNA codon table. As it has been shown, some amino acids can be made from more than one codon such as Proline. Translation is divided into 3 main steps. The first step is called initiation. In this step, the ribosome assembles around the target mRNA and the first tRNA is attached at the start codon and decoding is started. In the elongation step, the tRNA transfers an amino acid to the tRNA corresponding to the next codon. The ribosome then moves to the next mRNA codon to continue the process, creating an amino acid chain. Finally, in termination, when a stop codon is reached, the ribosome releases the polypeptide. Every chain of amino acids is decoded into one specific protein. Figure 2 shows these steps in more detail.

1.2 Proteins

This section is focused on the proteins characteristics. It explains the structure of a protein in detail and after that, it elucidates the motion of a protein and its impact on its function. Finally, we discuss how a protein does its job by binding to other proteins.

1.2.1 Amino Acids and Structure of a Protein

In the previous section, the procedure of getting amino acids from DNA was discussed. Each protein contains a sequence of 20 different amino acids. What makes different proteins depends on the combination of these amino acids. Here we explain the difference between each amino acid. Each amino acid is a molecule carrying amine (-NH2) and carboxyl (-COOH) functional groups, along with a side chain (R group) specific to each amino acid. Figure 3 shows the structure of an amino acid. All the amino acids have both amine and carboxyl groups and the only thing that is different between all amino acids are their side chain. There are multiple perspectives to classify these amino acids such as functional groups, biochemistry, Polarity, and so on. In section two and three we explain the corresponding representation of a protein in more detail.

The structure and function of a protein depends on its amino acid sequence. There are four main resolutions to represent a protein structure. Figure 4 depicts these four representations. The primary structure of a protein is its amino acid sequence. From the sequence we can extract information about the type of amino acids, and the order of residues, and the propensity of amino acid to bind to other types. The Protein secondary structure is the three-dimensional form of local segments of a protein. Secondary structures include $\alpha-$Helix, $\beta-$Sheets, beta turns, and omega loops. The Tertiary structure is the three-dimensional structure of a protein. Every protein is

Table 1.1: RNA Codon Table and Corresponding Amino Acids

1st base	2nd base								3rd base
	U		C		A		G		
U	UUU	(Phe/F) Phenylalanine	UCU	(Ser/S) Serine	UAU	(Tyr/Y) Tyrosine	UGU	(Cys/C) Cysteine	U
	UUC		UCC		UAC		UGC		C
	UUA	(Leu/L) Leucine	UCA		UAA	Stop (Ochre)	UGA	Stop (Opal)	A
	UUG		UCG		UAG	Stop (Amber)	(Ochre	(Trp/W) Tryptophan	G
C	CUU		CCU	(Pro/P) Proline	CAU	(His/H) Histidine	CGU	(Arg/R) Arginine	U
	CUC		CCC		CAC		CGC		C
	CUA		CCA		CAA	(Gln/Q) Glutamine	CGA		A
	CUG		CCG		CAG		CGG		G
A	AUU	(Ile/I) Isoleucine	ACU	(Thr/T) Threonine	AAU	(Asn/N) Asparagine	AGU	(Ser/S) Serine	U
	AUC		ACC		AAC		AGC		C
	AUA		ACA		AAA	(Lys/K) Lysine	AGA	(Arg/R) Arginine	A
	AUG	(Met/M) Methionine	ACG		AAG		AGG		G
G	GUU	(Val/V) Valine	GCU	(Ala/A) Alanine	GAU	(Asp/D) Aspartic acid	GGU	(Gly/G) Glycine	U
	GUC		GCC		GAC		GGC		C
	GUA		GCA		GAA	(Glu/E) Glutamic acid	GGA		A
	GUG		GCG		GAG		GGG		G

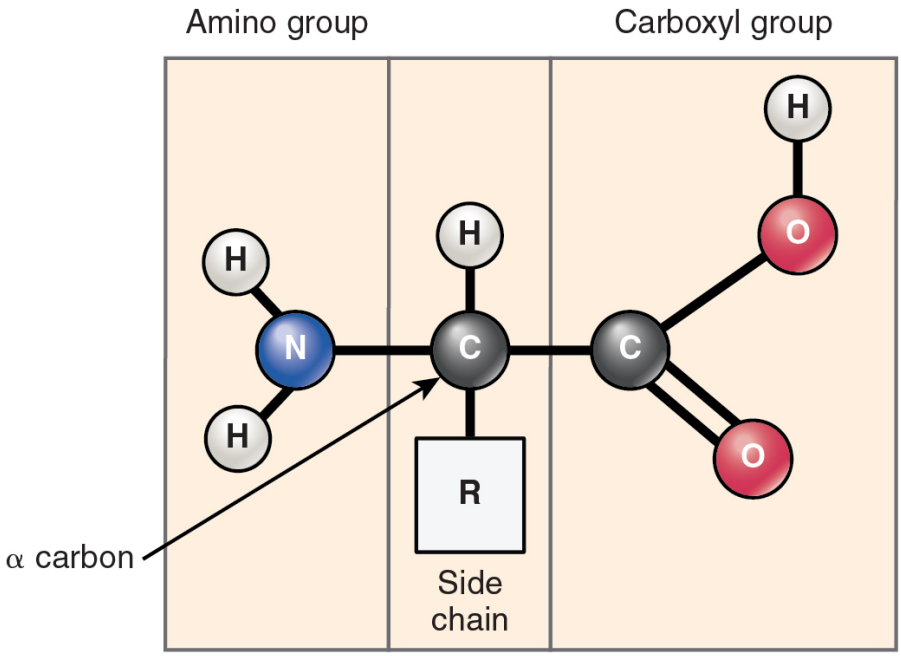


Figure 1.3: Structure of an amino acid. Figure from [19]

usually folded into a characteristic three-dimensional (3D) structure. Finally, there is the quaternary structure which is a complex, or an association, of two or more chains of a protein to form one functional unit.

Each of these representations can be used for different problems in bioinformatics. In this thesis, we use all the four representations to extract information for our model.

1.2.2 Proteins Function

In order to perform its function, a protein has to fold into its functional tertiary structure, called the *native structure*. A conformation of a protein is a possible 3D shape a protein can assume. Some proteins have two or more stable conformations (native structures) which have different functions, usually aided by binding to other molecules. The focus of the third chapter is to simulate the pathway that a protein traverses between these two conformations. Once a protein has folded into its correct form, it can perform its function by binding to other proteins, usually on a specific region on its structure. A protein performs its function if and only if it binds to its partner correctly.

1.3 Machine Learning

In recent years, the emergence of machine learning methods has greatly impacted research in data science. A sophisticated machine learning algorithm is built from a statistical model which can be applied to data to learn important features that contribute to its behavior. A good machine learning method can automate the process of gathering data, preprocessing, extracting features , evaluating, and making decisions.

In this thesis, we discuss some computational methods that can help us to understand and predict the motion of a protein between open and closed conformations and

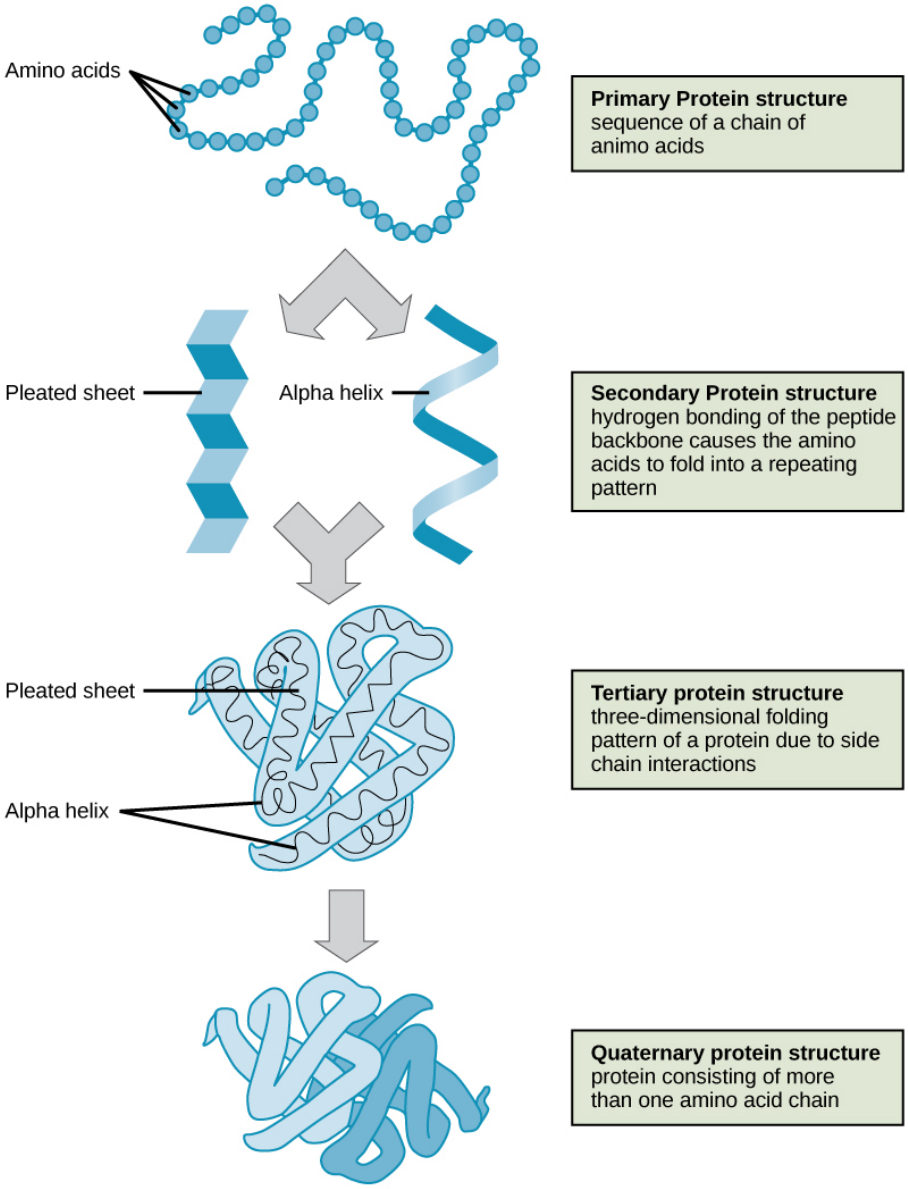


Figure 1.4: 4 main representations of a protein. Figure from [21]

clustering the resulting trajectories into intermediate conformations. Also, we develop probabilistic models to predict binding interfaces between two interacting proteins. Before we discuss these models, in the next subsections, we review the mathematical foundations.

1.3.1 Clustering High Dimensional Data

Clustering is the process of grouping a data set into subgroups such that the elements in the same subgroup are similar to each other with respect to some scoring function or metric. Clustering is an unsupervised method in machine learning. It means that we do not have any class labels. The most popular methods for clustering are hierarchical clustering, K-means, and Gaussian Mixture Models. Clustering can also have heuristic scoring functions or algorithms.

There are some challenges in clustering such as the nature of the task which is unsupervised, which makes building and evaluating a model particularly hard. Also, as in every machine learning method, feature selection plays a crucial role in the performance of a model. This problem becomes harder when the number of features increases. In the case of high dimensional space, there are some dimensionality reduction techniques that can help us to reduce the feature space greatly.

For example, consider clustering of proteins based on folding pattern of a protein. In order to perform this task, we need to represent each protein with a feature vector. This vector contains information about a protein's sequence of amino acids (an average protein has several hundreds of amino acids). However, this is not always sufficient to learn a protein function, and we need to add more features such as secondary structure elements, species and so on. As we add features the model will improve, but the computational cost will also increase. A good estimation of dimensionality reduction can be applied in this example by replacing each amino acid with 4-mers

(4 consecutive amino acids) which can reduce the feature dimension to 4 and still classify proteins. This one an example of dimensional reduction in complex space. Principle component analysis (PCA) is another example of dimensionality reduction.

1.3.2 Stochastic Simulation and Monte Carlo

Monte Carlo is a class of methods that relies on repeated random sampling to obtain numerical results. This randomness may help to solve the problem which may be deterministic in reality but it is hard to model. One category of Monte Carlo methods is called Monte Carlo tree search. The focus of Monte Carlo tree search is on the analysis of the best move, expanding the search tree based on random sampling of the search space. The idea is to start from root node and expand it randomly or based on probability distribution. This method consists of 4 main steps which represented in the the figure 1.5 also in the following:

- Selection: Start from root R select successive child nodes until a leaf node L is reached. L must be node that no simulation has yet been initiated from it.
- Expansion: If L is not terminal node, create one (or more) child nodes and choose node C from one of them. This needs to be valid move from L to C.
- Simulation: Expand node C until reach to terminal node T.
- Backpropagation: Use path R to T and update the information for next move.

These steps are base of our method for simulating conformational changes in a protein.

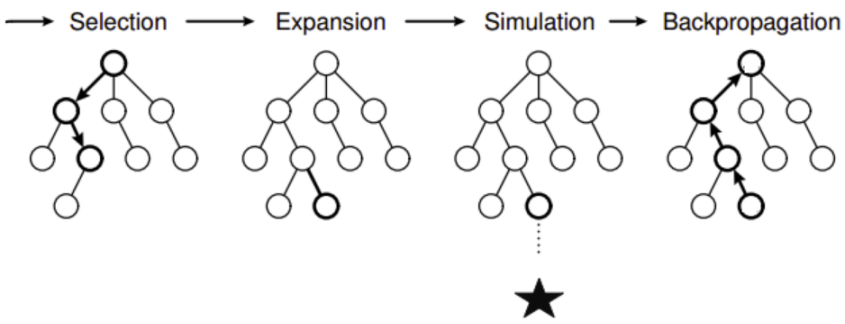


Figure 1.5: 4 main steps of Monte Carlo tree search method.

1.3.3 Undirected Graphical Modeling

Markov Random Field

One of the most powerful methods in machine learning is probabilistic graphical models (PGM) which models the relationship between variables as a directed or undirected graph and represents the potential of dependency between two or more variables over a multi-dimensional space. The graph is either compact or factorized representation of a set of independencies that hold in the specific distribution. PGMs can be divided into two main categories: if the graph is directed the model such as Bayesian graphical model. An exmple of undirected graphical model is Markov Random Field (MRF). In MRF, the probability distribution of a given variable X with length of L can be presented as:

$$P(X) = \frac{1}{Z} exp\left(\sum_{i=1}^{L}\left[V_i(x_i) + \sum_{j>i}^{L} W_{i,j}(x_i, x_j)\right]\right) \tag{1.1}$$

The above equation tries to model the relationship between every $i, j \in [1, L]$ as a graph. Here, V_i and $W_{i,j}$ are the potential functions, also called field and coupling, respectively in statistical physics, which is modeled as a probability function here. Z is called the partition function, to turn the score into probabilities.

Gaussian Graphical Model

Gaussian Graphical Models (GGM) are a subclass of MRF which restricts the random variables to have a normal distribution while GGM explicitly capture the statistical relationship between the variables of interest in the form of a graph [79]. This model can be applied in a lot of domains such as natural language processing, finance, and bioinformatics. Identifying the dependency of the variables is very important in more complex spaces, especially in network-based inference. A good example of application of GGM is Kramer's work which applied Gaussian Graphical Models to gene expression data to construct a network of gene-gene interactions[50].

In this subsection, we discuss some of the fundamental concepts that will be used in the next chapters. The full tutorial and details of the proofs can be found in Uhler's article about GGM [79]. The goal of graphical models is to capture the pairwise relationship between two nodes of a graph as a probability model. Equation 1.2 represents the density function of a GGM. It models a random variable $X \in R^L$ which has a normal distribution with mean (μ) and a covariance matrix of $\sum$. θ is the inverse of the covariance matrix (precision).

$$f_{\mu,\sum}(x) = (2\pi)^{\frac{-L}{2}}(det\sum)^{\frac{-1}{2}}exp\left\{\frac{-1}{2}(x-\mu)^T(\sum)^{-1}(x-\mu)\right\}, x \in R^L \qquad (1.2)$$

Equation 1.2 can be rewritten with respect to θ by applying the trace inner product on R^L as:

$$f_{\mu,\sum}(x) = exp\left\{\mu^T\theta x - \left\langle\theta, \frac{1}{2}xx^T\right\rangle - \frac{L}{2}log(2\pi) + \frac{1}{2}log(det(\theta)) - \frac{1}{2}\mu^T\theta\mu\right\} \qquad (1.3)$$

A graphical model which satisfies equation 1.3 is defined as $G = (V, E)$ such that G is an undirected graph with $|V| = L$ is the set of vertices and E is the set of edges

which satisfy the following condition:

$$\forall(i,j) \in E, (\sum)^{-1}_{i,j} \neq 0 \tag{1.4}$$

From equation 1.4, we can infer two properties: the sparsity pattern of G and the conditional independence of missing edges in G.

There are two perspective over Maximum Likelihood estimation (ML or MLE) problem in GGM. The first one allows the estimation of the edge weights given the graph structure. The Second allows learning the structure of the graph. In order to perform these two tasks, we will use the following lemma which is proved in [79].

Lemma 1.1- For $X \in R^L$ which is normally distributed with center of μ and covariance matrix of $\sum$ and

$\forall i,j \in [1,L]$ which $i \neq j$ and $K \subseteq [L] - \{i,j\}$ then the following statements are equivalent

a) $\mathrm{x}_i \perp\!\!\!\perp x_j | x_K$;

b) $\det((\sum)_{iK,jK}) = 0, \quad where \quad iK = \{i\} \cup K$;

c) $\det(\theta_{iR,jR} = 0) \quad where \quad R = [L] \backslash \{K \cup \{i,j\}\}$(1.5)

For ML estimation of a GGM, assume n observations $X^{(1)}...X^{(n)}$ from $\mathcal{N}(\mu, \sum)$ are given. The empirical covariance matrix is determined from

$$S = \frac{1}{n}\sum_{i=1}^{n}(X^{(i)} - \bar{X})(X^{(i)} - \bar{X})^T \quad where \quad \bar{X} = \frac{1}{n}\sum_{i=1}^{n} X^{(i)} \tag{1.6}$$

Then the Gaussian log likelihood can be expressed as:

$$\ell L(\mu, \sum) \propto -\frac{n}{2} log(det(\sum)) - \frac{n}{2} tr(S(\sum)^{-1}) - \frac{n}{2}(\bar{X} - \mu)^T (\sum)^{-1}(\bar{X} - \mu) \tag{1.7}$$

In order to reduce the complexity of the Maximum Likelihood (ML) estimation, we assume that the mean $\mu = \bar{X}$, then we estimate the ML using following objective function in a form of precision matrix:

$$\max_{\hat{\theta}} \quad logdet(\hat{\theta}) - tr(S\hat{\theta}) \tag{1.8}$$

Subject to $\hat{\theta} \in \theta_G$ which θ_G corresponds to Gaussian graph $G = (V, E)$ with following definition:

$$\theta_G \equiv \{\hat{\theta} \in R^L_{>0} | \hat{\theta}_{ij} = 0 \quad \forall i, j, \quad i \neq j, \quad with \quad (i, j) \notin E\} \tag{1.9}$$

It can be shown that the objective function in equation 1.6 is concave over the entirety of its domain[11]. Since the MLE may not exist if the likelihood is different from the global maximum, adding new constraints to change the objective function into a convex space is a common technique. Based on Lemma 1.3.3, in order to solve the objective function in 1.6, finding a feasible point is required. The identity matrix is a strong feasible point. So, we expand the set of edges to include all the self-loop edges such that, $\hat{E} = E \cup \{(i, i) | i \in V\}$.

Lemma 1.3.3- The MLE of an objective function does not exist if and only if there exist no feasible point for the dual optimization problem [2].

Given these conditions, the covariance matrix is positive definite which implies that the objective function 1.6 is a convex optimization problem. We are interested in a sparse model by solving the GGM which corresponds to a sparse underlying graph. In this case, we can use the L_1 norm penalty as a sum of the absolute values of the elements of precision matrix. So, the new objective function is given in 2.4

where θ is the precision matrix, S is the emperical matrix, and Λ is the penalty matrix.

$$\max_{\theta} log(det\theta) - tr(S\theta) - \Lambda||\theta||_1 \tag{1.10}$$

For solving 2.4, Meinshausen proposed an estimation-based method by fitting a Lasso model to each variable, using other predictors [58]. Applying interior-point method in optimization is another solution for exact maximization which was proposed in [88, 6]. In our work we used the blockwise coordinate descent method which originally developed by [6] and later modified and speed up by Friedman as a graphical lasso and its implementation in R which is called Glasso [34].

In the last part of this section, we explain the method that was used in Glasso [34]. The idea is the instead of estimating θ, estimate the covariance matrix based on empirical covariance(S). Let ω be the estimated covariance. By applying permutation on rows and columns on $\omega \in R^{L\times L}$ and $S \in R^{L\times L}$, they partition these matrices as:

$$\omega = \begin{pmatrix} \omega_{11} & \hat{\omega}_{12} \\ \hat{\omega}_{12}^T & \omega_{22} \end{pmatrix}, S = \begin{pmatrix} S_{11} & \hat{s}_{12} \\ \hat{s}_{12}^T & s_{22} \end{pmatrix} \tag{1.11}$$

Where $\omega_{11}, S_{11} \in R^{(L-1)\times(L-1)}$, $\hat{\omega}_{12}, \hat{s}_{12}$ are vectors of size $L-1$, and ω_{22}, s_{22} are scalars. Basically, the goal is to estimate $\hat{\omega}_{12}$ using ω_{11} values. Interestingly, the solution to $\hat{\omega}_{12}$, satisfies the 1.12 optimization problem and updates ω with new estimation of $\hat{\omega}_{12}$ until its convergence.

$$\hat{\omega}_{12} = \min_{y}\{y^T\omega_{11}^{-1}y : ||y - \hat{s}_{12}||_\infty \leq \Lambda\} \tag{1.12}$$

Also, the solution of β in 1.13 optimization problem is the same as 1.12 since $\hat{\omega}_{12} = \omega_{11}\beta$.

$$\min_{\beta}\{\frac{1}{2}||\omega_{11}^{\frac{1}{2}}\beta - b||^2 + \Lambda||\beta||_1\}, \quad where \quad b = \omega_{11}^{\frac{-1}{2}}\hat{s}_{12} \tag{1.13}$$

With the new setting, solving equation 1.13 can give the underlying graph of estimated covariance. Since the assumption is that ω is an estimation population of the covairance matrix, then $\omega\theta = I$ which is the same as:

$$\begin{pmatrix} \omega_{11} & \hat{\omega}_{12} \\ \hat{\omega}_{12}^T & \omega_{22} \end{pmatrix} \times \begin{pmatrix} \theta_{11} & \theta_{12} \\ \theta_{12}^T & \theta_{22} \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0 & 1 \end{pmatrix} \tag{1.14}$$

The 2.4 function is not differentiable over its entire domain, so by taking a subderivative we have $\omega - S - \Lambda\eta = 0$ where, η is defined in 1.15 and writing it for each partition in 1.14, we end it up with$\theta_{12} = -\theta_{22}\omega_{11}^{-1}\hat{\omega}_{12}$ (for more detail see [34]) therefore the solution to β in 1.13 implies $\theta_{12} = -\theta_{22}\beta$.

$$\eta_{ij} = \begin{cases} sign(\theta_{ij}) & \theta_{ij} \neq 0 \\ [-1, 1] & \theta_{ij} = 0 \end{cases} \tag{1.15}$$

Algorithm 1 represents how Glasso estimates the actual covariance with respect to S and Λ. In chapter two we explain a learning model based on this algorithm. In this algorithm, t is a soft-threshold function and cutoff is a fixed user defined variable.

Algorithm 1 Glasso(S,Λ)

1: $\omega = S + \lambda I$; //Do not change for diagonal elements in ω
2: **for** $j = 1..., L, 1, ..., L$ **do**
3: $current.\omega = \omega_{11}, current.s = s_{12}$
4: solve function 1.13 with respect to $current.\omega_{11}$ and $current.s_{12}$
5: $\hat{\beta}_j = \frac{t(current.s_j - \sum_{k \neq j} current.\omega_{kj}\hat{\beta}_k, \Lambda)}{current\omega_{jj}}$
6: **if** $|\omega - CutOff \times Average(S^{-diagonal})| < 0.001$ **then**
7: Break
8: **end if**
9: **end for**

Once this algorithm converges, we can calculate the precision matrix using 1.16 formula.

$$\theta_{ij} = (-1)^{i+j} \frac{det(\sum_{[L]\setminus\{i\},[L]\setminus\{j\}})}{det(\sum)} \tag{1.16}$$

1.4 LSIP-PSICOV: Structural Information as Penalty Helps Gaussian Graphical Models to Predict Protein- Protein Interface Better

1.4.1 Research Problem

A protein performs its function by binding to other proteins (*molecules*) on a specific location on its surface which is called a binding site (*interface*). Identifying interfaces is a challenging problem in reality due to a search space size. On average proteins have hundreds to thousands residues. All the possibilities of binding sites between two arbitrary proteins are between 10^4 and 1.5^6 pairs on average among two proteins. Only 5 to 400 pairs of these residues are actually binding sites. Also, every protein family can have a totally different pattern of binding which makes for a computationally hard problem. In the second chapter, we discuss the most important structural, physical, and chemical features that can impact the pattern of binding site between two proteins. Later we elucidate how to turn this knowledge into a penalty term in a Gaussian Graphical model and improve the task of protein-protein binding site prediction. Identifying the exact binding site can be used in a lot of domains such as identifying critical residues (hotspots) in a protein, identifying pathogenic mutations, and predicting protein folding, docking, and structure predictions.

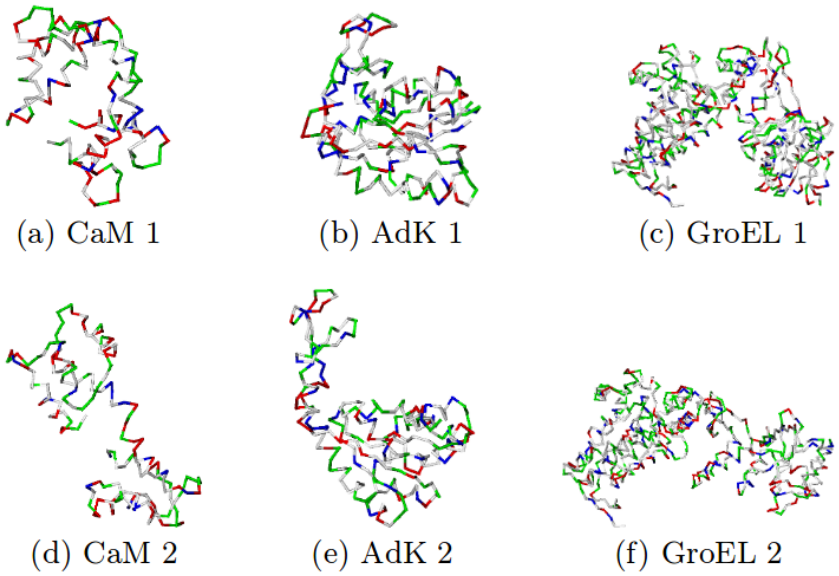


Figure 1.6: (a) CaM closed conformation (PDB: 1CTR); (c) CaM open conformation (PDB:1CFD); (b) AdK closed conformation (PDB:1AKE); (e) AdK open conformation (PDB:4AKE); (c) GroEL closed conformation (PDB:1SS8); (f) GroEL open conformation (PDB: 1SX4)

1.5 Identifying Clusters of Intermediate States in Conformational Changes

1.5.1 Research Problem

Many proteins change their structures from in-active to active state or bound to unbound in order to perform their function[35].

Figure 5 represents an example of open and closed conformation of three proteins. The conformational changes are usually a continuous and transient process making intermediate structures hard to be determined experimentally[53]. Understanding these intermediate conformations can improve protein binding models and also design a more accurate drug targeting methods. Due to the conformational search space (which could be proportional to the number of residues in a protein), it is not easy

to simulate these changes. In this work we apply a Monto Carlo based simulation method to model this conformational change between the open and closed forms of a protein.

1.6 Thesis Organization

This thesis is organized as follows. Chapter 2 describes the first research problem in detail by presenting GMM model that we used to accurately predict protein-protein binding sites. Additionally, this chapter provides several references to the literature related to the elements of the problem in question and concludes by presenting comparisons between the proposed method and a number of available methods in the community. Chapter 3 presents the second research problem, which is a stochastic method for simulation of protein conformational changes along with a geometric method for clustering the intermediate conformations. Lastly, Chapter 4 sums up the thesis contributions and concludes this dissertation.

Chapter 2

LSIP-PSICOV: STRUCTURAL INFORMATION AS PENALTY HELPS GAUSSIAN GRAPHICAL MODELS TO PREDICT PROTEIN- PROTEIN INTERFACE BETTER

2.1 Introduction

Experimental approaches for identifying binding sites between two interacting proteins include methods such as X-ray crystallography and mutagenesis, which are time consuming and expensive. Hence, there has been a rapid increase in computational methods that try to address this problem. Proteins are complex molecules and their binding depends on multiple factors. Identifying these characteristics of each protein family and later classifying them based on these features has been studied extensively. This chapter focused on reviewing some of these works and analyzing significant features. Then by extracting these features from the structure and sequence of two proteins and turn that information based on the protein family into a penalty matrix for GGM model. This penalty matrix works as a prior to our probabilistic model. In order to learn a penalty matrix appropriately, we need to understand the features that help to predict binding sites correctly otherwise this penalty will mislead the prediction. The main features can be divided into two categories. First, features that are related to the stability of a protein which are listed as:

- Distinguishing between surface and interior residues
- The distribution of the conformational substates
- Identifying the location of conformational changes

These features are mostly hard to quantify or measure which makes this task still challenging. The second group is less complex to measure, but the relationship between these features are complex and hard to model.

2.1.1 Common Interface Features Across Most Proteins

A practical approach to predict the function of a protein can be studied through predicting intra or inter protein contact regions, therefore, it is helpful to find the folding pattern and eventually how a protein functions. Application of Protein-Protein interaction can be found in a lot of domains such as drug discovery, protein dynamics, identifying hotspots residues and consequently mutational effect on those residues and so on [29, 28, 72].

From the Physical-Chemical point of view, any two proteins can interact [45] which is not the case in reality. Hence, the main question is about, is there any model that can capture this relationship accurately? Most studies showed that the most predictive features are Hydrophobic interactions, Hydrogen bonds, electrostatic interaction, conservation, solvent accessible surface area (SASA), propensity, and covalent bonds, but these features are general and their significance can vary in different protein families [27, 45]. Esmaielbeiki and others [27] reviewed these methods comprehensively. As a result, they compared more than 70 methods across multiple datasets and the conclusion was that the above features are crucial but also general in order to have a robust model with high performance where is not dependent on a dataset or protein family. A good model should identify binding site of a protein with respect to

its partner. Figure 2.1 represents an overview of these methods. In another work, Keskin and others [45] reviewed this problem from multiple perspectives. Some of the important features and conditions that every model needs to consider in order to have a reliable prediction will be discussted herel.

Protein complexes can be divided into two main categories as obligatory complexes, which are the proteins that perform their functions just inside a complex with partners belong to complex, and transient complexes proteins that depend on functional state of the partner [66, 45]. Additionally, each complex is either Homodimers or Heterodimers. In the context of protein binding sites, these two classes need to be studied separately since they have different properties. For example, most Homodimeric interfaces are hydrophobic, large, with a high value of a nonpolar buried surface area, and have a fitter complementarity geometry between two chains while that is not the case in Heterodimeric complexes[45]. Studies show obligatory complexes are very compact with stronger hydrophobic effect while transients are mostly polar/charge and the surfaces of the interface are not optimized[41]. All Homodimers and some of the Heterodimers belong to the obligatory class and an example of the transient complexes is the interaction between enzyme and inhibitor.

One main difference between these two classes of proteins that influence any methods is the rate of evolution. Interfaces of the obligatory complexes turn to evolve at a slower rate which increases co-evolving rate between a protein and its partner, on the contrary, transient complexes have a high rate of evolution and as a result, the score of co-evolving is low between two partners[59]. This difference is crucial for us since our proposed method is based on the co-evolution score.

Identify protein surfaces

Interfaces are mostly located in the surfaces of a protein. Therefore, the first step toward identifying the interface between two proteins, is to distinguish the surfaces

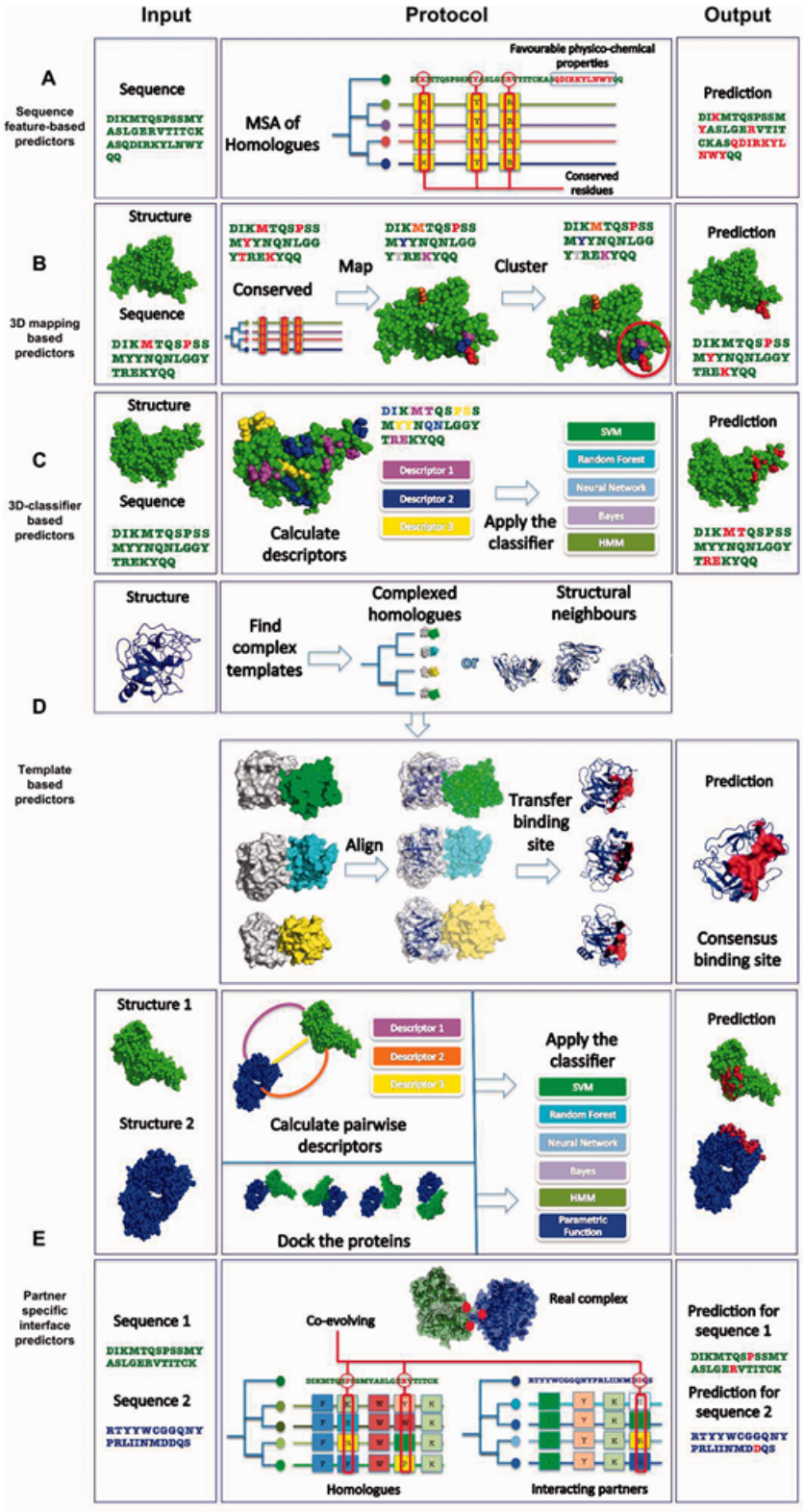


Figure 2.1: which are divided into 7 groups. Figure is from [27].

and interior residues among them. A common technique to determine surface residues is to calculate solvent accessible surface area (SASA) for each residue [51]. After measuring SASA, we cannot treat that as an absolute metric in the binding sites context, and it requires to correct for bias effect between the interface and non-interface surfaces, and it is unknown the difference between these two is because of either SASA or function [23]. QIPI quantify this difference by dividing surfaces into 2 types of interface and non-interface. A residue is in the surface if its SASA is greater than 1 $\mathring{A}^2$. The interface is made by spatially neighboring residues whose SASA were changed more than 1 $\mathring{A}^2$ between single domain and complex.

Hydrophobic effects

Table 2.1.1 associate each amino acid to corresponding side chain charge class. 20 amino acids based on the propensity of the side chain to be in contact with a polar solvent like water, can be divided as:

- hydrophobic (low propensity to be in contact with water)
- polar (usually participate in hydrogen bonds as proton donors or acceptors)
- charged (side chains often make salt bridges)

In an interesting study, Jones and others [44] tried to measure the correlation between interface residue and its side chain propensity. They found that a large portion of interface residues in Heterodimer complexes are hydrophobic or uncharges further they performed patch analysis and realized that the interface paths are more planar with considerably large SASA. Moreover, conservation across MSA is another important feature that had been reported by some people [65, 10]. Also, center residue in each patch turn to be more conserved than its neighbors and it is been suggested to divide the interface to its core and surrounding[3]. Nevertheless, side chain propensity is not enough for distinguishing between the interface and non-interface residues.

Table 2.1: Amino acids with corresponding side chain propensity.

Amino Acid	**Properties**
Alanine-Ala-A	Hydrophobic
Arginine-Arg-R	Positively Charged
Asparagine-Asn-N	Polar
Aspartate-Asp-D	Negatively Charged and Polar
Cysteine-Cys-C	No Charge, Non-polar, Hydrophilic
Glutamate-Glu-E	Negatively Charged and Polar
Glutamine-Gln-Q	Polar
Glycine-Gly-G	No Charge, Non-polar, Hydrophilic
Histidine-His-H	Positively Charged and Polar
Isoleucine-Ile-I	Hydrophobic
Leucine-Leu-L	Hydrophobic
Lysine-Lys-K	Positively Charged and Polar
Methionine-Met-M	Hydrophobic
Phenylalanine-Phe-F	Hydrophobic
Proline-Pro-P	Hydrophobic
Serine-Ser-S	Polar
Threonine-Thr-T	Polar
Tryptophan-Trp-W	Hydrophobic
Tyrosine-Tyr-Y	Polar
Valine-Val-V	Hydrophobic

Module, Hotspots and Protein Stability

A module is defined as those residues within a distance of 10 $\mathring{A}$. Residues belong to one module maybe cooperative, while residues located in different modules are additive [73]. It has been shown that for all proteins, not only the energy distribution is not uniform across a given interface of two interacting proteins, also a small subset of residues will have a higher contribution to the binding free energy than other residues[46]. Ma and others discovered the enrichment of polar residue hot spots in protein-protein binding sites, and also hotspots are what distinguishes binding sites from the remainder of the surface. They further show a conformity between energy hot spots and structurally conserved residues. The number of structurally conserved residues, especially high ranking energy hot spots, increases with the binding site contact size [55]. Therefore, conservation can help to identify hotspots. There is a direct relationship between protein stability and hotspot residues. Although identifying hotspot residues are very expensive and challenging, this has been investigated by some groups [22, 16], hence applying computational methods to estimate protein stability based on hotspot residues is not recommended due to the existence of a lot of exceptions [45].

Secondary Structure Elements

Protein-Protein interfaces have preferred architecture. Due to the fact that the number of secondary structures is limited and also the association between secondary structure and degree of freedom [31], it is important to consider its contribution in PPI problem. As a result, we consider four different secondary structure states as it is mentioned in table 2.2.1.

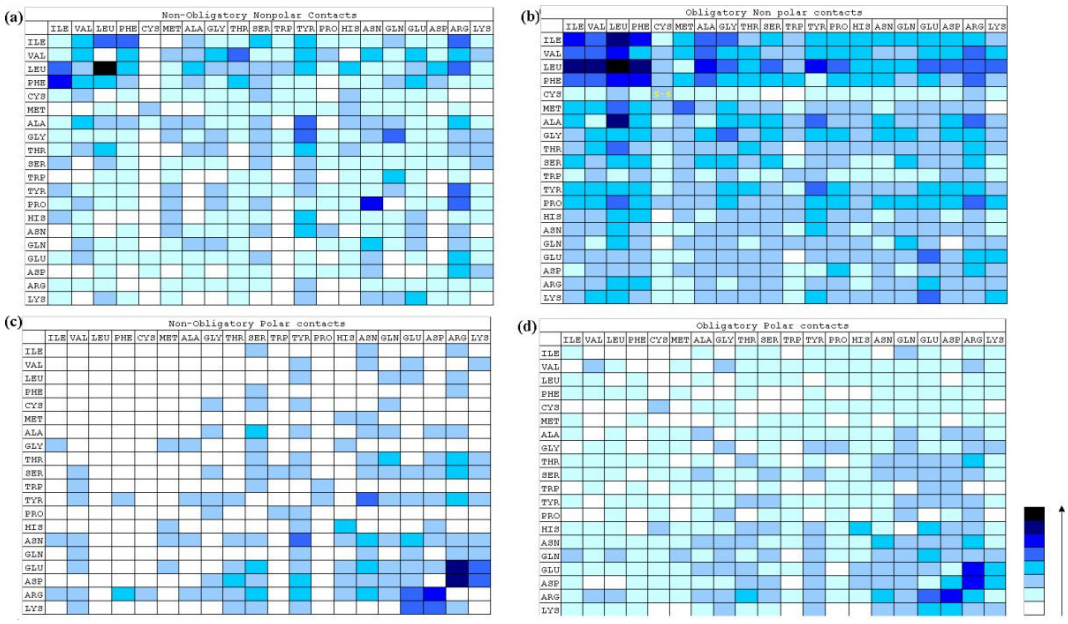


Figure 2.2: . Colour gradation: white- No interaction; Cyan to black- increasing gradation of interaction with normalized frequency varying between 014 in discrete steps. Figure from [24].

Hub and Lonely Proteins

In a study conducted by [45], the authors took protein-protein interfaces from PDB [68] and divided all the 103 clusters of interfaces into three categories.

- Interfaces that belong to proteins that have a global fold of parent between two chains and similar function.
- Proteins that are not similar in function and structure.
- These proteins have a similar binding site in one protein in front of many other partners.

They found the proteins in type three which sharing one interface across multiple interactions turns to interact on average 13 times while the average of other proteins was 5 times [68, 37]. Interestingly, those hub proteins have a smaller interface with enrichment in α-Helices in the interfaces.

2.1.2 Building Multiple Sequence Alignment for a Pair of Interacting Proteins

The first step toward measuring co-evolution in proteins residue is to build a multiple sequence alignment (MSA) of the ortholog proteins of the query protein. This is an important step which has an impact on a performance of the model. Let us assume we are interested in finding the binding sites between two proteins p and q in a species D. Building MSA for one protein is pretty straightforward. It can be done by performing ortholog search of the query protein against the same protein in different organisms, and then perform an alignment. In the case of a pair of proteins, it is a little bit tricky due to the fact that it may exist more than one copy of the protein in a genome (paralog), further we need to concatenate those pairs so that every protein p in a species is followed by protein q in the same species. This step is being studied extensively by some groups. Ovchinnikov et al, [70] showed that in order to extract less bias and more accurate information form the MSA, we need to have more sequences than the total number of residues in the MSA. In his method, first, they perform ortholog search with respect to two interacting proteins across other organisms folowed by taking those pairs that are at most 20 genes away in the genome to make sure they belong to the same operon. Basically, those genes that belong to the same Operon (co-located) are also co-regulated. $HH\Delta$ is used to measure similarity between two MSAs. Finally, eliminating those sequences that which are identical more than 90% or have a more than 75% gap in that position in the MSA. They do the alignment task using clustal omega. Hopf et al, [40], also provided a similar procedure for building MSAs, with more sequences in each MSA, but a little bit different in the tools that they applied. Our results and other methods show that the way Ovchinnikov used to build MSA is more appropriate for co-evolution methods.

2.1.3 Statistical Approaches for Measuring Co-Evolution in Proteins

In this subsection, review of other methods for inferring co-evolution from MSA is investigated. This technique is called Direct Coupling Analysis (DCA) and it may have different names in other fields. In the field of statistical mechanics (physics), it is called Inverse potts (Ising) model or Boltzmann Machine and in statistics or computer science, it is called Markov Random Field. Lapedes et al [8] proposed to estimate the maximum entropy from the covariation in the MSA and use this for a Boltzmann Machine by considering Monte Carlo based learning method. Due to the small number of sequences and to the computational runtime, this method did not become very popular.

Let us state the problem. Assume we want to model the distribution of the residues belong to a protein with sequence of length L by $P(X)$, where $X = (x_1, ..., x_L)$ and $x_i \in \{20 Amino Acids, Gap\}$. $P_i(a_k)$, denotes the marginal probability of a single amino acids a_k in position i^{th} which can be determined as a frequency of that amino acids in position i^{th} of the MSA. Similarly, $P_{ij}(a_k, a_r)$ corresponds to frequency of pair of amino acids (a_k, a_r) in positions i, j, respectively. The distribution of maximum entropy P_{ME} of an MSA is given as:

$$P_{ME}(X|h,j) = \max_{p(x)}[-\sum_x p(x)\log p(x) + \lambda(\sum_x p(x) - 1) + \sum_i [h_i(\sum_x p(x)\delta_{x_i a_k} - p_i(a_k))] +$$

$$\sum_i \sum_{j>i} [J_{ij}(a_k, a_r)(\sum_x p(x)\delta_{x_i a_k}\delta_{x_j a_r}] - P_{ij}(a_k a_r)] = \frac{1}{Z} e^{-H_{potts}(X|h,j)} \quad (2.1)$$

where, $\delta_{x_i a_k}$ is Kronecker delta, λ is the Lagrange multipliers and h and J are potentials which are also called fields and coupling, respectively. $P_{ij}(a_k, a_r)$ can capture both direct and undirect correlation between amino acids while $J_{ij}(a_k, a_r)$ carries causative correlation only [62].

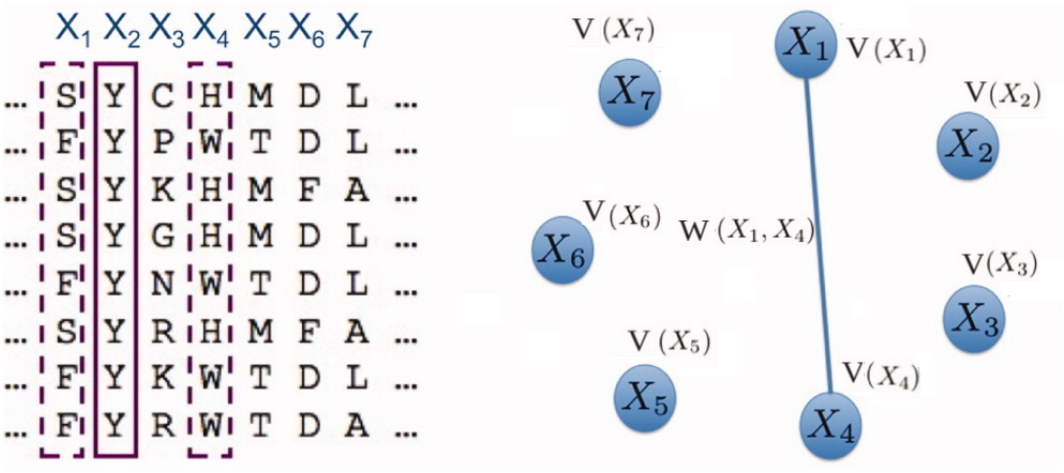


Figure 2.3: (left) with a graphical model in right figure and learn the parameters [4].

The main problem with equation 2.1 was the calculating the partition function Z, which corresponds to all sequences with length L that exist with 20 amino acids, L^{20}. With considering the average length of protein which is 500 residues, it is untraceable. A few years later Weight et al [84] applied message passing algorithm to learn the MRF. Unfortunately, it was not guaranteed to converge and was still expensive for large proteins. Two years later Balakrishnan et al [4], proposed the GREMLIN method to estimate the partition function Z which is called pseudo-likelihood. and instead of calculating the global Z, they used equation 1.1 to calculate the local Z by changing one position at a time and assume the rest of the residues in the sequence are fixed. GREMLIN models the pseudo-likelihood of θ, given an MSA X made up N sequences of length L as:

$$pll(\theta|X) = \sum_{n=1}^{N}\sum_{i=1}^{L} \log\left(\frac{exp\left(V_i(x_i^n) + \sum_{j\neq i,j=1}^{L} W_{i,j}(x_i^n, x_j^n)\right)}{\sum_{c=1}^{21} exp\left(v_i(c) + \sum_{j\neq i,j=1}^{L} W_{i,j}(c, x_j^n)\right)}\right) - R(\theta) \qquad (2.2)$$

Equation 2.1 and 2.2 are very similar in a way that (V,h) and (W,J) are trying to capture the same thing. R is the L_2 based norm regularization term which tries to encourage to sparsity of the network. In the MRF based model, the influence of regularization term decreases as number of sequences increase [69]. Once this objective function has converged, by calculating L_2 norm of W_{ij}, the energy that is distributed between i and j position can be captured. Later, in order to overcome to some phylogenetic biases and entropy effects, applying average product correlation [25] is recommended. plmDCA, GREMLIN and EVCouplings are all based on this idea with small modifications. They all were employed to predict protein structures based on sequences. CCMPred is a parallelized implementation of GREMLIN to measure co-evolving scores [26, 70, 40].

After GREMLIN, PSICOV applied the idea of estimating the inverse covariance matrix to measure co-evolution score by employing the L_1 penalty [42]. At the same time, mfDCA used mean-field approximation to perform the exact same task. PSICOV and mfDCA are very similar to each other [64]. Our methods just like PSICOV, convert a MSA to binary representation of it. For example, every position is replaced by a binary vector of size 21 (20 amino acids and gap) followed by calculating the covariance of the binary matrix. Finally, Graphical Lasso is used to solve it with fixed scalar L_1 penalty. On the estimated precision matrix, they follow the same post processing as GREMLIN by calculating the energy that distributed between each 20×20 submatrix of the precision matrix and using average product correlation for correction. At the end, they fit into the logistic curve to get strong correlations. Ovchinnikov benchmarked PSICOV with GREMLIN and he showed that moving from maximum entropy to PSICOV, improves the accuracy by 10% while by employing GREMLIN the accuracy improves 10% than PSICOV even though the speed of PSICOV is better.

An alternative to pseudo-likelihood was proposed by some groups [7, 17] as Adaptive Cluster Expansion (ACE) based on minimizing cross entropy for each cluster of sites and later it implies Boltzmann Machine to correct h, J in equation 2.1. The result shows that it can outperform plmDCA by a little bit.

All these DCA based methods have MSA biases, phylogenetics dependencies biases between different species, and indirect couplings which are captured as covariations may mislead the prediction. Sukowska et al [77] showed that not all the covariations that can be measured from MSA which also belong to the different branch of the phylogenetic tree, corresponding to co-evolution. Miyazawa proposed a method to remove phylogenetic biases and showed that the result will improve if we correct for these biases by determining partial correlation [61].

In 2015, multiple groups started to combine some prior knowledge into both pseudo-alignment and PSICOV either using machine learning approaches such as neural network, deep learning, random forest or as a new penalty. PconsC combines PSICOV and plmDCA into a random forest and in the second release they replace random forest with 5-layer neural network [76]. Meta-Psicov which was developed by the same group who developed PSICOV, considered 2 stages of a neural network. In the first stage, it combines PSICOV, mfDCA, and CCMPred to predict contact sites, later in the 2nd stage filters the result based on other features such as amino acid propensity, hydrogen bonds, and secondary structure [43]. Meta-PSICOV was the state of the art for predicting contact site between and within a protein until RaptorX was developed. They developed CoinDCA [56] at the first, by adding new group lasso penalty to the original PSICOV and also learned some supervised prior which came from protein propensity, sequence profile, mutual information and so on. This new prior information could outperform plmDCA. After coinDCA in 2017, same

group applied convolutional neural network along with CCMPred, as a result, it performs better than all other methods [83].

The main drawback of all these methods is that they are dependent on the quality of an MSA, and if a protein is conserved across multiple families, there is no signal for co-evolution to be captured. As a result, learning prior which came from other non-MSA based method can significantly improve these methods.

2.2 Method

A first step toward learning correct prior information is to extract some related features from both sequence and structure. This prior matrix made of 3 different methods and at the end, we apply a weighted ensemble (WE) learning to tune the coefficient for each model. At the same time we use binary matrix of msa to build covariance to pass along with penlaty to garphical lasso.

2.2.1 Extracting Potential Interfaces in a Protein Using Intpred

This model is based on a combination of sequence features and structural features. There are a lot of methods that try to do this task. After an extensive literature review and based on the features that contribute to the protein binding site, we decided to use Intpred method [67]. The input for Intpred is a PDB ID and corresponding chain ID and it extracts information from PDB file and applies a random forest to predict potential interface residues. Here we explain some of the key steps that Intpred considered to extract features.

The algorithm starts based on the following terms as defined in original work:

- **Patch centre atom** is the central atom that around the patch is built.
- **Patch radius** is the threshold distance from the patch centre atom used to select candidate residues for inclusion within the final patch.
- **Contact radius** is defined for a pair of atoms as the sum of their van der Waals radii, plus a tolerance (here set to 0.2 $\mathring{A}$). Two atoms are in contact if the distance between their centres is less than the contact radius.

Interestingly, these features can be used to approximately characterize the surface residues from non-surface residues.

The algorithm starts with identifying patch center atoms. Residues with relative SASA greater than 0.25 corresponds to patch center residues and among all atoms of that residue, the atom with largest SASA represents patch center atom. Then for each patch center atom c, and contact radius R (set to 14$\mathring{A}$) algorithm 2.2.1 builds patches. The output is set of all the patches P, with respect to patch centers.

Algorithm 2 Building-Patch(PDB,c,R)

```
1: P = {c}; // c is patch center atom
2: Find N from PDB // N is all the residues that have at least one atom within patch
   residue from c (Neighbour of c)
3: newP = P
4: while newP ≠ {} do
5:    newP = {}
6:    for i ∈ P do
7:       for j ∈ N do
8:          if Distance(i, j) < R   and   SolventAngle(i, j) < 120° then
9:             newP=P∪{j}
10:            N = N − {j}
11:            P = newP
12:         end if
13:      end for
14:   end for
15: end while
16: return P
```

Table 2.2: Features that are being used in Intpred method

Feature	**Description**	**Source**
Hydrophobicity	Kyte and Doolittle hydrophobicity scale	Sequence
Homology	Homology Conservation Score Based on Valader01 Score	Sequence
Conservation	FEP Score for finding functionally equivalent orthologues	Sequence
Propensity	Residue Propensity based on position and type	Sequence and Structure
Disulfide Bonds	Disulfide Bridge with in 2.2 $\AA$ Distance + 10% tolerance	Structure
Hydrogen Bonds	Binary Score if exist any H Bonds	Structure
α-Helix	if percentages of α-Helix >0.2 and β-Sheet$\leq$0.2	Structure
β-Sheet	if percentages of α-Helix $\leq$ 0.2 and β-Sheet>0.2	Structure
mix	if percentages of α-Helix >0.2 and β-Sheet>0.2	Structure
Coil	if percentages of α-Helix $\leq$ 0.2 and β-Sheet $\leq$0.2	Structure
Planarity	RMSD of all atoms in a patch from best fitted Plane	Structure

Then every residue that has at least one atom of it in P, belong to patch P. A residue is called interface if the difference between relative solvent accessible surface area (RSASA) of the residue and relative solvent accessible surface area with respect to all the residues in the patch is greater than 0.1. Then, the interface fraction for a patch P is calculated as sum over all RSASA of the interface residues in patch P over RSASA of all the residues in P. Then a label is assigned for each patch P as follows:

- I (interface), if fraction is larger than 0.5.
- S (surface), if fraction is equal to 0.
- U (unlabeled), otherwise.

Patches with U label are excluded from training in order to keep the problem as binary classification. Table 2.2.1 describes all the features that contributed as the predictors to random forest with considering the class labels (I and S) as the responses. lastly for those patches that predicted as I class are selected and the patch center residue (RSASA $>$ 0.25) represents the binding site. The features that contributed in Intpred are highly compatible with our literature review. However, the

drawback of this method is that it relies on the class labels. Although the method that is being used to determine the class labels is a necessary condition for being a patch to be an interface, it is not enough. In other words, if a protein has multiple interfaces with respect to other partners, the prediction of Intpred returns all of them. But, with high confidence, we can say usually the actual interface is a subset of whatever it predicts as an interface. Finally we use the probabilities obtained by Intpred and build a matrix of joint probability for every two residues between protein A and B. The joint probability matrix is $M^1 \in R^{n \times m}$ where n and m are number of residues in protein A and B, respectively. Then the $P_{i,j}$ element is constructed as $P_{i,j} = P_i \times P_j$ where P_i is the probability of i^{th} residue is interface for protein A. P_j is corresponding value for protein B in j^{th} residue.

2.2.2 Extracting Potential Interfaces From Docking Pattern

Docking algorithms are divided into two main categories: direct and template-based methods [49]. We used the ClusPro webserver [49] to get docking models of two proteins. ClusPro is a direct search based method which relies on thermodynamic constraints. As it is shown in figure 2.4, it is a three step hierarchical method which returns 10 clusters with the best scores. ClusPro has 6 energy functions that depend on the type of the complex which is another reason that we used this method. The first step is rigid body docking by simulating multiple random conformations. Next, clustering the top 1000 lowest energy complexes using an RMSD-based scoring function, and finally, filtering the structures based on energy minimization. In the following we explain each step briefly:

- **Rigid Body Docking**: this step relies on PIPER [48] method which is based on Fast Fourier Transform (FFT) correlation approach. It places protein A at a origin of the coordinate system on a fixed grid, and perturbs the second

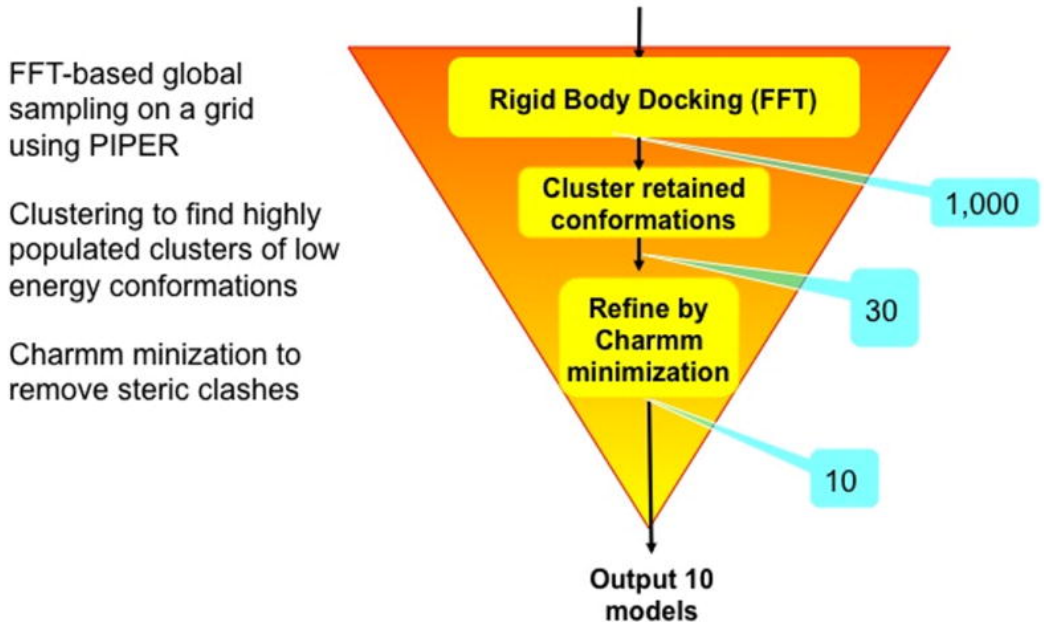


Figure 2.4: Overall view of ClusPro method [49].

protein on a moveable grid. Then the docking energy is calculated based on FFT correlation function. The correlation function made up electorstatic interaction and desolvation contribution. Considering a shape complementary is another advantage of ClusPro. As a result, it returns the 1000 lowest energy structures which are within 10 $\mathring{A}$ from the native structure as the candidates of docking.

- **Clustering of Highly Populated Conformations**: The goal of this step is to cluster the 1000 complexes that are generated in the previous step based on pairwise interface root mean square deviation (IRMSD) scoring function. Candidate complexes are divided into different clusters by calculating the pairwise IRMSD between every two structures. Then the structure with the largest number of neighbour structures that are within 9 $\mathring{A}$ IRMSD is denoted the center of the first cluster. All the structures that are within 9$\mathring{A}$ IRMSD from it are assigned to the first cluster. Then the first cluster is removed from the process

and the same procedure is applied to build the second cluster and so on. As a result, the top 30 clusters are returned in this step.

- **Refinement by energy Minimization**: Finally for each cluster, the Van der Waals energy is minimized using the Charmm potential function for up to 300 steps with a fixed backbone to remove small steric clashes. Finally, the top 10 populated cluster centers with cluster members are returned.

For each two given proteins, ClusPro returns the top potential docking models (on average 50 to 150 structures in top 10 clusters). We get these structures and perform voting count for every two residues between two proteins and normalize it. The residue pairs are scored according to the number of times they appear on the interface of the docked complexes. In other words, the residues are scored according to their probability of being placed on the interface by the docking program. We build the joint probability matrix $M^2 \in R^{n \times m}$ where n, m are the number of residues for proteins A and B, respectively. Initially all the elements are set to 0. Then for all the predicted docking complexes we measure the distance between every pair of residues (i, j) where $i \in [1, n]$ and $j \in [1, m]$. corresponds to the distance between two residues in a complex. If the distance is less than 8 $\mathring{A}$, we increment $M^2_{i,j}$ by one. By doing this calculation for all the predicted complexes of two input proteins, the matrix M^2 is determined where a large number for a position represents high probability of being on a binding site. In order to overcome uncertainly and to correct for noises, we apply a Gaussian smoothing filter on the M^2 matrix with a kernel size of 3, $\mu = 1$, and $\sigma = 0.5$, followed by dividing the smooth matrix by the maximum value in the matrix in order to turn these values into probabilities.

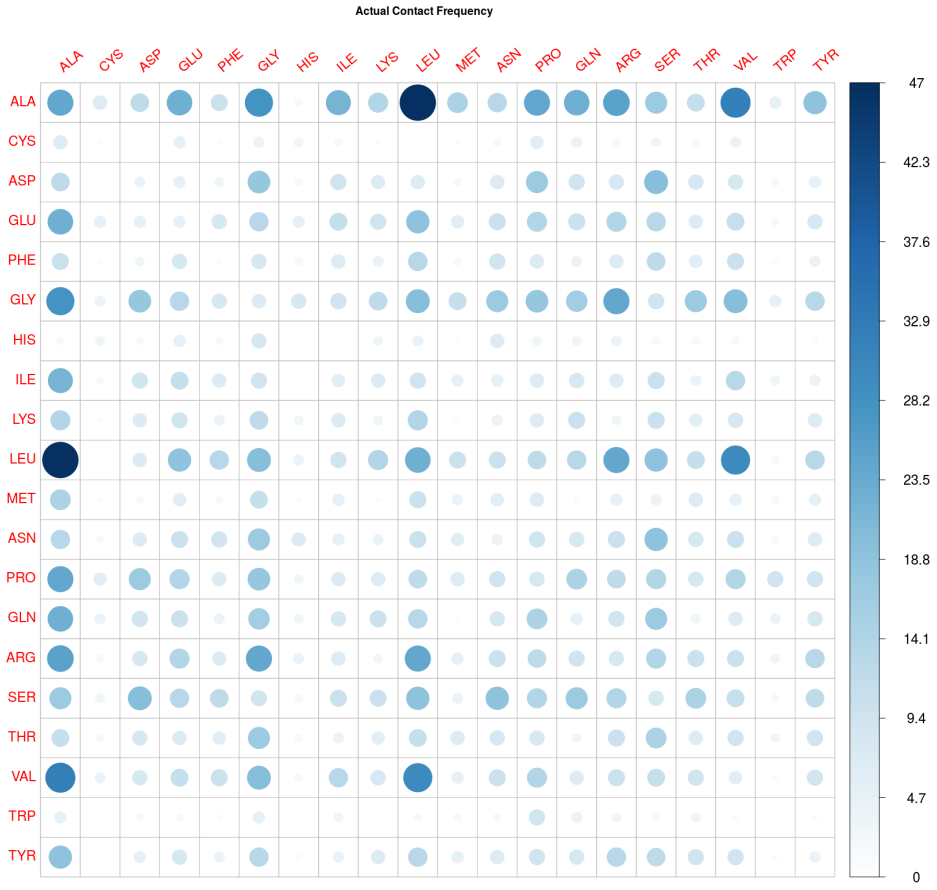


Figure 2.5: which was performed on the training data.

2.2.3 Measuring Amino Acid Contact Propensity

As discussed above, not all the amino acids have an equal tendency to interact with other amino acids. In order to measure this tendency more precisely, we used the training data to measure this correlation. Figure 2.5 represents the frequency of every two amino acids to be in contact (if the distance between two residues is $< 12\mathring{A}$) among the 58 pairs of interacting proteins in our set. We then normalized the counting matrix by dividing all the values by the maximum element in the matrix to turn the numbers into probabilities. The M^3 matrix was built based on the normalized amino acid propensities where element $M^3_{i,j}$ is directly updated from propensity matrix based on type of amino acids in $i \quad and \quad j$ and corresponding in the propensity matrix.

2.2.4 Data set and Simulated Data

Two benchmark data sets are used that provided by EVCoupling and GREMLIN [40, 70] papers which contains set of 58 and 28 homodimers from Escherichia coli (abbreviated as E. coli) bacteria, respectively. We used set of 58 proteins as a training set and learned the coefficient of each model and tested on the set of 28 proteins. We have also simulated some MSAs to test our hypothesis and measure the learning model which will be discuss later. The goal of simulation data is to have a set of MSAs with low co-evolution signal up to high co-evolution signal. This leads us to use both 1^{st} and 2^{nd} order Hidden Markov Model (HMM). The signal of co-evolving is controlled by three parameters: α, conservation, and bias which control the co-evolving, conservation of a co-evolving residue, and the changes in mutation, respectively. These MSAs were built based on PAM, blosum62, and also the distribution of amino acids in all the PDB structures. We tried to build an MSA of size 1000×200 based on this information. After that, we set 6 pairs of columns as a co-evolving pair. We modified the distribution of co-evolving columns by 3 parameters that mentioned in above. α is a score between 0 and 1 which controls the transition probability of a 21^2 states HMM. 0 means no co-evolution and 1 represents the full co-evolution. Another parameter is conservation which regulates the rate of amino acid changes from one type to another. 0 conservation means that we expect to see no conservation and 1 represents that co-evolving occurs between 2 amino acid types. Finally, bias controls the PAM matrix. We have a fair bias which represents an original PAM matrix. The motivation of this simulation is to investigate whether perturbation of the penalty matrix in poor MSA (small α) increases the co-evolution score in specific positions or not. For example by putting low penalty on co-evolving pairs and high on the rest of the matrix for a MSA with small α, we expect to see those co-evolving pair with low penalty on the contact pairs.

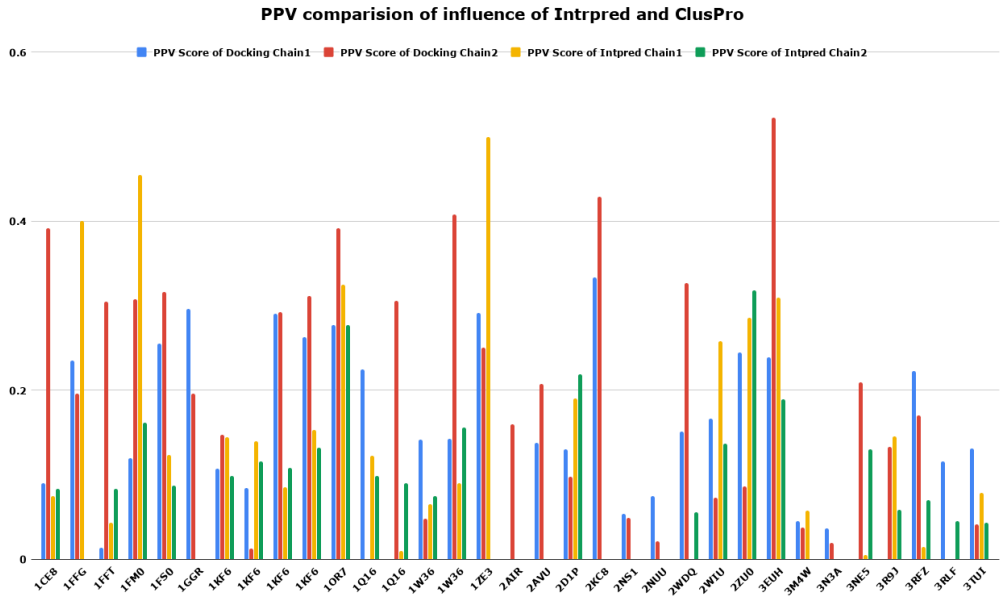


Figure 2.6: with respect to actual contact for result of docking (ClusPro) and Intpred on the test set of 38 proteins

2.2.5 Train Coefficient in Weighted Ensemble Model

So far, we build 3 independent models: M^1 based on structural and sequence features, M^2 based on docking models, and M^3 based on amino acid propensity. Each of these models can capture the probability that two residues are contacting in two proteins. In order to improve the accuracy and also measure the impact of each model separately, we used precision or Postive Predicted Value (PPV) score ($\frac{TP}{TP+FP}$) due to the fact that this is an unbalanced problem there the positive (interacting) residues are highly outnumbered by the negative (non-interacting) residues. Basically, the true negatives constitute the vast majority of the data, which could skew the results. Figure 2.6 represents the precision of training set associated with each model. The average precision score for Docking and Intpred were 18% and 10%, respectively in training set. This shows that the impact of Docking model is two times of Intpred, so we give the Docking model twice the weight of Intpred.

2.2.6 Learning Penalty Matrix and Turn Prior Information into a Penalty Matrix

After determining all the 3 models ($M^{(1,2,3)}$), the final probability is calculated as a linear combination of all the models which are given in equation 2.3. M is a $n \times m$ matrix where every element $M_{i,j}$ is corresponding to the probability of interaction between i^{th} residue from protein A to i^{th} residue from protein B given all the three models. $W_{1,2,3}$ is learned from training data.

$$M = \frac{w_1 \times M^1 + w_2 \times M^2 + w_3 \times M^3}{w_1 + w_2 + w_3} \tag{2.3}$$

The idea behind learning the probability model M is to impose this information as an L_1 penalty in estimating the precision matrix, θ such that those pairs with high probability of contacting are penalize less while pairs with low probability are penalized more, with the hope that this learned penalty would help co-evolving pairs with low evolution score to have a large value in the precision matrix. Converting the probability matrix M into a compatible penalty form for estimating a sparse precision matrix is a challenging part because of algorithm for learning the penalty which has a direct effect on the performance of the model, therefore, the connectivity of the underlying graph. Accordingly, having a way to learn the correct range for a penalty matrix is mandatory for what we are proposing.

As we explained in the introduction chapter, the graphical lasso algorithm enables us to estimate a sparse precision matrix based on the given empirical covariance matrix S and penalty matrix Λ. Λ controls the sparsity of a network, so having, an upper bound for Λ can improve the learning algorithm. An upper bound is defined as a $_{max}$ which is minimum value greater than 0 that can return a fully disconnected

graph, which means all the elements in the precision matrix (except for the diagonal elements) are zero.

$$\max_{\theta} log(det\theta) - tr(S\theta) - \Lambda||\theta||_1 \tag{2.4}$$

As shown in the introduction, the objective function of the graphical lasso problem is given in 2.4, where θ, S, and Λ are the precision, empirical covariance, and penalty matrices, respectively. Algorithm 1.3.3 can solve this optimization problem iteratively. By taking a closer look at the algorithm, especially in the updating formula, we can find λ_{max} in Λ matrix. Equation 2.5 represents updating statement for coefficients of each variable. λ_{max} can set $\hat{\beta}_j = 0$. As a result, the precision matrix would have a zero value for the corresponding element.

$$\hat{\beta}_j = \frac{t(current.s_j - \sum_{k \neq j} current.\omega_{kj}\hat{\beta}_k, \Lambda)}{current\omega_{jj}} \tag{2.5}$$

t is a soft-threshold function in equation 2.5 which is defined as:

$$t(x, \sigma) = \begin{cases} 0 & |x| \leq \sigma \\ x & x > \sigma \end{cases} \tag{2.6}$$

Based on equation 2.6, λ_{max} needs to be larger than $|current.s_j - \sum_{k \neq j} current.\omega_{kj}\hat{\beta}_k|$. Using proposition 2.1, an estimation of λ_{max} can be calculated from empirical covariance matrix S.

proposition 2.1- $\hat{\beta}_j$ is equal to zero in equation 2.5, if $\Lambda_j \geq ||S||_\infty$.

proposition 2.1 claims that if we set a penalty value for $\Lambda_{i,j}$ element as $\lambda_{max} \geq ||S||_\infty$ then the value of corresponding position in precision matrix is equal to zero.

This can be proof by tracing the algorithm 1.3.3. This can use this idea in order to set penalty for those pairs that have a low probability in M or pairs within a protein, close to λ_{max}. As a result, the correlation score for these pairs are 0.

Once λ_{max} is determined, matrix Λ is built which is a same size as S updated based on the matrix M as follows:

- $\Lambda_{j,i} = \Lambda_{i,j} = \lambda_{max}$, where i, j belong to only one protein
- $\Lambda_{j,i} = \Lambda_{i,j} = \lambda_{min} + C \times \lambda_{min}(1 - \frac{M_{i,j} - min(M)}{max(M) - min(M)})$, where, i and j belong to protein A and B, respectively. $C = \frac{\lambda_{max}}{\lambda_{min}}$ is constant that obtained from training set.

λ_{min} and C are equal to 0.0001 and 30, therefore those pairs with high probability of interaction based on M, have a penalty close to 0.0004, while the pairs with low probability get a penalty close to 0.003. The penalty for the remaining pairs is distributed linearly in this range with respect to corresponding value in M. After Λ is constructed, the algorithm 1.3.3 is called by passing S and Λ. The output of this algorithm is the estimated precision matrix. These conditions help to measure only the co-evolution score between two residues that are coming from two different proteins by blocking the contribution of co-evolution within a protein. This does not change anything in this problem due to independence of variables assumption.

In practice, setting a penalty to λ_{max} for a specific pair of residues $\Lambda_{i,j}$ is not a good idea specially in this context. Since it may eliminate the correlation score that either i^{th} or j^{th} may have had with other residues within a protein and consequently, it may affect the final precision matrix. This matter is investigated by normalizing using Average Product Correction (APC) which was introduced in 2008 by [25] for correction and normalization of phylogenetic tree biases.

2.2.7 Post Processing of the Precision Matrix

A precision matrix θ, which is the output of the graphical model, needs to be converted into a ranking based score with respect to the correlation score. This normalization is being performed in two steps just like PSICOV and CCMPred . Q_{ij} is defined as correlation energy that is being captured between position i and j in the MSA, which is determined by taking L_2 norm of all the 20×20 submatrices from θ (excluding correlation score for the column).

Then, the average product correction is applied as follows in order to remove phylogenetic biases:

$$\hat{Q}_{ij} = Q_{ij} - \frac{Q_{i.} \times Q_{.j}}{Q_{..}} \tag{2.7}$$

Where, $Q_{i.}$ is sum over all the columns in the i^{th} row. $Q_{.j}$ is similary defined with respect to all the rows in i^{th} column. $Q_{..}$ is a sum over all the Qs. Finally, we fit $\hat{Q}_{ij}$ into a logistic curve to avoid any extreme values.

2.2.8 Combining the Results with PSICOV

We also run PSICOV method with fixed penalty to measure the co-evolving score, since we can get highly co-evolving pairs between proteins A and B by taking into account those Q_{ij} where $\{i, j\}$ belong to one protein in the APC normalization step which are eliminated from our method. Finally, by taking the union between the results of Graphical Lasso with the learned penalty and PSICOV output, further sorting them by $\hat{Q}_{ij}$, the top L pairs correspond to the potential interfaces between the two proteins .

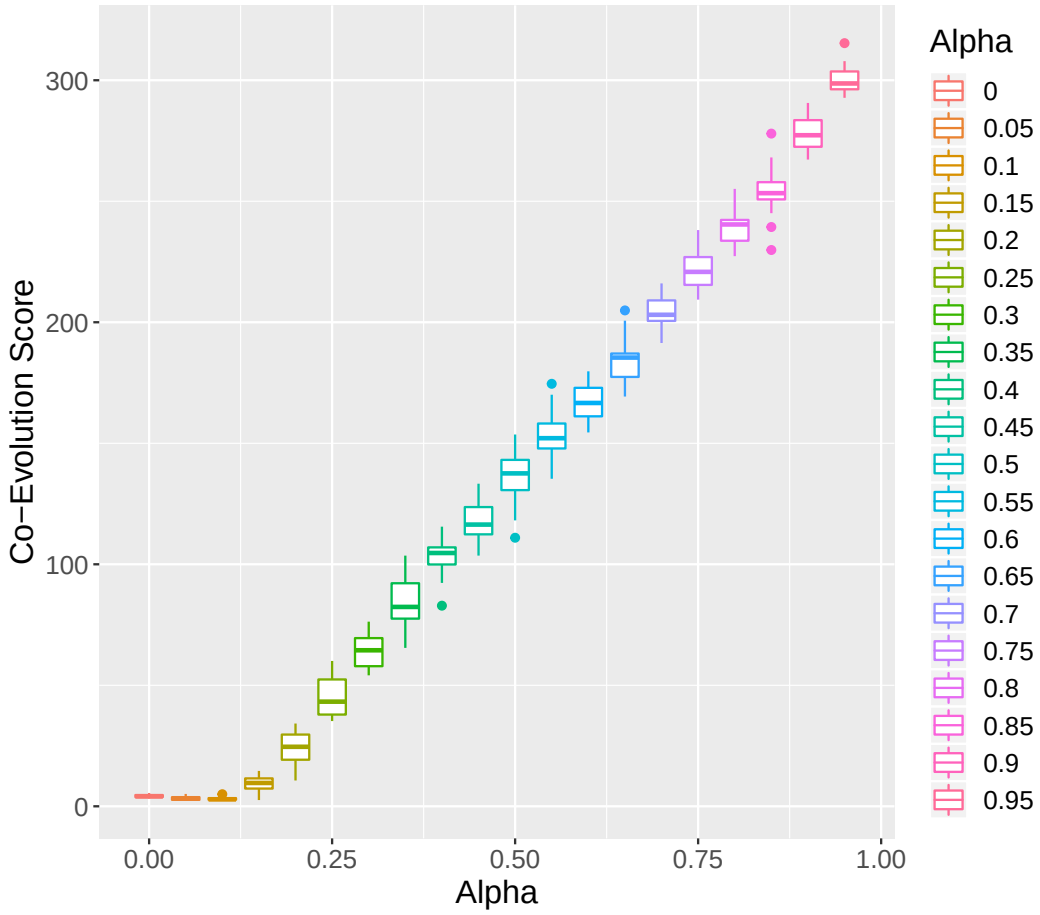


Figure 2.7: , the score of correlation increases among top 6 pairs as well

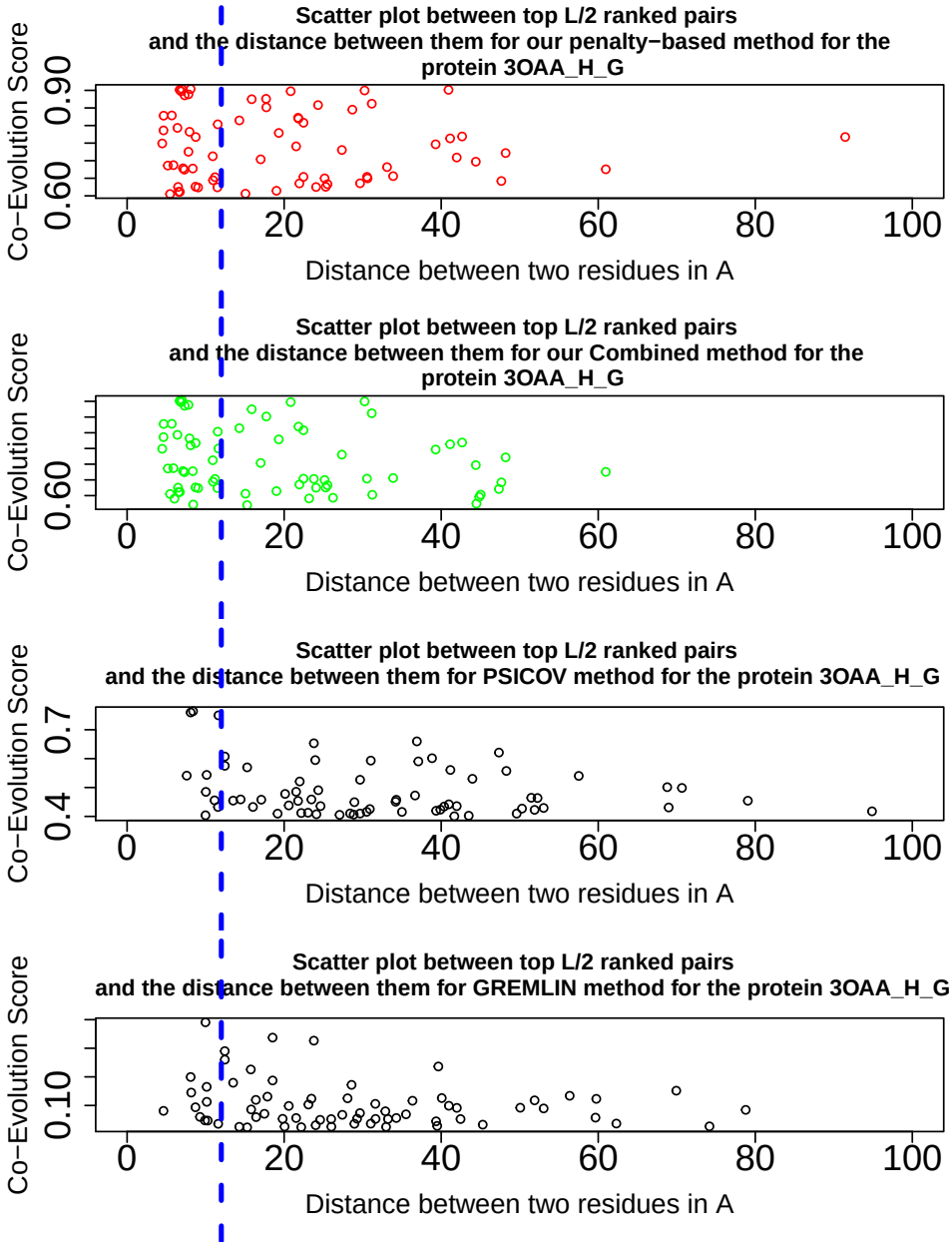


Figure 2.8: in top $L/2$ with contact cutoff 12 $\mathring{A}$ (blue line).

2.3 Results

2.3.1 Simulated Data Performance

In total, 21 MSAs generated with size of 1000×200, as it was described in data section. The motivation of this analysis is to see if we can design a penalty matrix that can help those MSAs with a low rate of correlation among pairs, get improved and stand out on the top ranked. These MSAs ranged from low co-evolution ($\alpha = 0$) up to high co-evolution ($\alpha = 1$). Figure 2.7 depict values of top $\hat{Q}_{ij}$ in y axis. x axis is representing 21 different MSAs. This curve represents that GREMLIN can capture the evolution score as α increases. Seeing improvement in the ranking of co-evolving pairs, is expected by penalizing low and high scores for co-evolving other pairs in Λ. This analysis showed by designing an appropriate penalty for each MSA. As a result we could get the co-evolving pairs on the top ranked pairs for MSAs with $\alpha > 0.3$ while we could not see those pairs with a fixed penalty.

2.3.2 Result of Test Data

Multiple models evaluated by eliminating each model at a time. Also, 3 different distances are considered as 8, 10, and 12 $\mathring{A}$. Finally, we measure PPV among top L, $L/2$, $L/5$, and $L/10$ ranked pairs where L is the number of residues in the smaller protein between two proteins. Figure 2.9 compares the performance of our algorithm in compare with GREMLIN and PSICOV methods.

16 out of 19 complexes, we performed better than PSICOV with average of relative improvement of 40 %. As it is been shown on other works, GREMLIN performs better than PSICOV. The relative improvement between our method and GRELMIN was 20 %, also, GREMLIN out performed better than us just in 6 complexes out of 19.

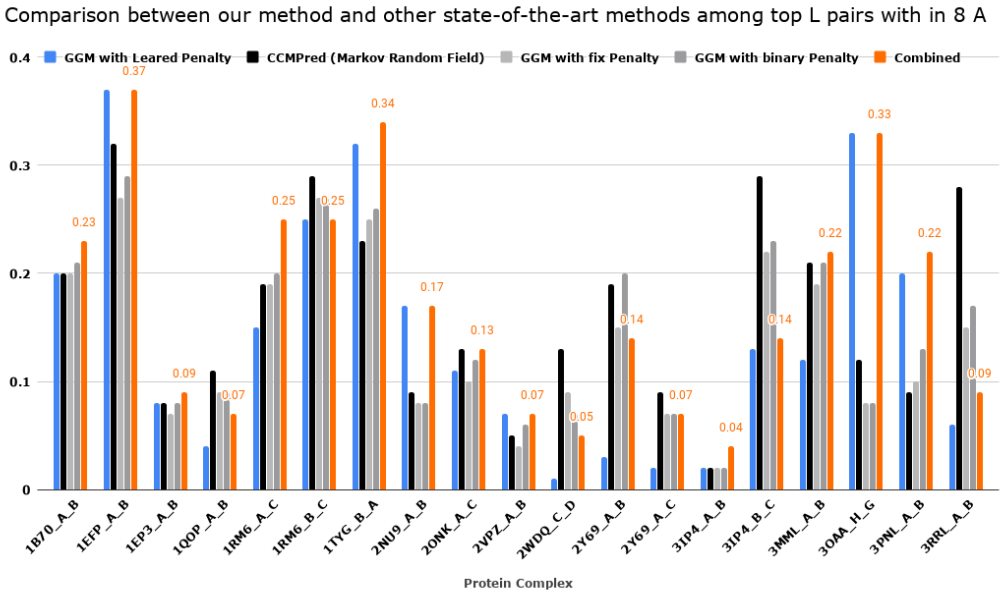


Figure 2.9: We outperformed all other method in more than 12 complexes out of 19.

Figure 2.8 compares the top $L/2$ ranked pairs against the actual distance in 3OAA between H and G chains in comapre with 2 other methods. As a result, our proposed method performed significantly better result in this case with less stringent condition (contact distance is $12\mathring{A}$).

Figure **??** represented this comparison between performance of other state-of-the-art methods for each complex. Relative comparison helps to understand how good we predict in compare with the performance of PSICOV and GREMLIN. From this figure, the impact of a prior knowledge in terms of penalty matrix improved the over performance of the model and helped poorly correlated residues in MSA stood out on the top ranked pairs.

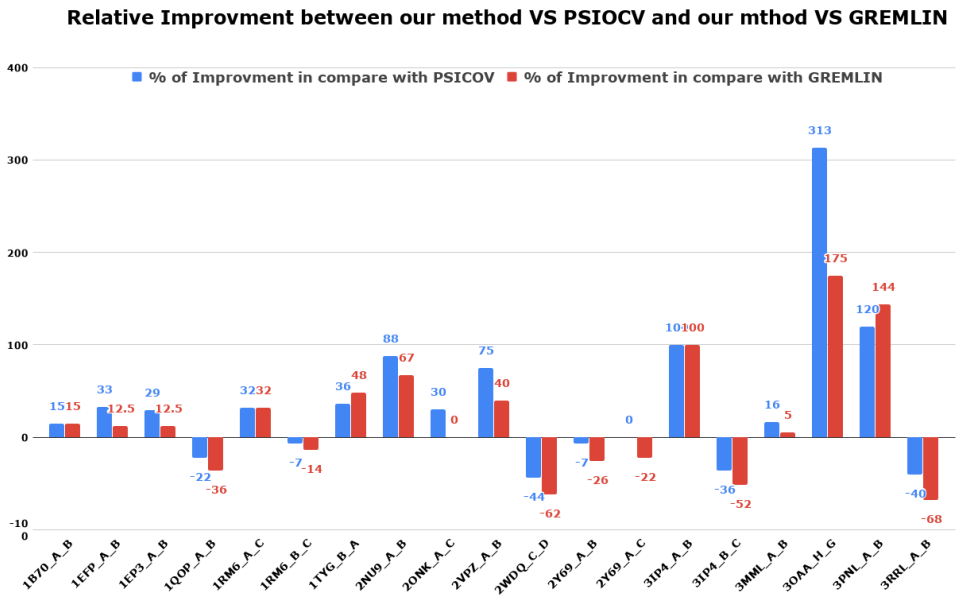


Figure 2.10: on average is 20 and 40 presents over GREMLIN and PSICOV methods, respectively.

2.3.3 Computational Run Time Compression

In general solving GGM is much faster problem than MRF which is used in PSICOV. We used PSICOV settings by applying Cholesky decomposition on covariance matrix to perturb it into a positive definite matrix. On the top of that, by setting a high penalty for within protein residues, the total number of iteration is decreased significantly. So the total run time is equal to one run of the PSICOV with fix penalty. We also can run the ClusPro part simultaneously with PSICOV.

2.4 Future Work

2.4.1 Rigidity Analysis of a Protein

Proteins in terms of dynamics are either flexible or rigid. As we discussed before, a protein in order to perform its function requires to conformational change between inactive and active. It is shown that the conformational changes are associated with protein function therefore, we applied rigidity analysis to investigate the association between rigid-flexible clusters and binding site clusters. Rigidity analysis can predict the cluster of residues that are likely to move together [32]. For this purpose, we choose Kinari algorithms. Figure 2.2 shows an overview of the method. From a molecule, it models as a mechanical structure called a body-bar-hinge where each the covalent bar represented as hings, and other elements such as hydrogen bonds and hydrophobic interactions are bar. Then an internal special multi-graph is built where each body represents a vertex and each hing corresponds to an edge. A pebble game algorithm calculates components in the multi-graph and rigid clusters are being measures. The output of Kinari is some clusters where all residues that belong to one cluster are more likely to move together.

2.4.2 Patch Building and Graph Analysis

Another idea is to build patches using Intpred on the top $L/10$ ranked pairs. After we have the patches, then they can be mapped into a graph using similarity by the Jaccard distance between two nodes (residues belong to patches) and finding the best patches which corresponds to interfaces.

Table 2.3: where contact distance is $8\mathring{A}$ among top L pairs. It also represents total number of residues in two proteins.

PPV of top L pairs within $8\mathring{A}$						
Complex	**GGM_Penalty**	**GGM_Binary**	CCMPred	PSICOV	**Combined**	Complex Length
1B70_A_B	0.2	0.21	0.2	0.2	**0.23**	1032
1EFP_A_B	**0.37**	0.29	0.32	0.27	**0.37**	554
1EP3_A_B	0.08	0.08	0.08	0.07	**0.09**	573
1QOP_A_B	0.04	0.09	**0.11**	0.09	0.07	699
1RM6_A_C	0.15	0.2	0.19	0.19	**0.25**	919
1RM6_B_C	0.25	0.27	**0.29**	0.27	0.25	481
1TYG_B_A	0.32	0.26	0.23	0.25	**0.34**	308
2NU9_A_B	**0.17**	0.08	0.09	0.08	**0.17**	671
2ONK_A_C	0.11	0.12	**0.13**	0.1	**0.13**	593
2VPZ_A_B	0.07	0.06	0.05	0.04	**0.07**	928
2WDQ_C_D	0.01	0.07	**0.13**	0.09	0.05	237
2Y69_A_B	0.03	**0.2**	0.19	0.15	0.14	487
2Y69_A_C	0.02	0.07	**0.09**	0.07	0.07	773
3IP4_A_B	0.02	0.02	0.02	0.02	**0.04**	968
3IP4_B_C	0.13	0.23	**0.29**	0.22	0.14	575
3MML_A_B	0.12	0.21	0.21	0.19	**0.22**	493
3OAA_H_G	0.33	0.08	0.12	0.08	**0.33**	423
3PNL_A_B	0.2	0.13	0.09	0.1	**0.22**	868
3RRL_A_B	0.06	0.17	**0.28**	0.15	0.09	425
Average of All Complexes	0.14	0.15	0.16	0.14	**0.18**	

2.5 Conclusions

In this chapter we proposed a new method for improving the prediction of binding site between two interacting proteins. We show that by inferring structural information from each protein we can impose this information as a penalty to a graphical lasso made from a multiple sequence alignment of the protein families. Further, we compared our method with two other state-of-the-art methods. The overall results show significant improvement in positive predicted value. Our method also performed better in large complexes where all other methods that rely on sequence alone cannot perform well. Table 2.3 represents this comparison for the most stringent condition where contact distance is $8\mathring{A}$ and we took top L ranked pairs. The average precision in the most least stringent condition is 58 % for our method where contact distance is $12\mathring{A}$ and top $L/10$ ranked pairs.

Chapter 3

CLUSTERING PROTEIN CONFORMATIONS USING A DYNAMIC PROGRAMMING BASED SIMILARITY MEASUREMENT

3.1 Introduction

Understanding the structure and dynamics of proteins is essential in order to understand their function. In particular, it is important to detect clusters of highly populated regions which could correspond to intermediate structures or local minima. The conformational space of proteins is complex and high dimensional, which makes its analysis a highly challenging task. We present a Dynamic Programming (DP) method for clustering and classification of protein conformations, based on their lower-dimensional representation. Previously, we used the similarity method to identify pairs of co-regulated genes based on their microarray expression data. In this chapter we demonstrate our method on trajectories obtained by a coarse grained protein conformational search of three different proteins. Our clustering method was extremely fast, and was able to produce compact, well separated clusters for all the tested examples, showing that both the DP-based method and the dimensionality reduction technique were able to preserve the inter-molecular distances and provide clusters that correspond to experimentally determined intermediates when such are available.

3.2 Background

Characterizing the conformational space of proteins is crucial for understanding the way they perform their function. Understanding the connection between protein structure, dynamics and function can contribute substantially to our understanding of cellular processes involving proteins. The question of how the structure and dynamics of proteins relate to their function has challenged scientists for several decades but still remains open. Conformational exploration methods aim to characterize the conformational space of proteins in order to find minimum energy regions corresponding to highly populated structures [60, 52, 38]. These intermediate states are transient and therefore hard to detect experimentally. However, they may be crucial to understanding dynamic events such as folding, docking, binding and conformational change processes. The potential energy landscape of a protein is often rugged and has a large number of local minima [12]. This makes it difficult to navigate. The problem becomes even more challenging due to the fact that a typical protein can contain several hundreds of amino acids or several thousands of atoms. Therefore, the search space made out of all possible conformations that a protein can assume is large and its enumeration is practically impossible. Existing physics-based computational methods that sample the conformational space of proteins include Molecular Dynamics (MD) [14], Monte Carlo (MC) [47] and their variants, as well as approximate methods based on geometric sampling [38, 71, 75, 1, 36], Elastic Network Modeling [86], normal mode analysis [33], morphing [85] and others.

Even after the conformational space is sampled, it should be filtered and clustered to extract meaningful information. Several clustering methods have been designed for protein conformational space [71, 82, 15]. Today, the majority of clustering methods for multi-dimensional data incorporate metric functions that evaluate the distance between objects in the dataset, or a lower-dimensional representation of these objects.

In this scenario, multiple dimensions are combined and are simultaneously considered according to a metric function in order to create a set of clusters. Due to the complexity and high dimensionality of protein structures and of events such as protein folding and binding, finding local and global energy minima becomes a problem of navigating and analyzing a complex, high dimensional space. In particular, we need:

- (a) A way to measure the similarity between two structures. This is not a trivial task. Standard methods such as Root Mean Square Deviation (RMSD) require a correspondence list between atoms of the two molecules, which may be a problem if comparing two instances of different molecules. RMSD also tends to average out localized changes. Other similarity measurements exist [5, 71], varying in their robustness and applicability to various types of molecules.

- (b) The conformational space of protein structures is very high dimensional. Most search and clustering methods do not scale up to hundreds or thousands of dimensions and therefore use a lower-dimensional projection of the search space, justified by the fact that the intrinsic dimensionality of protein structures is much lower due to the constraints between different parts of the protein. Dimensionality reduction methods aim to find a small set of collective coordinates that capture the main variability in the data. Such techniques include Principal Component Analysis (PCA) [9], Isomap [78] and more. These methods project the protein structure spaces onto a low-dimension space which captures desired properties in the structure. This topic is further discussed below.

- (c) The detection of outliers and determining the number of clusters, and in general measuring the quality of the clustering method. Some common clustering methods such as k-means [57] do not have outliers, and the number of clusters has to be defined in advance.

Hierarchical clustering methods result in a multi-scale view of the conformational space and enable us to view the hierarchical relationship between the local minima produced by the conformational search.

3.3 Method

3.3.1 Protein Conformational Search

Table (3.1) shows the proteins used in this work. Each conformational pathway was modeled in both directions, using a Monte-Carlo (MC) based search described below. Due to the size of the proteins a fully atomic representation of the structure is computationally costly. Therefore, the proteins were represented using their C-α atoms and the energy was estimated using a C-α based energy function [87]. During the search each intermediate conformation is projected onto a lower-dimension feature space for efficiency and each conformation is represented using an M-dimensional feature vector where $M << N$ (N is the number of amino acids in the protein. M is usually around 8-15 (See [38] and Section 3.3.2 for more details). The distance between a given conformation's feature vector and that of the goal structure is used as a score to measure the progress of the search. The lower the score, the closer a given conformation is to the goal structure. The search was run for a maximum of 10000 iterations and at every iteration a rotatable bond between two C-α atoms is selected. The bond to rotate is selected with a probability linearly proportional to the difference between this angle and its counterpart in the goal conformation, which serves as a bias of the search and a flexibility detection method. The selected angle was rotated by a random value between -5 and 5 degrees. The new conformation is validated by the potential energy function and considered further only if its energy is below a threshold. The feature vector score of the new conformation, FV_{new} is calculated and

Table 3.1: The tested conformational pathways. The PDB codes denote the end-points.

Name	RMSD	Residues	PDB	Conformations	no. clusters
AdK	6.95	214	1AKE→4AKE	5,235	20
			4AKE→1AKE	6,588	20
Calmodulin	14.72	144	1CLL→1CTR	11,483	47
			1CTR→1CLL	3,232	49
GroEL	12.21	525	1SS8→1SX4	1,689	41
			1SX4→1SS8	1,528	44

compared to that of current conformation, FV_{cur}. The new conformation is accepted according to the Metropolis criterion, if either of the following occurs:

1. $|FV_{new}| < |FV_{cur}|$
2. $r < e^{(-(|FV_{new}|-|FV_{cur}|)/(|FV_{new}|*a))}$

The result is a pathway leading from the start conformation to the goal conformation.

3.3.2 Feature Vector Representation

For the search, we project the conformations onto a lower-dimensional space that preserves much of the variance in the data. Inter-atomic interactions apply many constraints on protein motion, so the essential modes of motion can be captured using a small number of variables. The lower-dimensional projection was introduced by us in the past [38]. It is based on the distances and angles of the secondary structures with respect to one another and does not require the structures to be aligned. Given a conformation C, we first define a score for each manipulated secondary structure element i in C:

$$score(C^i) = \sum_{j \in K} \big(|\alpha_{ij} - \alpha'_{ij}| \times w_i + |d_{ij} - d'_{ij}| \times w'_i \big). \tag{3.1}$$

The summation is over the set K of manipulated secondary structures in C excluding i, α_{ij} is the angle and d_{ij} is the distance between secondary structure element i and secondary structure element j in C, α'_{ij} is the angle and d'_{ij} is the distance between the corresponding secondary structure elements in the goal structure, and w_i and w'_i are weight factors proportional to the size of secondary structure element i, such that the angle and distance components will be brought to the same order of magnitude. We used 1 for w_i and 5 for w'_i, which seem to give the best results. An angle between two secondary structure elements is defined as the angle between the two vectors representing them. A vector representing a helix is the least square straight line that passes through the helix atoms, and a vector representing a sheet is the normal to the surface best representing the sheet. The distance between two secondary structure elements is defined as the distance between their centers of masses. We then compute for a conformation C a feature vector:

$$v_C = \langle score(C^1), score(C^2), \ldots, score(C^k) \rangle \tag{3.2}$$

where the components of the vector are the scores of the K manipulated secondary structure elements of the conformation. The distance between two conformations, C_1 and C_2 is defined as the Euclidean distance between their feature vectors, i.e., $\|v_{C_1} - v_{C_2}\|^2$. By definition, when C_2 is the goal structure, the *score* of C_1 is the magnitude of its vector representation. The lower the score for a given conformation, the more similar it is to the goal structure. It should be noted that a secondary-structure based representation restricts this measurement to conformational changes where secondary structure elements do not drastically change. See [38] for more details.

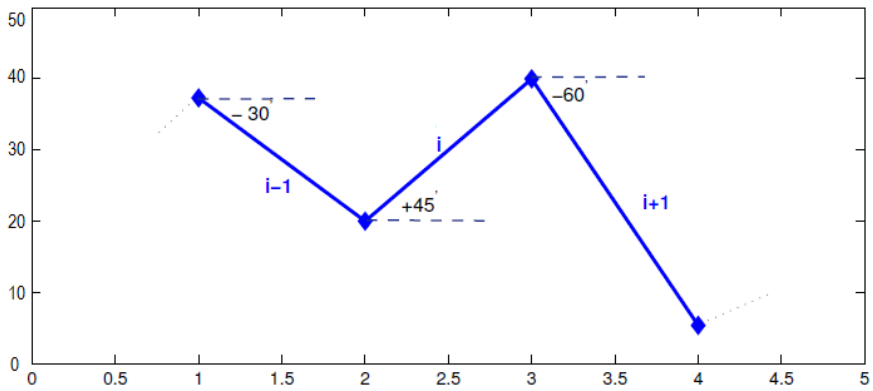


Figure 3.1: Each feature is located at a consecutive i value, and the magnitude of the feature is its y value. For segment i we measure (L_i, A_i), which is its length and angle with the x axis respectively. In this example $(L_i, A_i) = (8.03, 45)$

3.3.3 Similarity Measure

Recently, we developed a method to measure the distance between pairs of co-regulated genes based on the geometric characteristics of their gene expression data [80, 81]. In this work we apply this method to estimate the similarity of two feature vectors representing protein conformations. Each feature vector is represented as a polygon in a two-dimensional space as demonstrated in Figure 3.1. The features are represented consecutively on the x axis, and the y value represents the value of the feature in the feature vector. A line connects two consecutive points. We can represent each polygon by two attributes:

1. The length of the line i, denoted L_i, and

2. The angle of the line i with respect to x axis which represent as A_i.

This way, each conformation C is represented as follows:

$$C :< (L_1, A_1), (L_2, A_2), ..., (L_n, A_n) >$$

where n is the number of features. The similarity score compares two conformation based on the similarity of their representing polygons. Since the polygons are much

smaller than the number of atoms in a molecule, it is more efficient than RMSD, does not require the proteins to be aligned and is highly correlated with the RMSD (see Results below). Given two conformations C and D represented as polygons. In order to measure the similarity between line i of polygon C and line j of polygon D, the function $S(i, j)$ is defined as follows:

$$\begin{aligned} S(i,j) \;&=\; \omega_{length} \times (\; 1 - |L(i) - L(j)|\;) \\ &+ \omega_{angle} \times (\; 1/(\theta + |A(i) - A^(j)|\;) \end{aligned}$$

ω_{length} and ω_{angle} represent weight factors for the length and angle, respectively. The value of θ is determined by the slope of the lines based on different cases. The weight is determined by liner regression with respect to the input data. It can change based on data set and type of the problem. The main goal of the scoring function is to return an appropriate measure of similarity of the two lines.

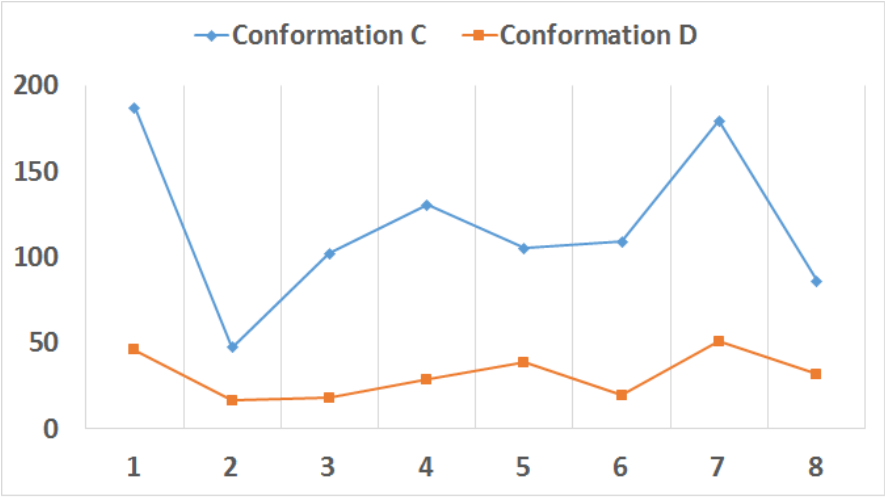
By applying this scoring function to all the pairs of conformations, we build a matrix M which each cell $M(i, j)$ represent the similarity score of conformation i and conformation j. The similarity between two conformations is measured using the Needleman-Wunsch DP algorithm [74]. The gap penalty is determined as the minimum similarity of two lines for the Needleman- Wunsch algorithm. An example is shown in Figure 3.2.

Algorithm 1 describes the similarity score. The input are two polygons representing conformations P and Q. The output of the algorithm is the score representing their similarity. The values of the parameters were determined experimentally. In this work we also did not use a gap penalty, but it can be used if needed.

It should be noticed that during the search we used the Euclidean distance between feature vectors to estimate their similarity since we used data produced in previous work (see above).

Algorithm 3 Score (P,Q)

for $i = 1...m$ **do**
  for $j = 1..n$ **do**
    if $P[i], Q[j]$ are on the same side **then**
      if $|P[i].angle - Q[i].angle| < 25$ **then**
        $\omega_{angle} = 90; \theta = 5$
      else
        $\omega_{angle} = 100; \theta = 30$
      end if
    else
      $\omega_{angle} = 10; \theta = 1$
    end if
    $score[i, j] = 30 * (1 - P[i].length - Q[j].length] + \omega_{angle} * (1/(\theta + angle(p_i) - angle(q_j)|)$
  end for
end for



C: 1 2 3 4 G 6 7 8
D: 1 2 3 G 5 6 7 8
Similarity Score: 0.87

Figure 3.2: In cases that the similarity of two lines is less than the gap score, the method select gap score for their similarity.

In the next section we describe the clustering methods applied to the feature vectors.

3.3.4 Clustering Methods

We applied K-means clustering. Our input was the similarity scores representing all the conformations generated for a given protein. It should be mentioned that the input to the clustering method is the DP-based similarity score. In other words, we are clustering one-dimensional data. This makes the clustering process extremely fast. Indeed some information is bound to get lost during the dimensionality reduction and similarity measurement process, but as we will see below, the results show that the clusters were still able to preserve most of the original properties of the structures. Additionally, the Pearson correlation coefficient between the RMSD and the similarity measure is very high, between -0.7 and -0.9 for all cases. It should be noted that the correlation is negative since RMSD is a distance measure and our method determines similarity.

Determining the number of clusters in K-means still is a big challenge in unsupervised learning. There exist many implementation of K-means algorithm for determining number of clusters and clustering. We used heuristic k-means algorithm [57]. Ckmeans.1d.dp is an R package tool for one dimension data which runs in $O(n^2k)$. In order to estimate the number of clusters, we used the Calinski-Harabasz criterion [13]. It creates several clusters for different values of k, and the number of clusters is estimating by the variance ratio criterion (VRC):

$$VRC = \frac{BGSS}{k-1} / \frac{WGSS}{n-k}, \tag{3.3}$$

where $BGSS$ is the between-cluster sum of squares, $WGSS$ is the within-cluster sum-of-squares, and n is the number of samples.

3.4 Results and Discussion

3.4.1 Cluster Properties

For AdK we produced 20 clusters using K-means to compare with our previous work [82]. For Calmodulin and GroEL we used the Calinski-Harabasz criterion mentioned above [13] to determine the ideal number of clusters. The number of clusters for each example is shown in Table 3.1 and is generally around 45-50.

Figure 3.3 shows the RMSD of the cluster centers with respect to the endpoints for two of our systems – AdK (4AKE→1AKE) and GroEL (1SX4→1SS8). Since K-means clustering assigns the cluster numbers arbitrarily, the clusters in the figure are re-numbered according to their RMSD from the respective endpoints. As seen, the clusters span the vast majority of the conformational space even when we measure the RMSD in the protein coordinate space, whereas the clustering was done in the one-dimensional similarity score space.

Figure 3.4 shows the cluster size distribution in three tested systems. Notice that due to the different number of conformations in each trajectory, the distribution of the number of conformations per cluster is different. The clusters vary in size, as expected, but the distances between the cluster centers are rather similar, and the cluster centers span the values of the similarity measurement described above rather uniformly. The variation in cluster sizes may be attributed to the sampling method, or to the fact that several intermediate conformations are more highly popular due to their low energy. This is the subject of on-going work.

3.4.2 Comparison with Known Intermediates

Experimental information about known intermediates is not always available due to lack of experimental knowledge about intermediate structures. However, AdK has several known mutants and intermediate structures [30]. We tested whether our clus-

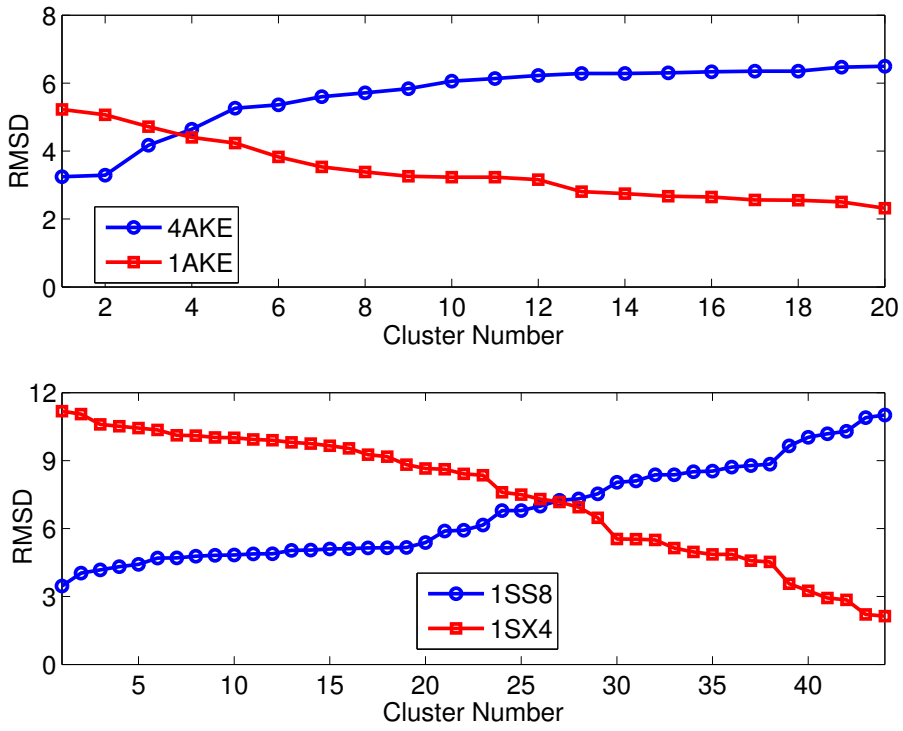


Figure 3.3: with respect to the two endpoints for AdK (top) and GroEL (bottom). The clusters are numbered according to their RMSD with the respective endpoint.

tering method can produce clusters similar to intermediate structures. Inspired by that study, we performed a similar test on our results. We focused on the following known intermediates: chains A, B and C of the hetero-trimer Adenylate Kinase from Aquifex Aeolicus (PDB accession code 2RH5), which are conformational change intermediates of the ligand free AdK [39], 1E4Y, which is an AdK mutant having 99% sequence identity with 4AKE and 1AKE and is a closed form of AdK binding with AP5A, and a mutant bound to an analog which shows domain closure over ATP (PDB code 1DVR). These intermediates were used successfully to validate conformational pathways for AdK [38, 1, 63]. We recorded for each path the closest conformation to any of our intermediates. The results are shown in Table **??**. For each intermediate, the table shows the average RMSD from the closest cluster, which is determined by center of clusters. Our results are in good agreement with previous work [30], as well as our earlier studies [54], which predicted 2RH5A-C to be close to the open conformation and 1E4Y to be closest to the closed conformation. Other structures are closer to intermediate conformations. In these cases we were able to find intermediate structures close to five intermediates (within about 3Å or less). The calculations were done with the UCSF Chimera software. Three of the intermediates and their closest cluster representatives are shown in Figure 3.5.

3.5 Conclusion

In this chapter we presented a DP-based method to measure the similarity between the lower-dimensional representation of protein conformations. We used the method to cluster trajectories of proteins which undergo large-scale conformational transitions.

The clustering is extremely fast, since it is done in a one-dimensional space generated by applying our similarity score on a lower-dimensional projection of the con-

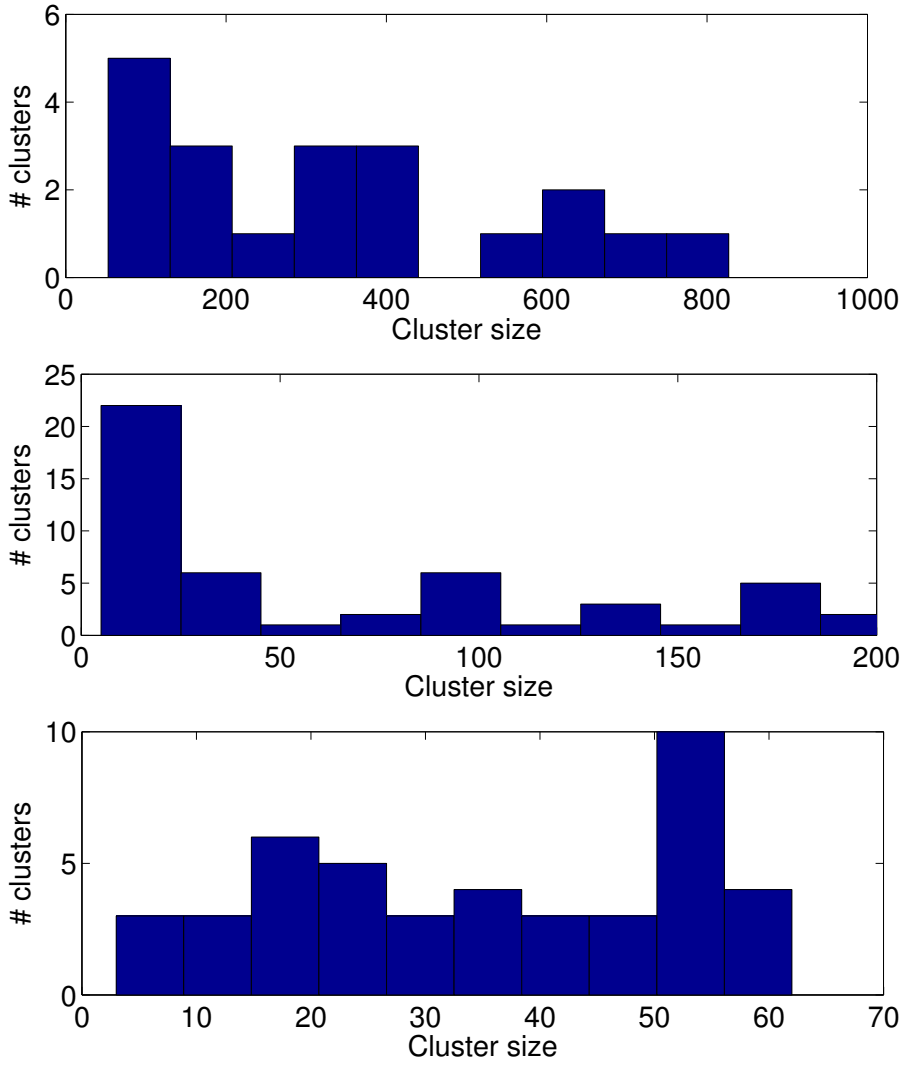
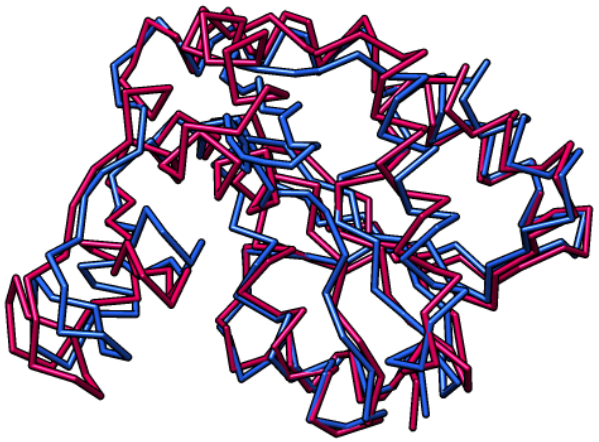
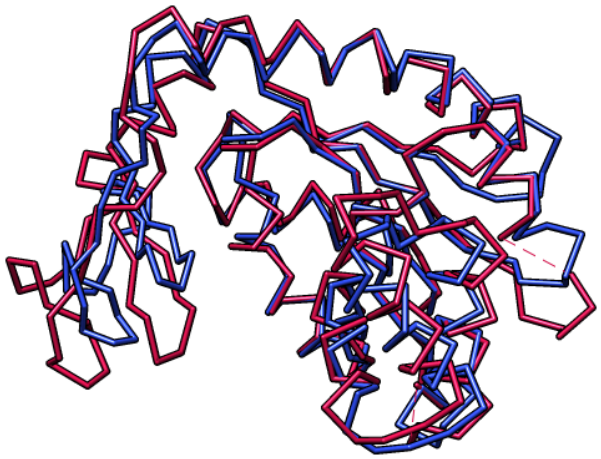


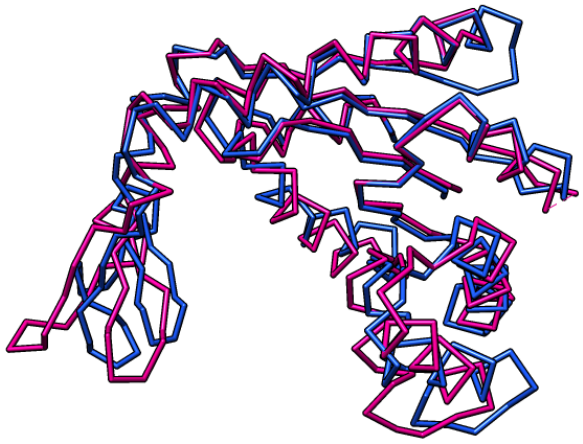
Figure 3.4: with for AdK (top), Calmodulin (middle) and GroEL (bottom).



(a)



(b)



(c)

Figure 3.5: on their closest cluster center (blue) (a) 1E4Y (b) 1DVR (c) 2RH5 (chain C).

formations. Yet, it is able to preserve a lot of the variance of the original data, since our similarity score correlates very well with the RMSD.

Current and future work includes optimizing the similarity measurement to improve clustering quality, comparison with other clustering methods such as hierarchical clustering which allows us to detect outliers, and testing more systems.

Chapter 4

CONCLUDING REMARKS

Understanding proteins function is still an open problem in computational biology. Proteins function is associated with conformational changes and binding partners. On the other hand, quantifying conformational changes is not an easy task. In this thesis, we investigate computational approaches for a better modeling of protein dynamics and function.

The main focus of this thesis was to propose a combination of machine learning methods to improve simulation of proteins dynamic along with utilizing statistical learning for predicting the binding site between two interacting proteins. In the second chapter, by extracting structural information from a protein, it turns into a likelihood matrix of binding for every two residues between two interacting proteins. This structural information converted into a penalty matrix for a graphical model to be learned from the protein sequence. By applying this prior to Direct Coupling Analysis method, a new set of co-evolving pairs stood out, therefore, the result improved significantly in comparison with current state-of-the-art. Furthermore, applying post-processing on the data has been proposed. In this way, every pair is converted into a node along with building patches round corresponding residues. The similarity between two nodes is calculated as an edge weight. Further graph diffusion can be applied to improve the prediction.

In chapter three, the goal was to identify the intermediate clusters between two conformations of a protein. This has been done in three steps. First, by utilizing

Monte Carlo tree search method, the pool of conformational changes between two proteins have been generated. Next, the pathways between the two conformations represented in smaller dimensions. This will help to reduce the complexity of the data. Finally, we proposed a novel clustering method for the coarse-grained model based on extracting geometry features and calculating scores. By applying this method to two conformations of a protein, up to the number of clusters conformations can be extracted and studied. These intermediate conformations can be used later as an information for the problem of binding site prediction.

Bibliography

[1] Al-Bluwi, I., Vaisset, M., Siméon, T., and Cortés, J. Modeling protein conformational transitions by a combination of coarse-grained normal mode analysis and robotics-inspired methods. *BMC structural biology 13*, Suppl 1 (2013), S2.

[2] Anderson, T. W., Anderson, T. W., Anderson, T. W., Anderson, T. W., and Mathématicien, E.-U. *An introduction to multivariate statistical analysis*, vol. 2. Wiley New York, 1958.

[3] Bahadur, R. P., Chakrabarti, P., Rodier, F., and Janin, J. Dissecting subunit interfaces in homodimeric proteins. *Proteins: Structure, Function, and Bioinformatics 53*, 3 (2003), 708–719.

[4] Balakrishnan, S., Kamisetty, H., Carbonell, J. G., Lee, S.-I., and Langmead, C. J. Learning generative models for protein fold families. *Proteins: Structure, Function, and Bioinformatics 79*, 4 (2011), 1061–1078.

[5] Ballester, P. J., and Richards, W. G. Ultrafast shape recognition to search compound databases for similar molecular shapes. *J. Comput. Chem. 28(10)* (2007), 1711–1723.

[6] Banerjee, O., Ghaoui, L. E., and dAspremont, A. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *Journal of Machine learning research 9*, Mar (2008), 485–516.

[7] Barton, J. P., De Leonardis, E., Coucke, A., and Cocco, S. Ace: adaptive cluster expansion for maximum entropy graphical model inference. *Bioinformatics 32*, 20 (2016), 3089–3097.

[8] Ben-Naim, E., and Lapedes, A. Genetic correlations in mutation processes. *Physical Review E 59*, 6 (1999), 7000.

[9] Benito, M., and Pena, D. A fast approach for dimensionality reduction with image data. *Pattern recognition* (2005), 2400–2408.

[10] Bordner, A. J., and Gorin, A. A. Protein docking using surface matching and supervised machine learning. *Proteins: Structure, Function, and Bioinformatics 68*, 2 (2007), 488–502.

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

[12] Bryngelson, J., Onuchic, J., Socci, N., and Wolynes, P. Funnels, pathways, and the energy landscape of protein folding: a synthesis. *Proteins 21* (1995), 167–195.

[13] Calinski, T., and Harabasz, J. A dendrite method for cluster analysis. *Comm. Statist. 3* (2009), 1–27.

[14] Case, D., Cheatham, T., Darden, T., Gohlke, H., Luo, R., Jr., K. M., Onufriev, A., Simmerling, C., Wang, B., and Woods, R. The amber biomolecular simulation programs. *J. Computat. Chem. 26* (2005), 1668–1688.

[15] Chang, H., Bacallado, S., Pande, V., and Carlsson, G. Persistent topology and metastable state in conformational dynamics. *PLoS ONE 8*, 4 (04 2013), e58699.

[16] Clackson, T., Ultsch, M. H., Wells, J. A., and de Vos, A. M. Structural and functional analysis of the 1: 1 growth hormone: receptor complex reveals the molecular basis for receptor affinity1. *Journal of molecular biology 277*, 5 (1998), 1111–1128.

[17] Cocco, S., and Monasson, R. Adaptive cluster expansion for inferring boltzmann machines with noisy data. *Physical review letters 106*, 9 (2011), 090601.

[18] Commons, W. Extended central dogma with enzymes.jpg, 2008.

[19] Commons, W. 223 structure of an amino acid-01.jpg, 2013.

[20] Commons, W. Rna-codons-aminoacids.svg, 2015.

[21] Commons, W. Figure 03 04 09.jpg, 2016.

[22] Cunningham, B. C., and Wells, J. A. High-resolution epitope mapping of hgh-receptor interactions by alanine-scanning mutagenesis. *Science 244*, 4908 (1989), 1081–1085.

[23] Dai, W., Wu, A., Ma, L., Li, Y.-X., Jiang, T., and Li, Y.-Y. A novel index of protein-protein interface propensity improves interface residue recognition. *BMC systems biology 10*, 4 (2016), 112.

[24] De, S., Krishnadev, O., Srinivasan, N., and Rekha, N. Interaction preferences across protein-protein interfaces of obligatory and non-obligatory components are different. *BMC Structural Biology 5*, 1 (2005), 15.

[25] Dunn, S. D., Wahl, L. M., and Gloor, G. B. Mutual information without the influence of phylogeny or entropy dramatically improves residue contact prediction. *Bioinformatics 24*, 3 (2007), 333–340.

[26] Ekeberg, M., Lövkvist, C., Lan, Y., Weigt, M., and Aurell, E. Improved contact prediction in proteins: using pseudolikelihoods to infer potts models. *Physical Review E 87*, 1 (2013), 012707.

[27] Esmaielbeiki, R., Krawczyk, K., Knapp, B., Nebel, J.-C., and Deane, C. M. Progress and challenges in predicting protein interfaces. *Briefings in bioinformatics 17*, 1 (2015), 117–131.

[28] Farahmand, S., Foroughmand-Araabi, M., Goliaei, S., and Razaghi-Moghadam, Z. Cytogta: A cytoscape plugin for identifying discriminative subnetwork markers using a game theoretic approach. *PloS one 12*, 10 (2017), e0185016.

[29] Farahmand, S., Goliaei, S., Ansari-Pour, N., and Razaghi-Moghadam, Z. Gta: a game theoretic approach to identifying cancer subnetwork markers. *Molecular BioSystems 12*, 3 (2016), 818–825.

[30] Feng, Y., Yang, L., Kloczkowski, A., and Jernigan, R. The energy profiles of atomic conformational transition intermediates of adenylate kinase. *Proteins 77(3)* (2009), 551–558.

[31] Finkelstein, A. V., and Ptitsyn, O. B. Why do globular proteins fit the limited set of foldin patterns? *Progress in biophysics and molecular biology 50*, 3 (1987), 171–190.

[32] Fox, N., Jagodzinski, F., Li, Y., and Streinu, I. Kinari-web: a server for protein rigidity analysis. *Nucleic acids research 39*, suppl_2 (2011), W177–W183.

[33] Frappier, V., Chartier, M., and Najmanovich, R. Encom server: exploring protein conformational space and the effect of mutations on protein function and stability. *Nucleic Acids Research 43* (2015), W395–W400.

[34] Friedman, J., Hastie, T., and Tibshirani, R. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics 9*, 3 (2008), 432–441.

[35] Gerstein, M., Lesk, A. M., and Chothia, C. Structural mechanisms for domain movements in proteins. *Biochemistry 33*, 22 (1994), 6739–6749.

[36] Gipson, B., Moll, M., and Kavraki, L. Sims: A hybrid method for rapid conformational analysis. *PLOS ONE 8*, 7 (2013), e68826.

[37] Grigoriev, A. On the number of protein–protein interactions in the yeast proteome. *Nucleic acids research 31*, 14 (2003), 4157–4161.

[38] Haspel, N., Moll, M., Baker, M., Chiu, W., and Kavraki, L. E. Tracing conformational changes in proteins. *BMC Structural Biology Suppl1* (2010), S1.

[39] Henzler-Wildman, K., Thai, V., Lei, M., Ott, M., Wolf-Watz, M., Fenn, T., Pozharski, E., Wilson, M., Petsko, G., Karplus, M., Hubner, C., and Kern, D. Intrinsic motions along an enzymatic reaction trajectory. *Nature 450*, 7171 (2007), 838–844.

[40] Hopf, T. A., Schärfe, C. P., Rodrigues, J. P., Green, A. G., Kohlbacher, O., Sander, C., Bonvin, A. M., and Marks, D. S. Sequence co-evolution gives 3d contacts and structures of protein complexes. *Elife 3* (2014), e03430.

[41] Janin, J., Rodier, F., Chakrabarti, P., and Bahadur, R. P. Macromolecular recognition in the protein data bank. *Acta Crystallographica Section D: Biological Crystallography 63*, 1 (2007), 1–8.

[42] Jones, D. T., Buchan, D. W., Cozzetto, D., and Pontil, M. Psicov: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. *Bioinformatics 28*, 2 (2011), 184–190.

[43] Jones, D. T., Singh, T., Kosciolek, T., and Tetchner, S. Metapsicov: combining coevolution methods for accurate prediction of contacts and long range hydrogen bonding in proteins. *Bioinformatics 31*, 7 (2014), 999–1006.

[44] Jones, S., and Thornton, J. M. Analysis of protein-protein interaction sites using surface patches1. *Journal of molecular biology 272*, 1 (1997), 121–132.

[45] Keskin, O., Gursoy, A., Ma, B., and Nussinov, R. Principles of protein- protein interactions: What are the preferred ways for proteins to interact? *Chemical reviews 108*, 4 (2008), 1225–1244.

[46] Keskin, O., Ma, B., and Nussinov, R. Hot regions in protein–protein interactions: the organization and contribution of structurally conserved hot spot residues. *Journal of molecular biology 345*, 5 (2005), 1281–1294.

[47] Kirkpatrick, S., Jr., C. D. G., and Vecchi, M. P. Optimization by simulated annealing. *Science 220* (1983), 671–680.

[48] Kozakov, D., Brenke, R., Comeau, S. R., and Vajda, S. Piper: an fft-based protein docking program with pairwise potentials. *Proteins: Structure, Function, and Bioinformatics 65*, 2 (2006), 392–406.

[49] Kozakov, D., Hall, D. R., Xia, B., Porter, K. A., Padhorny, D., Yueh, C., Beglov, D., and Vajda, S. The cluspro web server for protein–protein docking. *Nature protocols 12*, 2 (2017), 255.

[50] Krämer, N., Schäfer, J., and Boulesteix, A.-L. Regularized estimation of large-scale gene association networks using graphical gaussian models. *BMC bioinformatics 10*, 1 (2009), 384.

[51] Lee, B., and Richards, F. M. The interpretation of protein structures: estimation of static accessibility. *Journal of molecular biology 55*, 3 (1971), 379–IN4.

[52] Liwo, A., Czaplewski, C., Oldziej, S., and Scheraga, H. Computational techniques for efficient conformational sampling of proteins. *Curr. Opin. Struct. Biol. 18*, 2 (2008), 134–139.

[53] Luo, D., and Haspel, N. Multi-resolution rigidity-based sampling of protein conformational paths. In *Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics* (2013), ACM, p. 786.

[54] Luo, D., and Haspel, N. Multi-resolution rigidity-based sampling of protein conformational paths. In *in proc. of ACM-BCB (ACM International conference on Bioinformatics and Computational Biology)* (September 2013), pp. 787–793.

[55] Ma, B., Elkayam, T., Wolfson, H., and Nussinov, R. Protein–protein interactions: structurally conserved residues distinguish between binding sites and exposed protein surfaces. *Proceedings of the National Academy of Sciences 100*, 10 (2003), 5772–5777.

[56] Ma, J., Wang, S., Wang, Z., and Xu, J. Protein contact prediction by integrating joint evolutionary coupling analysis and supervised learning. *Bioinformatics 31*, 21 (2015), 3506–3513.

[57] McQueen, J. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* (1967), vol. 1, pp. 281–296.

[58] Meinshausen, N., Bühlmann, P., et al. High-dimensional graphs and variable selection with the lasso. *The annals of statistics 34*, 3 (2006), 1436–1462.

[59] Mintseris, J., and Weng, Z. Structure, function, and evolution of transient and obligate protein–protein interactions. *Proceedings of the National Academy of Sciences 102*, 31 (2005), 10930–10935.

[60] Miyashita, O., Wolynes, P. G., and Onucic, J. N. Simple energy landscape model for the kinetics of functional transitions in proteins. *Journal of Physical Chemistry B 109*, 5 (2005), 1959–1969.

[61] Miyazawa, S. Selective constraints on amino acids estimated by a mechanistic codon substitution model with multiple nucleotide changes. *PLoS One 6*, 3 (2011), e17244.

[62] Miyazawa, S. Prediction of structures and interactions from genome information. *arXiv preprint arXiv:1709.08021* (2017).

[63] Molloy, K., and Shehu, A. Elucidating the ensemble of functionally-relevant transitions in protein systems with a robotics-inspired method. *BMC Struct. Biol. 13*, Suppl 1 (2013), S8.

[64] Morcos, F., Pagnani, A., Lunt, B., Bertolino, A., Marks, D. S., Sander, C., Zecchina, R., Onuchic, J. N., Hwa, T., and Weigt, M. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences 108*, 49 (2011), E1293–E1301.

[65] Nimrod, G., Glaser, F., Steinberg, D., Ben-Tal, N., and Pupko, T. In silico identification of functional regions in proteins. *Bioinformatics 21*, suppl_1 (2005), i328–i337.

[66] Nooren, I. M., and Thornton, J. M. Structural characterisation and functional significance of transient protein–protein interactions. *Journal of molecular biology 325*, 5 (2003), 991–1018.

[67] Northey, T. C., Barešić, A., and Martin, A. C. Intpred: a structure-based predictor of protein–protein interaction sites. *Bioinformatics 34*, 2 (2017), 223–229.

[68] Ogmen, U., Keskin, O., Aytuna, A. S., Nussinov, R., and Gursoy, A. Prism: protein interactions by structural matching. *Nucleic acids research 33*, suppl_2 (2005), W331–W336.

[69] Ovchinnikov, S. *Protein structure determination using evolutionary information.* PhD thesis, 2017.

[70] Ovchinnikov, S., Kamisetty, H., and Baker, D. Robust and accurate prediction of residue–residue interactions across protein interfaces using evolutionary information. *Elife 3* (2014), e02030.

[71] Raveh, B., Enosh, A., Furman-Schueler, O., and Halperin, D. Rapid sampling of molecular motions with prior information constraints. *Plos Comp. Biol. 5(2)* (2009), e1000295.

[72] Razaghi-Moghadam, Z., Namipashaki, A., Farahmand, S., and Ansari-Pour, N. Systems genetics of nonsyndromic orofacial clefting provides insights into its complex aetiology. *European Journal of Human Genetics* (2018), 1.

[73] Reichmann, D., Cohen, M., Abramovich, R., Dym, O., Lim, D., Strynadka, N., and Schreiber, G. Binding hot spots in the tem1–blip interface in light of its modular architecture. *Journal of molecular biology 365*, 3 (2007), 663–679.

[74] S. B. Needleman, C. D. W. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Molecular Biology 48* (1970), 443–453.

[75] Shehu, A., and Olson, B. Guiding the search for native-like protein conformations with an ab-initio tree-based exploration. *The International Journal of Robotics Research 29*, 8 (2010), 1106–1127.

[76] Skwark, M. J., Abdel-Rehim, A., and Elofsson, A. Pconsc: combination of direct information methods and alignments improves contact prediction. *Bioinformatics 29*, 14 (2013), 1815–1816.

[77] Sułkowska, J. I., Morcos, F., Weigt, M., Hwa, T., and Onuchic, J. N. Genomics-aided structure prediction. *Proceedings of the National Academy of Sciences 109*, 26 (2012), 10340–10345.

[78] Tenenbaum, J., de Silva, V., and Langford, J. A global geometric framework for nonlinear dimensionality reduction. *Science* (2000), 2319–2323.

[79] Uhler, C. Gaussian graphical models: An algebraic and geometric perspective. *arXiv preprint arXiv:1707.04345* (2017).

[80] Vajdi, A., Banaee, H., and Haspel, N. A new dp algorithm for comparing gene expression data using geometric similarity. In *Workshop on Computational Regulatory Genomics and Metagenomics, in conjunction with IEEE-BIBM* (2015).

[81] Vajdi, A., and Haspel, N. Clustering protein conformations using a dynamic programming based similarity measurement. ISCA-BICOB, pp. 31–37.

[82] Vetro, R., Haspel, N., and Simovici, D. Characterizing intermediate conformations in protein conformational space. In *Proc. of the Ninth International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics* (Houston, TX, USA, July 2012).

[83] Wang, S., Sun, S., Li, Z., Zhang, R., and Xu, J. Accurate de novo prediction of protein contact map by ultra-deep learning model. *PLoS computational biology 13*, 1 (2017), e1005324.

[84] Weigt, M., White, R. A., Szurmant, H., Hoch, J. A., and Hwa, T. Identification of direct residue contacts in protein–protein interaction by message passing. *Proceedings of the National Academy of Sciences 106*, 1 (2009), 67–72.

[85] Weiss, D., and Levitt, M. Can morphing methods predict intermediate structures? *J. Mol. Biol. 385* (2009), 665–674.

[86] Yang, L., Song, G., and Jernigan, R. Protein elastic network models and the ranges of cooperativity. *Proceedings of the National Academy of Sciences 106*, 30 (2009), 12347–12352.

[87] Yap, E., Fawzi, N., and Head-Gordon, T. A coarse-grained alpha-carbon protein model with anisotropic hydrogen-bonding. *Proteins: Structure, function and bioinformatics 70* (2008), 626–638.

[88] Yuan, M., and Lin, Y. Model selection and estimation in the gaussian graphical model. *Biometrika 94*, 1 (2007), 19–35.