Discovering Protein Functional Regions and Protein-Protein Interaction using Co-occurring Aligned Pattern Clusters

by

Sanderz Fung

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirements for the degree of
Master of Applied Science
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2015

Declaration of Authorship

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

Papers related to this thesis

1. En-Shiun Annie Lee, Sanderz Fung, Ho-Yin Sze-To, and Andrew K.C. Wong. Discovering co-occurring patterns and their biological significance in protein families. BMC bioinformatics 15, no. Suppl 12:S2, 2014.

2. En-Shiun Annie Lee, Sanderz Fung, Ho-Yin Sze-To, and A. K. Wong. Confirming biological significance of co-occurrence clusters of aligned pattern clusters. In Proc. Int. Conf. Bioinformat. Biomed, pp. 422-427. 2013.

3. Sanderz Fung, En-Shiun Annie Lee, and Andrew KC Wong. Revealing Protein Structures by Co-Occurrence Clustering of Aligned Pattern Clusters. In Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics, p. 869. ACM, 2013.

4. Sanderz Fung, En-Shiun Annie Lee, and Andrew KC Wong. Comparing two Algorithms for Clustering Aligned Pattern Clusters. Mining Data Semantics in Information Networks (MDS2013) at ACM KDD 2013, Submission 7. ACM, 2013.

Substantial content of Paper 1, the accumulation of work from Papers 2-4, is incorporated into Chapter 3. For these papers, I was responsible in the formulation, the brainstorming, the development of APC co-occurrence and clustering of co-occurrence APCs, the testing of the co-occurrence score and the running of numerous experiments. The testing involved comparing different possible scores before picking the current calculation, Jaccard index, to be used as the co-occurrence score [1]. I was also responsible for the development and the comparisons of the several clustering algorithms used in co-occurrence clustering. Furthermore, I was involved and responsible for running all the experiments in all four papers, but not fully in the biological literature verification in Papers 1 and 2 since my co-authors are much more experienced in it.

5. Antonio Sze-To, Sanderz Fung, En-Shiun Annie Lee, Andrew K.C. Wong. Predicting Protein-Protein Interaction Using Co-Occurring Aligned Pattern Clusters. Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference on. Submission B472. IEEE, 2015 (Accepted)

A large part of Paper 5 is incorporated as Chapter 4. For this paper, I had 50% input into the formulation of all the new algorithms used in methods, but not the original pattern discovery work, the APCL algorithm and WEKA 3.7 [2]. Furthermore, I was solely responsible for the implementation of these new algorithms. Finally, I was responsible for the majority of the experiments, running more than 40 of them, not only to obtain our algorithm's results, but also to test for optimum parameters as well as to examine if applying a threshold to feature analysis was beneficial.

Abstract

Bioinformatics is a rapidly expanding field of research due to multiple recent advancements: 1) the advent of machine intelligence, 2) the increase of computing power, 3) our better understanding of the underlying biomolecular mechanisms, and 4) the drastic reduction of biosequencing cost and time. Since wet laboratory approaches to analysing the protein sequencing is still labour intensive and time consuming, more cost-effective computational approaches for analyzing protein sequences and their biochemical interactions are crucial. This is especially true when we encounter a large collection of protein sequences.

Aligned Pattern CLustering (APCL), an algorithm which combines machine intelligence methodologies such as pattern recognition, pattern discovery, pattern clustering and alignment, formulated by my research group and myself, is one such technique. APCL discovers, prunes, and clusters aligned statistically significant patterns to assemble a related, or specifically, a homologous group of patterns in the form of an Aligned Pattern Cluster (APC). The APC obtained is found to correspond to statistically and functionally significant association patterns, which corresponds as conserved regions, such as binding segments within and between protein sequences as well as between Protein Transcription Factor (TF) and DNA Transcription Factor Binding Sites (TFBS) in many of our empirical experiments. While several known algorithms also exist to find functionally conserved segments in biosequences, they are less flexible and require more parameters than what APCL requires. Hence, APCL is a powerful tool to analyze biosequences. Because of its effectiveness, the usefulness of APCL is further expanded from the assist of discovering and analyzing functional regions of protein sequences to the exploration of co-occurrence of patterns on the same sequences or on interacting patterns between sequences from the discovered APCs. Two new algorithms are introduced and reported in this thesis in the exploration of 1) APCs containing patterns residing within the same biosequences and 2) APCs containing patterns residing between interacting biosequences.

The first algorithm attempts to cluster APCs from APCs that share patterns on the same biosequences. It uses a co-occurrence score between APCs in a co-occurrence APC pair (two APCs containing co-occurrence patterns) to account for the proportion of biosequences of co-occurrence patterns they share against the total number of sequences containing them. Using this score as a similarity measure (or more precisely, as a co-occurring measure), we devise a Co-occurrence APC Clustering Algorithm to cluster APCs obtained from a collection of related biosequences into a Co-Occurrence Cluster of APCs abbreviated by cAPC. It is then analyzed and verified to see whether or

not there are essential biological functions associating with the APCs within that cluster. Cytochrome c and ubiquitin families were analyzed in depth, and it was validated that members in the same cAPC do cover the functional regions that have essential cooperative biological functions.

The second algorithm takes advantage of the effectiveness of APCL to create a protein-protein interaction (PPI) identification and prediction algorithm. PPI prediction is a hot research problem in bioinformatics and proteomic. A good number of algorithms exist. The state of the art algorithm is one which could achieve high success rate in prediction performance, but provides results that are difficult to interpret. The research in this thesis tries to overcome this hurdle. This second algorithm uses an APC-PPI score between two APCs to account for the proportion of patterns residing on two different protein sequences. This score measures how often patterns in both APCs co-occur in the sequence data of two known interacting proteins. The scores are then used to construct feature vectors to first train a learning model from the known PPI data and later used to predict the possible PPI between a protein pair. The algorithm performance was comparable to the state of the art algorithms, but provided results that are interpretable.

The results from both algorithms built upon the extension of APCL in finding co-occurring patterns via co-occurrence of APCs are proved to be effective and useful since its performance in finding APCs is fast and effective. The first algorithm discovered biological insights, supported by biological literature, which are typically unable to be discovered solely through the analysis of biosequences. The second algorithm succeeded in providing accurate and descriptive PPI predictions. Hence, these two algorithms are useful in the analysis and prediction of proteins. In addition, through continued research and development to the second algorithm, it will be a powerful tool for the drug industry, as it can help find new PPI, an important step in developing new drugs for different drug targets.

Acknowledgements

I would like to thank all the people that made this thesis possible. First I would like to thank my supervisor, Professor Andrew K. C. Wong, and my co-supervisor, Professor Daniel Stashuk, for letting me to study under them for the past two year. From Professor Wong, I learned on how to have a research-focused mindset and how to soak in his vast knowledge during the past two years. Moreover, I want to thank him on his many support and suggestions during my research process and his funding. I would also like to thank Professor Stashuk for the support during the Master's process.

Next, I thank my other two readers, Professor Kesen Ma and Professor Shi Cao for accepting and spending the time to be my thesis's reader. Furthermore, I would like to thank in advance for any possible future collaborations that I will have with the two readers.

I also thank my colleagues whom I have worked and collaborated with. Without their support and help this thesis would not have been possible. I would like to thank Dr. Annie En-Shiun Lee, for her tremendous leadership and support when I was her research assistant. Without her support, I would not have decided to partake in my Master's journey. Even during my Master's, she was a huge support for me. I would also like to thank Antonio Szeto for his excellent collaborative work, especially in Chapter 4. I also thank the following colleagues: Dr. Gary Li and Dr. D. Zhang.

Similarly, I thank my friends at graduate cell for their support and encouragement during my whole Master's journey.

Lastly, I would like to thank Professor Igor Ivkovic, whom I worked with as a teaching assistant. I would also like to thank graduate students and academic staff.

Contents

List of Figures

List of Tables

Abbreviations

APC	**A**ligned **P**attern **C**luster
APCL	**A**ligned **P**attern **CL**ustering
cAPC	**c**o-occurring **A**ligned **P**attern **C**luster
PDB	**P**rotein **D**atab**a**se
PPI	**P**rotein-**P**rotein **I**nteraction
PPI-DB	**P**rotein-**P**rotein **I**nteraction **D**atab**a**se
MSA	**M**ultiple **S**equence **A**lignment
SVM	**S**upport **V**ector **M**achine
MST	**M**aximum **S**panning **T**ree
TF	**T**ranscription **F**actor
TFBS	**T**ranscription **F**actor **B**inding **S**ites
CFTR	**C**ystic **F**ibrosis **T**ransmembrane Conductance **R**egulator

Symbols

Σ	alphabet of amino acids
S	a protein sequence
σ	an amino acid
$\mathbb{S}$	collection of protein sequences
p	pattern
$\mathbb{P}$	collection of patterns
C	an Aligned Pattern Cluster
$\mathbb{C}$	a set of Aligned Pattern Clusters
J	Jaccard index
K	a APC Cluster
B	protein pair
$\mathbb{B}$	collection of protein pairs
A	co-occurring Aligned Pattern Cluster pair
$\mathbb{A}$	collection of co-occurring Aligned Pattern Cluster pair
seg	Sequence segment
V	vertices
E	collection of edges
e	edge
W	adjacency matrix
I	identity matrix
L_{rw}	Laplacian matrix with random walk

Chapter 1

Introduction

Bioinformatics is a vastly growing research field which incorporates theory and methodologies in statistical pattern recognition, machine learning, and computational methods in the analysis of biological data, especially biosequence and micro-array data. Historically, biological experimental data has been analyzed through biochemical, physical and computational experiments, such as mass spectrometry which needs both biology wet laboratories and biology professionals. Such analyses are crucial for the final confirmation of the physical truth, yet usually involve numerous trial processes, which are very expensive and time consuming. However, since obtaining biosequence data is effective in the recent decades, there exists an attractive alternative. This alternative gets biological information directly from the sequence data coded for biomolecular structures and functions inherently in the biosequences, rather than going through numerous direct trials on wet laboratory experiments until the final confirmation. From such information, we could acquire more reliable results directly discovered and screened from sequence data via bioinformatic pattern discovery and analysis processes. Nevertheless, the use of bioinformatics to analyze biosequence data for revealing inherent conserved functionality in the past requires large amounts of data to justify the use of statistics. Even so, it has not rendered more specifically high quality predictive and interpretable results. Therefore, it needs more effective and intelligent computation to analyze the ample amount of such data and generate more accurate and specific results for further analysis and laboratory confirmation. Taking advantage of computational processing and computational capacity, we are able to make the analysis of biological data faster and cheaper using new and established algorithms developed in pattern recognition and machine intelligence [5].

One aspect of bioinformatics applications is the analysis of protein sequences, which are easy to obtain and are directly related to protein structures, functions and interactions.

With protein sequences, we aim to analyze them in order to understand more about their underlying functional and interaction mechanisms. For example, two characteristics that are worth analyzing are proteins' a) ability to fold and interact from their functional segments and b) involvement in interactions, i.e. in Protein-Protein Interactions (PPIs in brief). The first application enables us to have better understanding of the functional and mutational hot spots of the protein. The second application is important because by knowing how one protein interacts with others, we might be able to learn more about how proteins regulate disease control for better drug target discovery. These are the key questions when devising new drugs to combat diseases. Hence the analysis of protein sequences is important because any discovery based on bioinformatics techniques using only protein sequence data including those in the databases in the Web will be faster and cheaper when compared to biological wet laboratory experiments.

My research group has been developing new algorithms to further advance bioinformatics in protein sequence analysis. One such algorithm is known as Aligned Pattern Clustering (APCL). This algorithm is designed to analyse protein sequences and to reveal functionally conserved and mutated residue association and sites of protein local regions. Furthermore in [6], APCL has been found to be very effective in revealing and locating functionally important regions, and hence it is an effective algorithm for exploring and studying protein functionality. The objective of this thesis is to extend the capability of APCL to the discovery of not only the conserved functional regions but also the co-occurrence regions on the same sequences or between interacting sequences to reveal joint functionality such as site/residue binding and interacting sites between proteins respectively, and to find out if functional regions are likely candidates of binding or interacting sites between proteins. In view of these, I introduce and incorporate the idea referred to as co-occurrence of APCs to new algorithms which are able to cluster APCs if an appropriate co-occurrence measure between them can be used as a similarity measure to group them together. This idea aims to solve the following problem: can we create a quantifiable relationship between APCs that account for the majority of similar patterns co-occurring on the same sequences or interacting between sequences. Co-occurrence usually reflects inherent joint functionality such as containing binding or interaction sites/residues within and/or between protein sequences respectively. This thesis reports the formulation, implementation and experimentation of the proposed construct and renders interesting results for identifying and locating important regions of the same protein in the folded configuration and the interaction between two proteins in a PPI setting.

1.1 Thesis Contributions

The contributions of this thesis can be assessed in two aspects presented in Chapters 3 and 4. Both contributions are associated with the meticulous use of APCL in a new setting. Each aspect of this thesis's contribution can be described as follows:

1.1.1 Using Co-occurrence APCs to Reveal Interacting Regions within Protein Families

In this aspect, we extended the usefulness of APCs by incorporating a method to discover patterns and obtain APCs each of which may contain patterns residing on the same sequences. Because of the functional conservation characteristics of APCs, this relation of having patterns residing on the same sequence indicates that the co-occurrence regions are functionally important. Furthermore, there is joint functionalities between patterns of APCs in a co-occurrence APCs group. For example, their patterns may be associated with the cooperative interaction / binding function.

1.1.2 Using Co-occurrence APCs to Predict Protein-Protein Interactions

In this aspect of contribution, the usefulness of APCs is expanded by using them to identify and predict protein-protein interactions (PPI). Using only protein sequences and previous known PPI data, the devised algorithm takes advantage of patterns discovered in the form of APCs and the PPI knowledge acquired in the PPI database on the Web. It then obtains feature vectors from the training PPI datasets to build a classifier. For given pairs of protein sequences from the test set, it can use the classifier to output predictions with performance comparable to the state of the art algorithms, yet render interpretable sites which are not furnished by its counterparts.

1.2 Outline

The outline of the thesis is as follows. Chapter 2 will provide a biological background of the two major parts of this thesis, including the definition of biological terms. It will also provide the details of Aligned Pattern Clustering (APCL) and a related set of algorithms created by my research team, and the algorithms I contributed on top of the related works. Chapter 3 presents the first application of my APC co-occurrence algorithm on proteins of the same protein family, describing both the discovery and validation of the

results. Chapter 4 presents another application of my APC co-occurrence algorithm on proteins with known protein-protein interaction. We obtain a set of feature vectors to obtain a useful matrix in the learning phase and use it to predict the PPI in the testing case. Finally, Chapter 5 summarizes and concludes the thesis.

Chapter 2

Background

In this part, terminology that will be heavily used in the thesis, including protein, protein-protein interactions and Aligned Pattern Clusters (APCs) and other components, will be introduced. Furthermore, the various aspects of the thesis will address and discuss how the proposed methodology is used to analyze protein functions and interactions.

2.1 Biological Overview

2.1.1 Proteins

Proteins are biomolecules that have important and diverse biological functions, which include, but are not limited to, electron transfer, receptors and storage. Despite the various functions, shapes and sizes of proteins, they are all based on the same group of 20 **amino acids**. The proteins only differ in the composition and sequential association of the amino acids. (For example, one protein may have more of one type of amino acid than another protein) [7]. The sequential association of an amino acid segment govern not only the local biological function but also its folding, binding/interaction with other parts of the same protein or with sites of other proteins or other biosequences such as DNA. Research on proteins are important, not only because they are responsible for many important biological functions, but also the mutational impact of hot spots and regions. Considerable research has been conducted to study the loss or impairment of such functions due to mutation, even if the mutation is only a slight variation on the proteins. Such variation could lead to massive complications in the function of the protein and its interaction with other biosequences. For example, cystic fibrosis

is a disease that mutates a protein called cystic fibrosis transmembrane conductance regulator (CFTR). People with this disease have breathing and digestive problems [7].

2.1.2 Amino Acids

The twenty **amino acids** are twenty macro- molecules which contain the elements carbon, hydrogen, nitrogen, and oxygen. Two of the 20 amino acids additionally contain sulfur rendering a different composition of the elements. For simplicity, each amino acid can be represented by an English letter most of which are derived from their full names. They are: $\{A, R, N, D, C, E, Q, G, H, I, L, K, M, F, P, S, T, W, Y, V\}$ [7].

2.1.3 Protein Structure

To represent proteins, there are several structural levels of representation, namely, the primary structure, the secondary structure, the tertiary structure and the quaternary structure [7] (Figure 2.1), with each subsequent structure providing more detail of the protein than the previous one. In particular, the primary structure, or protein sequence, provides only the amino acid arrangement of the protein, and hence is a one-dimensional structure with complex amino acid associations. In comparison, all three subsequent structures are three-dimensional structures. Figure 2.2 shows both the primary and tertiary structure of the same protein, cytochrome c-553.

Different laboratory experiments are used to obtain the different protein structures. For protein sequences, sequencing techniques such as mass spectrometry and Edman sequencing [8] are used. For three-dimensional structures, techniques such as X-ray crystallography [9] are applied.

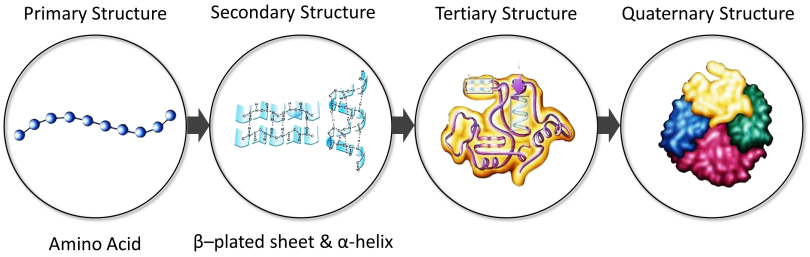

FIGURE 2.1: The four levels of protein structures

2.1.4 Protein Families

One way to categorize a protein is to use the sequence data and the acquired knowledge of **protein families**. Protein families are a collection of proteins that have evolved from the

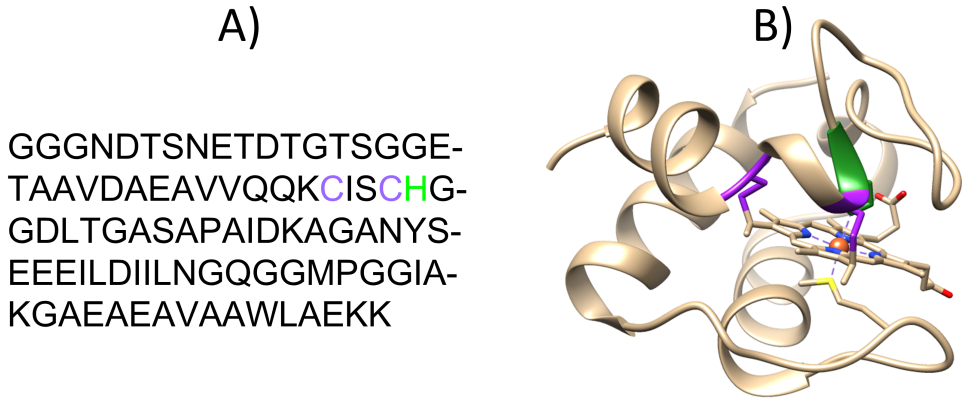

FIGURE 2.2: The primary and tertiary structure of a portion from the cytochrome c protein family, cytochrome c-553. The purple amino acids are the binding sites and the green amino acid is the metal binding site, as per [10, 11]. All three amino acids are part of the functional region of the protein. A) The primary structure of the protein [UniprotKB ID: P82599] [10, 11]. B) The tertiary structure of the protein [PDB ID: 1B7V] [12, 13]

same ancestor. Hence, proteins from the same protein family would have similar selection and arrangements of its amino acids, and have similar characteristics and functions [14]. Therefore, analyzing protein families is useful to reveal and understand common function across the proteins in a protein family.

2.1.5 Protein Databases

Thanks to the collaborative efforts in sequencing in the biologist community, there is currently a vast amount of known protein sequences on the Web. To facilitate possible computational analysis on these known protein sequences, several online databases exist to furnish public access to these protein sequences. For example, Uniprot [10] is a protein database that provides public access to 80 million protein sequences and their related annotations.

Another database is the Pfam [14] database, which provides access to thousands of protein families, including additional info such as a hidden Markov models of the protein family. Lastly, Protein Data Bank (PDB) [12] is a database that contains known protein three-dimensional structures.

The data used in the analysis in the subsequent chapters are collected from Uniprot and Pfam, and results are validated using PDB if necessary. This shows the importance of these databases in providing the necessary tools for bioinformatic analysis.

2.1.6 Protein Binding Sites and Protein-Protein Interactions

To facilitate the various functions of proteins, proteins need to have the ability to interact with other molecules (proteins, DNA, compound ions, etc.) or itself. In particular, when a protein interacts with another protein, it is referred to as **protein-protein interaction** (**PPI**) [15]. However, there are specific regions in the protein that are responsible for the protein's interaction with other molecules. These regions are referred as **protein binding sites**. In the example of cytochrome c-533, the protein needs to bind with a heme, an iron compound ion, for the protein to function as a part of the electron transport chain [16]. Figure 2.2 displays both the protein sequence and the three-dimensional structure of the cytochrome c-533, highlighting the only known protein binding site, or functional region, consisting of three amino acids that are crucial in interaction [10, 11], despite having 92 amino acids in this protein. Therefore, not only it is important to know the PPIs among interacting proteins, but also it is important to know the regions which are involved in those interactions.

2.2 Protein Sequence Analysis

Within the analysis of proteins, there are many analyses related to protein sequences. One group of protein sequence analysis is multiple sequence alignment. Multiple sequence alignment (MSA) takes multiple related protein sequences (for example, from the same protein family) and aligns the protein sequences, optimizing the similarity of amino acids on the aligned sites (or columns), inserting blanks within each protein sequence if necessary, such that the ideally aligning columns would have the same or almost the same amino acid for all protein sequences. The various MSA algorithms vary in the methods used to align the protein sequences. The method of progressive alignments, as opposed to exact alignments, does not align all sequences at the same time, but merges and aligns protein sequences based on a particular order. The advantage of progressive alignments is that it narrows down the possible alignment solution using a greedy algorithm. It thus greatly lowers the running time compared to that of exact alignment, though it might not provide the global optimal solution. Algorithms such as ClustralW [17], T-Coffee [18] and ProbCons [19] are examples of progress alignment MSA. In ClustralW, a tree is formed based on the protein sequence between all possible protein pairs. The tree level is then used as the order of which protein sequences are merged and aligned in progress alignments. However, one issue of ClustralW, as stated by Notredame [18], is that errors made in earlier merges will persist, and will be unable to be fixed in the subsequent merges. T-Coffee fixed the issue through the

use of a library that combines local and global alignment to ensure consistent alignment. Finally ProbCons introduced probabilistic alignment, adding a scoring function to measure alignment quality [19].

Another approach to protein sequence analysis is motif discovery. Instead of aligning protein sequences as a whole, motif discovery aims to find short amino acid patterns that occur repeatedly in the protein sequence. Similarly, the goal is to find regions in these protein sequences that are important, and hence, consistent across the protein sequences. Examples of motif discovery algorithms includes MEME [20], BLOCKS [21] and BLAST [22]. BLOCKS and BLAST compare input with known discovery for any similar motifs. BLOCKS compares the protein sequence to motifs in the BLOCK database, and BLAST compares the sequence with other sequences in Uniprot [10]. Our team's Aligned Pattern Clustering is an example of a motif discovery algorithm. However, the algorithm does not require any known motifs and has the ability to discover statistically associated segments as delimited motifs with slight variations, and therefore is a more flexible algorithm.

Outside of pure protein sequence analysis, one area of research is protein-protein interaction prediction (PPI Prediction). Using data of known related proteins, including protein family sequences, one of the major goals of bioinformatics is to predict PPIs of new protein candidates. Well known algorithms includes PIPE [23] / PIPE2 [24, 25] algorithm, and several that uses support vector machines (SVMs) [26–31]. While SVMs provide higher prediction performance, PIPE provides more interpretable results. However, this thesis aims to create an algorithm built on APCL that will have high prediction performance while providing interpretable results.

2.3 Aligned Pattern Clusters

Aligned Pattern Clustering (APCL) is composed of two algorithms that find potential conserved functional regions, including potential protein binding and interaction sites computationally. The two algorithms are 1) a pattern discovery algorithm that discovers statistically significant sequence patterns from a set of sequences of a protein family while pruning the redundant patterns [32]; and 2) an Aligned Pattern Clustering (APCL) algorithm that identifies homologous compact aligned groups of statistically significant patterns referred to as APCs. These APCs contain variations with adjustable low information entropy [6]. The details of the two algorithms are obtained from [3].

2.3.1 Input data

Based on the definition of protein structure, let $\Sigma = \{A, R, N, D, C, E, Q, G, H, I, L, K, M, F, P, S, T, W, Y, V\}$ be the protein alphabet containing twenty standard amino acids ($|\Sigma| = 20$). A protein sequence, or protein primary structure, $S = \sigma_1\sigma_2 \dots \sigma_{|S|-1}\sigma_{|S|}$ is an element of Σ^*, where each $\sigma_i \in \Sigma$ and S is of length $|S|$. Let the set of input protein sequences be defined as $\mathbb{S} = \{S_x | l = 1, ..., |\mathbb{S}|\} = \{S_1, S_2, \dots, S_{|\mathbb{S}|-1}, S_{|\mathbb{S}|}\}$. The input protein sequence set can either be from the same protein family (Chapter 3) or have known protein-protein interactions between each sequence pair (Chapter 4). Sequence patterns are then discovered from this input dataset in the next step.

2.3.2 Pattern discovery [3]

"Sequence patterns with statistically significant amino acid associations are first discovered [32]. They are defined as an ordered sequence of interdependent symbols $p = \sigma^1\sigma^2...\sigma^{|n|}$ from the alphabet Σ. The pattern p has length n, and the i^{th} symbol that appears in the sequence is σ^i. The list of patterns resulting from the pattern discovery algorithm is represented by $\mathbb{P} = \{p^i | i = 1, ..., |\mathbb{P}|\} = \{p^1, p^2, \dots, p^{|\mathbb{P}|-1}, p^{|\mathbb{P}|}\}$, and are pruned of redundant patterns." [3]

2.3.3 Aligned Pattern Clustering [3]

"An APC describes a set of sequence patterns that have been grouped due to their aligned similarities (as defined in [6]). Aligned patterns add gaps and wildcards to maximize the vertical similarity of amino acids between the patterns. Let an APC be defined as

$$C^l = \texttt{ALIGN} \begin{pmatrix} p^1 \\ p^2 \\ \vdots \\ p^m \end{pmatrix}, \tag{2.1}$$

$$= \begin{pmatrix} \sigma_1^1 & \sigma_2^1 & \dots & \sigma_n^1 \\ \sigma_1^2 & \sigma_2^2 & \dots & \sigma_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ \sigma_1^m & \sigma_2^m & \dots & \sigma_n^m \end{pmatrix}_{m \times n}, \tag{2.2}$$

where $sigma_j^i \in \Sigma \cup \{-\} \cup \{*\}$ is a symbol in pattern p^i with a new column index j. Note that $-$ denotes a gap character and $*$ denotes a wildcard character. Each of the $|\mathbb{P}^l| = m$ patterns in the rows of C^l is of length $|C^l| = n$." [3]

Let a set of APCs be defined as $\mathbb{C} = \{C^l | l = 1, ..., |\mathbb{C}|\} = \{C^1, C^2, \ldots, C^{|\mathbb{C}|-1}, C^{|\mathbb{C}|}\}$.

As seen in [33] APCs are successful in finding binding sites, or functionally important domains for protein-protein interactions. With this success, I want to extend APCL to expand its usefulness using a concept called co-occurrence of APC.

2.4 Motivations and Objectives

The motivation of my thesis is to extend the usefulness of APCs to provide a statistical and functionally conserved base to reveal more of the structural and functional relations between protein regions. Specifically, using an idea called co-occurrence, which quantifies the question: what proportion of patterns taken from each of the two APCs reside on the same sequence? This co-occurrence was addressed in two different algorithms covered in Chapter 3 and 4 (Figure 2.3) respectively. 1) The first algorithm considers patterns co-occurring on the same protein sequence in a collection of protein sequences. 2) The second algorithm considers that the co-occurrence APCs appear in protein-protein interactions in a collection of potential protein-protein interacting pairs. The current algorithms related to the second algorithm task either lack high prediction performance or lack interpretable results.

There are two objectives to this thesis: 1) to provide a quantitative basis for revealing site association relationship between the APCs, and 2) to conjecture possible biological association between protein regions based on supporting evidences on their pattern co-occurrence relationship.

‘

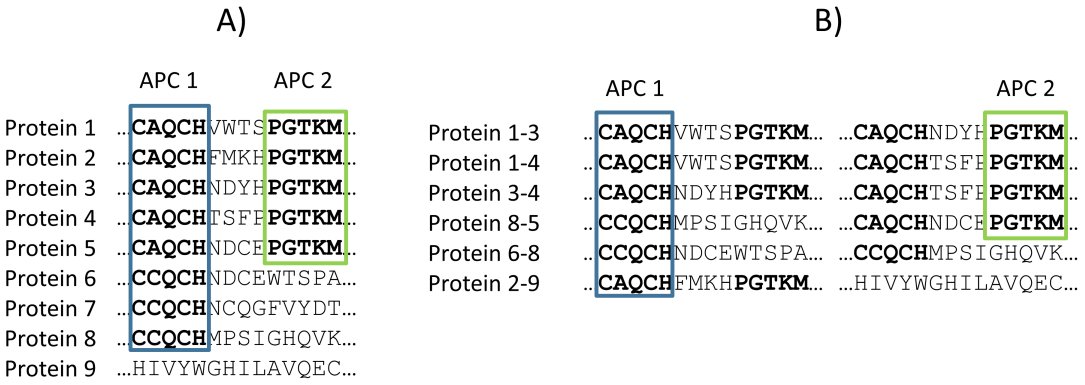

FIGURE 2.3: The two different co-occurrence in the APC data space to be discussed in Chapters 3 and 4. A) Calculates how often the patterns in the two APCs appear together in the protein sequences in the data space. Here they appear 5 times. B) Calculates how often the patterns in the two APCs data space appear together in the APC pairs of the protein-protein interactions. Here they appear 4 times. Of notice is that the protein-protein interaction between protein 2 and 9 is not counted despite both APCs appearing in protein 2. This is because both proteins must be represented by at least one of the two APCs.

Chapter 3

Applying Co-occurrence APCs to Discover Interacting Regions within a Protein in Protein Families

To expand the usefulness of APCs, we sought to extend the APCL algorithm by analyzing the potential relationships between the APCs. For this chapter, we want to retain one of the advantages of the APCL algorithm: obtaining results only from protein sequences. We hence use co-occurrence patterns in protein sequences and devise a co-occurrence score between APC pairs. Several protein families were analyzed for this algorithm, with two protein families, cytochrome c and ubiquitin, to provide further insights. The APCs were discovered, and the co-occurrence score between them were calculated. Then the APCs were clustered based on the co-occurrence relationship using a co-occurrence score as a similarity measure. The results shows that members of the same clusters are close together and that they cover more binding sites with the same biological function.

3.1 Introduction

Identifying functional regions on proteins is essential for understanding biological mechanisms and for designing new drugs. Due to the accessibility to protein sequences on the web, it is more effective to look for conserved segments from a set of functionally similar protein sequences than to perform laborious and time-consuming experiments and

computationally intensive modeling. The study of conserved functional regions relies on the assumption that amino acids in functional regions are integral and thus undergo fewer mutations throughout evolution than less functionally important amino acids [34]. Therefore, the functional regions of protein structures can be obtained from analyzing protein sequences that have similar biological functions.

Multiple sequence alignment (MSA) [17, 18] is a traditional computational method which is capable of aligning homologous protein sequences that are highly similar. However, it is unable to discover functional regions in more divergent protein sequences. Consequently, MSA is a global alignment method suitable for studying closely related proteins but not proteins that have only region-wise, partially functional similarities [35]. It has also been shown that finding the global optimal alignment is an NP-complete problem [36]. Coupling analysis [37–39] is a method based on MSA that examines the substitution correlation between two aligned columns within the MSA. This study hypothesizes that if two residues form a contact within a protein, then an amino acid substitution at one position is expected to be compensated for by a substitution in another position over the evolutionary time-scale. This observation suggests that co-occurring residues on the same protein can provide insight into the protein's structure. However, due to the dependence on MSA and the complexity of the method, determining the underlying statistical model requires a large number of homologous non-redundant protein sequences. Evolutionary tracing [34] is another method based on clustering alignments. The consensus within and across each group is identified to allow the study of divergent residues that are globally or functionally preserved in a protein family. Once again, evolutionary tracing is based on full sequence similarity requiring mutagenesis information for clustering [40]. Hence, it is not effective for revealing local functionality. Both coupling analysis and evolutionary tracing are based on examining pairwise amino acid correlations from MSA which focuses on two identified sites and does not take into account other sequence information.

In comparison to traditional methods, our algorithm finds and analyzes higher order sequence patterns in conserved regions, improving the capacity to reveal cross pattern association encompassing local and distant functionality. In our previous work, we introduced Aligned Pattern Clusters (APCs) [6] to represent functional regions as an alternative to position weight matrices [41]. Aligned Pattern Clusters are sequence patterns with variations and conservation without assuming independence between residues [6] at sites. Its strength lies in the retention of statistical significance along the amino acids on a sequence and also the tracking of distribution of their occurrences across the sequences. With this novel representation, we are now able to exploit the APC occurrences on a collection of sequences and study the co-occurrence between their patterns on the same protein sequence.

We hypothesize that co-occurring patterns reflect the joint functionality that are needed for co-operative biological functions such as chemical bonds or binding sites. Thus, we address the following two research questions: 1) given a set of homologous protein sequences, how can frequently co-occurring patterns be efficiently discovered? 2) How can the biological reasoning and significance of these co-occurrences be confirmed? To test these hypotheses, we used our co-occurrence clustering algorithm to find highly co-occurring patterns among a cluster of APCs and then studied their biological functions. First, we collect homologous protein sequences from the protein databases Pfam [14] and UniProt [42] as input. Next, we design an efficient algorithm based on our previous work [6, 32] to find and represent the frequently co-occurring patterns. Finally, we verify our results by comparing the three-dimensional distance between the co-occurring patterns against the average distance between the regions spanned by the patterns. To confirm the biological functions of the co-occurrences, we search the related scientific literature to support the conceived role of these co-occurring patterns.

In view of the above mentioned computational results and biological observations accomplished in this chapter, the contributions of this study mirror the answers to the research questions in two ways. First, we have established an algorithm that discovers co-occurring functional regions that are statistically reliable, measurable, and efficient. To our knowledge, this study is the first to identify the co-occurrence of patterns rather than residues. Compared to existing algorithms used to study correlations in amino acid residues, the novelty of our algorithm is that it does not require a large number of homologous protein sequences to identify pattern co-occurrences. Secondly, we have verified these co-occurrences by using the co-occurring patterns' three-dimensional closeness and by searching biological literature for support, enriching our understanding of the underlying mechanism. Novel co-occurrence relationships will provide new insight for the biological community for use in their study on protein functionalities.

3.2 Methods

The methodology proposed in this paper combines the pattern discovery algorithm and the APC algorithm presented in Chapter 2 and a new algorithm together to obtain the Co-occurrence Cluster of Aligned Pattern Clusters (Co-occurrence Cluster) (Fig. 3.1). In the third algorithm, Co-occurrence Clusters are obtained by clustering the APCs discovered using spectral clustering [43] with a co-occurrence score adopted as a measure of distance.

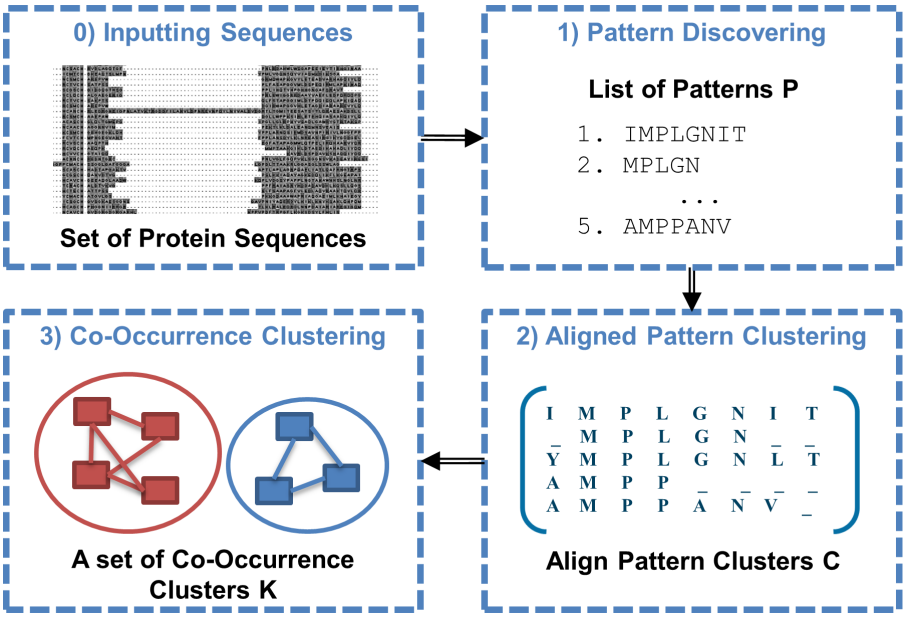

FIGURE 3.1: The overall process of our methodology is represented by a pipeline consisting of three algorithms. 0) the input is a set of sequences from the same protein family; 1-2) are the pattern discovery algorithm and the APC algorithm discussed in Chapter 2 and 3) the new Co-Occurrence Cluster algorithm, which cluster APCs by their co-occurrence scores.

3.2.1 Clustering APCs to Co-occurrence Clusters

Co-existence of patterns in different locations of the same protein may indicate that they are statistically significant and functionally related and important for the protein family. In Co-occurrence Clusters, we first apply a spectral clustering algorithm to cluster APCs using a co-occurrence score between APCs as the similarity measure. Let the graph $G = (V, E)$ be a relationship graph with APCs as vertices. Let each vertex v be an APC, and let each weighted edge e be the co-occurrence for two APCs; the edge weight is the co-occurrence score to be defined later between the two APCs. The spectral clustering algorithm is used to obtain Co-occurrence Clusters based on the co-occurrences of patterns between the APCs.

3.2.1.1 Co-occurrence score [3]

To tell how many patterns out of the total number of the discovered patterns co-occur in two APCs, we need a co-occurrence score which will be used as the similarity measure for clustering co-occurring APCs. "The co-occurrence score quantifies how often patterns in two APCs appear together on the same sequence. The Jaccard index is adopted [1]:

$$J = \frac{|C^1_{seq} \cap C^2_{seq}|}{|C^1_{seq} \cup C^2_{seq}|},$$

where C^1_{seq} = sequences that contain patterns from APC C^1 and C^2_{seq} = sequences that contain patterns from APC C^2." [3]

The APC pairs are ranked by the co-occurrence score and listed in descending order. When two or more APC pairs have the same score, the sequence count of the union of the two APCs ($|C^1_{seq} \cup C^2_{seq}|$) is used as a secondary ranking criteria, i.e., the pair with a higher union size indicates that it covers more sequences and, hence, should be ranked higher.

3.2.1.2 Spectral clustering

For spectral clustering [43], an adjacency matrix W is first created and filled with the co-occurrence scores between the APCs. Let W be an n by n matrix (n is the vertex count in G), where $W(i, j)$ is the adjacency weight between vertex v_i and v_j, i.e., the co-occurrence score between vertex v_i and v_j. The following matrix was first constructed:

$$d_i = \sum_j W(i, j).$$

$$D = diag(d_1, ..., d_n),$$

where D is an n by n diagonal matrix.

Next, using the adjacency matrix, a Laplacian matrix L is created, and L's eigenvectors are calculated. Using a random walk, construct the Laplacian matrix

$$L_{rw} = I - D^{-1}W$$

where I is an n by n identity matrix. Find both the eigenvalues and their corresponding eigenvectors for L_{rw} and sort the eigenvectors by the ascending order of their eigenvalues.

Finally, the eigenvectors are then used as positions for the APC vertices v, with the weighted edges e being the Euclidean distance between v in the vertex space of G and its neighbours. K-means clustering is applied to G, minimizing the Euclidean distance of the eigenvectors between the vertices. Let k be the final cluster count, defined as the count before the largest difference between consecutive eigenvalues [43]. We use the first k eigenvectors for clustering. We construct a n by k matrix eig, where each column corresponds to one of the first k eigenvector in vertical matrix form. Each row in eig corresponds to an APC vector, with each vector having k values. The vector represents the APC as a point in the k-dimensional space. As eig has a row count of n, there are n APC vectors. Apply the k-means clustering algorithm on the n APC vectors, minimizing the distances between the points within the clusters (Algorithm 1).

Algorithm 1 Spectral clustering

Input: A set of APCs $\mathbb{C}$, adjacency matrix W, and the final number of clusters required by the final k-means clustering algorithm
Output: APC Clusters $K_1...K_k$
for $i = 1$ to $|\mathbb{C}|$ **do**
  $d_i = \sum_j w_(i,j)$
end for
$D = diag(d_1, ..., d_n)$
Let I be a $|\mathbb{C}|x|\mathbb{C}|$ identity matrix
$L_{rw} = I - D^{-1}W$
Calculate the eigenvectors and their corresponding eigenvalues of L_{rw}
Sort the eigenvectors by their increasing eigenvalues
Take the first k eigenvectors and construct the matrix eig, where each column is an eigenvector in its vertical matrix form
Let each row of eig represent an APC,
and let each column of eig a dimension
Construct a k-dimension graph G_k based on eig
Apply k-means clustering on G_k, minimizing the Euclidean distance between the points within the clusters.
return $\{K_1...K_k\}$

3.2.1.3 Comparison of clustering algorithms

Two other clustering algorithms were implemented to compare with spectral clustering: that is, the k-means clustering and the hierarchical clustering.

A special variation of the k-means clustering algorithm called k-medoids [44] is used in this paper. APCs are used to represent the centroids since calculating a centroid with only co-occurrence scores between APCs is difficult. The medoids are initialized to be the first APC for each connected component due to the small number of APCs considered. During the clustering process, the medoids are updated by finding the APC that maximizes the co-occurrence score between itself and all the other APCs in the same cluster. Finally, to ensure that clustering provides the best possible results, five clustering indicators are computed to determine the optimal final number of clusters, i.e., optimum k, to be adopted for the k-medoids (Algorithm 2).

The hierarchical clustering algorithm uses a maximum spanning tree (MST) with minimal cut. First, an MST is built using Prim's algorithm (Algorithm 3). Next, the minimal weighted edge of the MST is cut to separate the vertices, which are APCs, into two co-occurrence clusters. The second step is repeated until an optimal solution is achieved, determined through different clustering indicators, such as the Dunn index [45] (Algorithm 4).

Algorithm 2 k-medoids clustering

Input: A set of APCs $\mathbb{C}$, and the co-occurrence scores between all pairs of APCs J, final number of clusters the k-means clustering is k
Output: APC Clusters $K_1...K_k$
Initialize centroids $M_1...M_k$, where each M_1 represent the center of APC Cluster K_i
Find number of components
Select first APC from each component as the centroid
for $i = |components| + 1$ to k **do**
  Identify the APC that forms the lowest co-occurrence score with known centroids
  Assign this APC as a new centroid
end for
repeat
  for all $APC\,C \in \mathbb{C}$ **do**
    Assign C to closest centroid M_j such that C and M_j are from the same component
  end for
  for all $clusterK_i \in \{K_1...K_n\}$ **do**
    Update centroid M_i by selecting APC that maximizes co-occurrence within all APCs in K_i
  end for
until convergence
return $\{K_1...K_k\}$

Algorithm 3 Maximum Spanning Tree (based on Prim's Algorithm)

Input: A set of APCs $\mathbb{C}$ as vertices V, and the co-occurrence scores between all pairs of APCs as edges E
Output: A set of $|\mathbb{C}| = |V|$ edges for the maximum spanning tree edges $E_M = \{e_1, e_2, ..., e_{|V|-|components|}\}$
repeat
  Add any edges e that connects to v to edge list,
  making sure that the other vertices connected to that edge is not already seen
  if edge list is not empty, **then**
    get the maximum edge from list
    Let the vertex that is connected by the maximum edge but currently not in MST be the new v
    Add the maximum edge to MST edge list
    Add the vertex to the seen vertices list
  else if edge list is empty **then**
    Find a random vertex that is not seen yet in the seen vertices list to be the new vertex
  end if
until all vertices are seen

Algorithm 4 Hierarchical Clustering

Input: A set of APCs $\mathbb{C}$ as vertices V, and the maximum spanning tree edges E_M
Output: APC Clusters $K_1 ... K_k$, where each K is an APC Cluster that contains a set of APCs C
repeat
  Compute clustering indicators for all MST edges
  Sort the clustering indicators
  Select the edge with the optimal cluster indicator
  Cut the edge if the current cluster indicator is better than the previous cluster indicator
until optimal clustering is reached: current cluster indicator is worse than the previous cluster indicator

Runtime Comparison The runtimes to find the optimal solutions for the three clustering algorithms are as follows: $O(n^4)$ for hierarchical clustering, $O(n^3)$ for spectral clustering, and $O(n^3)$ for k-medoids clustering. During the edge-cutting phase for hierarchical clustering the algorithm must evaluate all possible MST edges, a maximum of n edges, with each edge taking $O(n^2)$. Since there are a maximum of n MST edges to cut, the total running time is $O(n^4)$. K-medoids clustering takes $O(n^2)$ only if the cluster count is given. However, the algorithm is run n times to compare and obtain the optimal cluster count for the optimal clustering solution. Hence, the optimal solution has a runtime of $O(n^3)$. In comparison, spectral clustering takes $O(n^3)$ even with cluster count given, as the matrix multiplication that occurs when calculating the Laplacian matrix takes $O(n^3)$. However, the matrix is calculated only once, the optimal cluster count is obtained through the eigenvalues, and the algorithm uses the same that for the k-medoids algorithm to find the optimal cluster. Hence, the total runtime for spectral clustering is the same as k-medoids clustering, $O(n^3)$. Because of the quicker runtime, spectral and k-means clustering are preferred over hierarchical clustering.

Nature of the Dataset Moreover, the spectral clustering algorithm is selected over the k-means clustering algorithm used in [46] because of the nature of the data. Pfam [14] sequences are built from multiple sequence alignments with the help of a hidden Markov model; thus, the sequences have been pre-processed for correctness. UniProt [42] sequences are collected from a string query search of the database, so the quality of the sequences depends on the search terms. Therefore, the sequence quality of UniProt is less consistent, making it unsuitable for clustering using the global centroid of k-means since the low-quality sequences are heavily affected by outliers [47]. Closest neighbour characteristic in the spectral clustering algorithm is beneficial in handling noisy data. Therefore, this algorithm was selected to cluster co-occurring APCs.

3.2.2 Verification by three-dimensional structure

To evaluate the importance of the APC regions discovered, we use the three-dimensional distance between the protein segments corresponding to the APCs within the Co-occurrence Cluster. The rationale for using the three-dimensional distance is that if the APCs are close together in three-dimensional space then they will likely interact with one another. It thus provides biophysical support that these functional regions are of biological importance to the proteins in the protein family tested.

After applying co-occurrence clustering, we manually select the cluster that contains the lowest average eigenvector distance as the highly connected Co-occurrence Cluster. We relate these results to the corresponding three-dimensional protein structure from the Protein Data Bank (PDB) [12] using Chimera [48], highlighting the regions where the APCs, or parts of the APCs, appear. The distances between the APCs are calculated as follows: the positions of each carbon alpha in each APC region is averaged, creating an average centroid for each APC region. The Euclidean distance is then calculated amongst all centroids. Finally, the APC distance is compared to the average pairwise distance, which is the average Euclidean distance of all possible carbon alpha pairs in the structure.

Using only the highly connected Co-occurrence Clusters and finding their biological importance, we validate 1) that the co-occurrence score ranks important APC pairs over the less important ones, 2) that co-occurrence clustering is able to separate the less important APCs out and 3) that our algorithm can provide reasonably good results in a timely manner, i.e. by not having to search through all APCs discovered.

3.3 Datasets

The first dataset selected for our experiment contains two different protein families from UniProt, which are examined in subsequent detailed case studies. The first set is of ubiquitin protein sequences, downloaded on August 9th, 2012, with the following filters to obtain high quality sequences: having the name ubiquitin with a mnemonic starting with UB; and not containing the words ribosomal, modifier, factor, protein, conjugate, activating, or enzyme to remove other similar names. The second is of cytochrome c protein sequences, downloaded on December 20, 2013, similarly with the filters: having the name cytochrome c with the mnemonic CY*; not ending in "ase" to prevent the inclusion of oxidase or reductase; and not containing biogenesis or probability to remove other similar names. Each sequence from UniProt has an organism name, which is next

searched in UniProt Taxonomy to acquire the condensed taxonomy lineage. Finally, the top kingdom name is extracted as the class label.

Next, our method was run on the two UniProt datasets. For the 70 ubiquitin input sequences, the pattern-discovery step was executed with a minimal length of 5, a maximum length of 15, a minimum occurrence of 20, and a delta of 0.9 (for control of delta closed pattern pruning). The maximum length restricted long (or high order) patterns from being discovered in the highly conserved ubiquitin sequences. Aligned pattern clustering was then executed with the following settings: Global Alignment with Hamming Distance and heuristics conditions with a minimum consecutive column match of 3, a minimum conserved column of 1, and no relative position overlapping. For the 319 cytochrome c input sequences, the pattern discovery step was executed with a minimal length of 5, a minimum occurrence of 40, and a delta of 0.9. The increase in the minimum occurrence was due to the increase in the number of input sequences. Aligned pattern clustering was then executed with the same settings as above. Lastly, the co-occurrence score was computed, and the three clustering algorithms were run. For both datasets, spectral clustering and k-medoids resulted in producing the same Co-occurrence Cluster.

The second dataset contains nine different protein families downloaded from Pfam Release 3.2 for a large-scale study of the three-dimensional structure of proteins. Pfam was used due to its well curated and pre-processed data. The proteins are lipocalin [Pfam: PF00061]; bacterial rhodopsins [Pfam: PF01036]; bacterial antenna complex [Pfam: PF01036]; cytochrome c oxidase subunit I [Pfam: PF00115]; photosynthetic reaction centre protein family [Pfam: PF00124]; leptin [Pfam: PF02024]; G-alpha subunit [Pfam: PF00503]; protein kinase domain [Pfam: PF00069]; and tyrosine kinase [Pfam: PF07714]. The pattern-discovery and the aligned pattern clustering steps were executed with the same settings as above, except the minimum occurrence, which was adjusted based on the number of sequences and their sequence similarity as listed in Pfam. After clustering, we picked the Co-occurrence Cluster with the lowest average eigenvector distance to be evaluated for the three-dimensional distance.

3.4 Experimental results and discussions

3.4.1 Proteins verified by three-dimensional structure

We applied our method to nine protein families, confirming that our algorithm is effective at finding important regions on any protein family. Table 3.1 displays the Co-occurrence Cluster of closely related APCs in the PDB structure of the related protein family. We found that these APCs are close in Euclidean distance in the three-dimensional space.

TABLE 3.1: Results from the nine protein families. Displays the Co-occurrence Cluster with the lowest average eigenvector distance, and are used to verify the algorithm's effectiveness with a PDB structure. The shorter distance in the comparison is bolded. * means that one or more APCs were not found.

Protein Name	Pfam ID	Co-occurrence Cluster Count	Size of the Best Cluster	PDB ID of the Best Cluster	Average APC Distance of the Best Cluster	Average Pairwise Distance
Lipocalin	PF00061	6	4	2CZT	**16.77 Å**	19.26 Å
Bacterial rhodopsins	PF01036	2	2	1JGJ	**16.52 Å**	22.51 Å
Bacterial antenna complex	PF00556	4	5	1IJD	**0 Å**	19.92 Å
Cytochrome c oxidase subunit I	PF00115	2	25	3OM3	**26.78 Å***	30.00 Å
Photosynthetic reaction centre protein family	PF00124	2	7	1PSS	**27.87 Å**	30.19 Å
Leptin	PF02024	2	14	1AX8	**15.73 Å**	18.37 Å
G-alpha subunit	PF00503	3	8	4G5O	**15.78 Å**	27.45 Å
Protein kinase domain	PF00069	2	2	3OZ6	**15.32 Å**	27.51 Å
Tyrosine kinase	PF07714	2	8	4HW7	**14.43 Å**	24.99 Å

Of interest are the results from the bacterial antenna complex family [Pfam: PF00556], where there is an average APC distance of 0 Å. The reason is that, despite having 5 APCs in the maximum co-occurrence cluster, all APCs overlap with one another, creating one long continuous region highlighted in blue (Figure 3.2). Furthermore, the highlighted region covers positions 9 to 31 of the structure, and has only 46 amino acids, i.e., the maximum co-occurrence cluster continuously covers close to half of the whole structure. The figure also indicates that [Pfam: PF00556] might be highly conserved, exhibiting only minor variations in its primary structure across different proteins in the family, especially in the regions covered by the maximum co-occurrence cluster. Another result where the maximum co-occurrence cluster covers most of the amino acids in the PDB structure is Leptin [Pfam: PF02024, PDB: 1AX8], where only 14 amino acids are not covered by the APCs in the maximum co-occurrence cluster.

All the APCs within the cluster in all the experiments in Table 3.1 were closer in distance than the average pairwise distance, indicating a relation between co-occurring APCs and their distance in three-dimensional structures. We were able to observe some characteristics of the protein family, i.e., the conservation of its primary structure. Hence, our algorithm is proven to discover important conserved regions for protein families.

3.4.2 Biological validation

In this section, we investigated the biological significance of Co-occurrence Clusters. Our experimental results revealed the Co-occurrence Clusters of ubiquitin and cytochrome c. Here we would like to study why co-occurring APCs are close to one another in spatial distance despite being far from each other in the primary sequence. Our hypothesis is that they need to form chemical bonding or co-operate in essential biological functions.

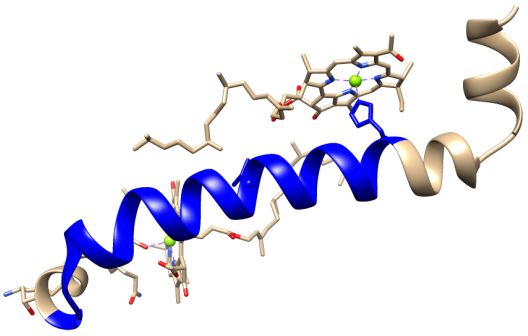

FIGURE 3.2: Three-dimensional structure of bacterial antenna complex [PDB: 1IJD]. The set of all the patterns in the APCs in the Co-occurrence Clusterinspected are all contained within one continuous highlighted blue region, indicating how the APCs overlap with one another.

3.4.2.1 Ubiquitin case study

Ubiquitin (UBI) is a small (8.5kDa) protein that consists of a single polypeptide chain of 76 amino acids [49]. It plays an important role in ubiquitination, which is a post translational protein modification process where either a single ubiquitin or multiple chains of ubiquitin are attached to a substrate protein. To form a chain, a ubiquitin connects to another ubiquitin by binding the diglycine in its C-terminal tail to one of the seven lysine amino acids of its linking partner.

Ubiquitination is widely used in regulating cellular signaling [50]. It does so by allowing the attached ubiquitin in substrate proteins to be bound through proteins with ubiquitin-binding domains (UBD) [50]. Either attaching a ubiquitin to a target protein or connecting it to another ubiquitin regulated by the sequential activity of ubiquitin-activating (E1), ubiquitin-conjugating (E2) and ubiquitin-ligating (E3) enzymes [50].

When the seven lysine amino acids were mapped to our APCs, they were all covered (Table 3.2). According to the results of our co-occurrence clustering algorithm in Figure 3.3, the optimum number of cluster of the six APCs is two. The first cluster includes APC 1, 2, 3, 4 and 5; the second cluster includes APC 6 only. Their biological significance is discussed in Figure 3.4.

The APCs in the first cluster to co-occur for two reasons. First, each APC covers at least one Lysine (K). The diglycine in the C-terminal tail, i.e., Gly(G)75 and Gly(G)76 (green shade), is also covered in APC 3. As discussed, Lysine (K) and the diglycine in the C-terminal tail are both important for the formation of multiple ubiquitin chains. Both APC 5 and APC 3 also cover important residues for facilitating the interaction of ubiquitin with E1 enzymes [52]. Mutagenesis experiments demonstrated that the mutation of Arg(R)42 or Arg(R)72 (red blocks) destabilizes the binding between Ubiquitin

TABLE 3.2: Key residues covered by APC and their roles in the Co-occurrence Cluster 1 of ubiquitin

APC	Residue(s)	Role(s)	Literature
1	K6, K11	Lys(K)6 and Lys(K)11 are used for forming ubiquitin chain(s) in ubiquitination.	[49]
	L8	Leu(8) facilitates the interaction between ubiquitin and E1 enzymes.	[50, 51]
2	K11, K27	Lys(K)11 and Lys(K)27 are used for forming ubiquitin chain(s) in ubiquitination.	[49]
	L8	Leu(8) facilitates the interaction between ubiquitin and E1 enzymes.	[50, 51]
3	K63	Lys(K)63 is used for forming ubiquitin chain(s) in ubiquitination.	
	H68, V70	His(H)68 and Val(V)70 facilitate the binding between ubiquitin and ubiquitin-binding proteins.	[50, 51]
	R72	Arg(R)72 facilitates the interaction between ubiquitin and E1 enzymes.	[52]
	G75,G76	Gly(G)75 and Gly(G)76 are used for forming ubiquitin chain(s) in ubiquitination.	[49]
4	R42	Arg(R)42 facilitates the interaction between ubiquitin and E1 enzymes.	[52]
	I44	Ile(I) 44 is the binding site between ubiquitin and the ubiquitin-binding proteins.	[50, 51]
	K48	Lys(K)48 is used for forming ubiquitin chain(s) in ubiquitination. It also facilitates the binding between ubiquitin and ubiquitin-binding proteins.	[49–51]
5	K27,K29,K33	Lys(K)27, Lys(K)29 and Lys(K)33 are used for forming ubiquitin chain(s) in ubiquitination.	[49]
	R42	Arg(R)42 facilitates the interaction between ubiquitin and E1 enzymes.	[52]

and E1 enzymes significantly, thus in turn, destroying the biological functions of ubiquitin [52]. Second, all APCs except APC 5 cover the ubiquitin-binding residues. These residues are important for the tight binding of ubiquitin with ubiquitin-binding proteins [50]. Therefore, the APCs in the Co-occurrence Cluster 1 are due to both ubiquitination and ubiquitin-binding.

There is only one APC, APC 6, in the second cluster (Figure 3.3) which has no co-occurrence with other APCs. We also observed a certain degree of overlapping between APC 6 and APC 5. We propose two reasons to explain why APC 6 is not merged with APC 5 but exists alone in another cluster. First, the conserved amino acid in residue 24 of APC 6 and APC 5 is Asp(D)24 and Glu(E)24 (yellow shade), respectively. We found that ubiquitin of Viridiplantae (plant kingdom) has mostly Glu(E)24, whereas ubiquitin of Metazoa (animal kingdom) has mostly Asp(D)24 in our dataset, this site is also well-known for differentiating human (containing Glu(E)24) ubiquitin from yeast (containing Asp(D)24) ubiquitin [53]. Hence, APC 6 and APC 5 are not merged in this study, because they cover patterns with different amino acids in different species.

Second, APC 6 does not include ubiquitination-related Arg(R)42 and covers the alpha helix 1, from residues 23 to 34, more precisely than APC 5. Previous literature has

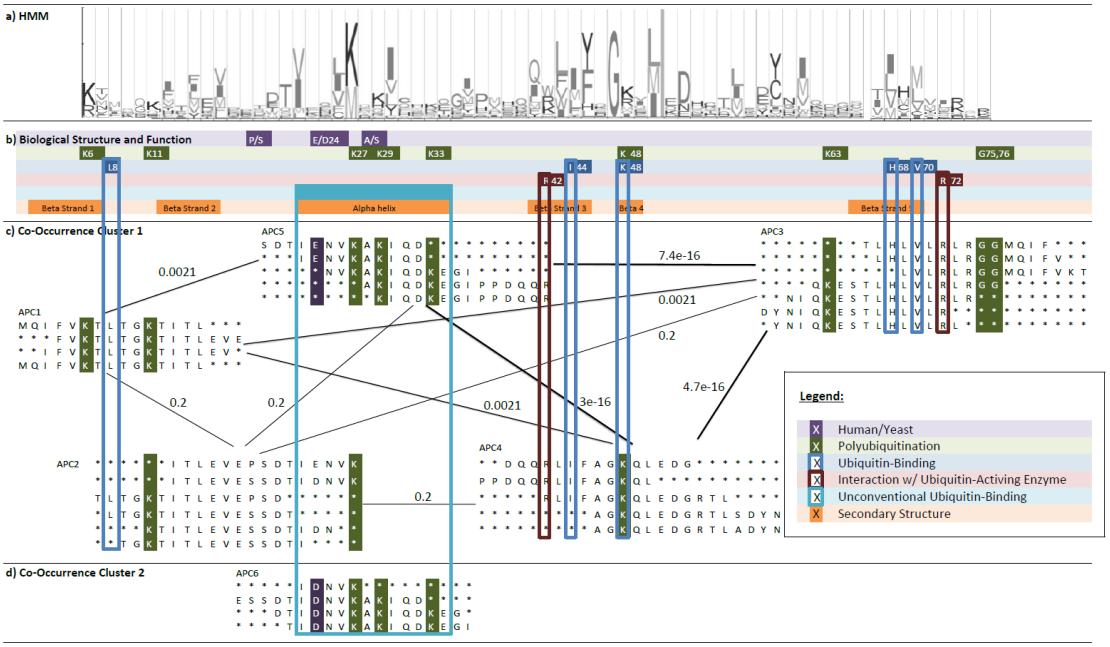

FIGURE 3.3: Co-occurrence clusters of ubiquitin. General Features: a) the top of the diagram is part of the HMM sequence profile of ubiquitin; b) the color shading blocks with legends immediately below mark the important amino acids and segments forming the important structure and function of the protein; c-d) the APCs discovered are represented by arrays of aligned amino acids; the color shaded columns correspond to the significant residues marked as in b); if the co-occurrences of patterns between APCs are frequent, the co-occurrence APCs are linked by an edge with weight representing co-occurrence score; treating APCs as vertices. A co-occurrence APC cluster is represented by a weighted graph linking co-occurring APCs; the important functional regions of the molecules as listed in Table 2 are highlighted in colored blocks specified by the legend. Specific Features: Note that APC 5 and APC 6 are not linked by co-occurrence since they belong to different taxonomical group and with different amino acids, Asp(D)24 and Glu(E)24, in the same column.

discovered that alpha helix 1 is an unconventional recognition site of ubiquitin-binding proteins [51]. Experiments in the same study revealed that, even if Ile(I)44 and His(H)68 were mutated, a high affinity binding between protein CKS1 and ubiquitin would still be identified, thereby proving that ubiquitin is unconventionally bound by CKS1 [51]. It should be noted that the conventional and unconventional ubiquitin-binding is not mutually exclusive [51]. Hence, APC 5 in the first cluster and APC 6 in the second cluster are not merged. Where APC 5 represents the scenario that either only conventional ubiquitin-binding occurs or conventional and unconventional ubiquitin-binding co-occur, APC 6 represents the scenario that only unconventional ubiquitin-binding occurs. Our experimental results from ubiquitin and literature search give us very strong support for the biological significance of the discovered Co-occurrence Cluster.

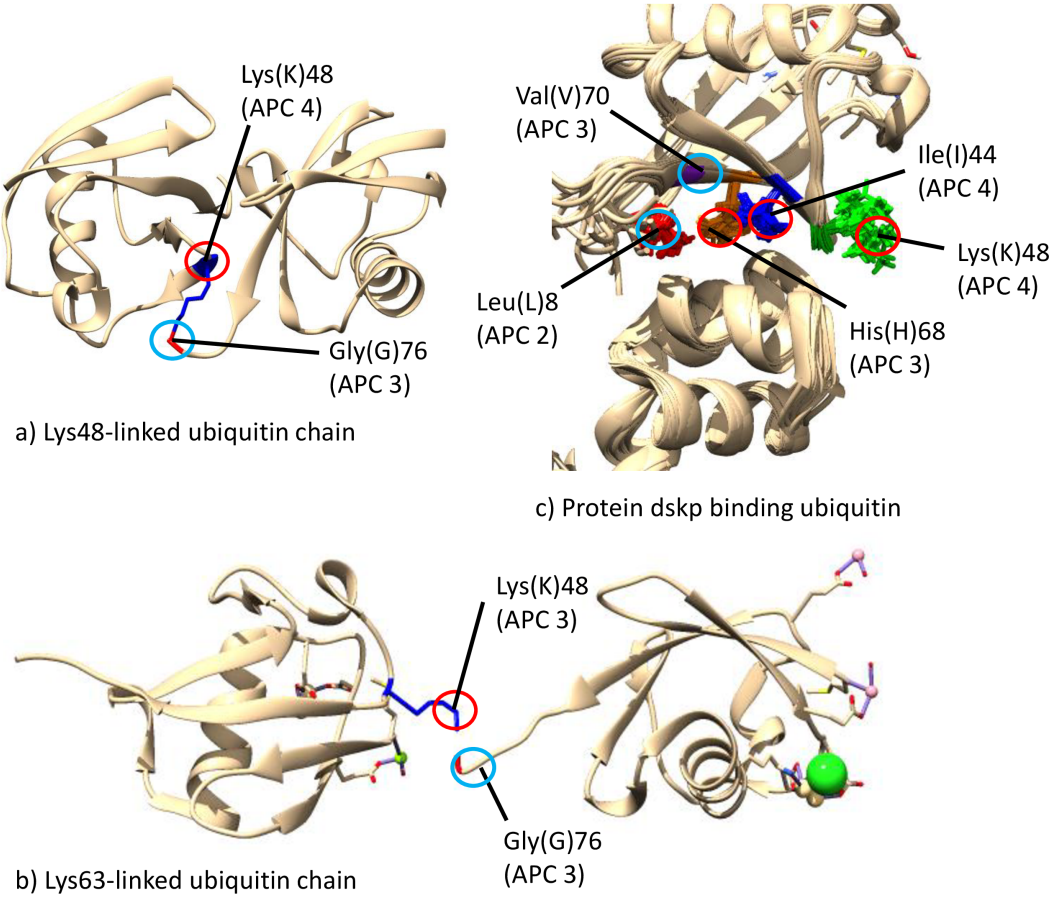

FIGURE 3.4: Three-dimensional structures of ubiquitin [PDB: 1AAR, 2JF5, 1WR1]. The binding residues discussed in Table 3.2 and their functions are displayed. a) is the ubiquitin chain linked by the Lys(K)48 in APC 4 to the diglycine, b) is the ubiquitin chain linked by the Lys(K)63 in APC 4 to the diglycine, c) is the binding between dskp binding ubiquitin and ubiquitin by Leu(L)8 of APC 2, Val(V)70 of APC 3, Ile44(I) and Lys(K)48 of APC 4, and His(H)68 of APC 3.

3.4.2.2 Cytochrome c case study

Cytochrome c (cyt-c) is a small (12.4kDa), heme-containing protein that consists of approximately 104 amino acids [54]. It is an essential component of the electron transport chain in the mitochondria. The heme group of cyt-c accepts electrons from the complexes III (cytochrome b-c_1 complex or cyt-bc_1) and transfers electrons to the complexes IV (cytochrome c oxidase or cyt-c_1) [54].

According to the results of our co-occurrence clustering algorithm (Figure 3.5), the optimum number of clusters of the 8 APCs is 2. The first cluster includes APCs 1 to 3; the second cluster includes APCs 4 to 8. Their biological significance is discussed as below.

For the first cluster, we found that all the APCs covered residues that contributed significantly to the binding of cyc-1 on cyc-bc_1. This is crucial for electron transfer.

TABLE 3.3: Key residues covered by APCs and their roles in Co-occurrence Cluster 1 of cytochrome c

APC	Residue(s)	Role(s)	Literature
1	Lys25, Lys27	The binding sites of cytochrome c cytochrome BC_1 complex	[55–57]
2	Lys27	The binding sites of cytochrome c cytochrome BC_1 complex	[55–57]
3	Lys5, Lys7, Lys8	The binding sites of cytochrome c cytochrome BC_1 complex	[55–57]

Experiments have established the importance of Lys(K)8, Lys(K)27 and, to a lesser extent, Lys(K)5, Lys(K)7, Lys(K)25 [55–57]. They are covered in the APCs in the first cluster (Table 3.3). Therefore, these APCs co-occur to facilitate the binding of cyc-1 on cyc-bc_1.

For the second cluster, we found that all the APCs covered residues that were mostly responsible for the stable axial ligand between cyc-t and the heme group (Figure 3.6), which is the component that takes part in the redox reactions for the electron transfer between cyt-c and other complexes. APC 4 covered Cys(C)14 [58, 59], Cys(C)17 [58, 59] and His(H)18 [60, 61]. His(H)18 [60, 61] forms an axial ligand with the heme from the proximal front. Cys(C)14 [58, 59] and Cys(C)17 [58, 59] enhance and maintain the axial ligand between His18 and the heme. APC 5 covered Tyr(Y)67 [54, 62], Pro(P)71 [63], and Pro(P)76 [64], Met(M)80 [61] and Phe(F)82 [65]. Met(M)80 [61] forms an axial ligand with the heme from the distal side. Tyr(Y)67 [54, 62], Pro(P)71 [63], Pro(P)76 [64] stabilize and coordinate the axial ligand between Met(M)80 and the heme. Phe(F)82 [65] stabilizes the native heme environment. APC 6 covered Gly(G)41 [66], which holds the axial ligand between Met(M)80 and the heme. APC 7 covered Asn(N)52 [67, 68], which maintains a hydrogen bond with the heme to stabilize the environment.

Although APC 8 did not cover any residues that are directly related to the axial ligands between cyt-c and the heme group, it covered residues that maintain the cyt-c structure. Among the 38 intra-molecular hydrophobic interactions reported in [67], APC 8 covered 17 (44.7%). It also covered Leu(L)94 [69] and Tyr(Y)97 [69], where one of them is required to provide a hydrophobic environment in order for cyt-c to function. Evidently, the APCs in the co-occurrence cluster 2 form and maintain stable axial ligands with the heme and also provide an appropriate structure and environment for cyt-c to function.

3.5 Summary

In this chapter, two research questions that were first posed in the introduction are addressed. We answer the first research question on discovering co-occurrences by creating

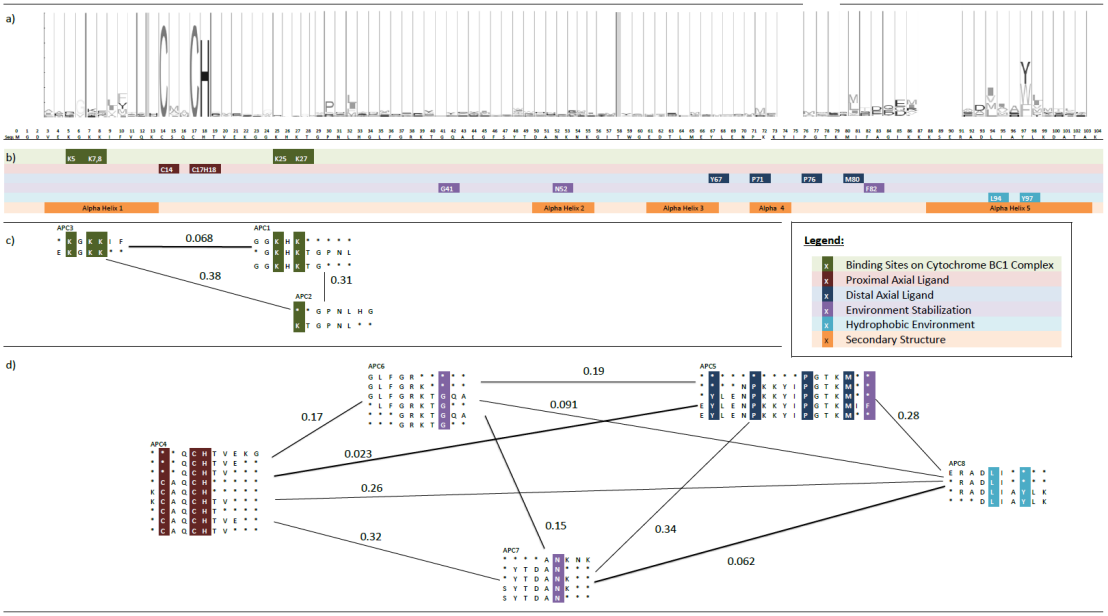

FIGURE 3.5: Co-occurrence clusters of cytochrome c. General Features is same as stated in Figure 3.3 c-d) Important functional regions as listed in Table 3 and 4, are highlighted here in color blocks as specified by the legend; Specific Features: Amino acids in Co-occurrence Cluster 1 facilitate the binding of cyc-1 on cyc-bc1 as listed in Table 3 and most of the amino acids in Co-Occurrence Cluster 2 are responsible for the stable axial ligand between cyc-t and the heme group.

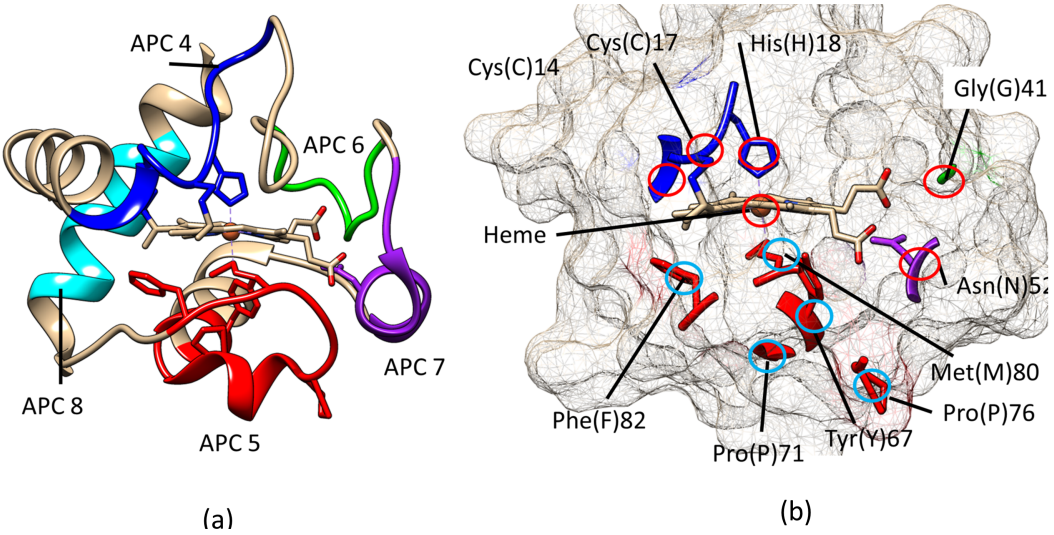

FIGURE 3.6: Three-dimensional structure of cytochrome c [PDB: 1HRC]. a) The APCs in Co-occurrence Cluster2 as listed in Table 3.4. b) The amino acids from APCs in Co-occurrence Cluster2 mostly interact with the heme to stabilize the axial ligand, as confirmed by biological literature listed in Table 3.4.

a novel algorithm that clusters APCs with a good proportion of co-occurring patterns into an effective, statistical, and measurable Co-occurrence Clusters. We respond to the second research question on the biological significance of these Co-occurrence Clusters by their three-dimensional closeness and also by their biological functionality and structural integrity. We confirm that the Co-occurrence Cluster with the lowest average

TABLE 3.4: Key residues covered by APCs and their roles in Co-occurrence Cluster 2 of cytochrome c

APC	Residue(s)	Role(s)	Literature
4	Cys(C)14	Cys(C) 14 enhances axial ligand strength between His18 and the heme.	[58, 59]
	Cys(C)17	Cys(C) 17 enhances axial ligand strength between His18 and the heme.	[58, 59]
	His(H)18	His(H)18 forms an axial ligand with the heme from the proximal front.	[60, 61]
5	Tyr(Y)67	Tyr(Y)67, its hydroxyl group, forms a H-bond with side chains of Met80 for structural stabilization.	[54, 62]
	Pro(P)71	Pro(P)71 helps coordinate the axial ligand between Met80 and the heme.	[63]
	Pro(P)76	Pro(P)76 helps coordinate the axial ligand between Met80 and the heme.	[64]
	Met(M)80	Met(M)80 forms an axial ligand with the heme from the distal side.	[60, 61]
	Phe(F)82	Phe(F)82 helps stabilize the native heme environment.	[65]
6	Gly(G)41	Gly(G)41 helps stabilize the axial ligand between Met80 and the heme.	[66]
7	Asn(N)52	Asn(N)52 maintains a hydrogen bond with the heme to stabilize the environment.	[67, 68]
8	Leu(L)94	One of Leu(L)94 or Tyr(Y)97 is required to provide a hydrophobic environment for the function of cyt-c.	[69]
	Tyr(Y)97	One of Leu(L)94 or Tyr(Y)97 is required to provide a hydrophobic environment for the function of cyt-c.	[69]

co-occurrence score is also closer in three-dimensional distance than the average amino acids in the three-dimensional structure. We also confirm that co-occurring APCs form chemical bonds or co-operate in essential biological functions as supported in biological literature. As a natural extension, we can use correlated amino acid variations to track evolutionary divergence and extend the algorithms to discover consistence and deviance of chemical properties. Since it is time-consuming to study the functional and structural sites for every target protein's drug interaction in detail, the ability to discover top-ranking Co-occurrence Clusters could also help to isolate the amino acids of biological significance. Hence, our method will have great potential to impact drug discovery and the biomedical community.

Chapter 4

Using Co-occurrence APCs to Predict Protein-Protein Interactions

Expanding on the co-occurrence of APCs idea from Chapter 3, we apply the idea to a widely researched bioinformatics problem: protein-protein interaction (PPI) prediction. For this chapter, the APC relationship between proteins is calculated by the amount of APC-PPI Score obtained from co-occurrence APC pairs from interacting proteins. These APC relationships are then used as features to be applied to the given protein-protein interaction data set to be learned using Random Forests [70]. The learned model is then used to predict potential PPI based on their target pair's protein sequences. The algorithm was applied to a yeast dataset, with the dataset between training set and testing set in 40 different cases. The results showed that our algorithm performed close to the state of the art algorithm, while providing highly interpretable results, and had higher prediction performance than PIPE.

4.1 Introduction

4.1.1 Background

Protein-protein interaction (PPI) is important for various biological processes and functions in living cells such as metabolic cycles, DNA transcription and replication, and signaling cascades [71]. Predicting PPI is thus critical for better understanding the molecular mechanisms inside the cell [71], particularly useful for discovering unknown functions of a protein [72]. Following [4], we refer to a PPI as an interaction that brings

two different proteins A and B into direct physical contact, i.e. heterodimeric interactions. In contrast, most homodimeric interactions, in which the interacting proteins are identical, are for maintaining the stability of the interacting complex but not for regulating cellular processes [73].

4.1.2 Literature Review

A number of experimental techniques have been developed for systematic and large-scale prediction of PPIs but they are costly, labor-intensive and time-consuming [74]. Thus, existing PPI data obtained by these methods covers only a small fraction of the complete PPI networks [75]. Moreover, these experimental methods usually suffer from high rates of both false positive and false negative predictions [76]. Hence, developing effective and reliable computational methods to facilitate prediction of PPIs is of fundamental importance [29]. Existing computational methods for PPI prediction can be classified according to the input data. Sequence-based methods are becoming popular, since sequence data is more readily available nowadays [72].

PIPE [23] / PIPE2 [24, 25] are well-established sequence-based methods, where PIPE2 is a faster version of PIPE. Given a protein A, a protein B and a database of positive PPIs, PIPE simply counts how frequently all fixed-length protein sequence segments in Proteins A and B co-occur in the database. For example, all combinations of 20-mers between them are first enumerated using a sliding window with a width of 20 amino acids. Then, the co-occurrence of each combination, e.g. MGIRRLVSVITRPIINKVNS from Protein A and GPEAIILTGTFDDWKGTLPM from Protein B, is counted in the database. The sum of all counts is obtained. If the sum is greater than or equal to a threshold, the algorithm then predicts that protein A and B would interact. However, in spite of the satisfactory prediction performance, there is ample room for improvement. The key drawback of PIPE/PIPE2 is their use of a fixed-window of 20 amino acids. This is biologically unrealistic since functional regions such as the Short Linear Motifs (SLiMs [77]) have variable length from 3 to 15 amino acids [77]), mostly less than 10 amino acids [78].

Another well-established sequence-based method involves the use of Support Vector Machine (SVM) with a Pairwise String Kernel [31]. They encode a PPI pair into a feature vector obtained from the co-occurrences of the k-mers (sequences of k residues) used for training the SVM to predict if a protein pair can interact. For example, assume k = 3, a selected feature could be the number of counts of how often the 3-mers, say WTG and LGA co-occur in a protein pair. Since all possible 3-mers are considered, the feature space could be as large as 20^3 x 20^3 (i.e. 64 millions) [79]. With an SVM,

even with such a high dimensionality, by using the kernel trick, neither computing nor storing the feature vector is needed. Thus, since no feature vectors are computed, in spite of achieving satisfactory prediction performance, it is hard to use the SVM results to reveal or interpret why the feature space leads to its good performance. Also, since the feature space is hardly interpretable, not much biological knowledge can be gained. Hence, to overcome this hurdle encountered when using an SVM is a key motivation of our proposed method. In WeMine-P2P, we utilize the local functionally conserved patterns [33, 80] and their co-occurring pattern clusters [81, 81] to obtain biologically realistic and interpretable features that are flexible in pattern length allowing variants. Experiments showed that our prediction results based on these features are comparable to those achieved by the SVM approaches, while being interpretable.

Motivated by the popular acceptance of sequence-based methods and realizing their aforementioned drawbacks, the objective of this study is to develop a new sequence-based prediction method that is (1) based on biologically interpretable discriminative features, (2) more biologically realistic such as allowing variable lengths and variations in the functional regions such as APCs, and (3) producing satisfactory prediction performance between protein sequences. In this study, we propose a new algorithm known as WeMine-P2P to achieve these objectives.

4.2 Method

We make use of our pattern search and integration strategies to discover and locate the "what" and "where" of the conserved regions, using them as discriminative features to construct the PPI classifier, as illustrated in Fig 4.1.

4.2.1 Input: PPI Database.

The input dataset, denoted PPI Database (PPI-DB), includes positive and negative PPI pairs, a pair of protein sequences A positive PPI pair is defined as a pair of protein sequences that can interact with each other, whereas a negative PPI pair is defined as a pair of protein sequences that cannot interact with each other.

The PPI-DB, in addition to the protein sequences as defined in Chapter 2, is use to train a predictive model to indicate if a protein pair would interact. The steps to train and test a predictive model are as follows:

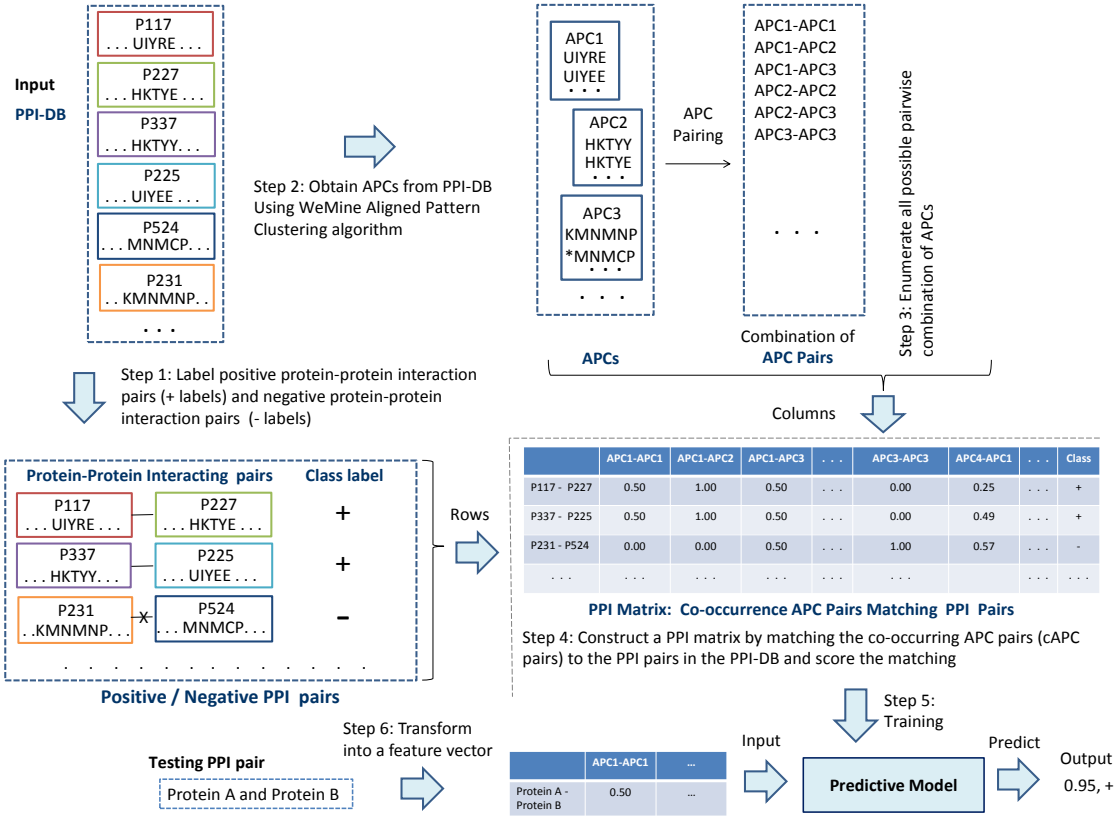

FIGURE 4.1: WeMine-P2P: a PPI Predictor. The input dataset, denoted the PPI Database (PPI-DB), consists of a set of protein sequences, positive and negative PPI pairs. Each protein sequence has a unique ID, e.g. P117, P227...etc. For illustration, only some segments on a protein sequence is shown. To train a predictive model, positive and negative protein-protein interaction pairs are labeled by "+" and "- " labels respectively (Step 1). For extracting features, APCs are obtained from PPI-DB using WeMine Aligned Pattern Clustering algorithm (Step 2). All possible pairwise combination of APCs are then enumerated as co-occurring APC pairs (cAPC pairs) (Step 3). To construct a PPI matrix, cAPC pairs are then matched to the PPI pairs in the PPI-DB and the matching is scored (Step 4). A predictive model is trained on the PPI matrix, where each of its rows is a feature vector (Step 5). Any protein pair can be turned into a feature vector by computing the APC-PPI Score of all extracted cAPC pairs to itself. The feature vector can then be inputted to the trained model to output the classification (Step 6).

4.2.2 Step 1: Label PPI pairs based on PPI-DB.

We label the positive and negative PPI pairs provided by PPI-DB as "+" class and "-" class respectively (Fig 4.1). This helps to form a supervised-learning based training set, in which a sample is a protein pair either in "+" or "-" class. Formally, we let $\mathbb{B} = \mathbb{S} \times \mathbb{S} = \{B_{1,1}, B_{1,2}, ..., B_{|\mathbb{S}|,|\mathbb{S}|}\}$, where each protein pair $B_{x,y}$ is composed of two protein sequences S_x and S_y such that $B_{x,y} = (S_x \times S_y)$.

4.2.3 Step 2: Obtain Aligned Pattern Clusters from PPI-DB.

The protein sequences from PPI-DB are then used to obtain APCs, using the algorithms defined in Chapter 2.

4.2.4 Step 3: Enumerate all possible cAPC pair.

We enumerate all possible pairs of APCs and call a pair of APC as a co-occurring Aligned Pattern Cluster pair (cAPC pair) (Fig 4.1). We define a set of cAPC pairs as $\mathbb{A} = \mathbb{C} \times \mathbb{C} = \{A_{1,1}, A_{1,2}, ..., A_{|\mathbb{C}|,|\mathbb{C}|}\}$, where there are in total $|C| \times |C| = N$ number of cAPC pairs. Each cAPC pair $A_{i,j}$ is composed of two APCs C^i and C^j such that $A_{i,j}$ is the Cartesian product $(C^i \times C^j)$. These cAPC pairs would be features extracted from PPI-DB on the fly to describe PPI pairs.

4.2.5 Step 4: Construct a Protein-Protein Interaction Matrix.

The PPI matrix M consists of rows of sample protein pair $B_{x,y}$ and columns of feature cAPC pair $A_{i,j}$ with the last column being the class label (Fig 4.1). Each cell of the PPI matrix $M(A_{i,j}, B_{x,y})$ has a value (between 0 and 1 inclusively) indicating the strength of occurrence of a cAPC pair $A_{i,j}$ on the protein pair $B_{x,y}$. APC-PPI Score is devised to determine the value. It ranges from 0 and 1 inclusively, and builds upon two other scores: the APCmatchingSegment Score and the APCoccurring Score.

4.2.5.1 APCmatchingSegment Score

We designed Algorithm 5 to match approximately an APC C, or more precisely the patterns in the APC, to a sequence segment E of the same length. We check if each character in the segment occurs in the APC column of the same index. The APCmatchingSegment Score is the sum of the total number of matches (in the segment) normalized by the length of the segment, as exemplified in Figure 4.2.

Algorithm 5 APCmatchingSegment Score

Input: APC C, Sequence segment seg
Output: Value in range [0 1]
for character c_i in seg **do**
  Add match count if c_i is found in σ_i^1 σ_i^2 ... σ_i^m of C { i: column index; m: number of rows in APC C}
end for
return match count / $|seg|$

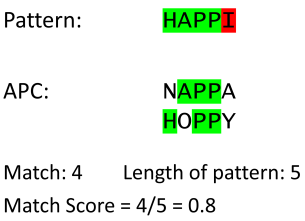

FIGURE 4.2: An example on how the APCmatchingSegment Score is calculated for a segment with 5 characters and an APC with 2 rows.

4.2.5.2 APCoccurring Score

We designed Algorithm 6 to output a score to represent the strength of occurrence of an APC C on a protein sequence S. We use a sliding window of the APC length over the sequence and compute an APCmatchingSegment Score for all segments (at an amount of $|S| - |C| + 1$) of segments. The maximum APCmatchingSegment Score is chosen as the APCoccurring Score.

Algorithm 6 APCoccurring Score

Input: APC C, Protein sequence S
Output: Value in range [0 1]
for $i = 1$ to $|S| - |C| + 1$ **do**
  Find APCmatchingSegment score of $S[i, i + |C|]$ and C
end for
return Maximum APCmatchingSegment score

4.2.5.3 APC-PPI Score

The APC-PPI Score is obtained from Algorithm 7 to measure the strength of occurrence of a cAPC pair on a PPI pair. It first calculates two APCoccurring Scores for each of the two possible APC-Protein combinations. Then the average of the APCoccurring Scores in each APC-Protein combination is calculated. Then, each of the two APC-Protein combinations is associated with a score. The APC-PPI Score is the maximum one among those two scores.

Algorithm 7 APC-PPI Score

Input: cAPC pair $A_{1,2}$, with APCs C^1 and C^2, and a PPI pair $B_{1,2}$, with Protein sequences S_1 and S_2.
Output: Value in range [0 1]
Let $Score1$ = Average of APCoccurring Score between (C^1,S_1) and (C^2,S_1)
Let $Score2$ = Average of APCoccurring Score between (C^1,S_2) and (C^2,S_2)
return Max of $Score1$ and $Score2$

4.2.6 Step 5: Train a predictive model based on the PPI Matrix.

We train a predictive model, specifically Random Forest [70], based on the constructed PPI matrix. Random forest is an ensemble learning method. In this study, we mainly use it for binary classification, i.e. to predict if a protein pair is a positive or negative PPI pair. It operates by constructing a number of decision trees in training, then outputting the class label by voting, i.e. the mode of individual trees. We choose Random Forest as our predictive model because 1) it runs efficiently on large training sets and is easily parallelized [70]; 2) it can handle lots of input variables without variable deletion [70]; 3) it seldom overfits the training set [70]. We adopt the machine learning package WEKA 3.7 [2] in training the Random Forest predictive model. It supports outputting the prediction probability in addition to the class label.

4.2.7 Step 6: Predict the testing protein pairs.

Given a testing protein pair, we first transform it into a feature vector by computing the APC-PPI Score of all extracted cAPC pairs to it. When the feature vector is constructed, we then input it to the predictive model to obtain a class label, and also the probability of the prediction (supported by WEKA [2]).

4.2.8 Feature analysis: cAPC pair Selection.

To analyze the features, we have developed a score to measure how discriminative a cAPC pairs column, $A_{i,j}$, in the PPI matrix is. The higher the score the cAPC pair could obtain, the more likely that it would co-occur in positive PPI pair but less in negative PPI pair. This score is built upon the APC-PPI Score but needs to be normalized to the number of PPI pairs (positive or negative) in the PPI matrix. We first define

$$tscore(A_{i,j}, B_{x,y}) = \begin{cases} \frac{score(A_{i,j}, B_{x,y})}{posPPI}, & \text{if +ve PPI pair,} \\ -\frac{score(A_{i,j}, B_{x,y})}{negPPI}, & \text{if -ve PPI pair,} \end{cases} \tag{4.1}$$

where $score(A_{i,j}, B_{x,y})$ is the APC-PPI Score, $posPPI$ is the number of positive PPI pair, and $negPPI$ is the number of negative PPI pair. The $tscore(A_{i,j}, B_{x,y})$ that relate to a cAPC pair $A_{i,j}$ is summed among all PPI pairs in $\mathbb{B}$. We define

$$hscore(A_{i,j}) = \sum_{\forall B_{x,y} \in \mathbb{B}} tscore(A_{i,j}, B_{x,y}) \tag{4.2}$$

We could then use *hscore* to rank the cAPC pairs.

4.3 Materials

Forty independent Yeast_Random datasets were downloaded from [4] at http://www.marcottelab.org/differentialGeneralization. The procedure to obtain these 40 datasets is described below. Yeast Protein-Protein Interaction (PPI) data ("Saccharomyces_cerevisiae-20100304.txt") containing the protein sequences and the positive PPI pairs was acquired from the protein interaction network analysis platform [82]. Further pre-processing was applied to the proteins therein. First, the proteins were clustered using CD-HIT2 [83] with the requirement that they shared sequence identity less than 40%. Second, the proteins with less than 50 amino acids as well as homo-dimeric interactions were also removed. In total, 6806 Yeast protein sequences remained after the pre-processing.

It is shown by [4] that predictive models perform much better for test pairs that share components with the training set than for those that do not. Traditional cross-validation, yet, overlooked this issue [4]. Hence, to prepare a training set with both positive and negative PPI pairs, a specific resampling process was conducted by [4] on the 6806 Yeast protein sequences to obtain 40 independent datasets. In each dataset, there are about 16000 PPI pairs for training and about 4000 PPI pairs (including C1, C2 and C3) for testing. It should be noted that the number of positive and negative PPIs is in equal amount. A simplified example dataset with training set and testing set C1, C2 and C3 is illustrated in Fig 4.3 with proteins existing in the training dataset in green and novel proteins not from the training dataset in red. The required generalization ability from the classifier increases with the number of novel protein sequences from C1 to C3.

4.4 Results and Analysis

4.4.1 Experimental Design and Parameter Setting

As mentioned in Section 4.3 Materials, we obtained in total 40 independent datasets provided by [4]. Each dataset was partitioned into a training set and a testing set. In our experiment, we first extracted features (Step 1, Step 2) from the training set, then used the features to construct PPI matrix (Step 3, Step 4) and trained a predictive model on the PPI matrix. In Step 1, we used WeMine Aligned Pattern Clustering algorithm [33, 80] to obtain APCs with length varying from 5 to 10 amino acids inclusively with the

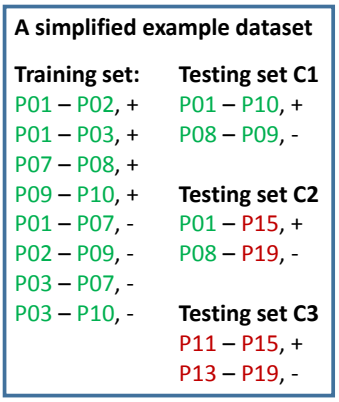

FIGURE 4.3: A simplified dataset example with a training set and a testing set with three distinct classes as defined in [4]. Each row is a pair of protein sequences with a class label. "+" means positive interactions and "-" means negative interactions. The positive PPI pairs are experimentally validated while the negative PPI pairs are sampled from the proteins within the same set that are not known to interact [84]. Proteins existing in the training dataset are in green and novel proteins not from the training dataset are in red. For example, in the training set, P01-P02 and P07-P08 are positive PPI pairs but P01-P07 is a negative PPI pair. All protein pairs in the testing sets are not found in the training set. However, all the protein sequences in C1 are in the training set, while in C2 only some protein sequences are in the training set, and in C3 no protein sequences are in the training set.

minimum support of 6, and the clustering threshold of 0.1. Other WeMine parameters remain default [33, 80]. We also trained 3000 trees in the Random Forest in Step 5 using Weka 3.7 [2]. Other Weka 3.7 parameters remain default [2]. We then transformed every PPI pair in the testing set into a feature vector (Step 6) and applied the trained model on it to output a class label and a score. We evaluated the predictive performance by computing the Area Under Curve (AUC) following [4]. We repeated the same procedure for all 40 independent datasets and computed the average AUC for comparison with Methods 1-7 in [4].

4.4.2 Comparison to PIPE2

To illustrate the improvement made by WeMine-P2P on the use of co-occurring sequence segments, we compared the average AUC with those obtained by PIPE2, provided by [4]. Recall that PIPE2 [24, 25] uses the short amino acid sequences (fixed at length of 20) that co-occur frequently in given positive PPI pairs to make predictions on a testing PPI pair. As shown in Table 4.1, our results demonstrate that WeMine-P2P achieves better performance in all three testing sets comparing to PIPE2, indicating that WeMine-P2P outperformed PIPE2 in this experiment. WeMine-P2P is novel in the sense that 1) the length of sequence patterns is allowed to vary, coping with inherent functional association in the form of statistically significant patterns; 2) sequence patterns are clustered and aligned as Aligned Pattern Clusters (APCs) to relate to inherent functional conservation

TABLE 4.1: Comparing PIPE2 and WeMine-P2P on the average Area Under Curve among 40 independent datasets

	Testing set C1	Testing set C2	Testing set C3
Method 6 (PIPE2)	0.75	0.59	0.52
WeMine-P2P	**0.79**	**0.61**	**0.58**

TABLE 4.2: Comparing SVM-based methods and WeMine-P2P on the average Area Curve Cover among 40 independent datasets

	Testing set C1	Testing set C2	Testing set C3
Method 1 [26, 27]	0.82	**0.61**	0.58
Method 2 [26, 27]	**0.84**	0.60	**0.59**
Method 3 [29]	0.61	0.53	0.50
Method 4 [30]	0.76	0.57	0.54
Method 5 [30]	0.80	0.58	0.55
Method 7 [31]	0.58	0.54	0.52
WeMine-P2P	0.79	**0.61**	0.58

and variations; 3) nonlinear predictive models can then be trained with the feature vectors. Since WeMine-P2P has overcome the drawbacks of PIPE2, it does outperform it in the experiment.

4.4.3 Comparison to SVM-based Methods

To further illustrate the strength of WeMine-P2P, we compared its average AUC to the SVM-based methods that are well-known for achieving the state-of-the-art predictive performances. The average AUC of SVM-based methods were obtained in [4]. As shown in Table 4.2, WeMine-P2P achieved comparable results, particularly for the testing sets C2 and C3, in which some testing protein sequences in C2 and all in C3 are new and not found in the training set (Fig. 4.3). (For details please refer to Section 4.3 Materials.) This illustrates that WeMine-P2P has similar predictive power comparing to SVM-based methods for novel testing protein sequences. We have to point out that while assuming the pattern length k = 3, the feature dimension of SVM-based methods with String Kernel [26, 27], though not computed nor stored, can be as large as $20^3 \times 20^3 =$ 64,000,000. In WeMine-P2P, the feature dimension is only around 50,000, while allowing the variation of residues with the pattern length varying from 5 to 10. It is a potential reduction of 1280x in feature dimension. Also, while the feature vector is fixed in SVM-based methods, WeMine-P2P could extract features from the input data on the fly, allowing them to be biologically interpretable as described in the next section. Note cAPC pair that Methods 5 and 7 do not use SVM directly but are variants of SVM-based methods [4].

TABLE 4.3: The top 10 cAPC pairs in *hscore*

Rank	1st APC ID	2nd APC ID	*hscore*
1	1465525	9692312	0.018337
2	1465525	9698509	0.018083
3	1465525	1465525	0.018030
4	1465525	9487593	0.017986
5	1465525	9728806	0.017978
6	1465525	8234623	0.017748
7	1465525	9590335	0.017430
8	1465525	9658538	0.017391
9	8234623	9658538	0.017231
10	9642970	9658538	0.017229

TABLE 4.4: The APCs in the top 10 cAPC pairs

APC ID	APC in Regular Expression
1465525	[**AQ**]**QAQ**[V**A**]
9692312	[GADSNEITV][VSANDGELK][EGDIQPLFNTS]EE[NTASGQVLRKID][DLGEIKANQTSVRYF][DRKA]
9698509	[**K**D**E**VIL][**E**GLSKNV]**E**[V**KE**LQRIAF]**K**[QR**EK**TS][**KQE**D]
9487593	[KI**E**ALNV][KD**E**NIA]**E**[LNSTIVGQRADK]**E**[Q**E**K][LAQ]
9728806	[IEDNLSFG][TIDLNSEFGKQ][LIKFTDEVR]DE[ANSIDFE][TDVSILKAYEQN][IAKLEDSQ][MD]
8234623	**E**[GLQTNKDSIVRA**E**]**EEE**[D**E**][GKQISTNRAL]**E**
9590335	[KSE][INE]VD[GLADKE][LD]
9658538	[NL][DS**E**V]**E**[GVKD**E**]**E**[ISGVDK**E**]**EE**
9642970	[**RK**E][**K**DI**R**][**R**AE**K**D]**RK**[LASE][AS**K**]**K**

4.4.4 Analysis of the discriminative features

We also investigate the discriminative features discovered by WeMine-P2P. We focus our analysis on the training data in the independent dataset (ID = 11). We adopted the *hscore* defined in Section 4.2. Methodology in order to compute a feature score (within -1 and 1 inclusively) for each feature (i.e. cAPC pair). The higher the score, the more likely the cAPC pair co-occur in positive PPI and less likely they co-occur in negative PPI. The features are ranked from the highest to lowest. The top 10 cAPC pairs are shown in Table 4.3 and their corresponding APCs are shown in Table 4.4.

We observed that 9 out of the top 10 APC pairs include an APC likely to represent a segment in the compositional bias region. For example, "AMAMAAMAMAMA" is a compositional bias region in which "A" and "M" are enriched. According to [85], compositional bias region is composed of amino acids that have locally shifted frequencies. Likely compositional bias region within the top-10 APCs are APC 1465525 (enriched for "A" and "Q"), APCs 9487593, 9692312, 8234623, 9658538 (enriched for "E"), APC 9642970 (enriched for "K" and "R"), and APC 9698509 (enriched for "K" and "E"), and

APC 9728806 (enriched for "D" and "E"). These enriched regions can play important roles in PPI [86].

4.5 Discussion and Summary

Protein-Protein Interaction (PPI) has been studied for years but discovering new PPIs remains challenging. Different types of biochemical experiments and computational methods have been proposed but each of them has their own limitations. Sequence-based machine learning methods are becoming more and more popular because they are readily applicable and achieve satisfactory performance. The approach of this thesis is not only able to produce quality predictive results but also to discover deeper statistical and functional knowledge in PPIs. It could analyze the interpretable discriminative features while existing methods could hardly produce both in accurate and biologically interpretable results as WeMine-P2P does. Furthermore, existing methods adopt features that are not biologically realistic such as fixing the pattern length and using exact patterns. However, WeMine-P2P could autonomously determine them within the discovered APCs. The technical contribution of this work is to furnish a new sequence-based method that overcomes these drawbacks while retaining the predictive performance. Since no prior information on PPI has been incorporated, WeMine-P2P is potentially extendable to other biosequence applications such as predicting Protein-DNA interactions [81] in the future.

Chapter 5

Conclusions

Understanding the characteristics of proteins is of great importance because of the many important functions that proteins have in living organisms. In addition, due to the drastic reduction of biosequencing cost and time, there is a large influx of protein sequences that can be analyzed upon for understanding of more about the underlying proteins. However, wet laboratory experiments are both labour intensive and time consuming. Hence, computational approaches are necessary to play an important role in analyzing protein sequences. This thesis is built on top of the success of Aligned Pattern Clustering (APCL), to further analyze protein sequences. Two algorithms were created based on the idea of co-occurrence of APCs to solve two important problems. The first algorithm focuses on finding and grouping functional regions of the same protein. It shows positive results in finding clusters of regions (APCs) that are either in close contact in three-dimensional structures (implying a possible interaction between the regions) or are shown to be co-operative in biological functions based on biological literature. The second algorithm focused on improving the current performance in predicting protein-protein interactions. After comparisons, the second algorithm displayed a similar performance with the state of the art methods for protein-protein interaction prediction (Table 4.2), and with more interpretable results. Hence, both algorithms provided positive results in its usefulness for analyzing protein sequences.

Future directions on the development of two algorithms includes continual refinement on the algorithms with the goal of their use in the field. In particular, our WeMine-P2P is potentially useful in drug industries such as drug target discovery with its protein-protein interaction prediction ability. The program can help to find new drugs, to be specific, the inhibitors for protein-protein interactions using the faster and more comprehensive search and predicting capability of WeMine-P2P. Furthermore, with interpretable details, the program can help drug companies gain further insight into PPI in

greater detail, giving researchers vastly more information to create drugs for fighting maligned proteins. While our current program is not ready for industrial use currently and demands further refinements, the results from this thesis have shown a great hope in reaching that goal.

Bibliography

[1] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Addison-Wesley, 2006.

[2] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[3] En-Shiun Annie Lee, Sanderz Fung, Ho-Yin Sze-To, and Andrew KC Wong. Confirming biological significance of co-occurrence clusters of aligned pattern clusters. *BIBM, p.(To Appear)*, 2013.

[4] Yungki Park and Edward M Marcotte. Flaws in evaluation schemes for pair-input computational predictions. *Nature methods*, 9(12):1134–1136, 2012.

[5] Arthur Lesk. *Introduction to bioinformatics*. Oxford University Press, 2013.

[6] En-Shiun Annie Lee and Andrew K. C. Wong. Revealing binding segments in protein families using aligned pattern clusters. *Proteome Science*, 2013.

[7] D Whitford. Proteins: structure and function. 2005.

[8] UniProt Consortium et al. The universal protein resource (uniprot). *Nucleic acids research*, 36(suppl 1):D190–D195, 2008.

[9] Nancy Craig, Rachel Green, Carol Greider, Gisela Storz, Orna Cohen-Fix, and Cynthia Wolberger. *Molecular biology: principles of genome function*. Oxford University Press, 2014.

[10] UniProt Consortium et al. Uniprot: a hub for protein information. *Nucleic acids research*, page gku989, 2014.

[11] Isabel HM Vandenberghe, Yves Guisez, Stefano Ciurli, Stefano Benini, and Jozef J Van Beeumen. Cytochrome c-553 from the alkalophilic bacterium bacillus pasteurii has the primary structure characteristics of a lipoprotein. *Biochemical and biophysical research communications*, 264(2):380–387, 1999.

[12] Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, TN Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank. *Nucleic acids research*, 28(1):235–242, 2000.

[13] Stefano Benini, Ana González, Wojciech R Rypniewski, Keith S Wilson, Jozef J Van Beeumen, and Stefano Ciurli. Crystal structure of oxidized bacillus pasteurii cytochrome c 553 at 0.97-å resolution. *Biochemistry*, 39(43):13115–13126, 2000.

[14] Robert D Finn, Jaina Mistry, John Tate, Penny Coggill, Andreas Heger, Joanne E Pollington, O Luke Gavin, Prasad Gunasekaran, Goran Ceric, Kristoffer Forslund, et al. The pfam protein families database. *Nucleic acids research*, 38(suppl 1): D211–D222, 2010.

[15] Ruth Nussinov and Gideon Schreiber. *Computational protein-protein interactions*. CRC Press, 2009.

[16] Oliver Maneg, Francesco Malatesta, Bernd Ludwig, and Viktoria Drosou. Interaction of cytochrome c with cytochrome oxidase: two different docking scenarios. *Biochimica et Biophysica Acta (BBA)-Bioenergetics*, 1655:274–281, 2004.

[17] Julie D Thompson, Desmond G Higgins, and Toby J Gibson. Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic acids research*, 22(22):4673–4680, 1994.

[18] Cédric Notredame, Desmond G Higgins, and Jaap Heringa. T-coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of molecular biology*, 302(1):205–217, 2000.

[19] Chuong B Do, Mahathi SP Mahabhashyam, Michael Brudno, and Serafim Batzoglou. Probcons: Probabilistic consistency-based multiple sequence alignment. *Genome research*, 15(2):330–340, 2005.

[20] Timothy L Bailey, Nadya Williams, Chris Misleh, and Wilfred W Li. Meme: discovering and analyzing dna and protein sequence motifs. *Nucleic acids research*, 34 (suppl 2):W369–W373, 2006.

[21] Jorja G Henikoff and Steven Henikoff. [6] blocks database and its applications. *Methods in enzymology*, 266:88–105, 1996.

[22] Gapped BLAST. Psi-blast: a new generation of protein database search programs altschul. *Stephen F*, pages 3389–3402.

[23] Sylvain Pitre, Frank Dehne, Albert Chan, Jim Cheetham, Alex Duong, Andrew Emili, Marinella Gebbia, Jack Greenblatt, Mathew Jessulat, Nevan Krogan, et al. Pipe: a protein-protein interaction prediction engine based on the re-occurring short polypeptide sequences between known interacting protein pairs. *BMC bioinformatics*, 7(1):365, 2006.

[24] S Pitre, C North, M Alamgir, M Jessulat, A Chan, X Luo, JR Green, M Dumontier, F Dehne, and A Golshani. Global investigation of protein–protein interactions in yeast saccharomyces cerevisiae using re-occurring short polypeptide sequences. *Nucleic acids research*, 36(13):4286–4294, 2008.

[25] Sylvain Pitre, Mohsen Hooshyar, Andrew Schoenrock, Bahram Samanfar, Matthew Jessulat, James R Green, Frank Dehne, and Ashkan Golshani. Short co-occurring polypeptide regions can predict global protein interaction maps. *Scientific reports*, 2, 2012.

[26] Shawn Martin, Diana Roe, and Jean-Loup Faulon. Predicting protein–protein interactions using signature products. *Bioinformatics*, 21(2):218–226, 2005.

[27] Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

[28] Jean-Philippe Vert, Jian Qiu, and William S Noble. A new pairwise kernel for biological network inference with support vector machines. *BMC bioinformatics*, 8 (Suppl 10):S8, 2007.

[29] Juwen Shen, Jian Zhang, Xiaomin Luo, Weiliang Zhu, Kunqian Yu, Kaixian Chen, Yixue Li, and Hualiang Jiang. Predicting protein–protein interactions based only on sequences information. *Proceedings of the National Academy of Sciences*, 104 (11):4337–4341, 2007.

[30] Yanzhi Guo, Lezheng Yu, Zhining Wen, and Menglong Li. Using support vector machine combined with auto covariance to predict protein–protein interactions from protein sequences. *Nucleic acids research*, 36(9):3025–3030, 2008.

[31] Matteo Bellucci, Federico Agostini, Marianela Masin, and Gian Gaetano Tartaglia. Predicting protein associations with long noncoding rnas. *Nature Methods*, 8(6): 444–445, 2011.

[32] Andrew KC Wong, Dennis Zhuang, Gary CL Li, and E-SA Lee. Discovery of delta closed patterns and noninduced patterns from sequences. *Knowledge and Data Engineering, IEEE Transactions on*, 24(8):1408–1421, 2012.

[33] En-Shiun Annie Lee and Andrew KC Wong. Ranking and compacting binding segments of protein families using aligned pattern clusters. *Proteome Sci*, 11, 2013.

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

[35] Julie D Thompson, Benjamin Linard, Odile Lecompte, and Olivier Poch. A comprehensive benchmark study of multiple sequence alignment methods: current challenges and future perspectives. *PloS one*, 6(3):e18093, 2011.

[36] Lusheng Wang and Tao Jiang. On the complexity of multiple sequence alignment. *Journal of computational biology*, 1(4):337–348, 1994.

[37] Martin Weigt, Robert A White, Hendrik Szurmant, James A Hoch, and Terence Hwa. Identification of direct residue contacts in protein–protein interaction by message passing. *Proceedings of the National Academy of Sciences*, 106(1):67–72, 2009.

[38] Lukas Burger and Erik van Nimwegen. Disentangling direct from indirect co-evolution of residues in protein alignments. *PLoS computational biology*, 6(1): e1000633, 2010.

[39] Faruck Morcos, Andrea Pagnani, Bryan Lunt, Arianna Bertolino, Debora S Marks, Chris Sander, Riccardo Zecchina, José N Onuchic, Terence Hwa, and Martin Weigt. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences*, 108(49):E1293–E1301, 2011.

[40] Srinivasan Madabushi, Alecia K Gross, Anne Philippi, Elaine C Meng, Theodore G Wensel, and Olivier Lichtarge. Evolutionary trace of g protein-coupled receptors reveals clusters of residues that determine global and class-specific functions. *Journal of Biological Chemistry*, 279(9):8126–8132, 2004.

[41] Xuhua Xia. Position weight matrix, gibbs sampler, and the associated significance tests in motif characterization and prediction. *Scientifica*, 2012, 2012.

[42] UniProt Consortium et al. Activities at the universal protein resource (uniprot). *Nucleic Acids Research*, 42(D1):D191–D198, 2014.

[43] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[44] Christopher M Bishop et al. *Pattern recognition and machine learning*, volume 1. springer New York, 2006.

[45] Joseph C Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. 1973.

[46] E.-S.A. Lee, S. Fung, Ho-Yin Sze-To, and A.K.C. Wong. Confirming biological significance of co-occurrence clusters of aligned pattern clusters. In *Bioinformatics and Biomedicine (BIBM), 2013 IEEE International Conference on*, pages 422–427, Dec 2013. doi: 10.1109/BIBM.2013.6732529.

[47] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.

[48] Eric F Pettersen, Thomas D Goddard, Conrad C Huang, Gregory S Couch, Daniel M Greenblatt, Elaine C Meng, and Thomas E Ferrin. Ucsf chimera-a visualization system for exploratory research and analysis. *Journal of computational chemistry*, 25(13):1605–1612, 2004.

[49] Senadhi Vijay-Kumar, Charles E Bugg, Keith D Wilkinson, and William J Cook. Three-dimensional structure of ubiquitin at 2.8 a resolution. *Proceedings of the National Academy of Sciences*, 82(11):3582–3585, 1985.

[50] Ivan Dikic, Soichi Wakatsuki, and Kylie J Walters. Ubiquitin-binding domains-from structures to functions. *Nature reviews Molecular cell biology*, 10(10):659–671, 2009.

[51] Denis Tempé, Muriel Brengues, Pauline Mayonove, Hayat Bensaad, Céline Lacrouts, and May C Morris. The alpha helix of ubiquitin interacts with yeast cyclin-dependent kinase subunit cks1. *Biochemistry*, 46(1):45–54, 2007.

[52] Timothy J Burch and Arthur L Haas. Site-directed mutagenesis of ubiquitin. differential roles for arginine in the interaction with ubiquitin-activating enzyme. *Biochemistry*, 33(23):7300–7308, 1994.

[53] S Vijay-Kumar, CE Bugg, KD Wilkinson, RD Vierstra, PM Hatfield, and WJ Cook. Comparison of the three-dimensional structures of human, yeast, and oat ubiquitin. *Journal of Biological Chemistry*, 262(13):6396–6399, 1987.

[54] Sobia Zaidi, Md Imtaiyaz Hassan, Asimul Islam, and Faizan Ahmad. The role of key residues in structure, function, and stability of cytochrome-c. *Cellular and Molecular Life Sciences*, 71(2):229–255, 2014.

[55] Tsunehiro Takano and Richard E Dickerson. Redox conformation changes in refined tuna cytochrome c. *Proceedings of the National Academy of Sciences*, 77(11):6371–6375, 1980.

[56] VB Sampson, T Alleyne, and D Ashe. Probing the specifics of substrate binding for cytochrome c oxidase a computer assisted approach. *West Indian Medical Journal*, 58(1), 2009.

[57] Oleksandr Kokhan, Colin A Wraight, and Emad Tajkhorshid. The binding interface of cytochrome c and cytochrome c1 in the bc1 complex: Rationalizing the role of key residues. *Biophysical journal*, 99(8):2647–2656, 2010.

[58] Paul D Barker and Stuart J Ferguson. Still a puzzle: why is haem covalently attached in c-type cytochromes? *Structure*, 7(12):R281–R290, 1999.

[59] Sarah EJ Bowman and Kara L Bren. The chemistry and biochemistry of heme c: functional bases for covalent attachment. *Natural product reports*, 25(6):1118–1130, 2008.

[60] Stephen J Hagen, Ramil F Latypov, Dimitry A Dolgikh, and Heinrich Roder. Rapid intrachain binding of histidine-26 and histidine-33 to heme in unfolded ferrocytochrome c. *Biochemistry*, 41(4):1372–1380, 2002.

[61] Tsunehiro Takano and Richard E Dickerson. Conformation change of cytochrome c: I. ferrocytochrome c structure refined at 1· 5 å resolution. *Journal of molecular biology*, 153(1):79–94, 1981.

[62] CJ Wallace, P Mascagni, BT Chait, JF Collawn, Y Paterson, AE Proudfoot, and SB Kent. Substitutions engineered by chemical synthesis at three conserved sites in mitochondrial cytochrome c. thermodynamic and functional consequences. *Journal of Biological Chemistry*, 264(26):15199–15209, 1989.

[63] Carmichael JA Wallace and Ian Clark-Lewis. A rationale for the absolute conservation of asn70 and pro71 in mitochondrial cytochromes c suggested by protein engineering. *Biochemistry*, 36(48):14733–14740, 1997.

[64] Karen M Black and Carmichael JA Wallace. Probing the role of the conserved β-ii turn pro-76/gly-77 of mitochondrial cytochrome c. *Biochemistry and cell biology*, 85(3):366–374, 2007.

[65] Gordon V Louie, Gary J Pielak, Michael Smith, and Gary D Brayer. Role of phenylalanine-82 in yeast iso-1-cytochrome c and remote conformational changes induced by a serine residue at this position. *Biochemistry*, 27(20):7870–7876, 1988.

[66] Tracy M Josephs, Matthew D Liptak, Gillian Hughes, Alexandra Lo, Rebecca M Smith, Sigurd M Wilbanks, Kara L Bren, and Elizabeth C Ledgerwood. Conformational change and human cytochrome c function: mutation of residue 41 modulates

caspase activation and destabilizes met-80 coordination. *JBIC Journal of Biological Inorganic Chemistry*, 18(3):289–297, 2013.

[67] R Sanishvili, KW Volz, EM Westbrook, and E Margoliash. The low ionic strength crystal structure of horse cytochrome c at 2.1 å resolution and comparison with its high ionic strength counterpart. *Structure*, 3(7):707–716, 1995.

[68] A Schejter, TI Koshy, TL Luntz, R Sanishvili, I Vig, and E Margoliash. Effects of mutating asn-52 to isoleucine on the haem-linked properties of cytochrome c. *Biochem. J*, 302:95–101, 1994.

[69] Zoey L Fredericks and Gary J Pielak. Exploring the interface between the n-and c-terminal helixes of cytochrome c by random mutagenesis within the c-terminal helix. *Biochemistry*, 32(3):929–936, 1993.

[70] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[71] Anne-Claude Gavin, Markus Bösche, Roland Krause, Paola Grandi, Martina Marzioch, Andreas Bauer, Jörg Schultz, Jens M Rick, Anne-Marie Michon, Cristina-Maria Cruciat, et al. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415(6868):141–147, 2002.

[72] Lun Hu and Keith Chan. Discovering variable-length patterns in protein sequences for protein-protein interaction prediction. 2015.

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

[74] Zhu-Hong You, Keith CC Chan, and Pengwei Hu. Predicting protein-protein interactions from primary protein sequences using a novel multi-scale local feature representation scheme and the random forest. 2015.

[75] Zhu-Hong You, Ying-Ke Lei, Jie Gui, De-Shuang Huang, and Xiaobo Zhou. Using manifold embedding for assessing and predicting protein interactions from high-throughput experimental data. *Bioinformatics*, 26(21):2744–2751, 2010.

[76] Xin Luo, Zhuhong You, Mengchu Zhou, Shuai Li, Hareton Leung, Yunni Xia, and Qingsheng Zhu. A highly efficient approach to protein interactome mapping based on collaborative filtering framework. *Scientific reports*, 5, 2015.

[77] Richard J Edwards and Nicolas Palopoli. Computational prediction of short linear motifs from protein sequences. In *Computational Peptidology*, pages 89–141. Springer, 2015.

[78] Floriane Montanari, Denis C Shields, and Nora Khaldi. Differences in the number of intrinsically disordered regions between yeast duplicated proteins, and their relationship with functional divergence. *PloS one*, 6(9):e24989, 2011.

[79] Tobias Hamp and Burkhard Rost. Evolutionary profiles improve protein-protein interaction prediction from sequence. *Bioinformatics*, page btv077, 2015.

[80] Andrew KC Wong and En-Shiun Annie Lee. Aligning and clustering patterns to reveal the protein functionality of sequences. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 11(3):548–560, 2014.

[81] En-Shiun Annie Lee, Ho-Yin Sze-To, Man-Hon Wong, Kwong-Sak Leung, Terrence Chi-Kong Lau, and Andrew KC Wong. Discovering protein-dna binding cores by aligned pattern clustering. In *Bioinformatics and Biomedicine (BIBM), 2014 IEEE International Conference on*, pages 125–130. IEEE, 2014.

[82] Jianmin Wu, Tea Vallenius, Kristian Ovaska, Jukka Westermarck, Tomi P Mäkelä, and Sampsa Hautaniemi. Integrated network analysis platform for protein-protein interactions. *Nature methods*, 6(1):75–77, 2009.

[83] Weizhong Li and Adam Godzik. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, 22(13):1658–1659, 2006.

[84] Yungki Park and Edward M Marcotte. Revisiting the negative example sampling problem for predicting protein–protein interactions. *Bioinformatics*, 27(21):3024–3028, 2011.

[85] Kirill S Antonets and Anton A Nizhnikov. sarp: A novel algorithm to assess compositional biases in protein sequences. *Evolutionary bioinformatics online*, 9:263, 2013.

[86] Vladimir N Uversky and A Keith Dunker. Understanding protein non-folding. *Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics*, 1804(6):1231–1264, 2010.