Characterization and prediction of B-cell epitopes

Liang, Zhao

2013

Zhao, L. (2013). Characterization and prediction of B-cell epitopes. Doctoral thesis,
Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/54752>

<https://doi.org/10.32657/10356/54752>

CHARACTERIZATION AND PREDICTION OF B-CELL EPITOPES

LIANG ZHAO

School of Computer Engineering

A Dissertation Submitted to the School of Computer
Engineering of Nanyang Technological University in
Partial Fulfilment for the Degree of Doctor of Philosophy

2013

Acknowledgments

I heartily thank my supervisor, Dr. Steven C.H. Hoi, for his excellent guidance and invaluable suggestions. His kind support, in both spirit and finance, allows me to eventually accomplish this work.

I also would like to express my sincere gratitude to my previous supervisor, Dr. Jinyan Li, for introducing me to the field of bioinformatics and investing huge amount of time in advising my doctoral research.

My heartfelt gratitude also goes to Dr. Limsoon Wong, for his enthusiastic encouragement and great advices on my research.

I want to thank Dr. Tao Zeng, for his patiently guidance at the beginning of my doctoral study, Dr. Zhenhua Li and Dr. Qian Liu, for their helpful technical discussions. I would also like to pay my sincere gratitude towards the colleagues of Bioinformatics Research Center, for their valuable comments and generous help.

Finally, I would like to pay my wholehearted appreciation to my elder brother, Ming, for everything he has done for me. Without his constant encouragement and strong support, I would not be able to make this achievement.

Abstract

A B-cell epitope is a set of antigen surface residues which can be recognized by an antibody. Identifying epitopes facilitates the understanding of the basic recognition mechanism of immune responses, which in turn guides disease diagnosis, vaccine design and drug development. However, the identification of epitopes is challenging due to the complicated nature of antigen-antibody interactions as well as the context-awareness principle behind the interaction. Context-awareness highlights the existence of multiple epitopes in antigens and the reconfiguration of epitope residues when an antigen interacts with a different antibody. A coarse binary classification of antigen regions into epitopes or non-epitopes without specifying antibodies may not accurately reflect this biological reality. To accurately and rationally detect the epitopes of an antigen, comprehensive analysis of multiple epitopes is carried out. This is followed by antibody-specific epitope prediction in line with the principle of context-awareness and antibody-agnostic epitope prediction which is capable of predicting one or multiple epitopes that are consistent with the rule of context-dependence.

A multi-interface domain is one that can shape multiple and distinctive binding sites to contact with many other domains, forming a hub in domain-domain interaction networks. Graph theory and algorithms are applied to discover fingerprints of interfaces, explore relations between interfaces, and establish associations between interfaces and their functions of multi-interface domains retrieved from the PDB. Experimental results show that about 40% of proteins have multiple interfaces; however, the involved multi-interface domains account for only a tiny fraction (1.8%) of the total number of domains. The interfaces of these multi-interface domains are distinguishable in terms of their fingerprints, indicating the functional specificity of

the multiple interfaces in a domain. Furthermore, both cooperative and distinctive structural patterns are observed in the interfaces of multi-interface domains.

Based on the fact that multiple interfaces exist in antigens and that these interfaces associate with different antibodies, a two-dimensional association-based model is established to predict antibody-specific epitopes. The two kinds of associations revealing the contextual awareness are: (i) residues-residues pairing preference, and (ii) the dependence between sets of contacting residue pairs. Preference plays a bridging role to link interacting paratope and epitope residues, while dependence is used to infer new interacting residue pairs. Experiments conducted on a non-redundant data set containing 80 antibody-antigen structural complexes have found that the proposed model yields good performance in antibody-specific epitope prediction. In addition, this model predicts antibody-specific epitopes from antigen-antibody sequences, although it is trained on antigen-antibody structural complexes, hence indicating its broad applicability in epitope prediction.

The two-dimensional association can capture the context-awareness of paratope-epitope interacting complexes, but it cannot cover the contacts within a paratope or an epitope. Thus, a new concept — coupling graph — is introduced to include both inter-protein contacts between a paratope and an epitope as well as intra-protein contacts within a paratope or an epitope. The coupling graph is a two-layered graph with each node in one graph connecting with nodes in the other graph. The coupling graph can represent the context-awareness principle well; however, it is very challenging to mine frequent coupling subgraphs which are used to reveal the context-awareness. Therefore, a new algorithm for coupling graph mining, based on graph transformation, has been designed. Experiments show that the innovative algorithm significantly reduces the time cost and memory consumption in coupling graph mining, and its application in antibody-specific epitope prediction outperforms the association-based model.

A novel graph based on the antibody-agnostic epitope prediction model is built to predict one or multiple epitopes of an antigen, which overcomes the problem that

existing models predict all the antigenic residues of an antigen as a single epitope although these antigenic residues may belong to totally different epitopes. This model divides an antigen surface graph into subgraphs by using the Markov Clustering algorithm, and then a classifier is constructed to distinguish these subgraphs as epitopes or non-epitopes. The classifier is then taken to predict epitopes for a test antigen. On a big data set of 92 antigen-antibody PDB complexes, the proposed method significantly outperforms the state-of-the-art epitope prediction methods, achieving 24.7% higher averaged f-score than the best existing model. In particular, this model performs equally well on protrusive epitopes and planar epitopes which are hardly addressed by existing models. Furthermore, it can also detect multiple epitopes whenever they exist.

In summary, we have comprehensively analyzed the property of multi-interfaces from both structural and functional perspectives, and have built both antibody-specific and antibody-agnostic epitope prediction models which are consistent with the principle of context-awareness. We observe that multi-interface proteins are ubiquitous, consolidating the principle of context-awareness. Both the association-based and the coupling graph-based antibody-specific epitope prediction models are effective, and the graph based antibody-agnostic epitope prediction model significantly improves prediction performance by identifying one or multiple epitopes.

Author's Publications

- **Liang Zhao** and Jinyan Li* (2009). *Sequence-based B-cell epitope prediction by using associations in antibody-antigen structural complexes*, BIBM CSBW.
- **Liang Zhao** and Jinyan Li* (2010). *Mining for the antibody-antigen interacting associations that predict the B cell epitopes*, BMC structural biology, 10(Suppl 1): S6.
- **Liang Zhao**, Limsoon Wong and Jinyan Li* (2011). *Antibody-specified B-cell epitope prediction in line with the principle of context-awareness*, IEEE/ACM TCBB, 8(6):1483-94.
- **Liang Zhao**, Steven C.H. Hoi*, Limsoon Wong, Tobias Hamp and Jinyan Li* (2012). *Structural and functional analysis of multi-interface domains*, PLoS ONE, 7(12): e50821.
- **Liang Zhao**, Limsoon Wong, Lanyuan Lu, Steven C.H. Hoi* and Jinyan Li* (2012). *B-cell epitope prediction through a graph model*, BMC Bioinformatics, 13(Suppl 17):S20.
- **Liang Zhao**, Limsoon Wong, Zhenhua Li, Hung Nguyen, Steven C.H. Hoi* and Jinyan Li*. *Coupling graphs, efficient algorithms and application in bioinformatics*, (under revision).

* Corresponding author.

Abbreviations and Symbols

Ab	Antibody
Ag	Antigen
ANN	Artificial Neural Network
ASA	Accessible Surface Area
CDR	Complementarity-determining region
C_H	Constant domain of antibody heavy chain
C_L	Constant domain of antibody light chain
Fab	Antigen-binding fragment
FN	False Negative
FP	False Positive
FR	Framework region
HMM	Hidden Markov Model
IEDB	Immune Epitope Database
Ig	Immunoglobulin
MCL	Markov CLustering algorithm
PDB	Protein Data Bank
SCOP	Structural Classification of Proteins
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
V_H	Variable domain of antibody heavy chain
V_L	Variable domain of antibody light chain

Contents

Acknowledgments	1
Abstract	3
Author's Publications	7
Abbreviations and Symbols	8
List of Tables	13
List of Figures	15
1 Introduction	19
1.1 Background	19
1.1.1 Antigen and its Epitope	20
1.1.2 Antibody and its Paratope	22
1.1.3 Multi-interface Protein and Context-awareness	25
1.2 Motivation and Objective	28
1.3 Main Contributions	31
1.4 Thesis Organization	32
2 Literature Review	35
2.1 Epitope Definition	35
2.2 Data Sets for Epitope Prediction	36
2.3 Continuous Epitope Prediction	38
2.4 Discontinuous Epitope Prediction	47
2.5 Metrics for Evaluation	52

3	Multi-Interface Domain Analysis	55
3.1	Background	55
3.2	Materials and Methods	60
3.2.1	Compilation of multi-interface proteins	60
3.2.2	Aggregation of multi-interface proteins	64
3.2.3	Construction of interface graphs	65
3.2.4	Mining structural patterns from interfaces	65
3.3	Results	67
3.3.1	Distributions of multiple interfaces	67
3.3.2	Analysis on multiple interfaces	71
3.3.3	Profiling of cross-domain multiple interfaces	79
3.4	Summary and Discussion	81
4	Association-based Antibody- specific Epitope Prediction	83
4.1	Motivation	83
4.2	Materials and Methods	87
4.2.1	Data set	87
4.2.2	Model construction	87
4.2.3	Sequence-based epitope prediction	91
4.3	Experimental Results	98
4.3.1	Resides relative composition and cooperativity	98
4.3.2	Associations in paratope-epitope interacting pairs	101
4.3.3	Performance of antibody-specific epitope prediction	103
4.3.4	Simplifying ABepar for antibody- <i>ignored</i> epitope prediction . . .	105
4.4	Discussion	109
4.5	Summary	111

5	Coupling Graph and Antibody- specific Epitope Prediction	113
5.1	Background	113
5.2	Related Works	117
5.3	Definitions and Algorithms for Mining Frequent Coupling Subgraphs .	119
5.3.1	Transformation of a coupling graph into a generic graph	120
5.3.2	Restoration of coupling graph from transformed generic graph .	122
5.3.3	Frequent coupling subgraph mining	122
5.3.4	Transformation and restoration with duplicate node labels . . .	124
5.4	Efficiency Results on Real Protein Complex Data	125
5.4.1	Coupling graph database compilation	125
5.4.2	Efficiency results	126
5.4.3	Irrelevant frequent subgraphs generated by gSpan	127
5.4.4	Statistics on the frequent coupling subgraphs	128
5.4.5	Scalability	129
5.4.6	Discussion	131
5.5	Application: Pattern Discovery and Epitope Prediction	131
5.5.1	Frequent coupling subgraph patterns in Ab-Ag complexes	132
5.5.2	Epitope prediction using frequent coupling graphs in Ab-Ag complexes	132
5.6	Conclusion	134
6	Graph-based Antibody-agnostic Epitope Prediction	137
6.1	Background	137
6.2	Materials and Methods	141
6.2.1	Collection of antigen protein data	141
6.2.2	Construction of epitope prediction model	142
6.2.3	Prediction of epitopes for an unknown antigen	148
6.3	Results and Discussions	148
6.3.1	Significant improvement of prediction accuracy	148

6.3.2	Locating epitopes with planar formations	151
6.3.3	Identifying multiple epitopes from an antigen	153
6.3.4	Graphical triplet patterns for epitopes	156
6.4	Summary	158
7	Summary and Future Work	159
7.1	Summary	159
7.2	Future Work	162
7.2.1	Epitope prediction with low- and high-level features	162
7.2.2	Paratope construction through context-awareness	163
7.2.3	Vaccine design with the assistance of epitope prediction	164
7.2.4	Coupling graph mining with concise graph transformation	165
	References	166

List of Tables

1.1	Amino acids classification	22
1.2	CDR definition	25
2.1	Data sets for epitope prediction	38
2.2	Summary of continuous epitope prediction approaches	46
2.3	Summary of discontinuous epitope prediction approaches	53
3.1	Multi-interface protein distribution	68
3.2	Distribution of multi-interface domains	70
3.3	Preference of multi-interface protein interactions	71
3.4	Top ten multi-interface domains with the largest numbers of proteins .	72
3.5	Number of isomorphic fingerprints between different interfaces	74
4.1	Proportion of each CDR's contribution in antigen binding	100
4.2	Top preferences between paratope and epitope	101
4.3	Top dependencies of paratope-epitope interacting residue pairs	102
4.4	Antibody-specific epitope prediction performance of ABepar	106
4.5	Epitope prediction performance comparison	107
5.1	The numbers of frequent coupling subgraphs	129
6.1	F-score <i>t</i> -test <i>p</i> -values between BeTop, DiscoTope, SEPPA and ElliPro	150
6.2	The averaged performances comparison on epitope prediction	150
6.3	The detail performance of BeTop	151
6.4	BeTop performance on multiple epitopes prediction	154

6.5	Top ten features favor of epitopes and non-epitopes	157
6.6	Selected features for epitope and non-epitope classification	158

List of Figures

1.1	The general structure of amino acid	21
1.2	Formation of discontinuous and continuous epitopes	23
1.3	Stretch size distribution of conformational epitopes	24
1.4	The X-ray crystallography of an antibody molecule	25
1.5	Example of the six CDRs	26
1.6	Epitope similarities versus CDR mutations	28
1.7	Multiple interfaces and context-awareness	29
1.8	Protrusive and planar epitopes	30
1.9	The logic and main contents of the thesis	34
2.1	Flowchart of linear epitope prediction	39
2.2	Flowchart of discontinuous epitope prediction	47
3.1	Multi-interface domain illustration	56
3.2	Multiple interfaces in the catalytic domain of plasmin	57
3.3	Flowchart of multi-interface protein data set construction and data analysis.	60
3.4	Multi-interface domain distribution	69
3.5	Amino acid distributions of interfaces in Ig VH domain	73
3.6	Amino acid distributions of interfaces of proteasome β -subunit domain	73
3.7	Example of a fingerprint in the proteasome β -subunit domain	74
3.8	Example of cooperative fingerprints	75
3.9	Fingerprints assortativity of Ig VH domain and Ig VL- κ domain	78
3.10	Example of a cross-domain interface	80

3.11	Distribution of cross-domain interfaces	80
4.1	Diagram of association based epitope prediction	86
4.2	Association-based model construction and epitope prediction	88
4.3	First order hidden Markov model for epitope prediction	96
4.4	Paratope residues' relative compositions	99
4.5	Epitope residues' relative composition	99
4.6	Log odds ratio of residues' cooperativity in epitope and H3	100
4.7	Example of association between a paratope and an epitope	103
4.8	Performance of ABepar on 80 Ab-Ag complexes	105
4.9	Pairwise sequence similarity of all the 80 antigen sequences	105
4.10	Example of epitope reconfiguration	109
5.1	Examples of coupling graphs in bioinformatics	115
5.2	An example of residue patterns in antibody-antigen interaction	116
5.3	Frequent subgraphs and frequent coupling subgraphs of a graph data set	117
5.4	Example of coupling graph transformation	121
5.5	Running time comparison on frequent coupling graph mining	127
5.6	Memory consumption comparison on frequent coupling graph mining .	127
5.7	Comparison of the number of connected subgraphs	128
5.8	Examples of frequent coupling subgraph patterns	129
5.9	Time cost in scalability exploration	130
5.10	Memory consumption in scalability exploration	130
5.11	Performance comparison on antibody-specific epitope prediction	134
5.12	An example of coupling graph in antibody-antigen complex	135
5.13	Frequent connected coupling subgraphs used for epitope prediction . . .	135
6.1	A hydrophilic island with a hydrophobic core in an antigen	139
6.2	An example of a multiple-epitope antigen	140
6.3	Pair-wise antigen sequence similarities distribution	142

6.4	Diagram of an antigen surface graph	143
6.5	Epitope prediction performance comparison	149
6.6	Comparison of prediction performance in terms of AUC value	149
6.7	Performances comparison at different levels of non-planarity	153
6.8	Example of multiple epitopes	155
6.9	Features that favor the epitope class	155
6.10	Examples of triplet feature patterns rich in the epitopes	157
6.11	Negative features enriched in non-epitope clusters	157
7.1	Flowchart of paratope construction	164
7.2	An example of full and concise graph transformation	165

Chapter 1

Introduction

1.1 Background

The immune system is a layered defense system that protects organisms from infections. In vertebrates, the system is comprised of the physical barriers, innate immune system and adaptive immune system with increasing specificity. The physical barriers prevent bacteria or viruses from entering an organism, such as the skin, respiratory tract and gastrointestinal tract. In case pathogens breach the physical barriers of an organism, the innate immune system will be activated immediately to perform dominant and non-specific responses. Furthermore, if pathogens escape recognition by the innate immune response, the adaptive immune system with high specificity acts as the last line of defense against invasions. Among the components of the adaptive immune system, B-cell plays a critical role in recognizing pathogens, presenting pathogens and memorizing pathogen signatures. However, it is yet to be fully understood as to how B-cells recognize pathogens and the way through which they distinguish self proteins from foreign proteins or infected self proteins. Therefore, a thorough understanding of the mechanism of the B-cell immune response is important to comprehend the basis of immune response, and is also beneficial for drug development and vaccine design. The following part provides a brief introduction of antigens, antibodies and the basic recognition mechanism between them.

1.1.1 Antigen and its Epitope

In the field of immunology, an ***antigen*** is defined as a molecule that can be specifically recognized by an antibody. This definition can be extended to any molecules or molecular fragments that are presented by a major histocompatibility complex (MHC) to a T-cell receptor [1]. Many types of molecules can serve as antigens including lipids, nucleic acids, proteins, and polysaccharides [2]. Among the various kinds of antigens, protein plays the most important role in inducing immune response. Thus, in this study we focus on exploring the basis of antibody-antigen recognition with antigen type of protein by computational methods. A ***protein*** is a large biological molecule consisting of single or multiple chains of amino acids. Proteins play the essential and fundamental role of maintaining biological functions. For ease of presentation, an antigen is always referred to as the type of protein without particular specification in this report hereinafter.

An antigen is linked by *peptide bonds*, formed by the dehydration synthesis reaction, between parts of or all of the twenty standard amino acids in nature. These amino acids are arginine (ARG, R), lysine (LYS, K), asparagine (ASN, N), aspartate (ASP, D), glutamine (GLN, Q), glutamate (GLU, E), histidine (HIS, H), proline (PRO, P), tyrosine (TYR, Y), tryptophan (TRP, W), serine (SER, S), threonine (THR, T), glycine (GLY, G), alanine (ALA, A), methionine (MET, M), cysteine (CYS, C), phenylalanine (PHE, F), leucine (LEU, L), valine (VAL, V), and isoleucine (ILE, I). The three-letter codes and one-letter codes in the parentheses are the abbreviations of the corresponding amino acids. Amino acids that release hydrogens and a hydroxyl groups during the dehydration synthesis reaction are usually called residues. We use the terms “amino acid” and “residue” interchangeably in this report. For the twenty different standard amino acids, the same frame is shared, as shown in Figure 1.1 (courtesy of YassineMrabet [3]). The amino group, the carboxylate group and the α -carbon are the same for all twenty standard amino acids. The side chain group R , which attaches to the α -carbon, differs from each other. Based on the different physicochemical properties of the R -group, amino acids can be classified into

Figure 1.1: The general structure of amino acid

several groups: positively charged (basic), negatively charged (acidic), polar (overall uncharged, but has uneven charge distribution, which can form hydrogen bonds with water) and non-polar (overall uncharged and uniform charge distribution, and cannot form hydrogen bonds with water) [4, 5, 6]. Alternatively, other classification schemes have been proposed based on different measurements [7, 8]. These properties, especially hydrophobicity and polarity, have been widely used to predict epitopes of antigens [9, 10, 11, 12, 13, 14]. The detailed classification is shown in Table 1.1

An *epitope* is defined as a cluster of residues of an antigen which can be recognized by antibodies [15]. Due to polypeptide folding, the residues of an epitope do not have to connect with each other in the primary sequence. Therefore, epitopes are usually categorized into continuous (linear) epitopes and discontinuous (conformational) epitopes [16, 17]. A continuous epitope consists of residues that are contiguous in the primary sequence of an antigen, while a discontinuous epitope is formed by residues that are separated, sometimes far away, from each other in the primary sequence but are close to each other in 3D space through polypeptide folding (see Figure 1.2 for illustration). It is reported that more than 90% epitopes belong to the category of discontinuous epitopes [18]. The classification of epitopes into continuous epitopes and discontinuous epitopes is not as precise as it seems to be. On one hand, some residues within a continuous epitope are indifferent in antibody binding; on the other hand, a discontinuous epitope is usually composed of several stretches of contiguous residues [19]. The stretch size distribution of discontinuous epitopes is shown in Figure 1.3. The average size is 2.23 ± 1.8 with the maximum length 12.

Table 1.1: Classification of amino acids according to their side chain properties

AA	hydropathy index	charge	polar	aromatic	aliphatic
R	-4.5	positive	✓	-	-
K	-3.9	positive	✓	-	-
N	-3.5	-	✓	-	-
D	-3.5	negative	✓	-	-
Q	-3.5	-	✓	-	-
E	-3.5	negative	✓	-	-
H	-3.2	positive	✓	✓	-
P	-1.6	-	-	-	-
Y	-1.3	-	✓	✓	-
W	-0.9	-	-	✓	-
S	-0.8	-	✓	-	-
T	-0.7	-	✓	-	-
G	-0.4	-	-	-	-
A	1.8	-	-	-	-
M	1.9	-	-	-	-
C	2.5	-	-	-	-
F	2.8	-	-	✓	-
L	3.8	-	-	-	✓
V	4.2	-	-	-	✓
I	4.5	-	-	-	✓

1.1.2 Antibody and its Paratope

One important type of protein of the adaptive immune system that neutralizes pathogens is the *antibody*. The term “antibody” was first clearly demonstrated by Maurice Nicolle under an imbroglio in which many terms were used to address antitoxicity [20]. There are two types of antibodies: B-cell surface antibody and B-cell secreted antibody. Surface antibody has a tail anchored into the membrane of the B-cell which can transduce signals to the cell upon interaction with a pathogen. As a result, a huge volume of specific antibodies are secreted from B-cells to neutralize that particular pathogen. Antibody is an efficient and powerful weapon to eliminate antigenic instances, which can bind to almost all the biological molecules ranging from simple intermediary metabolites to very complicated protein complexes. Although a tremendous number of pathogens can be neutralized by antibodies, the interaction between

Figure 1.2: Formation of discontinuous and continuous epitopes. Green chains represent antibody heavy chain and light chain, while purple chains represent antigens. The orange stretches of antigens are epitope peptides.

antibody and pathogen is very specific. Research has shown that the great specificity between antibody-pathogen interactions guarantees the sharp discrimination between self cells and foreigners with tremendous accuracy [2].

The specific binding between the antibody and pathogen is essential for recognizing invasions and tolerating self proteins, but it also suggests that the entire volume of antibodies should be very huge in order to neutralize various pathogens. It is estimated that more than 2×10^7 kinds of antibody-combining sites exist due to the recombination of variable (V), diversity (D) and joining (J) segments (antibody light chain only contains V and J segments) [21]. Although the entire antibody repertoire has an extraordinarily large number, all the basic units share the same frame as shown in Figure 1.4. Each antibody basic unit is comprised of two identical heavy chains (green chains in Figure 1.4) and two identical light chains (orange chains in Figure 1.4), in which two symmetric subunits are formed with each containing one heavy chain and one light chain. The two symmetric subunits (the surface rendered structure and the ribbon rendered structure in Figure 1.4) then form the “Y” shape structure of the antibody (see Figure 1.4). The constant fragment (Fc) of an antibody is universally shared by all antibodies, while the variable fragment differs across all

Figure 1.3: The stretch size distribution of 80 non-redundant discontinuous epitopes.

kinds of antibodies.

The binding site of an antibody that recognizes the antigen is mainly resided in the Complementarity Determining Regions (CDR) of the variable region [19]. There are six CDRs within an antibody basic unit—L1, L2, L3 are from the antibody light chain, and H1, H2, H3 are from the antibody heavy chain [22]. Various approaches have been proposed to define the CDRs of an antibody, including the Kabat definition [22], Chothia definition [23] and contact definition [24]. The Kabat definition is based on sequence variability of multiple aligned antibody sequences; the Chothia definition detects the loop region as CDRs based on structural alignments of antibodies; and the contact definition identifies CDRs as the residues involved in antigen binding. Among these definitions, the Kabat one is the most widely used due to its simplicity and satisfactory accuracy. However, the Kabat CDR definition is sensitive to the sample size and sequence alignment quality [25]. Therefore, “canonical structures”, which are the common main chain conformations, are used to refine the CDR definition [23]. The Kabat and Chothia CDRs definitions based on the Chothia numbering scheme are shown in Table 1.2. Figure 1.5 shows an example of the six CDRs of the PDB 1TQB.

Figure 1.4: The X-ray crystallography of an antibody molecule with PDB ID of 1IGT. Antibody heavy chains are colored in dark forest, while antibody light chains are colored in orange. One subunit is rendered as surface while the other is rendered as ribbon.

The set of residues on an antibody that interact with a specific antigen is defined as the *paratope*. Paratope residues are mainly located in the six CDRs of an antibody.

1.1.3 Multi-interface Protein and Context-awareness

A protein can interact with different proteins at non-overlapping binding sites synchronously or asynchronously. For example, the prion protein can interact with two

Table 1.2: Kabat and Chothia CDR definitions based on Chothia numbering scheme

light chain			heavy chain		
CDR	Kabat	Chothia	CDR	Kabat	Chothia
L1	L24–L34	L24–L34	H1*	H31–H35	H26–H32
L2	L50–L56	L50–L56	H2	H50–H65	H52–H56
L3	L89–L97	L89–L97	H3	H95–H102	H95–H102

* H1 should be H31–H35B and H26–H32..34 based on Kabat numbering scheme, where the “B” in “H35B” is the insertion position starting from “A”.

Figure 1.5: An example of the six CDRs of the antibody in PDB 1TQB.

different antibodies at two distinctive binding sites as shown in Figure 1.7. A protein with “multiple interfaces” is defined as a “*multi-interface protein*”. The phenomenon of multiple interfaces on a protein has been reported in several studies although they are slightly different in meaning. These include the “multi-interface hub” defined by Kim to illustrate a protein interacting with multiple proteins synchronously or asynchronously [26]; the “multibinding protein interface” defined by Tyagi to describe the same interface that has interactions with several partners asynchronously [27]; and the “multi-ligand interface” defined by Dasgupta to depict the union of overlapping interfaces on a protein [28]. To this study, the term “multi-interface” is slightly different from “promiscuous” [29] in which the former excludes the interface having multiple partners but without another distinguishable companion.

Antigens, which are restricted as proteins, can have multiple epitopes. The epitope of an antigen can sometimes be reconfigured with a different set of residues when the interaction changes to be with another antibody. This is known as “*context-awareness*” in the recognition between antigens and antibodies [30]. Thus, epitopes and paratopes are inter-dependent entities identifiable only by their interactions.

An epitope is an extrinsic feature of a protein. As mentioned, this feature can be recognized only when one antigen interacts with a specific antibody. Some researchers

have also argued that structural analysis skews the study of protein antigenicity and the fourth dimension should be considered to render the functional analysis of protein antigenicity [31, 19]. A more sophisticated facet of the binding mechanism is that a cluster of residues that act as an epitope under one set of circumstances will not necessarily behave as an epitope under another set of conditions [30]. That is, the epitope of an antigen may be reconfigured with a different set of residues when different antibodies are in the interaction. Therefore, a coarse binary classification of antigen regions into epitopes or non-epitopes without specifying antibodies and paratopes may not accurately reflect biological reality [30].

Upon interaction the epitope structure may undergo induced adaptation [32] due to the conformational plasticity. Thus, the reconfiguration of epitope residues is possibly solely caused by the change of antibodies. To check to what extent that the change of antibody affects the epitope reconfiguration, an experiment is conducted on a set of 311 non-redundant antibody-antigen complexes, shown in Figure 1.6. Surprisingly, with at least one single residue mutation on CDRs of antibodies, 99.3% corresponding epitopes reconfigured to another one with overlapping less than 50%. This means: (i) one epitope almost only interact with one kind of antibody and, (ii) another multi-interface epitope can hardly be accommodated in the very close region of one existing epitope. It also implies that the antigen structure, e.g., disorderability, has negligible impact on inducing multi-interface epitopes. In another word, the conformational plasticity has marginal effect on epitope reconfiguration.

An example of this context-awareness is shown in Figure 1.7. This antigen—prion protein—can interact with two antibodies. The epitope residues within PDB:1TQB are G127, L128, N156, Y158, R159, I185, K188, Q189, T191, V192, T193, T194, T195, T196, K197, G198 and E199, while epitope residues identified in PDB:2W9E are G145, S146, D147, Y148, D150, R151, Y152, R154, E155, N156, H158, R159, N200, T202 and K207. (The numbering in PDB:2W9E has been shifted in alignment to the antigen numbering in PDB:1TQB). It is clear that there are only two epitope residues (N156 and R159) overlapping between these two epitopes.

Figure 1.6: Scatter plot of pair-wise epitope similarities versus the number of mutations between the corresponding CDRs.

1.2 Motivation and Objective

Epitopes are indispensable for immune response, and evasion of recognition by the immune system could induce malfunction and disorder. Therefore, identifying epitopes within an antigen is critical for vaccine design, disease diagnosis and disease therapy. As the cornerstone of these applications, epitope identification has attracted intensive efforts in the past few decades on both continuous and discontinuous epitope predictions.

Conventionally, a continuous epitope is identified by averaging a particular property scale within a slide window along the antigen sequence, and epitope residues are deemed as those with very high or low scores. Usually, these properties include the residue's hydrophobicity, back bone flexibility, residue's antigenicity [12, 13, 33], and so on. As for the discontinuous epitope, it is determined by a single feature or combination of features extracted from antigen sequences and structures, such as accessible surface area, residue contact, and geometrical formation [34, 35, 36, 37, 38, 39, 40]. Besides, sophisticated machine learning approaches have been employed to tackle this problem, including the artificial neural network, Bayesian network, support vector

Figure 1.7: Multiple interface and context awareness of an antigen interacts with two different antibodies. The epitope residues of the antigen (prion protein) that interact with an antibody (PDB ID: 2W9E) are of color purple, while the epitope residues of the antigen interacting with another antibody (PDB ID: 1TQB) are colored in orange. The forest green structure is the six CDRs of the antibody with PDB entry 2W9E, and the blue colored structure is the six CDRs of the antibody with PDB entry 1TQB. The antigen is rendered in black ribbon, while the antigen binding fragments of each antibody is rendered in transparent surface. The two antibody-antigen complexes are superimposed based on the antigen structure.

machine, random forest, and kernel methods [41, 42, 14, 43, 33]. With these tremendous efforts, both continuous and discontinuous epitope predictions have advanced significantly. However, many limitations still exist in the available methods, and there remains much room for improvement in the performance.

Firstly, the existing methods predict all the antigenic residues of an antigen as a single epitope without distinguishing which antibody the epitope belongs to. These methods ignore the fact that multiple epitopes may exist in an antigen and antibody-antigen interaction involves context-awareness; thus, the predictions may not reflect the biological reality. For example, the conventional methods predict the two different epitopes of the antigen, shown in Figure 1.7, as a single epitope although they are markedly different from each other. Therefore, the results generated by the existing

Figure 1.8: Protrusive and planar epitopes on ubiquitin. Epitopes are colored by pink and sandy brown, respectively.

methods are just the combination of epitopes instead of an epitope itself.

Secondly, the existing methods using geometrical properties to identify epitope residues prefer epitopes with protrusive formations, and do not identify epitopes with planar shapes. These properties include protrusive index, contact number, and accessible surface area. Intuitively, protrusive epitopes have distinctive values on these properties and are thus easier to identify. However, epitopes may not be protrusive at all. In fact, epitopes can have various geometrical formations with a large range of non-planarities. For example, the non-planarity (measured by root mean square deviation) of the epitopes colored in pink and sandy brown, as shown in Figure 1.8, are $1.04\text{\AA}$ and $3.14\text{\AA}$, respectively. Hence, the epitope with pink color is too flat to be detected by the existing approaches.

Last but not least, the existing models perform poorly in epitope identification, especially in discontinuous epitope prediction. The best AUC (area under the curve) value is reported as 0.644 for discontinuous epitope prediction [40], and this result is far from satisfactory. Therefore, more efforts are needed to significantly improve the prediction performance.

The ultimate goal of this research is characterizing and predicting B-cell epitopes. In detail, our main objectives are: (i) characterization of B-cell epitopes, including exploring properties of multiple interfaces of proteins and discovering distinctive patterns of epitopes; (ii) identification of antibody-specific B-cell epitopes that are in

line with the principle of context-awareness; and (iii) prediction of antibody-agnostic B-cell epitopes with one or multiple epitopes whenever they exist, that are consistent with the principle of context-awareness.

1.3 Main Contributions

This research is mainly focused on characterizing and predicting B-cell epitopes, and the specific contributions are as follows:

- (i) Systematically analyze multi-interface domains of proteins from both structural and functional perspectives. Antigen interacting with antibody constitutes context-awareness, and multiple interfaces exist in an antigen. Therefore, we comprehensively analyzed the distributions of multi-interface domains and proteins, the structural fingerprints of interfaces of multi-interface domains, the competitive and cooperative relations between interfaces of a multi-interface domain, and the association between multiple interfaces and their biological functions.
- (ii) Development of the antibody-specific B-cell epitope prediction model incorporating two-dimensional association and the Hidden Markov Model. Two-dimensional association—viz., residues-residues pairing preference and dependence between contact residue pairs—is introduced in this study to explore the context dependence principle between antigen and antibody. By using two-dimensional association to seed candidate epitope residues together with the Hidden Markov model to complete epitope identification, we have successfully identified antibody-specific B-cell epitopes. Besides, the proposed model is trained on antibody-antigen complexes, but it requires antibody-antigen sequence complexes as inputs to identify epitopes, which has greater applicability.
- (iii) Design of a new algorithm to mine frequent coupling subgraphs, which are further applied to identify antibody-specific B-cell epitopes. A coupling graph is

more appropriate to represent the inherent context-awareness principle between paratopes and epitopes; however, it is very challenging to mine frequent coupling subgraphs. In this study, we designed an efficient algorithm to mine frequent coupling subgraphs by graph transformation. By using the frequent coupling subgraphs in antibody-antigen complexes and the Hidden Markov Model, we achieved better performance on antibody-specific B-cell epitope prediction. In addition, the new algorithm is not only suitable for antibody-specific epitope prediction but is also applicable for pairing structural motifs identification and binding partner engineering.

- (iv) Construction of a graph-based model to predict antibody-agnostic B-cell epitopes. A novel framework is proposed to identify generic B-cell epitopes, which is constituted by antigen surface triangulation, edge weight calculation, graph clustering and subgraph classification. This model is superior to existing antibody-agnostic epitope prediction models in several aspects: (i) it significantly improves prediction performance; (ii) it performs equally well on both protrusive and planar epitopes, while existing models can hardly identify planar epitopes; (iii) it can identify multiple epitopes whenever they exist, while conventional models mix all antigenic residues as a single epitope.

1.4 Thesis Organization

This thesis consists of seven chapters. Chapter 1 introduces the immediately basic concepts, including the antibody, paratope, antigen, epitope, multi-interface protein and context-awareness. The motivations are presented subsequently, and it is followed by our objectives and main contributions of this research.

Chapter 2 mainly reviews the previous in-depth works on both continuous epitope prediction and discontinuous epitope prediction. The prevalent data sets and evaluation metrics are also described.

In chapter 3, the property of multi-interface, which is the basis of the context-awareness principle, is comprehensively explored. This includes profiling multi-interface domains distribution at different structural levels, identifying fingerprints of interfaces, examining relations between multiple interfaces, and uncovering associations between multi-interfaces and their functions.

In Chapter 4, a two-dimensional association that echoes the principle of context dependence together with its application in antibody-specific epitope prediction is explained. The two-dimensional association consists of residues-residues pairing preference and dependence between sets of contacting residue pairs. To predict antibody-specific B-cell epitopes, the two-dimensional association seeds candidate epitope residues at the first stage, while the Hidden Markov Model completes epitope identification at the second stage.

Chapter 5 completes the representation of context-awareness with a coupling graph on antibody-specific epitope prediction together with the introduction of an innovative algorithm for frequent coupling subgraph mining. With the novel idea of graph transformation, we are able to effectively mine frequent coupling subgraphs from a coupling graph database. Based on the frequent coupling subgraphs of a set of antibody-antigen complexes, a model is built to identify seed epitope residues. This is followed by epitope completion with the Hidden Markov Model being used to complete epitope identification.

Chapter 6 introduces a graph-based antibody-agnostic B-cell epitope prediction model. This model incorporates statistical ideas, graph clustering algorithm and graph classification approach to determine one or multiple epitopes from an antigen that is consistent with the principle of context-awareness. Intensive results are reported as well.

In Chapter 7, we summarize this research and propose some interesting topics that we could explore in future.

The main body of this thesis contains chapter 3, chapter 4, chapter 5 and chapter 6. Chapter 3 explores the context-awareness principle, while chapter 4 and chapter

Figure 1.9: The logic and main contents of the thesis.

5 aim at predicting antibody-specific epitopes which coincide with this principle. In chapter 6, a new model is designed to identify antibody-agnostic epitopes which also satisfies the principle. Figure 1.9 shows the logic and the main contents of these four chapters.

Chapter 2

Literature Review

B-cell epitope prediction has attracted much attention since the late 80s because of its great potential in therapeutic and prophylactic applications [44, 45, 46, 47]. In this section, the two types of epitope prediction approaches, i.e., continuous epitope prediction approaches and discontinuous epitope prediction approaches, are reviewed. Various epitope definitions, commonly used data sets and popular performance evaluation measurements are also reported.

2.1 Epitope Definition

Various computational methods have been proposed to define an epitope, including:

- Lost ASA $\geq 1\text{\AA}^2$ of antigen residue upon binding with antibody [36, 34];
- Distance $\leq 4\text{\AA}$, $5\text{\AA}$, or $6\text{\AA}$ between antigen and antibody [35, 38, 48, 49];
- Residue contact types of antigen-antibody interacting complex [50].

Epitope defined by ASA. ASA is the short form of accessible surface area which was first coined by [51]. ASA is typically calculated by the “rolling ball” algorithm developed by [52]. More specifically, a particular ball with a specific radius is used to probe the surface of a molecule and the tracking area of the probe center is considered as the accessible surface area. Usually, an antigen residue is deemed as an epitope

residue if its accessible surface area loses more than $1\text{\AA}^2$ after binding with an antibody. Several epitope prediction approaches have adopted this definition to determine epitope residues, such as Rapberger’s approach [36] and Kulkarni’s approach [34].

Epitope defined by atom-atom distance. Atom-atom distance is calculated based on the Euclidian distance of two heavy atoms (excluding hydrogen), whereby one is from an antigen and the other is from an antibody, with three-dimensional coordinates. Hydrogens are usually excluded from the epitope definition because they are not reported in most of the cases. The typical distance threshold used to define epitope residues is $4\text{\AA}$ [35, 38]. Other choices, such as $5\text{\AA}$ [48] and $6\text{\AA}$ [49], have also been used to define epitope residues.

Epitope defined by residue contact type. Epitope residues defined by types of residue contacts include hydrogen bond, salt bridge, disulfide bridge, van der Waals interaction, water-mediated hydrogen bond and hydrophobic interaction [50].

Amongst these definitions, the distance threshold of $4\text{\AA}$ is the most preferable one. This is due to the simplicity and effectiveness of this definition as well as the negligible difference in epitope residues generated by various definitions [53].

2.2 Data Sets for Epitope Prediction

Large-scale carefully curated data sets play a pivotal role in computational model construction and validation, and inconsistent description of data aggravates the prediction performance of models and perplexes the comparison between different predictors [49]. Currently, several data sets are available for constructing epitope prediction models although the volume of the data, especially structural data, still remains scarce.

The earliest linear epitope data set, containing 82 epitopes, was collected by Pellequer [54]. These epitopes were extracted from 14 proteins which were well studied by many researchers; thus, it was widely used for linear epitope predictor training and testing.

Another epitope data set consisting of 67 antigens was constructed by Alix in 2000. The antigens homology within this data set was less than 50% [55].

The AntiJen database, mainly composed of continuous epitopes, was assembled by Toseland [56]. This database not only provides the antigenic peptides, it also presents their catalogues according to their relative observed immunodominance. In addition, residues which are important for antibody binding are recorded as well.

Another linear epitope database is Bcipep established by Saha in 2005 [57]. The immunogenic peptides, determined by experiments which have very high confidences, stored in this database are collected from publicly available literatures. In addition, this database covers a wide range of organisms including bacteria, viruses, and fungi. This database is not only useful for epitope model training and testing, but is also appropriate for analysis of epitope evolution.

The Epitome database [49] contains antigenic epitopes which are compiled from the protein data bank (PDB) [58] by a semi-automatic process. This database categorizes antibody structures according to the Structural Classification of Proteins database (SCOP) [59]. Epitope residues are inferred from antibody-antigen interacting complexes with structure information by the distance threshold of 6Å.

The Immune Epitope Database and Analysis Resource (IEDB) [60] is the most comprehensive epitope database. It includes both well annotated epitopes and non-epitopes. The significance of the negative data parallels the importance of the positive data for model construction. In addition to the functional assays, IEDB also maintains structurally inferred epitopes from the antibody-antigen complexes that are available in PDB [58].

AntigenDB is a recently developed database for antigens [61]. It contains more than 500 antigens coming from 44 important pathogenic species. In addition to the basic sequence and structural information of antigens, it also contains extensive information including epitope, function, gene expression and post translational modification, where available.

Very often, a database contains duplicate items and entries, thus Ponomarenko compiled two non-redundant data sets of discontinuous epitopes in 2007, with one containing of 62 antigen structures and the other containing 82 antibody-antigen

complexes [53]. This benchmark data set has been widely accepted for model training and testing.

The list of available data sets for B-cell epitope prediction is shown in Table 2.1.

Table 2.1: Data sets and their brief descriptions for the problem of epitope prediction

Resource	Ref.	Brief description
Pellequer	[54]	82 continuous epitopes extracted from 14 well studied proteins.
Alix	[55]	67 antigens with homology less than 50%.
AntiJen	[56]	Over 3,500 entries of both continuous and discontinuous epitopes with immunodominance catalogue.
Bcipep	[57]	Experimental assay determined linear epitopes with more than 3,000 entries.
Epitome	[49]	Computational approach determined epitopes from 150 antigen-antibody complexes with structural classification information.
IEDB	[60]	A comprehensive database contains $\geq 80,000$ B-cell epitopes and $\geq 62,000$ B-cell non-epitopes.
AntigenDB	[61]	More than 500 antigens with epitope, function, gene-expression, post translational modification information.
Ponomarenko	[53]	Benchmark data sets containing 62 representative 3D structures of antigens and 82 non-redundant 3D structures of antigen-antibody complexes with different structural epitopes.

2.3 Continuous Epitope Prediction

Epitopes can be categorized into continuous epitopes and discontinuous epitopes. Although this classification is untidy [19, 62] and less than 10% epitopes belong to the category of continuous epitopes [18], many researchers still focus on continuous epitope prediction. Several reasons account for this preference. Firstly, it is much easier to identify continuous epitopes; secondly, a large volume of continuous epitope data sets are available while it is very scarce for discontinuous epitopes; and thirdly, the antigenic peptides can be used to devise vaccines.

In the last few decades, amino acid propensities, protein secondary structures, protein tertiary structures and even machine learning approaches—such as Artificial Neural Network (ANN), Hidden Markov Model and Support Vector Machine (SVM)—were used to predict continuous epitopes.

Figure 2.1: Flowchart of Hopp and Woods linear epitope prediction model. Positions colored as yellow in the last step are considered as epitope residues.

A representative method for predicting continuous epitopes by using single property-scale—hydrophobicity—is proposed by Hopp and Woods [63]. This method assigns a hydrophobicity value to each residue, then averages the hydrophobicity value of each position within a slide window of six residues (three to the left and three to the right),

and finally determines the epitope residues as those positions with high hydrophobicity values. The flowchart of this method is shown in Figure 2.1. Hydrophobicity is a good feature to distinguish epitope residues from non-epitope residues to some extent, but many other features also take part in governing antibody-antigen interactions, e.g., polarity, hydrogen bond, flexibility, etc.

Different properties with the same strategy introduced by Hopp and Woods have been used to identify continuous epitopes. Kyte and Doolittle used their own hydrophobicity index to identify linear epitopes in 1982 [5]. Parker and Hodges designed a new method called high-performance liquid chromatography (HPLC) to calculate amino acid hydrophilicity, and then used this property to predict antigenic regions in an antigen sequence [10]. Evaluation carried out on seven protein antigens have shown that the new hydrophilicity parameters generated by HPLC produced the best accordance between hydrophilicity and antigenicity. Jameson and Wolf proposed a method of determining antigenic residues by combining the scales of accessibility, flexibility and secondary structure [64]. Later on, Pellequer *et al.* developed a new method for predicting epitope peaks by turning scales [54] with the shifting window of five. 87 high resolution ($\leq 2\text{\AA}$) X-ray crystallography proteins were used to train the model, and 14 extensively studied antigens were employed to test it. It was argued that the proposed approach achieved higher accuracy compared with the other existing methods (the correct prediction rate was about 70%). Alix presented a representative combination method called PEOPLE [55] to identify antigenic residues. PEOPLE linearly combined four attributes, i.e., secondary structure, hydrophilicity, accessibility [65] and flexibility [66, 67], to predict antigenic determinants of an antigen. Secondary structure was computed by GOR [68], hydrophilicity was obtained from [63, 10, 5, 69], surface accessibility was calculated by [70], and flexibility was determined by the tools reported in [66, 67]. Applications on tropoelastin and P30 of *Toxoplasma gondii* showed that the method achieved very good correlation between the predicted epitope residues and the known experimentally determined reference. Huang and colleagues developed an epitope identification model based on four amino

acid indices originated from residue network topology [71], i.e., relative connectivity, clustering coefficient, closeness and betweenness [72]. They argued that the network topology based indices achieved better performance than the best single property scale based models.

Extensive effort has been made on continuous B-cell epitope prediction by using physicochemical properties. However, Blythe and Flower have shown that the single propensity scales only perform marginally better than random guessing [73]. Therefore, several sophisticated machine learning approaches have been proposed.

BepiPred, a combinative method based on the Parker hydrophilicity scale [10] and the HMM, has shown significant improvements in linear epitope prediction [12]. This model was trained on a data set of 127 protein sequences extracted from AntiJen [56] with the parameter optimized on the Pellequer data set consisting of 82 well annotated epitopes. It was validated unbiasedly on a separate data set with 10 protein sequences retrieved from the HIV Molecular Immunology Database of the Los Alamos National Laboratory [74]. Results showed that the AUC value is as high as 0.671. This machine learning approach takes the advantage of learning from data, but it only incorporates a single feature. Thus, other methods proposed try to overcome this shortcoming.

ABCpred, another machine learning approach, was developed to predict continuous epitopes [41]. ABCpred was a recurrent artificial neural network based method with three layers that had a single output unit and 35 hidden units. The optimal input window size was determined to be 16. To fix the input sequence size, the peptide with more than 16 residues was truncated symmetrically in both the N- and C-terminals. Meanwhile, the peptide sequence with less than 16 residues was extended symmetrically in both the N- and C-terminals from the original protein sequence. A non-redundant data set containing 700 unique epitopes extracted from Bcipep [57] and 700 non B-cell epitopes retrieved from Swiss-Prot [75] were used in the evaluation. The accuracy of this model was 65.9% evaluated by five-fold cross validation. This method learns the prediction power from the fixed peptides; however, epitope

never has a fixed size on one hand, and non-epitope residues included weakens the prediction power on the other hand.

Epitope prediction model which overcomes the problem of using fixed length of epitope is also available, such as BCPred [42]. BCPred used subsequence kernel to evaluate the similarity between two strings. Based on a set of 701 homology-reduced epitopes retrieved from Bcipep [57] and an equal number of non-epitopes obtained from Swiss-Prot [75], BCPred showed the best predictive performance with AUC of 0.758. Subsequently, EL-Manzalawy extended their string kernel method to identify flexible length linear epitopes, called FBCPred [76]. This method achieves very good performance comparing with existing models, but the negative data set they used is not necessarily negative as cross-interaction may exist. One possible solution for fixing this problem is that removes the homologous chains from the negative data comparing with the positive data.

Sollner and colleagues developed a random forest method to identify continuous epitopes based on the residues' antigenicity, variability and post-translational modification [77]. By maximizing antigenicity, and minimizing variability as well as the likelihood of post-translational modification, they performed consistently better than existing machine learning approaches for half of the 57 curated epitopes downloaded from IEDB [60]. Machine learning approach is used in this model, which is trained on a very limited number of samples; thus, overfitting problem is hard to avoid. More data, especially negative data, should include to train the model.

In 2008, Rubinstein *et al.* constructed two naïve Bayes classifiers to discriminate epitopes from antigen sequences and antigen structures [14]. The classifier for epitope identification from antigen structures was trained on 47 non-redundant protein antigens extracted from the SPIN server [78]. The features used by this classifier include the amino acid's frequency ratios of A, R, C, H, L, F, S, T, V, secondary structure frequency ratio of Helices, patch average relative-accessibility to the solvent, patch average accessibility to a large probe (9Å), patch average curvature, and patch atoms' proportion of antigen hull within 4Å. For an input antigen with structures, the model

determined epitope residues as follows: (i) divide the antigen surface residues into patches with 20 residues; (ii) score each patch based on the Bayes predictor and assign this value to the central residue; and (iii) identify epitope residues with high values. The sequence-based classifier was trained and tested following the same strategy as the structure-based classifier. The slight differences are: (i) the sequence-based classifier was trained on 194 non redundant antigen sequences retrieved from Bcipep [57]; (ii) the antigen sequence was divided into stretches with lengths of seven instead of patches; and (iii) the properties obtained from the antigen structures were abandoned while new features from antigen sequences were used instead. Both structure-based and sequence-based classifiers were validated by leave-one-out cross validation, and they achieved successful prediction rate of 70.3% and 70.6%, respectively. This model has its own merits of bombing lots of features with Bayes classifier. However, it has the same problem with Sollner’s approach [77] that too many features are used while the number of sample is very small.

COBEpro, another continuous epitope predictor, was developed by Sweredoski and Baldi in 2009 [33]. This method built a two-stage prediction model to screen epitope residues. In the first stage, fragments of antigenic peptide were fished out from antigen sequences by using SVM; and in the second stage, secondary structure information and solvent accessibility were included to improve the specification of antigenic residues. The properties that input to the SVM were a vector of similarities between the query peptide and each of the positive peptides in the training library, thus the size of the input vector was equal to the number of positive peptides in the training repertoire. The similarity was calculated by the number of identical substrings between the query peptide and the given positive peptide. Cross-validation was used to evaluate the performance of COBEpro on several data sets, consisting of 872 epitope sequences from Bcipep [57], 701 redundancy-reduced sequences curated by El-Manzalawy *et al.* [42] as well as the same number of non-epitope sequences extracted from Swiss-Prot [75], 103 epitopes from 10 protein antigens curated by Korber *et al.* [74], and 83 epitope fragments of 14 antigens compiled by Pellequer *et al* [54]. Results have

shown that COBEpro achieved up to 0.829 of AUC on identifying antigenic fragments from antigen sequences and 0.628 of AUC on specifying antigenic residues. Moreover, COBEpro was free of input sequence length. One major limitation here is that the single amino acid change could make the whole epitope non-antigenic, indicating the library query is not always competent.

Wee and colleagues designed a SVM-based epitope prediction model, BayesB, using Bayes Feature Extraction (BFE) to classify linear B-cell epitopes with diverse lengths [79]. Features that used to distinguish epitopes from non-epitopes were the position-specific propensities of amino acids. Performance comparison on the data set in [42] has shown that BayesB achieved the best performance with AUC as high as 0.84. The length of epitope varies a lot, thus the position-specific propensities only work well on epitopes have similar size.

Another interesting approach for linear epitope identification is GFSMLP (Group Feature Selecting Multilayered Perceptron) [80]. GFSMLP incorporates hydrophilicity [10], accessibility [65], flexibility [66], surface exposed scale [81], polarity [82], turn [54], antigenicity [11] and Beta-turn [83] to predict linear epitopes. Intriguingly, they took all the twenty amino acids as a group on each feature to train the multilayered neural network. They found that polarity and Beta-turn were the most significant determinants in epitope identification while turn and surface exposed scale were the least preferred determinants. The observations here contribute a lot for advanced model construction, but the model itself has limitation on the large size of input vector against the small size of epitope. Therefore, few amino acids, instead of all the twenty amino acids, are possibly used to reduce the size of input vector.

Recently, Gao and co-workers devised a linear epitope prediction model, BEST (B-cell Epitope prediction using Support vector machine Tool), by using SVM [84]. The features used in this model are sequence conservation, similarity to known (training) epitopes, predicted secondary structure and relative solvent accessibility. The major difference between BEST and the existing linear epitope predictors is that BEST predicts epitopes from the whole antigen sequence while the existing machine learning

approaches classify based on epitope peptides. It is claimed that BEST outperforms several existing epitope prediction models. BEST has a good try on working with the whole antigen sequence. But the features chosen to achieve this goal is not good enough. Firstly, epitope changes frequently, which indicates the similarity search only including a small number. Secondly, the secondary structure prediction itself is not so accurate, thus the use of this feature is not very convincing.

The aforementioned machine learning approaches show plausible prediction power, but none of them provides easy-to-understand classification rules nor do they calculate the prediction confidence. Therefore, Söllner and Mayer used another machine learning approach, the decision tree, to predict epitopes along with explainable rules [13]. This method adopted hundreds of features to capture the antigenicity of a given peptide, such as amino acids propensities, secondary structures, and amino acid neighborhood propensities. Subsequently, these features were used to train a k-nearest neighbor classifier and a C4.5 decision tree classifier to predict continuous epitopes. Cross-validation performed on 1201 epitopes together with the same number of references showed superior performance of this model. Moreover, rules of classification and the confidence (post test probability (PTP)) of each epitope were also provided. The rules produced by this model are very useful to understand the biological reality in depth; however, the feature space is very large and it is difficult to provide essential rules that separate epitopes from non-epitopes.

In addition, bi-residues, tri-residues and even tetra-residues were used as features to distinguish epitope peptides from non-epitope peptides [85, 86].

Besides machine learning approaches, a mathematical model was also introduced to identify linear B-cell epitopes [87]. This model adopted mathematical morphology of eroding and dilating antigenicity scales of epitope residues. Based on two data sets, that one contains 691 epitopes collected from AntiJen [56] and the other contains 94 epitopes compiled by Pellequer [54], the proposed model showed much better performance on low-to-moderate antigenicity residues identification, but the performance was still very poor.

The summary of existing continuous B-cell epitope prediction models are shown in Table 2.2.

Table 2.2: Summary of continuous epitope prediction approaches

Resource	Ref.	Features	machine learning model
Hopp	[63]	hydrophlicity	-
Kyte	[5]	hydropathy index	-
Parker	[10]	hydrophilicity, accessibility, flexibility	-
Jameson	[64]	accessibility, flexibility, secondary structure	-
Pellequer	[54]	turn	-
PEOPLE	[55]	secondary structure, hydrophilicity, surface accessibility, flexibility	-
BepiPred	[12]	hydrophilicity	Hidden Markvo Model
ABCpred	[41]	-	recurrent neural network
Söllner	[13]	propensity scales, neighborhood propensities, sequence complexities	k-nearest neighbor, decision tree
Huang	[71]	residues' relative connectivity, clustering coefficient, closeness, betweenness	-
BCPred	[42]	string kernels	support vector machine
Söllner	[77]	antigenicity, variability and post translational modification	random forest
LE-LP/NLP	[87]	antigenicity	mathematical morphology
Rubinstein	[14]	residue frequency, helix frequency, relative solvent accessibility, curvature	Naïve Bayes
Wang	[85]	residue triplets	-
COBEpro	[33]	secondary structure, solvent accessibility	support vector machine
BayesB	[79]	residue's position specific propensity	support vector machine
Wang	[86]	residue pair, residue triplet, tetrapeptide	-
BEST	[84]	sequence conservation, sequence similarity to known epitopes, predicted secondary structure, relative solvent accessibility	support vector machine
GFSMLP	[80]	hydrophilicity, accessibility, flexibility, surface exposed scale, polarity, turn, antigenicity, Beta-turn	multilayered perceptron

2.4 Discontinuous Epitope Prediction

In contrast to tens of continuous epitope prediction models, only a few approaches are available for discontinuous epitope prediction. This gap is mainly caused by the fact that discontinuous epitopes are more complicated than continuous epitopes due to the nature of polypeptide folding. The state-of-the-art discontinuous epitope prediction models are outlined as follows.

Figure 2.2: Flowchart of discontinuous epitope prediction by CEP. Magenta residues are predicted as epitope residues.

CEP, or *conformational epitope prediction*, is a pioneering work of discontinuous epitope prediction made by Kulkarni-Kale *et al.* in 2005 [34]. It is a representative

method for predicting discontinuous epitopes by using the residue's accessibility and contact information. As depicted in Figure 2.2, when an antigen structure is given, CEP calculates the accessible residues in the first step, and then determines the antigenic residues with high accessible surface area in the second step. Antigenic determinants are then fished out from the antigenic residues in the third step, and finally the discontinuous epitopes are formed by extending and collapsing the antigenic determinants in a distance of 6Å. ASA can be used as a feature which is combined with other features to identify epitopic residues; however, not all of the residues with large ASA are antigenic residues. Besides, the rationale of residue extending and collapsing has seldom concrete data to support. One can use connecting weight to differentiate multiple epitopes instead of taking the union of all exposed residues.

Another discontinuous epitope predictor, DiscoTope, was developed almost at the same time [35]. DiscoTope was a combinative method comprising of smoothed amino acid's log-odds ratios based predictor and contact number based predictor. The log-odds ratio between epitope residue and non-epitope residue was calculated from a non-redundant data set with 76 antibody-antigen complexes constructed by the author, while the contact number of a given residue was determined by the number of surrounding residues within 10Å between their C_α atoms. Five-fold cross-validation carried out on the 76 antibody-antigen structural complexes has shown that DiscoTope produced good performance on discontinuous epitope prediction with sensitivity of 15.5% when the specificity was set to 95%. DiscoTope has improved performance comparing with CEP, but the performance is still far from satisfactory considering the substantial imbalanced distribution of epitope residues versus non-epitope residues.

A more sophisticated method developed by Ponomarenko *et al.* in 2008 was dubbed as ElliPro [37]. ElliPro took antigen structure as input, or antigen sequence if the structure was not available. If antigen sequence was used to predict discontinuous epitopes, then the corresponding antigen structure was modeled by MODELLER [88] or replaced by a homologue structure searched by BLAST [89] from PDB. When an antigen structure was available, ElliPro constructed an ellipsoid for the protein at first

[90], and then the protrusion index (PI) for each residue was calculated later [91]; at last neighborhood residues clustered agglomeratively with respect to their PI values. Evaluation performed on a benchmark data set compiled by Ponomarenko and Bourne [53] has shown that ElliPro achieved a very high AUC value of 0.732. ElliPro makes considerable advancement on discontinuous epitope prediction on protruding area of antigen, but it has a constitutional drawback that it only favours protrusive epitopes but not identifies planar epitopes.

Amino acid propensity scales together with half sphere exposure (both down and up) were linearly combined to predict discontinuous epitopes [38]. Half sphere exposure for a residue was measured as the number of residues within a specific distance between their C_{α} s. It was a simple but effective method according to the report. Similar to ElliPro, this model also only favours protrusive epitopes.

Epitopia was another web-server developed for identifying antigenic determinants [39]. This server was implemented based on a Naïve Bayes classifier that was described in [14] with the features of physicochemical and geometrical properties. For an input antigen, its surface residues were clustered into patches and the values of properties of each patch were calculated. Subsequently, each patch was classified as either an epitope or a non-epitope based on the property value. Epitopia predicts both continuous and discontinuous epitopes based on the input data. Epitopia classifies each residue patch as an epitopic patch or non-epitopic patch, which is very good starting point. However, the combination of all the antigenic patches as a single epitope overlooks the fact that multiple epitopes may exist on the same antigen.

A new feature—unit patch of residue triangle—was proposed by Sun [40] to predict discontinuous epitopes. For an input antigen with structural information, SEPPA calculated each residue’s solvent accessible surface area, and formed unit patch of residue triangles with Euclidian distance of 4Å. The antigenicity of a residue was the sum of averaged unit patch indices and the cluster coefficient of patches within 15Å, where the unit patch index is defined as the ratio of propensities between the epitope and non-epitope, and the cluster coefficient is the ratio between the number

of real contacts and the number of complete contacts of a cluster. Epitope residues were selected as the ones with large antigenicity indices. SEPPA was claimed to be better than CEP [34], DiscoTope [35] and PEPITO [38] according to the comparison. SEPPA also identifies antigenic patch of an antigen at first; however, it has the same problem of considering all the candidate residues as one epitope.

Liang designed a consensus scoring method, EPCES [92], to predict discontinuous B-cell epitopes based on six features: residue epitope propensity, conservation score, side-chain energy score, contact number, surface planarity score, and secondary structure composition. A residue was predicted to be an epitope residue if at least five of the six scoring functions considered it as an epitope residue. The performance of this method on epitope prediction was comparable to other published models. Subsequently, Liang improved the consensus scoring method by incorporating the support vector regression model to prediction discontinuous epitopes, called EPSVR [93]. Moreover, EPMeta, an epitope prediction server was implemented based on five existing models: EPSVR, EPCES [92], EPITOPIA [39], SEPPA [40], PEPITO [38], and Discotope1.2 [35]. A residue was considered as an epitope residue if at least two models predicted it to be so. EPMeta was consistently better than other existing models based on extensive comparisons, according to the report. Theoretically, the performances of Liang’s models are better than other models. This is because the probability of making an error is smaller if the number of predictors is larger. However, Liang’s models is based on the assumption that all the employed modes have the minimum performance of 0.5, which is not guaranteed.

PREL (Patches and Regression Epitopes Localization), a patch-based epitope localization method, utilized geometric properties, conservation score, chemical properties and single layer perceptron to detect discontinuous epitopes [94]. For each property, two relevance scores were computed to associate the property with epitopes. The first relevance score was the minus logarithm value of the t -test p -value between the epitope patch and non-epitope patch, and the second relevance score was the correlation coefficient between the patch overlapping index and the property. The highly

associated properties were further selected by single layer perceptron. For a given antigen structure, all the patches were constructed with a patch diameter of 50Å, and then the probability of each patch becoming an epitope was estimated based on the selected properties. Residues with high probability were considered as epitope residues. By comparing with existing methods, PREL showed significantly better performance on Ponomarenko’s data set [53] and on Liang’s data set [92]. However, on another of Liang’s data set [93], PREL was slightly better than other models except EPSVR and EPMeta [93]. PREL considers both the patch property (the prototype of epitope) and the correlation property (the nature of multiple epitopes), but the control data which is extracted from antigens is problematic because of the incomplete exploration of antigenic residues [34, 35, 14].

Another existing sequence-based discontinuous epitope prediction model was conceived by Ansari and Raghava [95]. The astonishing AUC was reported to be as high as 0.9. In this study, three SVM models were reported, i.e., binary profile of patterns (BPP) based SVM, physico-chemical profile of patterns (PPP) based SVM, and composition profile of patterns (CPP) based SVM. The first two generated only competitive results, while the third one, CPP based SVM, significantly improved the prediction performance. Intuitively, CPP works well with small number of highly conserved patterns, but it is unpredictable on astronomical variant real data.

Relative accessible surface area and B-factor were used to predict discontinuous epitopes based on the logistic regression model [96]. Although comparable results were achieved by this model, an obvious defect—the use of B-factor—greatly limited its rationality. Since antibody-antigen complexes were used to train the model, the B-factor of interface residues were obviously smaller than other unbound residues.

Shape complementarity was also adopted to identify discontinuous epitopes [36]. When a host and a guest protein were given, the shape complementarity between these two proteins were calculated by the routine fast atomic density evaluation (FADE) [97], and then the contact and binding free energies for this antigen-antibody complex were derived by the FastContact algorithm [98]. In the final step, epitope residues

were determined by the properties obtained from the previous routines. The author observed that they correctly identified 12 out of 26 given antigens with known discontinuous epitopes. Intuitively, this method can provide good performance, but the lack of structural data becomes its main bottleneck.

A summary of discontinuous B-cell epitope prediction approaches is shown in Table 2.3.

At present, many continuous and discontinuous epitope prediction models, as listed in Table 2.2 and Table 2.3, have been proposed. However, none of them can provide satisfactory results on epitope prediction. More importantly, the context dependence which is always ignored by all the aforementioned methods should also be highlighted. Therefore, there is still a long way to go to meliorate epitope predictors.

2.5 Metrics for Evaluation

Evaluation metrics can simplify the comparison between different prediction methods on one hand, and present the reliability of a particular tool for experiment assessment on the other [30]. However, a single metric can hardly represent all qualities of a predictor and the same metric may not be appropriate for all situations [99]. Therefore, a variety of evaluation metrics are necessary. We describe two types of measurements here: parameter-dependent measurements and parameter-independent measurements.

Parameter-dependent measurements:

- sensitivity (sen) (or recall)
- specificity (spe)
- positive predictive value (PPV) (or precision)
- F-measure (F1)
- accuracy (acc), and

Table 2.3: Summary of discontinuous epitope prediction approaches

Resource	Ref.	Features	machine learning model
CEP	[34]	accessible surface area, spatial distance	-
DiscoTope	[35]	residue log-odds ratio, contact residue number	-
Rapberger	[36]	shape complementarity, binding free energy	-
ElliPro	[37]	protrusion index	-
PEPITO	[38]	propensity scale, half sphere exposure	-
Epitopia	[39]	physicochemical scales, geometrical properties	Naïve Bayes classifier
SEPPA	[40]	unit patch of residue triangle, clustering coefficient	-
EPCES	[92]	residue epitope propensity, conservation score, side-chain energy score, contact number, surface planarity score, secondary structure composition	-
EPSVR	[93]	(the same as EPCES)	support vector regression
Ansari	[95]	binary, physicochemical, composition profile of pattern	support vector machine
PREL	[94]	geometric properties, conservation score, chemical properties	single layer perceptron
Liu	[96]	Relative accessible surface area, B-factor	logistic regression

- Pearson correlation coefficient (PCC).

The corresponding definitions are as follows:

$$sen = TP / (TP + FN)$$

$$spe = TN / (TN + FP)$$

$$PPV = TP / (TP + FP)$$

$$F1 = 2 * sen * PPV / (sen + PPV)$$

$$acc = (TP + TN) / (TP + FP + TN + FN)$$

$$PCC = \frac{\sum_i (x_i - \bar{x})(p_i - \bar{p})}{\sqrt{\sum_i (x_i - \bar{x})^2} \sqrt{\sum_i (p_i - \bar{p})^2}}$$

where TP is the number of correctly predicted epitope residues, TN is the number of correctly predicted non-epitope residues, FP is the number of incorrectly predicted epitope residues which are non-epitope residues in fact, and FN is the number of incorrectly predicted non-epitope residues which are epitope residues actually. Regarding PCC, x represents the actual value and p is the predicted value. This measure manifests how well the predicted values coincide with the actual values.

Parameter-independent measurements:

- area under the receiver operating characteristic (ROC) [100] curve (AUC);

AUC is a parameter-independent and robust measurement recommended by researchers for performance evaluation [30].

Parameter-independent measurements are preferred for comparing the prediction capability; however, they are not applicable for some non scale-based methods. Thus, it is important that an appropriate measurement should be chosen to properly qualify the performance of a given model.

Chapter 3

Multi-Interface Domain Analysis

Context-awareness highlights the phenomenon that multiple interfaces could exist in the same domain — multi-interface domain. A multi-interface domain is a domain that can shape multiple and distinctive binding sites to contact with many other domains, forming a hub in domain-domain interaction networks. The functions played by the multiple interfaces are usually different, but there is no strict bijection between the functions and interfaces as some subsets of the interfaces play the same function. This work applies graph theory and algorithms to discover fingerprints of the multiple interfaces of a domain and to establish associations between the interfaces and functions, based on a huge set of multi-interface proteins from PDB.

3.1 Background

A protein domain is usually a contiguous segment in a protein’s primary sequence that can be independently folded to form a stable tertiary structure. Domains vary in length from about 25 amino acids to about 500 amino acids. The number of domains is huge—for example, there are 110,800 domains currently stored in the Structural Classification of Proteins (SCOP) database [59]. As building block of proteins, each domain can be used by a variety of different proteins. When two proteins A and B have an interaction, it is usually an interaction between a domain a of A and a domain b of B. The domain a of protein A may have an interaction with protein C at a different binding site as shown in Figure 3.1. Thus, there can be multiple interfaces

in a domain. Those domains with multiple interfaces are defined as “multi-interface domains”. The concept of multi-interface has been proposed elsewhere previously but differing in meaning, such as “multi-interface hub” defined by Kim to illustrate a protein interacting with multiple proteins synchronously or asynchronously [26], “multibinding protein interface” defined by Tyagi to describe the same interface with interaction to several partners asynchronously [27], and “multi-ligand interface” defined by Dasgupta to depict the union of overlapping interfaces on a protein [28]. Figure 3.2 shows three different interfaces of the catalytic domain of plasmin. This domain in fact has thirteen different interfaces playing five molecular functions according to Gene Ontology (GO) annotations [101]. Note that, the term “multi-interface” is different from “promiscuous” with respect to the number of distinctive interfaces. The former one excludes the isolated interfaces having multiple interacting partners at the same site, while the latter one does not have such constraint. Nevertheless, the attributes of promiscuous interfaces obtained from physical/chemical perspective are very helpful in characterizing multi-interface proteins/domains. For instance, some excellent works done by Fernández on promiscuity show that the order-disorder intermediate region in soluble proteins is vulnerable [146], the induced disorder increases the binding affinity [147] and the possibility of turning promiscuous proteins into drugs [148, 149].

Figure 3.1: Multi-interface domain illustration. Domain a of protein A interacting with domain b of protein B produces interface $a-b$ on domain a , and domain a binding to protein C generates interface $a-c$ on domain a . Interfaces $a-b$ and $a-c$ are distinguishable on domain a , thus domain a is a multi-interface domain.

Figure 3.2: Multiple interfaces in the catalytic domain of plasmin. The interfaces are colored in limegreen (46 residues), marine (6 residues), and red (7 residues) for plasmin interacting with a streptokinase, a protein inhibitor, and another plasmin symmetric unit, respectively. The two overlapping residues are colored orange, and the five molecular functions of this domain retrieved from GO are shown at the top right corner.

Multi-interface domains can be found in the following situations: (i) One domain of a protein interacts with multiple other proteins at the same time. For example, as mentioned above, the catalytic domain of plasminogen can interact with three other plasmins simultaneously to shape three non-overlapping interfaces. (ii) The same domain of a protein interacts with other different proteins at different time or environments. For instance, an antigen peptide can interact with a major histocompatibility complex (MHC) molecule, while it can be also recognized by a T lymphocyte after being presented to the cell surface by a MHC molecule. This phenomenon is previously highlighted by Narayan *et al.* [102] as well, in a discussion on the multiple binding sites of the interferon-regulated transcription factor IRF-1, which is involved in several biological processes such as antiviral response and tumor suppression. (iii) Multiple copies $a_1, a_2, \dots, a_n$ of a domain a are included in different proteins $X_1, X_2,$

..., X_n , and each of the n copies of the domain a has an interaction with another protein at a different binding site. For example, domain D-maltodextrin-binding protein is contained in several proteins, such as in maltose-binding periplasmic (MBP) (PDB ID 3LC8), in MBP/NEDD8-activating enzyme compound 10 E1 catalytic subunit chimera (PDB ID 2NVU) and in maltose binding-A1 homeodomain protein chimera (PDB ID 1MH4). The aforementioned three situations have unique feature on their own although they are similar to each other. The first two situations are classified based on temporal information, while the third one—distinct from the first two—is based on genetic information.

The multiple interfaces in a domain can be grouped into subsets such that each of them share a unique biological function. These functions are usually distinguishable and non exchangeable, i.e., one interface cannot be substituted by others that still hold the same function. As an example shown in Figure 3.2, the three functions of the plasmin catalytic domain cannot be swapped with regard to their interfaces. More interestingly, sometimes the number of such subsets of the interfaces in a domain can be large. As protein functions are currently annotated at the domain level, such as those by GO [101], it is difficult to figure out which interface in a domain possesses what function based on the current annotations. The study of multi-interface domains can associate interfaces with their functions more precisely. Yet almost all past studies of proteins are either at the protein level (*e.g.*, protein-protein interaction, protein complex identification) [103, 104], or at the domain level (*e.g.*, domain-domain interaction, domain transitivity analysis) [105, 106], or at residue level (*e.g.*, interface residue identification, hot spots prediction) [107, 108]. Studies are seldom undertaken at the interface level, which is in the middle between the domain and residue levels. This is probably attributable to uncertainties in locating an interface due to the adaptation, context-awareness, or re-configuration of interfaces [102, 30, 109], in contrast to the clear boundaries possessed by a protein, domain or residue. Some notable exceptions are the works of identifying conserved interface patterns by interface alignment [110], detecting common 3D sites in proteins by frequent graph patterns

of stereochemical atom groups [111], uncovering functional sites in protein families by recurring graph patterns [112], and delineating biological functions of proteins by common atomic motifs of interfaces [113, 114]. However, all these methods mix up interfaces from different domains even if they are remarkably different. In addition, multiple interfaces in one domain are deemed as independent as no relation has been unveiled. Further more, associations between multiple interfaces and their biological functions still remain unanswered.

In this study, we address the following questions: (i) What kind of domains prefer the multi-interface property? That is, we want to know the distribution of domains that have multiple interfaces. (ii) What are the fingerprints of an interface, or a subset of interfaces, in a domain? That is, we want to discover unique structures in a domain that distinguish the multiple interfaces from each other. (iii) What are the relationships between the multiple interfaces in a domain? That is, we want to see whether the multiple interfaces in a domain have any cooperative or competitive relations. (iv) What are the associations between multiple interfaces and their molecular functions?

Our study thus aims to unveil some facets of protein interaction mechanisms by looking at the multiple interfaces of a domain. This is new because past interface analysis [115, 116, 117] generally provides only generic profiles of interface residue organization (e.g., residues contacting graph, hydrophobicity distribution, polarity scattering) and interface residue preference (e.g., favorable and unfavorable residues on interfaces). Our multi-interface properties and the generic profiles of binding interfaces can be combined to facilitate many applications such as protein function analysis [118, 119, 120], protein engineering [121, 122], drug development [123, 124].

To this end, we have compiled a representative data set of multi-interface domains from PDB [58]. Based on this comprehensive data set, graph theory and algorithms are applied to construct interface graph, to mine fingerprints of multiple interfaces, and to explore relations within multiple interfaces as well as associations between interfaces and their molecular functions.

3.2 Materials and Methods

Our data and method are outlined in Figure 3.3, in which the upper part shows the steps for the data set construction, while the lower part is a diagram of our data analysis.

Figure 3.3: Flowchart of multi-interface protein data set construction and data analysis.

3.2.1 Compilation of multi-interface proteins

We compile multi-interface proteins in the following four steps:

- Retrieve PPI (Protein-Protein Interaction) complexes from PDB;
- Preprocess PPI complexes by eliminating transformed chains and short chains;
- Cluster protein chains by sequence similarity;

- Identify representative multi-interface proteins from each protein cluster.

3.2.1.1 Retrieve PPI complexes

We retrieve those PDB complexes produced by X-ray crystallography that satisfy the following criteria: (i) Their macromolecule type is protein but not DNA or RNA. We exclude DNA/RNA because the difference between protein-protein binding and protein-DNA/RNA binding may cause confusion in defining multi-interface proteins. (ii) The number of protein chains within each biological unit in one complex is bigger than 1. (iii) The chain length is larger than or equal to 30. (iv) Their X-ray resolution is better than 3.0Å. Using these selection criteria, 35,760 PDB entries containing 109,672 protein chains are obtained.

3.2.1.2 Preprocess PPI complexes

A few chains in the PDB complexes have transformed coordinates (458 out of 109,672 chains in our data set). Hence the structural relation between these chains and other chains in the same complex may not be correctly represented by their original coordinates. We thus remove those protein chains with the non-identity transformation matrix to clean the noise in the final data set. For example, there are two chains (A and B) with transformed atom coordinates in biomolecule 2 of PDB entry 3HZN, which contains a total of four chains (A, B, C and D); thus the interaction between chain A and chain D is incorrect without coordinates transformation. Besides excluding transformed chains, those protein chains with less than 30 amino acids are also removed from each PDB complex. With the removal of transformed chains and short chains, some chains do not have interaction partner anymore. These chains or complexes are further eliminated from the data set. Finally, a total of 24,664 PDB entries with 87,395 protein chains are used for analysis.

3.2.1.3 Cluster similar chains

The same protein chain interacting with different protein chains can be included in different PDB entries. Therefore, different instances of the same chain have to be

clustered together to identify multiple interfaces. We cluster similar protein chains by the following steps: (i) For each chain in the compiled data set containing 87,395 protein chains, we search the other 87,394 chains using BLAST [125] with e-value of 0.001, and store the similar chains satisfying the e-value criterion. E-value of 0.001 or smaller value indicates the alignment is unique and not caused by error. In our final data, all the e-values are less than e^{-120} . (ii) We build a graph $G^{chain} = \langle V^{chain}, E^{chain} \rangle$ for the entire set of 87,395 chains based on their sequence similarity, and cluster the graph G^{chain} into subgroups based on connected components—*i.e.*, each connected component forms a cluster. Sequence similarity S_{ij} between two chains i and j is defined as $S_{ij} = N_{ij}^{identical} / N_{ij}^{aligned}$, where $N_{ij}^{identical}$ is the number of identical aligned residues between chains i and j , and $N_{ij}^{aligned}$ is the length of semi-global alignment between chains i and j . The alignments used are the ones produced by BLAST. Nodes in G^{chain} are protein chains, and edges are similar chain-pairs with similarity larger than or equal to 97.5%. Pair-wise sequence similarity is calculated based on aligned length instead of the whole sequence length. The similarity of 97.5% is chosen because at this level the interface similarity of their binding pattern is no less than 80% which is the popular threshold used to distinguish interfaces [109]. This strategy can detect multi-interface domains included in proteins with more than one domain. Although it may lead to clustering by short peptides, 99.4% of the clusters have the minimum pair-wise sequence similarity of 0.9, indicating that this situation rarely happens in our data. Besides, almost all the clusters have the minimum pair-wise sequence similarity of 0.9, confirming that our data is nearly free of chaining events due to using a single-linkage clustering algorithm. This yields 13,295 clusters. 11,681 clusters among these have more than one chains, indicating that each of the 11,681 clusters has at least two potential interfaces. In this study, sequence similarity instead of structure similarity is used to cluster protein chains because different proteins may have very similar structures and we cannot group different proteins together to identify multiple interfaces of the same chain.

3.2.1.4 Identify representative multi-interface proteins

For the 11,681 clusters, we first identify the interface residues for each chain of every cluster. Interface residues in the surface of a chain are determined by using a Euclidian distance of 5Å which is commonly used to determine inter-chain contacts [48]. A residue is considered as an interface residue if at least one of its heavy atoms is within 5Å to a heavy atom of its ligand residue. Protein chains that are near each other in 3D space but not in the same biological unit are excluded from this interface residue determination procedure. For instance, the six chains in PDB entry 1UT1 are A, C, and E of bio-molecule 1 and B, D, and F of bio-molecule 2. By definition, we only consider the interfaces between chains of bio-molecule 1 or bio-molecule 2 but not those between bio-molecule 1 and bio-molecule 2. Since the same protein chain can be included in different PDB entries, redundant interfaces may exist in one cluster. Therefore, we further calculate the similarities between different interfaces in one protein cluster to eliminate duplicates or similar interfaces. This is done in five steps: (i) Align all the chains together in one cluster using ClustalW [126]. (ii) Adjust position label of interface residues for all the interfaces in each cluster according to the multiple sequence alignment. This is necessary as the same chain can be numbered diversely in different entries. (iii) Calculate pair-wise interface similarities. Interface similarity is defined as $S_{ij}^{interface} = N_{ij}^{identical} / N_{[i,j]}^{smaller}$, where $N_{ij}^{identical}$ is the number of identically aligned interface residues and $N_{[i,j]}^{smaller}$ is the cardinality of the smaller one of the two sets of interface residues between interface i and interface j . Here, two interface residues are considered as identical if they have the same amino acid type and the same position according to the multiple sequence alignment. (iv) Construct a graph for all the interfaces in a cluster, in which nodes are interfaces and edges represent similar interface pairs with similarity equal to or larger than 0.8. (v) Determine connected components for every interface graph and choose one representative interface for each component. The representative interface is chosen based on the best X-ray resolution of all the chains in that connected component. In case

more than one chains have the same best resolution, then the first one encountered is chosen. Finally, all interfaces in each representative protein of one cluster will be mapped to one of these representative proteins with the best resolution. Following these steps, we have collected 5,222 multi-interface proteins for further use in our in-depth data analysis. For the other 6,459 ($= 11681 - 5222$) clusters, each interface graph is actually a clique—*i.e.*, all the chains in a cluster share a similar interface.

3.2.2 Aggregation of multi-interface proteins

To explore which domains have multiple interfaces as well as their distributions in PDB, we aggregate all these 5,222 multi-interface proteins according to their structural annotations.

Protein structural annotations are obtained through following steps. First we directly retrieve each multi-interface protein’s structural annotations from SCOP [59]. Then, for those proteins that do not have SCOP annotations, we employ PDBeFold [127] to search for annotations of similar proteins stored in SCOP. Among the results generated by PDBeFold for a given protein, we chose the one with the best Q-score as the target domain and retrieve the complete information of the protein containing this domain from SCOP.

Based on structural annotations, multi-interface proteins are further aggregated into several groups in accordance with SCOP classification, as per the following steps: (i) Align each multi-interface protein sequence to its target domain sequence. (ii) Categorize each interface to a domain by the interface residues’ position and domain range. If the entire set of interface residues fall into one domain for a given interface then it is annotated by that domain identifier; otherwise, multiple domain identifiers are tagged to that interface. (iii) Aggregate multi-interface proteins into clusters at different SCOP classification levels, *i.e.*, class, fold, superfamily, family, and domain, according to their annotations.

3.2.3 Construction of interface graphs

Interface analysis is carried out on their fingerprints, where each fingerprint is represented by closed frequent interface residue contacting graph. Each interface in the 5,222 multi-interface proteins is represented as a graph $G = \langle V, E \rangle$, where V is a set of interface residues and E is a set of edges representing the spatial closeness between residue pairs. Edges in each interface satisfy two criteria: the Delaunay triangulation rule and the distance threshold of 5Å. Delaunay triangulation, aiming to maximize the minimum angle of all triangles in the interface graph, is perceptually more meaningful and widely used to build biological structural networks [128, 129]. While distance threshold is used to eliminate those contacts constructed by Delaunay tessellation but are improbable real contacts in practice. Each interface graph is built according to the following three steps: (i) Retrieve all heavy atoms' 3D coordinates for each surface residue stored in its PDB file and transform them into Qhull [130] input format. Only heavy atoms are considered because very few hydrogen atoms' coordinates are reported. (ii) Construct atom contacting graph by Qhull [130]. (iii) Upgrade atom contacting graph into residue contacting graph and pick out interface contacting graph from residue contacting graph. Upgrading is conducted as follows: atom contacts in the same residue are ignored and atom contacts between different residues are kept. For multiple contacts between two residues produced by upgrading, they are further merged together into one contact. That is, connection between two residues is captured in the resulting interface residue contacting graph but not the number of connections between them.

3.2.4 Mining structural patterns from in-domain multiple interfaces

Since every interface is represented by a graph, all interfaces in a domain form a graph database. We are interested in closed frequent subgraphs, paired cooperative subgraph sets and distinctive subgraph sets between interfaces to uncover fingerprints for interfaces and to identify relations between multiple interfaces of one domain.

We introduce some additional notations. A graph $H = \langle V(H), E(H) \rangle$ is a subgraph of a graph $G = \langle V(G), E(G) \rangle$, denoted by $H \subseteq G$, if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. The support of a subgraph H in a graph database D_G is the number of graphs in D_G that contain H as a subgraph. A subgraph H is said to be frequent in D_G under a given support threshold count δ if the support of H is at least δ . A frequent subgraph H is closed in D_G if H cannot be extended by any additional node or edge without changing its support in D_G . A paired cooperative subgraph set P^G is a paired set of graphs $(\{G_1^x, G_2^x, \dots\} : \{G_1^y, G_2^y, \dots\})$ such that G_i^x occurs in interface x and G_j^y occurs in interface y of the same domain simultaneously. A distinctive graph set $S_G^x : \{G_1^x, G_2^x, \dots\}$ says that each graph G_i^x in S_G^x only occurs in the interface x of a given domain but never occur in other interfaces of the same domain. The detailed procedures for mining these graph patterns are described below.

Closed frequent subgraphs of one interface cluster, which are deemed as fingerprints of one interface, are mined from an interface graph data base, where the graph database is a set of interface residue contacting graphs in one domain. Closed frequent subgraphs instead of frequent subgraphs are mined since we aim to figure out the largest structures that can be used to identify a specific interface for a given domain. To explore the fingerprints of an interface of a domain, we first align all protein's structures together by the CE algorithm [131], and then cluster the interfaces together based on their spatial similarity. Subsequently, fingerprints for each interface are mined by ParMol [132] with the FFSM algorithm [133]. The local support of mining fingerprints is set to 20% in this study, thus the global support δ is set as the number of graphs times 0.20.

Interface relation in one domain is described by paired cooperative graph sets and distinctive graph sets. Paired cooperative graph sets between two different interfaces in a domain are identified using the following steps: (i) Build graph database for each set of interfaces. (ii) Mine closed frequent subgraphs by ParMol [132] for each graph database with local support of 20%. (iii) Transform sets of closed frequent subgraphs into a transactional data set and mine closed frequent item set with LCM [134]. In this

transactional data set, each transaction is a closed frequent subgraph and the items in each transaction are the different interfaces. With these steps we can identify all paired co-existing subgraph sets in a sets of interfaces. Exploring distinctive relations between interfaces is much easier than mining cooperative relations. To obtain a unique set of fingerprints for each interface, we first simply mine all closed frequent subgraphs from a set of interfaces, and then pick out those graphs that belong to one type of interfaces but not to others. Finally, we group them together according to the interface type they belong to. Cooperative relation tells the connection between two different interfaces and distinctive relation discriminates one interface type (function) from other interfaces. Distinctive relation can be used to identify specific interface and cooperative relation can be useful to infer new interfaces based on known interfaces.

3.3 Results

In this section, we present results to show the distributions of multi-interface proteins and multi-interface domains, fingerprints of interfaces, cooperative and distinctive relations between multiple interfaces, associations between interfaces and molecular functions, and some properties of cross-domain interfaces.

3.3.1 Distributions of multi-interface proteins and multi-interface domains

The 87,395 protein chains are grouped into 13,295 clusters according to their sequence similarities generated by BLAST [125]. For the 13,295 clusters, there are 5,222 multi-interface proteins with a total of 15,345 interfaces—*i.e.*, 3 interfaces each on average. We note that the number of multi-interface proteins differs with regard to the change of interface similarity threshold due to promiscuity of interface [135]. For example, Kim and colleagues identified 873 multi-interface proteins based on structural exclusion in which similarity threshold is 0 [26], and Kar and coworkers only obtained 79 cancer-related multi-interface proteins with threshold of 0.2 [136]. The distribution of multi-interface proteins of our data in terms of the number of interfaces of a protein is shown

in Table 3.1, indicating that most of the multi-interface proteins have a very small number of interfaces and only a few of them have more than 5.

Table 3.1: Multi-interface protein distribution in terms of the number of multiple interfaces.

# interfaces	# protein chains
2	2735
3	1349
4	608
5	241
6	119
7	62
8	45
9	28
10	11
>10	24

Figure 3.4 is produced by PHYLIP [137] based on 2,517 of the 5,222 multi-interface proteins that have SCOP annotations (where the rest 2,705 do not have SCOP annotations). It shows the number distribution of multi-interface proteins at different SCOP classification levels. Obviously, multi-interface domains can appear in a broad range of clusters in terms of SCOP classification. Among all the eleven classes in SCOP, α/β proteins, $\alpha + \beta$ proteins, all- β proteins, and all- α proteins account for 90.3% of all the multi-interface proteins. Figure 3.4 also indicates that all- β proteins, or at least part of them, are less conservable since they have the largest number of multi-interface proteins in one domain. It can be also seen that multi-interface proteins with a large variability tend to aggregate to a small number of clusters instead of uniformly spread out to each cluster as shown in Figure 3.4.

Table 3.2 gives the distribution of the 2,517 multi-interface proteins at different levels of SCOP classification. The complete number of sub-levels for each classification level is retrieved from SCOP [59], while the number of sub-levels with multiple interfaces for each level is determined by the number of multi-interface domains “upgraded” from the domain level to the class level. It can be seen from Figure 3.4 that,

Figure 3.4: Multi-interface domain distribution at different SCOP levels. Length of lines represents the normalized number of multi-interface proteins at each classification level, and multiple lines under the same level of one cluster represent different sub-clusters. The clusters are organized as a rooted tree structure from a higher level to a lower level, and the clusters at the same level are plotted in the clockwise descendant order. Here a represents α and b represents β .

while multi-interface proteins exist over all classes of SCOP classification, they clearly favor a few of the sub-levels. In particular, although there are more than 110,000 domains with annotations in SCOP [59], only a very small proportion of these domains (1,730/97,178) have the multi-interface property. This phenomenon also suggests that

all biological processes have their own small set of pivotal proteins [138].

Table 3.2: Distribution of multi-interface domains at different SCOP classification levels.

class	fold		superfamily		family		domain	
	$N_{all}^\bullet$	N_{int}°	N_{all}	N_{int}	N_{all}	N_{int}	N_{all}	N_{int}
All α proteins (349) [†]	259	89	459	128	772	174	- [‡]	255
All β proteins (577)	165	85	331	138	679	207	-	336
α and β proteins (α/β) (684)	141	91	232	137	736	293	-	527
α and β proteins ($\alpha+\beta$) (662)	334	161	488	209	897	318	-	476
Multi-domain proteins (45)	53	21	53	21	74	26	-	36
Membrane and cell surface proteins (53)	50	24	92	35	104	38	-	43
Small proteins (67)	85	24	122	29	202	38	-	57
Total	1087	495	1777	697	3464	1094	97178*	1730

[•]number of clusters under the given SCOP classification level; [◦]number of clusters that have multi-interface proteins under the given SCOP classification level; [†]number of protein chains that have multi-interfaces; [‡]data is not available in SCOP; *the total number of domains is 97,178 in SCOP version 1.73, but the total number of domains listed above is slightly smaller since there are still four classes with very few number of domains are not shown here. SCOP version 1.73 instead of 1.75 is used in this study because the PDBeFold [127] is based on SCOP version 1.73, which is used to search SCOP to get similar domains for a given protein.

Interfaces between proteins can be categorized into homo-oligomer, homo-complex, hetero-oligomer and hetero-complex in terms of sequence similarity and interaction lifetime [139]. Therefore, we classified the interactions of the 5,222 multi-interface proteins into the aforementioned four types by the method described in [140]. Then, we analyzed the preference for each interaction type, with the preference defined as $\frac{N_i^m}{\sum N_{i'}^m} / \frac{N_i^a}{\sum N_{i'}^a}$, where N_i^m is the number of multi-interface protein interactions of type i and N_i^a is the whole number of interactions of type i in our data. This was followed by mining fingerprints of every interaction type and exploring relations between fingerprints of different interaction types. According to preferences shown in Table 3.3, proteins with multiple interfaces are favored in homo-interactions (homo-oligomer and homo-complex). Fingerprints mined from each type of interaction with a minimum frequency of 5% show that homo-oligomers have 96 non-trivial fingerprints and that the same number is significantly lower for hetero-complexes (9). This indicates that interfaces of homo-oligomers share some common structural patterns, although homo-oligomers are different from each other, and interfaces of hetero-complexes rarely have

recurring structural patterns. Surprisingly, despite only 9 fingerprints mined from hetero-complexes, 4 of them are isomorphic to (other 4) fingerprints (of the 96 mined) from homo-oligomers. The p -value of this number of isomorphic fingerprints against randomly generated graph pairs is $6.7\text{e-}4$. This shows that sharing of interface patterns between homo-oligomers and hetero-complexes is significantly more frequent than expected by chance. Due to a lack of homo-complex and hetero-oligomer interactions in multi-interface proteins, we cannot get reasonable results from these two types of interactions.

Table 3.3: Preference of multi-interface protein interactions of four interface types.

Interface type	# interaction	# multi-interface interaction	preference
Homo-oligomer	647	127	1.42
Homo-complex	13	3	1.67
Hetero-oligomer	38	4	0.76
Hetero-complex	6695	890	0.96

3.3.2 Analysis on multiple interfaces within the same domain

The top ten multi-interface domains with the largest numbers of proteins are shown in Table 3.4. Unexpectedly, domains from the immune system have the most number of multi-interface proteins. This observation in part can be explained by the fact that a small portion of hypervariable regions exist in these proteins [2]. That is, although these proteins have the same domain, they do contain different interfaces due to mutations occurring in the hypervariable regions.

Table 3.4 also unveils the wide coverage of biological functions played by some domain. For example, the proteasome beta subunit (catalytic) domain has around seven different interfaces which are involved in different biological processes. This is quite different from the Ig VH domain with the typical two interfaces playing the Ig VL protein binding role and the role of antigen recognition.

Since there are as many as 1,730 multi-interface domains—see Figure 3.4 and Table 3.2—and exploring properties of every multi-interface domain is not our purpose

in this study, we undertake analysis on two domains: Ig VH domain and proteasome beta subunit domain. The former is contained in the largest number of multi-interface proteins and the latter has many binding sites besides a sufficient number of multi-interface proteins holding this domain.

Table 3.4: Top ten multi-interface domains with the largest numbers of proteins.

Domain name	# multi-interface proteins	avg # interfaces
Ig heavy chain variable domain, VH	43	2.4(± 0.7)
Ig light chain κ variable domain, VL- κ	35	2.5(± 0.8)
Ig heavy chain γ constant domain 1, CH1- γ	28	2.3(± 0.6)
Ig light chain κ constant domain, CL- κ	18	2.4(± 0.6)
Hemoglobin, beta-chain	17	2.8(± 0.9)
Proteasome beta subunit (catalytic)	17	7.6(± 3.2)
T-cell antigen receptor	16	3.1(± 1.6)
Hemoglobin, alpha-chain	14	3.1(± 0.5)
Nucleoside diphosphate kinase, NDK	13	2.9(± 0.3)
Dodecameric ferritin homolog	12	5.4(± 1.4)

3.3.2.1 Fingerprints of interface

Given a set of multiple interfaces in a domain, we fish out interface-specific fingerprints by the mining of closed frequent subgraphs (substructures) from the corresponding interface graph database. These frequent substructures capture the natural organizations of interface residues. In the past, frequent sub-structures have been successfully applied to study protein structure and function [141].

Our experiments on identifying interface fingerprints are carried out on Ig VH domain and proteasome beta subunit domain separately. Generally, we obtained a great number of non-trivial closed frequent substructures for each interface of the two domains. We then examine whether these graph patterns (fingerprints) are domain-specific or not. To this end, we compared fingerprints between interfaces in various dimensions. First, we directly compared residue composition of different interfaces. The detailed results are shown in Figure 3.5 and Figure 3.6 for Ig VH domain and proteasome beta subunit domain, respectively. It is obvious that residue preferences are divergent for different interfaces. Second, we explored fingerprint isomorphism

between different interfaces. Not surprisingly, we identified just one isomorphic fingerprint between the interfaces of Ig VH domain and very few isomorphic fingerprints between interfaces of proteasome beta subunit domain shown in Table 3.5. Based on these observations, we can safely draw the conclusion that the fingerprints are interface-type specific and they indeed can distinguish different interfaces of one domain.

Figure 3.5: Amino acid distribution of the antigen-binding interface and the protein-binding interface in the Ig VH domain.

Figure 3.6: Amino acid distribution of the six binding sites in the proteasome beta subunit domain. *bs* means binding site.

Figure 3.7 shows an example of a fingerprint in proteasome beta subunit domain. The real data of this structure contained in PDB entry 3NZX is shown in Figure 3.7(b). With hydrophobicity information labeled to each node in this structure, we found that the center of binding sites is filled with hydrophobic residues which is surrounded by hydrophilic residues as can be observed from Figure 3.7(b). This result is consistent with the previous wet-rim-dry-core observation of binding site [142].

Table 3.5: Number of isomorphic fingerprints between different interfaces within the proteasome β -subunit domain.

	Binding site A (271 [†])	Binding site B (49)	Binding site C (685)	Binding site D (21)	Binding site E (251)	Binding site F (237)
Binding site A		7 (<5.0e-4*)	10 (2.0e-3)	0 (7.3e-1)	6 (1.9e-2)	5 (1.3e-2)
Binding site B	7 (<5.0e-4)		3 (7.6e-2)	0 (9.1e-1)	0 (5.0e-1)	4 (1.0e-3)
Binding site C	10 (3.3e-3)	3 (8.7e-2)		2 (9.1e-2)	17 (<5.0e-4)	9 (5.0e-4)
Binding site D	0 (7.4e-1)	0 (9.1e-1)	2 (8.3e-2)		1 (2.4e-1)	0 (8.3e-1)
Binding site E	6 (1.3e-2)	0 (5.2e-1)	9 (2.0e-3)	1 (2.2e-2)		3 (1.0e-1)
Binding site F	3 (3.3e-2)	4 (<5.0e-4)	4 (5.3e-2)	0 (8.3e-1)	3 (9.0e-2)	

Values in the upper triangle are calculated based on interface overlapping threshold of 0.8, while the lower triangle values are computed with no overlapping residues between interfaces. [†]Number of fingerprints, and * p -value. p -value is calculated against randomly generated graphs based on fingerprints of two arbitrary binding sites i and j with equal graph size and number of edges. Node labels of generated graphs are determined based on amino acid frequency.

Figure 3.7: An example of a fingerprint in the proteasome beta subunit domain. (a) is a diagram of the given fingerprint, where the filled circles represent the interface residues, and the lines represent the contacts. Color and color shade represent residue hydrophobicity index defined by Kyte and Doolittle [5]. (b) is the real structure of (a) presented in PDB entry 3NZX. The dashed lines represent the contacts determined by Delaunay triangulation, and the highlighted lines are the contacts shown in (a).

3.3.2.2 Cooperative and distinctive relations between multi-interfaces

Closed frequent substructures characterize the common organization of interfaces in a domain. However, they cannot reveal the relations between multiple interfaces in that domain. In addition, one domain is a perfectly assembled structure constituted by all kinds of residues. It is believed that protein structures are not randomly assembled together and they should incorporate some cooperative or competitive relations

Figure 3.8: An example of pairs of fingerprints from multiple interfaces within the Ig VH domain. The dash line separated structures are fingerprints, and the solid orange lines between these fingerprints represent cooperative relations between them. The filled circles represent interface residues and the solid green lines represent residue contacts. Color of residues indicates their hydrophobicity.

[143, 144]. Thus we wonder whether correlations exist between different interfaces in a domain. To this end, co-existing paired fingerprints in different interfaces are used to describe cooperative relations, and unique fingerprints are employed to depict distinctive relations between multiple interfaces in one domain. Please refer to the methods section for detailed descriptions of these ideas.

Co-existing paired fingerprints somehow could reveal reenforcement of interface residues from a physicochemical perspective which could help wet-lab experiments. Our experimental results reveal that a large number of co-existing fingerprints of different interfaces exist both in Ig VH domain and proeasome beta subunit domain. An example of co-existing paired fingerprints in Ig VH domain is shown in Figure 3.8, which is contained in two thirds of all multi-interface proteins of this domain.

Unique fingerprints of each interface are the signatures that distinguish one interface from other interfaces in the same domain. Observations show that, besides cooperative fingerprints, a large number of fingerprints are interface-specific as shown in Table 3.5. Statistical analysis shown in Table 3.5 also reveal that, although very

few isomorphic fingerprints can be identified from different binding sites of proteasome beta subunit domain, some pairs of binding sites indeed have significant number of common fingerprints as indicated by the very confident p -values, which do consolidate the argument that cooperative relation exists between different binding sites of proteasome beta subunit domain.

Overlapping residues between interfaces distort the study of co-existing paired fingerprints and common fingerprints between interfaces. Thus we further investigated the influence of overlapping, measured by Jaccard index, between multiple interfaces. Our experimental result on Ig VH domain shows that overlapping between antigen binding interface and protein binding interface is 0.04 ± 0.05 which means the overlapping has marginal effect on cooperative fingerprints analysis. Besides, cooperative fingerprints in each pair are quite different, indicating they are not from the overlapping region of interfaces, as shown in Figure 3.8. For control, we carried out isomorphic fingerprints testing on non-overlapping interfaces and interfaces with overlapping threshold of 0.8 on proteasome beta subunit domain. Results are shown in Table 3.5. The change of p -values before and after the removal of overlapping residues implies a little influence of the overlapping residues, but the very slight change indicates that cooperative fingerprints are not caused by overlapping, as shown in Table 3.5.

Co-existing paired fingerprints and distinctive fingerprints shown in our data do reveal that interface is distinguishable and cooperates with each other to some extent in the same domain.

3.3.2.3 Association between interface's fingerprints and its function

Organization of interface residues determines its capable binding partners, which further specifies its associated molecular function. Therefore, fingerprints of interfaces and their molecular functions are examined to uncover the associations between them in Ig VH domain and proteasome beta subunit domain.

Molecular functions of Ig VH domain retrieved from GO [101] are protein binding and antigen binding, while functions of proteasome beta subunit domain are threonine-

type endopeptidase activity, peptidase activity, hydrolase activity, endopeptidase activity, protein binding, endopeptidase activator activity, RNA binding, and NF- κ B binding. Unfortunately, these functions are assigned to each domain and we cannot figure out which interface is associated with which function. By analyzing multi-interface profile in the above section, we found that interfaces are distinguishable by their fingerprints. Although a few of the fingerprints are shared by multiple interfaces, lots of them are unique to their interface type. For example, 73 and 1182 fingerprints are identified for the antigen binding interface and protein binding interface from Ig VH domain, respectively. But the number of isomorphic fingerprints between the two sets of fingerprints is only one. Hence almost all of the fingerprints belonging to antigen binding interface and protein binding interface can be used to specify their functions, except the common one. Regarding proteasome beta subunit domain, the number of fingerprints for the six interfaces ranges from 21 to 685 as shown in Table 3.5; but the largest number of isomorphic fingerprints between each pair of these interfaces is only 17. Therefore, it is possible to work out which type of interfaces is for what biological function based on their fingerprints in a multi-interface domain. This observation can be further extended to determine protein binding sites and their molecular functions based on fingerprints.

3.3.2.4 Interface fingerprints between different domains

We have identified interface fingerprints and analyzed their similarities between interfaces within a domain. But it is still unknown whether these fingerprints are domain-specific or not for multi-interface domains. To address this issue, we take a small trick of comparing the similarity of fingerprints of two very similar domains in the same protein family. The assumption is that if the similarity is low for the two very close domains, then the similarity should be much lower for two randomly selected domains. Based on this assumption, we analyzed the fingerprints similarity between the Ig VH domain and Ig VL kappa domain which have similar number of multi-interface proteins, and both belong to the V-set domain family. We employ

Figure 3.9: Fingerprints assortativity of Ig VH domain and Ig VL-kappa domain.

graph assortativity to measure the similarity between different sets of fingerprints. Assortativity is a preferred metric to quantify the equivalence of a network's nodes and connections in graph theories. Assortativity between fingerprints of this two domains is calculated using NetworkX [145]. Figure 3.9 shows the assortativity of fingerprints in the Ig VL-kappa domain and Ig VH domain. The null hypothesis says that the mean assortativities of fingerprints between the two domains are the same. Then we calculated the t-test p -value of graph label-assortativity (residue type) and graph degree-assortativity of the fingerprints from the Ig VL-kappa domain and Ig VH domain. The p -values, $1.7e-4$ and $1.0e-2$, under the two tests suggest the significant difference of the fingerprints between Ig VL-kappa domain and Ig VH domain.

However, difference between the fingerprints of two domains could be induced by the fingerprint topology or the number of fingerprints with their combinations. Therefore, we conducted additional experiments to understand the influence of fingerprint's topology and fingerprint's volume on analysing fingerprints' domain-specificity, by exploring the difference within randomly generated fingerprints and examining disparity between subsets of fingerprints of the two domains. To explore the effect of fingerprint's topology on this domain-specificity, we randomly generated two sets of fingerprints with one having an equal number of fingerprints to that of Ig VL-kappa

domain and the other having the same number as that of Ig VH domain. The t -test p -values of degree-assortativity and label-assortativity between the two sets of randomly generated fingerprints are $8.2\text{e-}1$ and $9.2\text{e-}1$, respectively. These insignificant p -values indicate that the randomly generated fingerprints are very similar; and thus this test verifies that the difference between the fingerprints of Ig VL-kappa domain and Ig VH domain is not trivial. To examine the influence of fingerprint's volume on this domain-specificity, we sampled a small set of fingerprints (50, in our experiment) for each domain, and calculated the disparity between them in terms of isomorphic fingerprints. This sampling with replacement was carried out for thousands of times and were subsequently used to compute the p -value of the difference between subsets of fingerprints of the two domains. The significant p -value of $7.0\text{e-}2$ suggests that the difference between the two domains is not dominantly caused by the large number of fingerprints and their combinations. Based on the above analysis, we can conclude that the fingerprints of multi-interface domains are domain-specific. As this study focuses on multi-interface domains, the domain-specificity is not examined for the single-interface domains.

3.3.3 Profiling of domains with cross-domain multi-interface

In protein-protein interacting complexes, most of their interfaces are located within one single domain. But there are still a few interfaces that spread over other domains; these are named cross-domain interfaces in this study. Our results show that among the 2,517 multi-interface proteins with SCOP annotations, 301 proteins contain cross-domain interfaces. This seems a bit surprising as it is usually believed that domain is the basic functional unit in cell. This, however, is not always true as in some circumstances they have to be combined together to play a certain function. Figure 3.10 shows an example of a cross-domain interface in PDB entry 1DE4, in which the protein-binding interface in chain G spans over Hemochromatosis protein Hfe α -1, α -2, and α -3 domains. The numbers of cross-domain interfaces in various SCOP class levels are shown in Figure 3.11. It can be seen that cross-domain interfaces are favored

Figure 3.10: An example of a cross-domain interface. The forest colored interface is a cross-domain interface, while the interface with orange color is another interface.

Figure 3.11: Distribution of cross-domain interfaces at various SCOP class levels. A line between two classes means a cross-domain interface in these two classes. The weight on each line indicates the number of instances that have cross-domain interfaces in our data set. Cross-domain multi-interfaces with very small numbers are not shown here.

in $\alpha + \beta$ and α/β proteins. However, cooperative fingerprints between cross-domain interfaces at SCOP class level can be rarely obtained. This could be attributed to the large number of domains in each class (see Table 3.2) with a very small number of cross-domain interfaces (301 in total). Based on this observation, we can conclude that domain is mainly the functional unit, but with some exceptions.

3.4 Summary and Discussion

Multi-interface domains have been demonstrated to have multiple non-exchangeable interfaces in every domain. That is, one of the interfaces of a multi-interface domain cannot be changed to others that still hold the same function.

This work has conducted comprehensive analysis on the following aspects of multi-interface domains: (i) which domains have multiple interfaces; (ii) what the fingerprints of the multiple interfaces are; (iii) the relations of the multi-interfaces in a domain; (iv) the associations between multi-interface and their molecular functions, and (v) profiles of cross-domain multi-interfaces. Our data is a set of 5,222 multi-interface proteins obtained from 35,760 PDB entries. Interface geometric information, graph theories, closed frequent item set mining, and association mining techniques are utilized together to reveal interface signatures, associations between multiple interfaces in a domain, and relation between interface and its molecular function. Based on our systematic analysis, we found that around 40 percent of proteins have multiple interfaces which are distributed to a very small set of domains over all available domains, and that the multiple interfaces in one domain can have the same or different function types. We observed that the multiple interfaces of these domains were distinguishable in terms of their fingerprints, which further indicated the function-specific property of these interfaces in a domain. Moreover, we observed both unique and co-existing structural patterns existing between multiple interfaces of one domain, highlighting the distinctive and cooperative relations between multiple interfaces. The number of multi-interface domain is still very large although it accounts for a very small portion in the entire number of domains. Therefore, analysis is undertaken on other selected domains.

Future works include building interface-function association database to facilitate a lower level analysis and to relate specific multi-interface domains to real-life applications, for example, multiple interface predictions in antigen-antibody interactions.

Chapter 4

Association-based Antibody-specific Epitope Prediction

Context-awareness characterizes the reconfiguration of epitope residues when an antigen interacts with a different antibody. A coarse binary classification of antigen regions into epitopes, or non-epitopes without specifying antibodies may not accurately reflect this biological reality. In Chapter 3, we systematically explore the characteristics of multiple interfaces; in this Chapter, we take a step further to predict antibody-specific epitopes in line with the context-awareness principle, which has significant impact on prophylactic and therapeutic applications.

4.1 Motivation

A B-cell epitope is a small subset of residues in an antigen sequence which can be recognized by a specific antibody [15]. Epitopes can be categorized into two types: continuous and discontinuous [19]. A continuous epitope consists of residues that are contiguous in the primary antigen sequence, while a discontinuous epitope is formed by residues that are separated, sometimes far away, from each other in the primary sequence but they are close to each other in 3D space through polypeptide folding. For example, the epitope in the antigen chain E of PDB:1ZTX is constituted by five distant stretches: TYR302, SER306 LYS307 ALA308 PHE309, THR330 GLY331 THR332 ASP333, ALA365 THR366 ALA367 ASN368, and GLY389 GLU390 GLN391. All these short segments are required for the recognition by the antibody and thus they

are not five epitopes on their own. Due to the importance of identifying epitopes in vaccine design, disease diagnosis and disease therapy [150], intensive efforts have been made to predict both continuous and discontinuous epitopes over the past few decades.

Context-awareness in the recognition between antigens and antibodies [31, 30], also known as context dependence, is the key idea that motivates the whole work. Context-awareness is a property highlighting that epitope is a dependent entity—an epitope cannot be formed without a corresponding antibody [30]. Therefore, epitope can be detected only at the point when an antigen interacts with a specific antibody area which is called the paratope of the antibody. A more intriguing facet of such binding mechanism is that the cluster of residues that acts as an epitope under one set of circumstances will not necessarily behave as epitope under another set [30]. In other words, an epitope of an antigen may be reconfigured with a different set of residues when a different antibody is presented in the interaction. For instance, the epitope of the prion protein as shown in PDB:1TQB consists of residues GLY127, LEU128, ASN156, TYR158, ARG159, ILE185, LYS188, GLN189, THR191, VAL192, THR193, THR194, THR195, THR196, LYS197, GLY198 and GLU199. However, the epitope of this antigen is reconfigured remarkably when the antibody is changed to another protein as shown in PDB:2W9E. The reconfigured epitope consists of residues GLY145, SER146, ASP147, TYR148, ASP150, ARG151, TYR152, ARG154, GLU155, ASN156, HIS158, ARG159, ASN200, THR202 and LYS207, having only two common residues (ASN156 and ARG159) overlapping with the first epitope. Therefore, a coarse binary classification of antigen regions into epitopes, or non-epitopes without specifying antibodies and paratopes may not accurately reflect biological reality [30].

The aim of this work is to predict the location of the epitope residues when we are given the sequence of an antigen and the heavy chain and light chain sequences of an antibody. Existing methods [63, 66, 10, 11, 34, 35, 37] generally overlooked the property of context-awareness, and studied instead an antibody-ignored B-cell epitope prediction problem. Given an antigen, these earlier methods just tell the union of all

antigenic residues in this antigen. They were unable to specify which subset of the antigenic residues that can form an epitope binding to an antibody.

In light of this context-awareness, we introduce two kinds of associations: (i) the residues-residues pairing preference between epitopes and paratopes, and (ii) the dependence between two sets of contact residue pairs. Both of them are mined from antibody-antigen complexes for the training of our model. When an antigen-antibody complex without structural information is presented for epitope prediction, the preference association plays a bridging role to link the residues from the paratope to the epitope, and the dependence association is used to infer new set of contact paratope-epitope residue pairs. Our prediction begins by identifying antibody paratope residues. This idea is based on the fact that antibody paratopes are regularly structured and are easy to be identified. The residues of a paratope are usually located in six complementary determining regions (CDRs), namely L1, L2, L3, H1, H2 and H3 [151]. The first three CDRs are in antibody light chain, and the other three are in antibody heavy chain. The six CDRs can be identified by the Chothia CDR definition [23], and the paratope residues can be easily located by its residues' relative composition and cooperativity information. Figure 4.1 shows a diagram of our prediction method and illustrates how the preference association is used to predict epitope residues and how a dependence association is used to infer a new set of contact residue pairs. As there are still possibly many uncertain residues in the antigen, a semi-supervised Hidden Markov Model (HMM) is proposed to complete the process of epitope prediction.

Our technical ideas differ from traditional ones. Very often, physico-chemical properties, such as individual amino acid's hydrophilicity, backbone flexibility, residue's antigenic propensity, exposed surface area and so on, were used to tackle the problem of predicting continuous epitopes [63, 66, 10, 11]. Single or combinations of different properties including solvent-accessible surface area, spatial information, contact distance and amino acid statistics had also been proposed as features to study the

Figure 4.1: Antibody-specific epitope identification by using associations and a semi-supervised Hidden Markov Model. Here, the unfilled circles represent non-paratope/epitope residues. The paratope residues are colored by orange, and the epitope residues are colored by light green and purple. We distinguish them by light green and purple because the light green circles are the epitope residues that interact with this specific antibody, while the purple circles are in a different epitope that interacts with another antibody when the environment changes. The solid arrows represent our epitope identification by the preference between paratope and epitope residues, while the dashed arrow and two boxes represent our epitope detection through dependence within two sets of paratope-epitope interacting residue pairs. The circles colored by green are the epitope residues that we can identify. The purple residues are not predicted as restricted by the context awareness..

discontinuous epitope prediction problem [34, 35, 37]. Some advanced machine learning approaches were also exploited [13, 41, 152, 14, 33]. Overall, the performance on the traditional antibody-ignored epitope prediction problem was not satisfactory for either continuous or discontinuous epitopes [73, 53, 153].

We name our method **ABepar** short for **Antibody-specified B-cell epitope prediction** through **association rules**. As mentioned, our method is trained on antibody-antigen PDB complexes, but it can be applied to any antibody-antigen sequence pairs that do not contain structural information. By simplifying the computational steps, our ABepar method can also be used to conduct the traditional epitope prediction when only antigen sequence is provided. The software and supplementary data are available at <http://sunim1.sce.ntu.edu.sg/~s080011/>

4.2 Materials and Methods

4.2.1 Data set

We compiled a set of 80 non-redundant X-ray crystallographic antigen-antibody complexes with the resolution better than $3.0\text{\AA}$ after an exhaustive search from PDB [58]. Each complex in this data set satisfies: (i) it contains three or more polypeptides each with a length ≥ 30 amino acids; (ii) it contains antibody variable regions, such as Fab, VHH or Fv fragments; and (iii) the paratope residues of each complex are mainly situated in the six CDRs. Besides, if a structure contains more than one asymmetric units but with no structural difference, only one unit is selected. To remove redundancy, an antigen structural pair-wise alignment was carried out by the combinatorial extension (CE) algorithm [131]. An antigen is redundant if the following criteria are met: (i) the alignment z -score is ≥ 4.0 ; (ii) the root mean square deviation (RMSD) of the alignment is $\leq 3\text{\AA}$; (iii) the proportion of the matching residues against the aligned positions is $\geq 80\%$; and (iv) the proportion of overlapping epitope residues is $\geq 80\%$ against the epitope residues of the shorter sequence. The first criterion is based on the CE algorithm, while the middle two are suggested by [53]. Here the structural alignment instead of sequence alignment is used because two very similar structures can come from low similarity, some times very different chains.

4.2.2 Construction of our training model

Given an antibody-antigen PDB complex, the epitope and paratope residues are determined by using a distance threshold $4\text{\AA}$. (This threshold was recommended by [35] as it can capture the epitopes and paratopes with a high precision.) One residue is considered as an epitope residue if there exists an atom of this residue, except hydrogen, that is separated within a distance of $4\text{\AA}$ from an atom of an antibody residue except hydrogen. This antibody residue is correspondingly called a paratope residue.

Taking all of the epitopes and paratopes from the training data, we mine the residues-residues pairing preference in these binding sites, as well as the dependency

Figure 4.2: Flow chart of model construction and epitope prediction.

between sets of contact residue pairs. We also compute epitope and paratope residue's relative composition and cooperativity values. These compositions and associations, formally defined later, are used when a pair of antibody-antigen sequences is given for epitope prediction. The detailed flowchart of training and testing is shown in Figure 4.2. The upper part of the flow chart shows our steps in model construction and the lower part depicts the steps for epitope prediction.

4.2.2.1 Associations: residues-residues pairing preference and dependence between two sets of contact residue pairs

A residues-residues pairing preference in paratope-epitope interacting complex is defined as a frequent biclique $G = \langle V_b, V_g, E \rangle$, where V_b is a set of vertices representing paratope residues, and V_g is a set of vertices representing epitope residues. E is a set of edges constituted by paratope-epitope contact residue pairs, satisfying the condi-

tion $|E| = |V_b| * |V_g|$, i.e. $\forall v_b \in V_b, \forall v_g \in V_g, \langle v_b, v_g \rangle \in E$. This concept of residue pairing preference underlies the pairing or full interaction between a group of paratope residues and a group of epitope residues. It extends the common pairing between two single residues, and it can expand the statistical and biological significance of residue pairing as demonstrated in our early work [154] where protein binding hotspots are studied.

Residues-residues pairing preferences are identified from a set of antibody-antigen PDB complexes through the following steps: (i) determine all paratope and epitope residues of each antibody-antigen complex by using the distance threshold 4Å; (ii) construct a bipartite graph from the paratope-epitope interacting residue pairs of each complex. The vertices are the paratope and epitope residues, and the edges stand for the contacts between paratope and epitope residues; (iii) identify the maximal bicliques from every bipartite graph by using the LCM-MBC algorithm [155]; and (iv) select the frequent bicliques from all the antibody-antigen complexes with a minimum occurrence level of 7%.

A dependence between two sets of interacting residue pairs is described by an association rule. It highlights the extent to which one set of interacting residue pairs is dependent on the other set. In other words, if one set of interacting residue pairs is observed in a binding site, then what is the probability that (at which confidence level) the residue pairs in the other set also occur in this binding site. Formally, such an association rule is in the form $\{\langle p_1^o-e_1^o \rangle, \langle p_2^o-e_2^o \rangle, \dots\} \rightarrow \{\langle p_1^i-e_1^i \rangle, \dots\}$, where $\langle p-e \rangle$ represents a contact residue pair in the paratope-epitope binding site without position reference. The superscript o means observed paratope-epitope interacting residue pair, while the superscript i means the implied interaction.

Four steps are taken in detecting dependence from a set of antibody-antigen interacting complexes, i.e., (i) identify all the paratope-epitope interacting residue pairs of each complex by using the distance threshold 4Å; (ii) transform each interacting residue pair $\langle p-e \rangle$ into an item $I_{p,e}$ by using $I_{p,e} = I_p * 20 + I_e$, where I_p is the index

of residue p , and I_e is the index of residue e . A residue index is the position number of this residue in the Kyte and Doolittle's increasing hydrophathy index [5]. This pre-processing is necessary as required by the software [156] for association rule mining; (iii) form a transaction for each binding site of one complex with the transformed items; and (iv) mine all frequent association rules from this transactional data set by an association rule mining software developed by [156]. The support and confidence level were set as 7% and 80% respectively during the dependence mining.

4.2.2.2 Relative composition and cooperativity

Associations describe the relationship between antibody and antigen at a high level, but they cannot reveal the connections of residues inside paratopes or epitopes at a low level. Therefore, residue composition and cooperativity of epitope/paratope are introduced. Composition reflects the contribution of a residue in binding to its ligand residue, and cooperativity discloses the tendency of one residue directly neighboured by another residue.

Residue relative composition of paratope is defined as $RC_j^i = P_j^i * 2 * \log_2(P_j^i/Q_j^i)$, where RC_j^i is the relative composition of residue j in CDR i , P_j^i is the composition of residue j over the paratope residues in CDR i , and Q_j^i represents the composition of residue j over all residues in CDR i . For residue relative composition of epitope, it is computed similarly, defined as $RC_j = P_j * 2 * \log_2(P_j/Q_j)$, where RC_j is the relative composition of residue type j , P_j is the composition of residue type j over the epitope residues, and Q_j represents the composition of residue j over all antigen residues.

Regarding paratope residue's cooperativity, it is defined as a ratio $CO_{jk}^i = (P_{jk}^i/Q_{jk}^i)$, where CO_{jk}^i is the cooperativity of residue pair jk with respect to CDR i , P_{jk}^i is the composition of residue pair jk over the paratope residues in CDR i , and Q_{jk}^i is the composition of residue pair jk over all residues in CDR i . Epitope residues' cooperativity is calculated similarly by $CO_{jk} = (P_{jk}/Q_{jk})$, where CO_{jk} is the cooperativity of residue pair jk within antigens, P_{jk} is the composition of residue pair jk over the

epitope residues and Q_{jk} is the composition of residue pair jk over all antigen residues. Here, a residue pair specifies two neighborhood residues.

Following is a toy example of calculating residue relative composition and cooperativity of an epitope. For the sake of simplicity, suppose the epitope is “RKKQD” and residues are uniformly distributed (although they are not), then the relative composition for residue type “R” is computed as $\frac{1}{5} * 2 * \log_2(\frac{1/5}{1/20})=0.8$, and the cooperativity for residue pair “RK” is computed by $\frac{1/4}{1/(20 * 20)}=100$, where 20 is the number of standard types of amino acids.

4.2.3 Sequence-based epitope prediction

Using the model components constructed from the training data of antibody-antigen PDB complexes as described in the above subsection, the epitope in a given antigen sequence can be predicted through the following steps when an antibody sequence is also available:

- Identifying seed paratope residues by using the paratope residues’ relative composition;
- Detecting neighborhood paratope residues of the seeds by using the paratope residues’ cooperativity ratios;
- Specifying candidate epitope residues by using the paratope-epitope interacting preference;
- Inferring new sets of paratope-epitope interacting residue pairs from the observed contact residue pairs by dependence;
- Seeding epitope residues from the candidate epitope residues by utilizing the epitope residues’ relative composition;
- Screening out neighborhood residues of the seeds by using the epitope residues’ cooperativity ratios;
- Completing epitope identification by using a semi-supervised Hidden Markov Model.

The pseudo code for these steps is shown in the algorithm 1. The meaning of the input symbols are:

S^b : antibody sequence;

S^g : antigen sequence;

A_p : association preference obtained in the training stage;

A_d : association dependence generated in the training stage;

τ_{rc} : threshold of residue's relative composition;

τ_{co} : threshold of residue's cooperativity.

4.2.3.1 Identifying seed paratope residues

Given an antibody-antigen complex without structure information, the antibody heavy chain and light chain are numbered by the modified-Chothia numbering scheme [157], then the six segments of CDRs are determined according to the Chothia CDR definition [23].

Every residue within the six CDRs is tested against a CDR-dependent paratope residue's relative composition threshold. Residues that meet the constraint are marked as seed paratope residues. Empirically, the thresholds are set to allow only the top three types of residues to pass the test. In this study, the seed paratope residues instead of CDR residues are used for epitope identification because only CDRs are known and partial CDR residues other than all the CDR residues interact with antigen.

4.2.3.2 Detecting neighborhood paratope residues of the seeds

Having identified the seed paratope residues, the neighborhood paratope residues of these seeds can be screened out by using the paratope residues' cooperativity ratios. More exactly, if a residue i has been marked as a paratope residue, then the cooperativity between its left neighbor $i-1$ and itself is tested against a CDR dependent threshold T_{co} . The left neighbor residue $i-1$ is marked as a paratope residue if and only if $CO_{i-1,i} \geq T_{co}$. This test is also applied to the right neighbor of i . Both the left and right neighbors are required to be in the same CDR as i 's. The neighborhood residues are iteratively expanded during candidate epitope identification. Usually, the threshold is set to be such a proper number that only the top ten percent cooperativity ratios of the pair-wise neighborhood residues can pass the test.

Algorithm 1 PredictEpitope($S^b, S^g, A_p, A_d, \tau_{rc}, \tau_{co}$)

Initialize: $E := \emptyset, C_p := \emptyset, C_e := \emptyset$ **Procedure:**

```
for  $s_i^b$  in  $S^b$  do
  if  $RC_{s_i^b} \geq \tau_{rc}$  then
     $C_p \leftarrow s_i^b$ 
    if  $CO_{s_i^b s_{i+1}^b} \geq \tau_{co}$  then
       $C_p \leftarrow s_{i+1}^b$ 
    end if
    if  $CO_{s_{i-1}^b s_i^b} \geq \tau_{co}$  then
       $C_p \leftarrow s_{i-1}^b$ 
    end if
  end if
end for
flag := 1
while flag do
  flag := 0
  for  $a_p = \langle V_p, V_e, E \rangle$  in  $A_p$  do
    if  $\forall v_p^i$  in  $V_p, v_p^i \in C_p$  then
       $C_e \leftarrow V_e, \text{flag} := 1$ 
    end if
  end for
  for  $a_d = \{\langle p_1^o - e_1^o \rangle, \langle p_2^o - e_2^o \rangle, \dots\} \rightarrow \{\langle p_1^i - e_1^i \rangle, \dots\}$  in  $A_d$  do
    if  $\forall p^o \in C_p \ \&\& \ \forall e^o \in C_e$  then
       $C_p \leftarrow p^i, C_e \leftarrow e^i, \text{flag} := 1$ 
    end if
  end for
end while
for  $s_i^g$  in  $S^g$  do
  if  $RC_{s_i^g} \geq \tau_{rc} \ \&\& \ s_i^g \in C_e$  then
     $E \leftarrow s_i^g$ 
    if  $CO_{s_i^g s_{i+1}^g} \geq \tau_{co} \ \&\& \ s_{i+1}^g \in C_e$  then
       $E \leftarrow s_{i+1}^g$ 
    end if
    if  $CO_{s_{i-1}^g s_i^g} \geq \tau_{co} \ \&\& \ s_{i-1}^g \in C_e$  then
       $E \leftarrow s_{i-1}^g$ 
    end if
  end if
end for
complete  $E$  by using HMM
return  $E$ 
```

4.2.3.3 Specifying candidate epitope residues by preference

A subset of paratope residues can be picked out from the first two steps, then the candidate epitope residues that could interact with these paratope residues are de-

terminated by residues-residues pairing preference which plays a subtle bridging role in linking paratope residues to epitope residues. To achieve this goal, all paratope residues of V'_b in each preference pattern $\langle V'_b, V'_g, E' \rangle$ are checked against the already identified paratope residues, and then the epitope residues V'_g in this pattern are deemed as candidate epitope residues if all of the residues in V'_b are observed.

4.2.3.4 Inferring new sets of contact residue pairs by dependence

Based on the paratope-epitope interacting residue pairs specified by preference, new sets of paratope-epitope contact residue pairs can be implicated by dependence. For each strong dependence pattern $\{\langle p_1^o-e_1^o \rangle, \dots, \langle p_m^o-e_m^o \rangle\} \rightarrow \{\langle p_1^i-e_1^i \rangle, \dots, \langle p_n^i-e_n^i \rangle\}$ mined from the training data set, all the interacting pairs $\langle p_i^o-e_i^o \rangle$ from the observed set are checked against the identified interacting residue pairs. The implied interacting residue pairs $\langle p^i-e^i \rangle$ are considered as true interaction in this antibody-antigen binding site if all of the observed pairs $\langle p^o-e^o \rangle$ have been identified by the preference associations. For example, suppose paratope-epitope interacting residue pairs $S'_{identified} = \{\langle D-K \rangle, \langle S-E \rangle, \langle W-K \rangle\}$ have already been identified, then we can infer that $\langle Y-K \rangle$ is also in this binding site if dependence pattern $P'_{dependence} = \{\langle D-K \rangle, \langle W-K \rangle\} \rightarrow \{\langle Y-K \rangle\}$ is known.

Preference combines paratope and epitope together, while dependence implicates new interactions between the paratope and epitope from the interactions produced by preference, and in return the result generated by dependence can provide new clues for preference. This reciprocal enhancement works well in identifying interactions between an antibody and an antigen. These two steps and the neighborhood residue detecting step are terminated until no more paratope/epitope residues can be identified from a given antibody-antigen sequence pair.

4.2.3.5 Seeding epitope residues

The seed epitope residues are identified through a similar strategy as seed paratope residues identification with only a slight difference. First, when determining the

seed epitope residues, both epitope residues' relative composition and the candidate residues are considered. More exactly, a residue from the antigen sequence is marked as an epitope residue if and only if its relative composition exceeds the preset threshold T_{rc}^{eh} and this type of amino acid has already been fished out by the paratope-epitope associations (preference and dependence). Second, due to that the whole antigen sequence could be antigenic, the residues are checked one-by-one along the whole antigen sequence. Third, besides identifying the seed epitope residues, those residues with very low RC values with respect to a threshold T_{rc}^{el} are marked as non-epitope residues. Empirically, only the best four and the worst three types of residues can pass through these two thresholds respectively.

4.2.3.6 Screening out neighborhood epitope residues

Neighborhood epitope residues are detected by using exactly the same strategy of neighborhood paratope residue detection. One more thing is to screen out the non-epitope residues based on the non-epitope seeds. A neighborhood residue is marked as a non-epitope residue if the cooperativity between this neighbor and itself is within the lowest ten percent of all the observed cooperative residue pairs according to the epitope residues' cooperativity ratios.

4.2.3.7 Completing epitope identification

A stringent threshold based epitope identification strategy can identify a subset of epitope residues and non-epitope residues with high accuracy, but leaves a large number of antigen residues still hidden. Thus we introduce a semi-supervised Hidden Markov Model (HMM) to complete the prediction of epitope residues.

An HMM is a statistical Markov model with unobserved states Π and observed variables X . In our case, the unobserved states are epitope and non-epitope, and the observed variables are the twenty standard amino acids. Figure 4.3 illustrates the diagram of the first-order hidden Markov model for epitope prediction which is quantified by parameters of transition probability and emission probability. The

transition probability and emission probability are calculated by equation (Eq. 4.1) and equation (Eq. 4.2), respectively.

$$\hat{a}_{k,l} = \frac{A_{k,l}}{\sum_{l' \in \Pi} A_{k,l'}} \quad (\text{Eq. 4.1})$$

$$\hat{b}_k(\sigma) = \frac{B_k(\sigma)}{\sum_{\sigma' \in \Omega} B_k(\sigma')} \quad (\text{Eq. 4.2})$$

where $A_{k,l}$ is the number of transitions from state k to state l , Π is a set of states, $B_k(\sigma)$ is the number of emitting symbol σ at state k , and Ω is a set of symbols standing for the set of twenty standard amino acids.

Figure 4.3: First order hidden Markov model for epitope prediction. Arrows represent transition, and the numbers on arrows stand for transition probability from one state to another. Representative emission probabilities for state N and state E are shown at their right hand.

In this work, the first-order HMM and second-order HMM are utilized together to build a better classifier. For the first-order HMM, $\Pi = \{begin, N, E, end\}$, and for the second-order HMM $\Pi = \{begin, NN, NE, EN, EE, end\}$. The symbol N represents the state of non-epitope residue, while E represents the state of epitope residue.

$$\delta_k(i) = \max_{\pi_1, \dots, \pi_{i-1}} P(\pi_1, \dots, \pi_{i-1}, \pi_i = k, x_1, \dots, x_i) \quad (\text{Eq. 4.3})$$

$$\delta_l(i+1) = b_l(x_{i+1}) \max_k \{\delta_k(i) * a_{k,l}\} \quad (\text{Eq. 4.4})$$

$$\begin{cases} \pi_n^* = \arg \max_k \{\delta_k(n)\} \\ \pi_i^* = \arg \max_k \{\delta_k(i) * a_{k,\pi_{i+1}^*}\} \end{cases} \quad (\text{Eq. 4.5})$$

For a given sequence of antigen residues $x = (x_1, x_2, \dots, x_m)$, the most probable sequence of states (or say labels) $\pi^* = (\pi_1^*, \pi_2^*, \dots, \pi_m^*)$ is estimated by the Viterbi algorithm [158]. Suppose $\delta_k(i)$, which is calculated by equation (Eq. 4.3), is the maximal probability of the path ending at sequence position i with state of k when observation is known, then the induction of $\delta_l(i+1)$ is carried out by equation (Eq. 4.4). After determining the maximal probability for each site, then the most likely state for each site is determined by equation (Eq. 4.5) if the state of site i has not been determined yet. If the state of site i has already been determined by the epitope seeding and neighborhood epitope residue screening, then it keeps the state unchangeable. We employ this semi-supervised learning method because of the significantly uneven distributed transition probability from non-epitope residue to epitope residue.

4.2.3.8 Example of epitope prediction

Take the PDB complex 1ZTX as example. To identify the antibody-specific epitopes, the following steps are used to achieve this goal. (i) The candidate paratope residues $C_p = \{Y_{\{H27, H32, H98, L92\}}, D_{\{H31, H100\}}, W_{\{L50\}}\}$ are determined by using the process S1 in the Figure 4.2. The letter in the subscript is the chain type, and the number is the residue position. (ii) The neighborhood residues of C_p that have high cooperativities are selected by the step S2 in Figure 4.2. These residues include $S_{\{H95, H97, L30\}}$ (by using cooperative residues WS) and $G_{\{H99\}}$ (by using YG). After updating, C_p is $\{Y_{\{H27, H32, H98, L92\}}, D_{\{H31, H100\}}, W_{\{L50\}}, S_{\{H95, H97, L30\}}, G_{\{H99\}}\}$. (iii) Partial candidate epitope residues $C_e = \{S_{\{E306\}}, G_{\{E331\}}, E_{\{E390\}}, K_{\{E307\}}, N_{\{E368\}}\}$ are determined with the preferences $\langle \{D\}, \{K\} \rangle, \langle \{S\}, \{E\} \rangle, \langle \{Y\}, \{S, G\} \rangle$ and $\langle \{Y,$

Y}, {N}} according to S3 in the Figure 4.2. (iv) By using the step S4, candidate paratope residue $R_{\{H56\}}$ and epitope residues $Q_{\{E391\}}$ and $D_{\{E333\}}$ can be found by using the dependences $\{\langle\{D, Y\}-\{K\}\rangle\} \rightarrow \{\langle\{W, S\}-\{Q\}\rangle\}$ and $\{\langle\{Y\}-\{K, E\}\rangle\} \rightarrow \{\langle\{R, R\}-\{D\}\rangle\}$. Iteratively applies step (iii) and step (iv) to the antibody and antigen sequences, we get the final candidate epitope residues $C_e = \{Y, S, K, A, F, T, G, D, N, E, Q\}$. (v) Based on the step S5 in Figure 4.2, antibody-specific epitope residues $K_{\{E307\}}$, $D_{\{E333\}}$ and $Q_{\{E391\}}$ are fished out. Consequently (vi), the neighborhood epitope residues $Y_{\{E302\}}$, $S_{\{E306\}}$, $E_{\{E390\}}$ and $T_{\{E330, E390\}}$ are screened out by using the cooperativities KY, DS, QE and QT. Finally (vii), the complete epitope residues $E = \{Y_{\{E302\}}, S_{\{E306\}}, K_{\{E307\}}, A_{\{E308, E365, E367\}}, F_{\{E309\}}, T_{\{E330, E332, E366\}}, G_{\{E331, E389\}}, D_{\{E333\}}, N_{\{E368\}}, E_{\{E390\}}, Q_{\{E391\}}\}$ are identified by using a semi-supervised HMM model as shown in the step S7 of Figure 4.2.

4.3 Experimental Results

4.3.1 Residues' relative composition and cooperativity in paratope and epitope

The paratope and epitope residues' relative compositions show that the same type of residue has diverse tendencies in antigen binding. Residue Y has a very high probability to become a paratope residue in CDR L1 and L2, while this tendency is weakened in other CDRs; residue D favors all the CDRs except H3 which is exactly a reverse tendency for residue G. Intriguingly, the same residue has very different relative composition in the six CDRs and in the antigen. For example, residues Q, H and P are more likely to be epitope residues rather than paratope residues, while residues N, Y, W and S are in favor of paratope residues instead of epitope residues. Particularly, residues Y and W have much higher tendency to serve as paratope residues rather than epitope residues. Our finding on residues Y and W does not agree very well with the result reported by [?]. The reason should be that Y and W are rich in antigen sequence although they have a relatively large proportion in epitopes. In addition, our results clearly illustrate that epitopes prefer very much hydrophilic residues to

hydrophobic residues. Detailed information is presented in Figure 4.4 and Figure 4.5, respectively.

Figure 4.4: Paratope residues' relative compositions in six CDRs.

Figure 4.5: Epitope residues' relative composition.

Our findings suggest that each CDR makes different contribution in antigen binding and that the same residue can exhibit diverse roles and profiles in the six CDRs and in antigen. This confirms our idea of treating the six CDRs separately. The proportion of each CDR's contribution in antigen binding is presented in Table 4.1. The value is calculated based on the count of each paratope residue that interacts with epitope residues. For example, if one paratope residue interacts with two epitope residues then its contribution is counted twice instead of once. By this definition,

Table 4.1: Proportion of each CDR's contribution in antigen binding

CDR	L1	L2	L3	H1	H2	H3
Proportion	0.121	0.066	0.161	0.130	0.187	0.336

H3 makes the highest contribution in antigen binding and L2 presents the lowest. Besides, the three CDRs from the antibody heavy chain take about two thirds contribution in antigen binding. Although a different and simple definition of calculating contribution is used here, our observations are consistent with the findings reported in [159].

The cooperativity ratios of epitope residue pairs and those in the paratope residue pairs in CDR H3 which contributes about one third in the antigen binding are shown in Figure 4.6. The high values at the top-left corner of the epitope cooperativity picture emphasize that epitope residues prefer pairs of hydrophilic residues. With regard to the residue pairs within CDR H3, unexpectedly they cover only a few residues, e.g. Y, W, S, T and G. Although hydrophobic residues are scarce in the paratopes, they tend to cooperate with other residues once they occur in a paratope. This is interesting.

Figure 4.6: Log odds ratio of residues' cooperativity in epitope (left panel) and in CDR H3 (right panel).

4.3.2 Associations in paratope-epitope interacting pairs

The key idea of this work is to use two kinds of associations, the preference and dependence, to address the problem of antibody-specific epitope prediction. As introduced, preference can reflect the contextual relation between paratope and epitope residues in a local manner, while dependence can span this contextual relation between two local sub-regions. The combination of these two rules, which extends association from one-dimension to two-dimension, can well capture the concept of context-awareness in the recognition between paratopes and epitopes.

The residues-residues pairing preferences with frequency (or, support) $\geq 12\%$ are shown in Table 4.2. As an example, the second preference in Table 4.2 means that paratope residues $\{D, Y\}$ interact frequently (15 out of 80) with epitope residue K. Overall, these most frequent paratope preference residues are enriched by Y, D, S and G, while the epitope residues are rich of K, R, Q and E. This observation consolidates the findings given by residues' relative composition in another way. Interestingly, residue G does not have a very high relative composition, however G is rich in H3 which takes about one third contribution in antigen binding. Therefore residue G is also frequent in paratope. We take another example of preference to illustrate how the preference information is used in epitope prediction. Using $\langle \{G, Y\}, \{E\} \rangle$, we can say that once residues G and Y have been identified as paratope residues, then residue E can be marked as a candidate epitope residue.

Table 4.2: Preferences between paratope residues and epitope residues with frequency (or, support) $\geq 12\%$.

Paratope	Epitope	Frequency
$\{D\}$	$\{K\}$	21.3%(17/80)
$\{D, Y\}$	$\{K\}$	18.8%(15/80)
$\{S\}$	$\{E\}$	16.3%(13/80)
$\{D\}$	$\{R\}$	13.8%(11/80)
$\{D, Y\}$	$\{R\}$	12.5%(10/80)
$\{G, Y\}$	$\{E\}$	12.5%(10/80)
$\{S\}$	$\{K\}$	12.5%(10/80)

The dependencies with support level $\geq 10\%$ and confidence level $\geq 90\%$ are listed in Table 4.3. Here support is defined as the proportion of antibody-antigen complexes in the data set that contain the given dependence, and confidence is a probability indicating to what extent that one set of paratope-epitope interacting residue pairs is accompanied by another set in the training data. Taking $\{\langle D-K \rangle, \langle W-K \rangle\} \rightarrow \{\langle Y-K \rangle\}$ as an example to demonstrate the usefulness of dependence, we can say that once paratope-epitope interacting residue pairs $\langle D-K \rangle$ and $\langle W-K \rangle$ are predicted or observed in the binding site, then $\langle Y-K \rangle$ is believed to occur in this site as well with 100% confidence. Compared with preference shown in Table 4.2, one can see that dependence involves a much broader interactions between paratope and epitope residues. Thus the complementary enhancement between these two associations facilitates the accurate antibody-specific epitope identification.

Table 4.3: Dependencies of paratope-epitope interacting residue pairs with support $\geq 10\%$ and confidence $\geq 90\%$

Obs. Pair(s) ^l	Imp. Pair(s) [→]	confidence
$\{\langle D-K \rangle, \langle W-K \rangle\}$	$\{\langle Y-K \rangle\}$	100.0%
$\{\langle T-K \rangle\}$	$\{\langle Y-K \rangle\}$	100.0%
$\{\langle Y-G \rangle, \langle T-K \rangle\}$	$\{\langle Y-K \rangle\}$	100.0%
$\{\langle D-K \rangle, \langle T-K \rangle\}$	$\{\langle Y-K \rangle\}$	100.0%
$\{\langle W-K \rangle\}$	$\{\langle Y-K \rangle\}$	93.33%
$\{\langle Y-G \rangle, \langle S-E \rangle\}$	$\{\langle Y-E \rangle\}$	90.00%
$\{\langle N-G \rangle\}$	$\{\langle Y-G \rangle\}$	90.00%

^l: Observed paratope-epitope interacting pairs;

[→]: Implied paratope-epitope interacting pairs.

An example of the two-dimensional associations between an antigen and its antibody's six CDRs (in PDB:1EGJ) are partially shown in Figure 4.7. The prediction of the epitope residues is proceeded as follows: (i) residues TYR:32:H, TYR:30B:L, ASN:91:L, ASN:92:L, TRP:96:L are fished out in the step of identifying seed paratope residues, (ii) residue GLU:93:L is selected by using cooperativity detection, and (iii) epitope residues R, K and E are identified by utilizing these paratope information,

residue’s relative composition and the paratope-epitope association patterns, such as $\langle\{N\}\cup\{R\}\rangle$, $\langle\{Y\}\cup\{E\}\rangle$, and $\{\langle Y-R\rangle, \langle Y-E\rangle\}\rightarrow\{\langle Y-K\rangle\}$. Other epitope residues within this antigen sequence are identified by using the same strategy. To complete the identification, a semi-supervised HMM is used in the final step to label the undetermined residues.

Figure 4.7: Associations between the paratope and epitope residues within PDB:1EGJ. The epitope residues are rendered by red spheres, while the other antigen residues are colored by green. (Only a small segment of the antigen sequence is shown here.) The paratope residues are colored by yellow and orange for the antibody heavy and light chain respectively, meanwhile other residues within six CDRs are rendered by their surface. The dashed blue lines are used to illustrate the preference between the paratope and epitope residues. This picture is generated by PyMol [160].

4.3.3 Performance of ABepar for antibody-specific B-cell epitope prediction

Leave-one-out cross validation is carried out to evaluate the performance of our method. In every iteration, one complex without its structural information is tested against the model constructed by using the rest 79 PDB complexes in the training. Four metrics are adopted to quantify the performance which are sensitivity (sen), specificity (spe), accuracy (acc), partial AUC and F-measure (F1).

The detailed performance on the 80 complexes are reported in Table 4.4. Figure 4.8 plots a visualization for the performance of sensitivity versus specificity on all

of the 80 complexes. The average sensitivity is 0.434 ± 0.238 , the average specificity is 0.781 ± 0.126 , the average partial AUC is 0.611 ± 0.095 , and the overall accuracy is 0.749 ± 0.103 . We note that the distribution of the non-epitope and epitope residues for every antigen is extremely un-balanced with the ratio 12.65 ± 10.15 on average. This nature of imbalance increases the complexity of the prediction problem and skews the competence of accuracy. Thus the F1-scores are also presented in Table 4.4 (columns 9 and 18) for a more meaningful measurement. To avoid inflated performance caused by the high similarity between antigen sequences, we further analyzed the antigen sequence similarity by using local pairwise sequence alignment with the BLOSUM62 substitution matrix. The pairwise sequence's similarities of all the 80 antigen sequences are shown in Figure 4.9. We found that about 98.9% of the antigen sequence pairwise similarities are less than or equal to 0.25. That is, there is a faint over-fitting problem caused by the sequence similarity and the results well reflect the capability of our method. Although there are still a couple of antigen sequences having a very high similarity, their epitopes are quite different from each other. For instance, the epitope residues of PDB:1A2Y are 19N, 22G, 23Y, 24S, 27N, 102G, 103N, 116K, 117G, 118T, 119D, 120V, 121Q, 124I and 125R, while the epitope residues of PDB:1J1O are 15H, 16G, 19N, 20Y, 21R, 62W, 63W, 73R, 75L, 77N, 89T, 93N, 96K, 97K, 100S, 101D, 102G and 103N. Clearly there are only three overlapping residues between these two antigens, which are 19N, 102G and 103N. According to the performance on PDB:1A2Y and PDB:1J1O shown in Table 4.4, we can conclude that our method can well distinguish the antibody-specific epitopes.

The performance comparison between our method and those methods described by Ponomarenko and Bourne [53] shows that our method outperforms others according to the sensitivity and specificity values. Regarding the comparison with continuous epitope predictors, we note that there is no direct comparison because of the different problems addressed. We also note that the comparison with the methods in [53] is indirect as we are dealing with the new, antibody-specific epitope prediction problem rather than the traditional antibody-ignored epitope prediction problem. Therefore,

Figure 4.8: Sensitivity versus specificity performance on the 80 complexes by ABepar.

Figure 4.9: Pairwise sequence similarity of all the 80 antigen sequences.

a direct comparison under the strictly same platform is presented in the following subsection where a simplified version of ABepar is taken in the comparison.

4.3.4 Simplifying ABepar for antibody-*ignored* B-cell epitope prediction

To our best knowledge, all previous methods for B-cell epitope prediction do not use the context-dependence principle in the recognition between antigens and antibodies.

Table 4.4: The performance for antibody-specific epitope prediction on 80 antibody-antigen complexes by ABepar

PDB	Ab	Ag	#E ¹	#NE ²	sen	spe	acc	AUC	F1	PDB	Ab	Ag	#E	#NE	sen	spe	acc	AUC	F1
1A2Y	BA	C	15	114	0.600	0.711	0.698	0.656	0.316	1ZTX	HL	E	16	85	0.563	0.882	0.832	0.723	0.514
1AR1	CD	B	15	237	0.600	0.865	0.849	0.733	0.321	2ADF	HL	A	15	174	0.000	0.960	0.884	0.480	0.000
1BGX	HL	T	35	793	0.371	0.808	0.790	0.590	0.130	2AEP	HL	A	19	369	0.474	0.767	0.753	0.621	0.158
1BJ1	HL	W	16	78	0.438	0.923	0.840	0.681	0.483	2ARJ	HL	Q	18	97	0.222	0.866	0.765	0.544	0.229
1BQL	HL	Y	12	117	0.000	0.923	0.837	0.462	0.000	2B2X	HL	A	18	170	0.556	0.859	0.830	0.708	0.385
1EGJ	HL	A	11	90	0.909	0.733	0.752	0.821	0.444	2BDN	HL	A	16	52	0.688	0.731	0.721	0.710	0.537
1EO8	HL	A	15	304	0.133	0.849	0.815	0.491	0.063	2CMR	HL	A	17	178	0.235	0.972	0.908	0.604	0.308
1EZV	XY	E	17	168	0.412	0.798	0.762	0.605	0.241	2DD8	HL	S	19	173	0.474	0.832	0.797	0.653	0.316
1FE8	HL	A	19	167	0.263	0.850	0.790	0.557	0.204	2FD6	HL	U	12	237	0.667	0.755	0.751	0.711	0.205
1FJ1	BA	F	17	234	0.588	0.791	0.777	0.690	0.263	2H9G	HL	S	12	52	0.583	0.615	0.609	0.599	0.359
1FNS	HL	A	12	184	0.750	0.783	0.781	0.767	0.295	2J4W	HL	D	11	23	0.636	0.826	0.765	0.731	0.636
1FSK	CB	A	17	142	0.118	0.711	0.648	0.415	0.067	2J5L	CB	A	11	23	0.273	0.826	0.647	0.550	0.333
1H0D	BA	C	16	106	0.875	0.500	0.549	0.688	0.337	2J88	HL	A	9	310	0.778	0.703	0.705	0.741	0.130
1IQD	BA	C	16	140	0.188	0.829	0.763	0.509	0.140	2JEL	HL	P	15	70	0.533	0.686	0.659	0.610	0.356
1J1O	HL	Y	18	111	0.444	0.739	0.698	0.592	0.291	2NR6	DC	A	14	315	0.357	0.905	0.881	0.631	0.204
1JHL	HL	A	11	118	0.818	0.729	0.736	0.774	0.346	2NY7	HL	G	22	268	0.500	0.840	0.814	0.670	0.289
1JPS	HL	T	19	181	0.579	0.867	0.840	0.723	0.407	2NYY	DC	A	20	1247	0.350	0.874	0.866	0.612	0.076
1JRH	HL	I	15	80	0.067	0.975	0.832	0.521	0.111	2Q8B	HL	A	18	274	0.500	0.723	0.709	0.612	0.175
1LK3	HL	A	18	118	0.389	0.763	0.713	0.576	0.264	2QQK	HL	A	12	529	0.167	0.711	0.699	0.439	0.024
1N8Z	BA	C	16	565	0.000	0.894	0.869	0.447	0.000	2QQN	HL	A	12	142	0.500	0.718	0.701	0.609	0.207
1NCA	HL	N	20	369	0.250	0.780	0.753	0.515	0.094	2R56	HL	A	21	138	0.381	0.703	0.660	0.542	0.229
1NFD	HG	D	12	227	0.833	0.784	0.787	0.809	0.282	2UZI	HL	R	18	148	0.722	0.561	0.578	0.642	0.271
1NMB	HL	N	19	369	0.421	0.751	0.735	0.586	0.134	2VXQ	HL	A	14	78	0.286	0.782	0.707	0.534	0.229
1NSN	HL	S	17	121	0.647	0.554	0.565	0.601	0.268	2VXS	IM	A	7	79	0.429	0.443	0.442	0.436	0.111
1OAZ	HL	A	17	98	0.235	0.867	0.774	0.551	0.235	2VXT	HL	I	17	139	0.412	0.806	0.763	0.609	0.275
1OB1	AB	C	13	83	0.769	0.771	0.771	0.770	0.476	2W9E	HL	A	15	84	0.800	0.571	0.606	0.686	0.381
1ORS	BA	C	10	122	0.000	0.975	0.902	0.488	0.000	2ZCH	HL	P	16	221	0.563	0.855	0.835	0.709	0.316
1OSP	HL	O	20	231	0.400	0.827	0.793	0.614	0.235	3B2U	HL	A	17	178	0.412	0.787	0.754	0.600	0.226
1OTS	CD	A	9	435	0.000	0.945	0.926	0.473	0.000	3B9K	LH	B	19	98	0.368	0.837	0.761	0.603	0.333
1P2C	ED	F	16	113	0.750	0.690	0.698	0.720	0.381	3BN9	FE	A	22	219	0.545	0.653	0.643	0.599	0.218
1R3J	BA	C	13	90	0.538	0.844	0.806	0.691	0.412	3CVH	HL	A	15	259	0.467	0.780	0.763	0.624	0.177
1RJL	BA	C	12	83	0.667	0.663	0.663	0.665	0.333	3D9A	HL	C	18	111	0.444	0.739	0.698	0.592	0.291
1SY6	HL	A	11	157	0.727	0.631	0.637	0.679	0.208	3D85	BA	C	16	117	0.000	0.915	0.805	0.458	0.000
1TQB	BC	A	17	85	0.706	0.506	0.539	0.606	0.338	3EOA	HL	I	14	165	0.429	0.909	0.872	0.669	0.343
1UJ3	BA	C	18	187	0.556	0.818	0.795	0.687	0.323	3G04	BA	C	23	205	0.435	0.854	0.811	0.645	0.317
1V7M	HL	V	16	129	0.563	0.775	0.752	0.669	0.333	3GBN	HL	B	12	161	0.000	0.770	0.717	0.385	0.000
1W72	HL	A	14	260	0.429	0.846	0.825	0.638	0.200	3GI9	HL	C	16	421	0.188	0.938	0.911	0.563	0.133
1WEJ	HL	F	11	93	0.636	0.333	0.365	0.485	0.175	3GRW	HL	A	33	173	0.515	0.671	0.646	0.593	0.318
1YJD	HL	C	14	104	0.357	0.894	0.831	0.626	0.333	3H42	HL	B	22	470	0.182	0.855	0.825	0.519	0.085
1YYM	RQ	P	13	286	0.308	0.916	0.890	0.612	0.195	3HI6	HL	A	22	158	0.455	0.665	0.639	0.560	0.235

¹: number of epitope residues, and ²: number of non-epitope residues.

Antigen is the only issue and data that is taken for the prediction. Given an antigen, the prediction just tells whether a residue is antigenic or not. The prediction result is an unordered union of the antigenic residues of the different epitopes that are formed when this antigen interacts with different antibodies. Such an approach can narrow down from the whole antigen sequence to the smaller antigenic region, but it cannot pinpoint the constituent residues of a specific epitope which is usually much shorter than the antigenic region of an antigen. An example is shown later. Therefore, such a generic epitope prediction is quite different from the problem we are addressing.

To make a fair comparison between our method and the previous ones, we compile a new data set from PDB in which every antigen has multiple antibody partners. The purpose of constructing such a data set is aimed at alleviating the underestimation of current epitope predictors caused by incomplete epitope exploration [34, 35, 14]. This data set is shown in Table 4.5.

Table 4.5: Performance evaluation for ABepar, ElliPro, DiscoTope, and SEPPA on an epitope relatively completely explored data set

antigen	corresponding PDB ID	#Ab [†]	F-measure			
			F1 ^a	F1 ^e	F1 ^d	F1 ^s
Hen egg white lysozyme	1A2Y 1BQL* 1BVK[C,F] [‡] 1C08 1DQJ 1FDL 1G7H 1G7I 1G7J 1G7L 1G7M 1IC4 1IC5 1IC7 1J1O 1J1P 1J1X 1KIP 1KIQ 1KIR 1MLC[E,F] 1NDG 1NDM 1P2C[C,F] 1UA6 1UAC 1VFB 2DQC 2DQD 2DQE 2DQF[C,F] 2DQG 2DQH 2DQI 2DQJ 2EIZ 2EKS 2IFF 2YSS 3D9A	44	0.747	0.683	0.519	0.722
Vascular endothelial growth factor	1BJ1[V,W] 1CZ8[V,W] 2FJG[V,W]	6	0.464	0.571	0.540	0.567
Ubiquitin	3DVG[X,Y] 3DVN[U,V,X,Y]	6	0.367	0.375	0.456	0.430
Influenza virus neuraminidase	1NCD 1A14 1NMB 1NCB 1NCC 1NCA	6	0.244	0.336	0.387	0.378
Von willebrand factor	1FE8[A,B,C] 2ADF	4	0.343	0.343	0.150	0.483
Outer surface protein A	1FJ1[E,F] 1OSP	3	0.381	0.347	0.379	0.367
Integrin alpha-L	3HI6[A,B] 3EOA[I,J]	4	0.385	0.381	0.386	0.479
Prion protein	1TPX 1TQB 1TQC 2W9E	4	0.630	0.355	0.526	0.480
Tissue factor	1JPS 1UJ3	2	0.426	0.00	0.241	0.195

^a: ABepar, ^e: ElliPro, ^d: DiscoTope, and ^s: SEPPA. [†]: number of antibodies interacting with this antigen. Some of the antibodies are mutants of its wild type. [‡]: multi-antigen chains in the same PDB complex. *: it is Bobwhite quail lysozyme, but it has very similar sequence and structure with hen egg white lysozyme.

Three structure-based epitope prediction methods ElliPro [37], DiscoTope [35], and SEPPA [40] are used in the comparison. ElliPro predicts antigenic residues by means of the protrusion index (PI) which is calculated based on antigen structures. The performance here by ElliPro was calculated over all the predicted discontinuous epitopes with the default parameters. With regard to DiscoTope, the epitope residues from a testing antigen structure is predicted based on features that are extracted from

a set of training antigen structures, such as residue solvent accessibility, residue spatial distribution and residue propensity. The performance of DiscoTope was achieved by using the default cutoff threshold -7.7. SEPPA identifies the antigenic residues by clustering the triangle residues of unit patches, and the result shown here was generated with the default parameter of 1.8.

To identify antibody-ignored epitope residues by our method, we modified our computational process by skipping the first three steps which are specially for identifying paratope associated candidate epitope residues.

From the F1-scores shown in Table 4.5, we can see that our method is competitive to, sometimes much better than the three structure based methods even though we take only sequence information as input. In particular, our method can provide much better results on identifying epitope residues from those more completely explored antigen sequences. For example, we can correctly identify 37 of the 55 epitope residues and include only 7 non-epitope residues as epitope residues from hen egg white lysosyme, achieving an F1-score of 0.75. This performance is much better than 0.68 by ElliPro, 0.52 by DiscoTope, and 0.72 by SEPPA. All these demonstrate that even without 3D structural information, the simplified version of ABepar for antibody-ignored prediction is also powerful to identify epitope residues.

We once again stress the importance and complexity of the antibody-specific epitope prediction problem. We show some extreme pairwise dissimilarities between different epitopes of an antigen. Taking hen egg white lysozyme again, this antigen can interact with as many as 44 different antibodies (some are mutants of the wild type antibody, for instance 1G7H, 1G7I, 1G7J, 1G7L, 1G7M are mutants of 1VFB), and there are 55 antigenic residues in total in the 129 residues of this antigen. However, the average number of residues over these antibody-specific epitopes is only 16.3 ± 2.0 . The average pairwise epitope similarity of these 44 epitopes is 0.359. More remarkably, 58.4% of these pairwise epitope similarities are < 0.1 , i.e. with less than 2 residues in common between two epitopes. For example, the epitope residues of PDB complex 1A2Y and 2IFF are distributed on two entirely non-overlapping locations on

the surface of hen egg white lysozyme as shown in Figure 4.10. Here the similarity between two epitopes A and B is calculated by $|A \cap B|/(|A| * |B|)$, where $A \cap B$ represents the overlapping epitope residues, and $|X|$ means the residue number of X . These findings do manifest the extreme reconfiguration of epitope when an antigen binds to different antibodies. Therefore, it is important to specify the corresponding antibody when predicting epitope residues on a given antigen. We do not mean that identifying epitope regardless of antibody is less important, in fact we emphasize that identifying an epitope with the specification of its antibody in some applications is necessary as also stressed by [30].

Figure 4.10: Reconfiguration of epitope when hen egg white lysozyme interacts with different antibodies (PDB:1A2Y and PDB:2IFF). Epitope residues in PDB:1A2Y are colored by magenta while those epitope residues within PDB:2IFF are colored by orange. All the rest residues are rendered by cyan..

4.4 Discussion

In this study, the antibody-specific epitope prediction problem is newly formulated. It is a more meaningful but more challenging problem than the traditional epitope prediction problem that does not consider antibody information in the prediction. There are two main differences. First, antibody-antigen PDB complexes are used in our model training, and antibody-antigen sequence pairs without structure information are required for our model testing. However, just antigen sequences or structures without antibodies are needed for epitope identification by the traditional epitope

prediction models. Second, the output of our model is a set of antigenic residues that interact with the specific antibody excluding all other antigenic residues that interact with other antibodies. However, the output of the traditional model is a residue union of all possible epitopes without any clarity to label the constituent residues of any epitope.

Antibody-specific epitope prediction is first time proposed by this work, hence there is no direct comparison between the new problem and the traditional problem. Therefore a simplified version of our method is introduced for a fair comparison, and it is tested on a data set containing complexes which have been relatively completely explored for epitopes. The purpose of creating this data set is to avoid as much as possible the underestimation problem caused by incomplete epitope exploration which has been argued by many researchers [34, 35, 14]. The more complete epitope exploration of one antigen is, the more convincing of the result conducted on the data. Therefore hen egg white lysozyme, whose epitopes are relatively completely explored, is taken as example to demonstrate the prediction capability of different models.

In this study, we adopted a semi-supervised hidden Markov model to complete the process of epitope prediction. This approach can alleviate the wrong predictions caused by the stringent cutoff thresholds of residues' relative composition and cooperativity, and it also overcomes the thorny problem raised by the imbalanced distribution of non-epitope residues and epitope residues.

The proposed antibody-specific epitope prediction model is novel and promising, but it still suffers from the scarce of training data set. Thus the leave-one-out cross validation was used to evaluate this model. In every iteration, one sample was left out for testing and the remaining samples were taken for model construction. Along with the increasing available samples, we believe our model will show more powerful performance in antibody-specific epitope prediction.

4.5 Summary

We summarize the whole work as follows. In this study, we have proposed a method called **ABepar** to predict antibody-specific epitopes with a high accuracy and broad applicability. This method is trained on a relatively small set of PDB complexes that contains antibody-antigen structures, but it can be applied to any antibody-antigen sequences without 3D structural information. The novel idea of this method is based on the context-awareness in the recognition between paratope and epitope which is implemented by two kinds of associations (preference and dependence). In addition, the semi-supervised HMM also plays an important role in completing the process of epitope identification in comparison with our previous study where HMM did not take into consideration [161]. It can capture the sequential relationship within epitope residues at one hand, and also it overcomes the insurmountable obstacles caused by the imbalance distribution of the epitope and non-epitope residues at the other hand. Furthermore, we also assessed the performance of our model on epitope prediction when antibody information is not given. The comparison results show that our sequence-based epitope prediction method has a competitive performance compared with structure-based prediction methods which have much smaller applicability.

Chapter 5

Coupling Graph and Antibody-specific Epitope Prediction

Two-dimensional association has been introduced to capture the principle of context-awareness in Chapter 4. Although it is sufficient to represent the interactions between paratopes and epitopes, contacts within paratope and epitope cannot be revealed. Therefore, a new concept—coupling graph—is introduced to include both inter-connections between paratopes and epitopes and intra-connections within paratopes and within epitopes. Based on the newly designed coupling graph mining algorithm, a finer model is built to mining antibody-specific epitopes. We report the efficient new algorithm for frequent coupling graph mining and its application on antibody-specific epitope prediction in this chapter.

5.1 Background

Graph representation and graph data analysis have been widely used in many bioinformatics studies. Protein-protein interaction network is a well-known example. In a protein-protein interaction network, its nodes denote unique proteins and its edges represent physical contacts between the pairs of proteins [162]. Another example is about genetic regulatory networks. The nodes in a genetic regulatory network represent genes, and the edges stand for gene regulatory relations, such as a relation that gene A inhibits gene B, or a relation that gene B activates gene C [163].

A more interesting type of graphs used in bioinformatics are those graphs which contain two sets of nodes having different meanings. For example, a gene-phenotype association network contains two different sets of nodes. Each node in one set represents a gene, while each node in the other set stands for a phenotype. The edges in such a network also have different meanings, and can be grouped into: (i) those relation edges within the genes only, (ii) those similarity edges within the phenotypes only, and (iii) the association edges between the genes and phenotypes [164]. An illustration of a gene-phenotype network is shown in Figure 5.1 (a). It can be seen that the nodes in this network belong to two categories (gene and phenotype) and that the edges have different meanings (i.e., inter-gene interactions, inter-phenotype similarities, and gene-phenotype associations). This kind of two-layer graph complexes are referred to as *coupling graphs* in this work. Each layer in a coupling graph is defined as a subgraph and every node in one layer has at least one edge connecting with a node in the other layer. A coupling graph is not necessarily a bipartite graph, as there usually exist many edges within each layer subgraph of a coupling graph. However, a coupling graph can be easily reduced to a bipartite graph by removing all of the edges in the same layer subgraph.

Many other bioinformatics problems also involve coupling graphs. For example, an antibody-antigen interaction complex [166] can form a coupling graph when the residues (the basic elements linked in a 3-dimensional antigen or antibody) are represented by nodes, and the physical contacts between the residues are represented by edges. As shown in Figure 5.1(b), the interactions of some residues in the antibody-antigen complex (PDB entry 1TJG) forms a coupling graph, where the nodes are the contacting residues and the edges are the residue contacts. As another example, the expression regulation network of microRNAs and genes can be constructed as a coupling graph. One layer of this coupling graph represents the similarity network of the microRNAs' expression, while the other layer is a gene expression similarity network. The edges between these two networks are functional regulation relationships [167], as shown in Figure 5.1(c).

Figure 5.1: Examples of coupling graphs in bioinformatics. (a) is a diagram of gene-phenotype association network, in which genes are represented by light green nodes and phenotypes are depicted by light purple nodes. The solid lines are interactions within the genes or phenotypes, while the dash lines are associations between the genes and phenotypes. (b) shows partial interactions between antigen gp41 and antibody 2F5. The interactions between this antibody and antigen are represented by dash lines. (c) illustrates the role of microRNAs in regulating TGF β signaling pathway. The regulations between the microRNAs and their targets are represented as dash lines [165].

Compared to generic bipartite graphs, the integrative notion of coupling graphs has advantages for deciphering biological associations, identifying structural motifs in protein complexes, predicting context-awareness binding sites of proteins, and constructing binding partners for an input protein [168, 161].

First, the two special subgraphs (the two layers) in a coupling graph can preserve topology information. For example, in the two layers—the paratope and epitope subgraphs—of an antibody-antigen interacting complex (Figure 5.1(b) and Figure 5.2), important biological and geometrical information are preserved for the hydrophilic core structure surrounded by a neutral rim in the paratope subgraph (upper layer), and for the hydrophobic core structure enclosed by a hydrophilic rim in the epitope subgraph (lower layer). Secondly, the edges between the two subgraphs of a coupling graph capture the contact details between the nodes of the two subgraphs. As shown in Figure 5.2, the residue VAL166 of the antigen and GLU96 of the antibody clearly shows the contact between this antibody and antigen. Thirdly, the unification of the contacts within the two subgraphs and the contacts between the two subgraphs not only keeps the topology within the subgraph and inter-contacts between the subgraphs, but also uncovers systematical structures of the contacts. For instance, the

interactions shown in Figure 5.2 not only reveal special organizational patterns of epitope residues and paratope residues but also imply the complementarity contacts (i.e., hydrophobic core of epitope versus hydrophilic core of paratope, and hydrophilic rim of epitope versus neutral rim of paratope). However, if bipartite graphs are used for data representation, many important neighborhood and topological information and biological properties in the two subgraphs of coupling graphs are lost.

Figure 5.2: A hydrophobic residue which is surrounded by hydrophilic residues in an antigen interacts with a hydrophilic residue that is encircled by neutral residues in an antibody. This hydrophobic residue is shown in orange, while the hydrophilic residues are colored in blue. White color residues are those residues with a neutral hydrophobicity

This work is focused on efficient mining of coupling subgraphs that occur frequently in a coupling graph database. There exist efficient algorithms for mining frequent subgraphs from a generic graph database, including AGM [169], FSG [170], MoFa [171], gSpan [172], FFSM [133] and Gaston [173]. However, these algorithms cannot be directly used to mine frequent coupling subgraphs from a coupling graph database. If a coupling graph is treated as a generic graph, difficulties will arise when the aforementioned subgraph miners are used to find frequent coupling subgraphs. On the one hand, a frequent subgraph generated by these algorithms may contain nodes from only one layer of a coupling graph or include irrelevant subgraphs. For example, the frequent subgraph “1—3” in Figure 5.3 is not a frequent coupling subgraph because it only exists in the upper-layer graph of the graphs H_2 and H_3 , and the frequent subgraph “2—1—3” contains a subgraph “1—3” which is not a *frequent* coupling

Figure 5.3: Some frequent coupling subgraphs and frequent subgraphs of a graph data set. A solid line represents an edge within a layer subgraph, while a dash line represents an edge between the two-layer subgraphs of a coupling graph.

subgraph. The detailed description of frequent coupling subgraph is presented later. On the other hand, a coupling graph $A=(G_1^A, G_2^A, E^A)$ is isomorphic to a coupling graph $B=(G_1^B, G_2^B, E^B)$ if they are regarded as generic graphs, but their corresponding constituent graphs may not be isomorphic, i.e., G_1^A may not be isomorphic to G_1^B and G_2^A is not necessary to be isomorphic to G_2^B .

We propose new algorithms and make the following contributions to the efficient mining of frequent coupling subgraphs from coupling graph databases. We define and formulate the new concepts related to coupling graphs. We design an efficient algorithm to mine frequent coupling subgraphs from coupling graph database by novel graph transformation and graph restoration techniques. We prove that the transformation and restoration are reversible. We also evaluate the efficiency of our algorithm by comparing it to the performance of generic subgraph mining algorithms on large-scale real data. Furthermore, we are able to predict antibody-specific B-cell epitope with higher accuracy by using the proposed coupling graph model.

5.2 Related Works

Frequent coupling subgraph mining is quite different from generic subgraph mining and bipartite graph mining. However, several graph mining algorithms are closely related, and some of their ideas are useful for developing coupling graph mining algorithms.

AGM [169] is a representative Apriori-based approach for mining frequent subgraphs, which can identify both connected and unconnected graphs. It employs an adjacency matrix to represent graphs, and breadth-first search (BFS) to discover frequent graph patterns. Other Apriori-based algorithms have also been proposed for mining frequent subgraphs, including FSG [170], gFSG [174] and DPMine [175]. Although the same strategy is adopted by these algorithms, different graph representation and repeat count ideas are used. The BFS search strategy performs strong pruning during subgraph expansion; however, it consumes huge volume of memory. Therefore, the depth-first search (DFS) method, which takes less memory, is developed. MoFa [171] uses a fragment-local numbering scheme to expand subgraphs. Although structural pruning and molecular knowledge are used to reduce support calculation, MoFa still produces duplicates. Another well-established algorithm for frequent subgraph mining based on pattern growth is gSpan [172]. gSpan uses the minimum DFS code to represent each graph and only expands a frequent subgraph with minimum DFS code. Instead of recording the embedding list for all the frequent subgraphs, gSpan only keeps the transaction identifiers for discovered patterns. Therefore, it significantly saves the memory consumption. The canonical adjacency matrix (CAM) graph representation is used by FFSM [133] to mine frequent subgraphs. This algorithm uses an embedding list to record the discovered frequent patterns in CAM format, which avoids graph isomorphism testing. Gaston [173] incorporates a progressive model, from path, tree to graph, to reduce the mining time. Graph isomorphism testing is only performed on subgraphs instead of trees and paths. Various graph expansion and support counting methods have been proposed to mine frequent subgraphs; which, however, cannot be directly used to mining frequent coupling subgraphs as the edges in a coupling graph have different meanings.

Besides frequent subgraph mining, attempts have been made on correlated graph pattern mining. The *correlated graph search* is formulated by Ke *et. al.* [176], in which Pearson’s correlation coefficient is used to measure the correlation between graphs. Later on, the *frequent correlated subgraph pairs* mining algorithm is established by Ke

et. al. [177], in which a theoretical bound on the minimum correlation is determined to discover correlated subgraph pairs. HSG [178] is proposed to discover frequent *hyperclique patterns* in graph databases, where a hyperclique pattern is defined as a set of items with high affinity measured by h-confidence [179]. Another related work is *pairs of graph pattern* mining, which discovers rules to classify graph pairs by estimating the tight upper bound on a statistical metric. An attempt has also been made on frequent subgraph-subsequence pair mining [180]. However, these problems are different from coupling graph mining—the correlated graphs are separate in the former, while they are tightly connected in the latter.

In this paper, we formulate the concept of coupling graphs to represent connected graph pairs, such as protein-protein interaction complexes, gene regulatory networks, and microRNA-gene regulation networks. We also design an efficient coupling graph mining algorithm, as existing subgraph mining methods are not directly usable.

5.3 Definitions and Algorithms for Mining Frequent Coupling Subgraphs

A graph G is an ordered pair denoted by $G = (V, E)$, where V is a set of nodes and $E \subseteq V \times V$ is a set of edges. An edge e in E is denoted by $e = (v_i, v_j)$.

Definition 5.1 A coupling graph H is a graph complex denoted by $H = ((V^1, V^2), (E^1, E^2, E^{12}))$, where $E^{12} \subseteq V^1 \times V^2$, V^1 and E^1 forms a subgraph G^1 , V^2 and E^2 forms a subgraph G^2 , and the two subgraphs satisfy that $\forall v_i^1 \in V^1, \exists v_j^2 \in V^2$ such that $(v_i^1, v_j^2) \in E^{12}$, and $\forall v_j^2 \in V^2, \exists v_i^1 \in V^1$ such that $(v_i^1, v_j^2) \in E^{12}$.

We note that every node v^1 in G^1 is required by definition to connect to at least one node v^2 in G^2 for a coupling graph H . This constraint guarantees that all the nodes are involved in the interaction between the two subgraphs of a coupling graph. This modeling constraint is motivated by some real application needs. For example, to characterize antibody-antigen interactions, only paratope residues in an antibody and epitope residues in an antigen are needed, and the rest can be ignored.

A coupling subgraph is a coupling graph which is a subgraph of a coupling graph.

Definition 5.2 *A coupling graph H is frequent in a coupling graph database $\mathbb{H}$ if H is a coupling subgraph in not less than δ number of coupling graphs in $\mathbb{H}$.*

We take the following three steps to mine frequent coupling subgraphs: (i) transform a coupling graph into a generic graph; (ii) mine frequent subgraphs from the transformed generic graphs by using an existing graph mining method; and (iii) restore the coupling graphs from the set of transformed frequent subgraphs. The detailed description for each step is presented in the following subsections.

5.3.1 Transformation of a coupling graph into a generic graph

For a coupling graph $H = ((V^1, V^2), (E^1, E^2, E^{12}))$, we transform it into a generic graph $H' = (V', E')$ in two steps: node construction and edge construction.

- *Node construction*: for each edge $e_i^{12} = (v_i^1, v_i^2)$ in E^{12} , we use $v_i^1 v_i^2$ as a label to form a new node of V' ;
- *Edge construction*: two nodes $v'_i = v_i^1 v_i^2$ and $v'_j = v_j^1 v_j^2$ of V' are connected by an edge with a label l defined as:

$$l = \begin{cases} 11, & \text{iff } v_i^1 = v_j^1 \text{ \& } (v_i^2, v_j^2) \in E^2 & (a) \\ 11, & \text{iff } (v_i^1, v_j^1) \in E^1 \text{ \& } v_i^2 = v_j^2 & (b) \\ 11, & \text{iff } (v_i^1, v_j^1) \in E^1 \text{ \& } (v_i^2, v_j^2) \in E^2 & (c) \\ 01, & \text{iff } (v_i^1, v_j^1) \notin E^1 \text{ \& } (v_i^2, v_j^2) \in E^2 & (d) \\ 10, & \text{iff } (v_i^1, v_j^1) \in E^1 \text{ \& } (v_i^2, v_j^2) \notin E^2 & (e) \end{cases}$$

For the label “ l ” of an edge in E' , the first code “1” means that the edge (v_i^1, v_j^1) exists in E^1 , and the first code “0” represents that there is no edge between v_i^1 and v_j^1 , and similarly for the meaning of the two second codes. Condition (a) represents that one same node in G^1 connects with two different nodes v_i^2 and v_j^2 in G^2 between which there is an edge, while condition (b) is a similar situation of condition (a). In

Figure 5.4: Coupling graph transformation. (a) is the original coupling graph, where solid lines represent edges within G^1/G^2 and dash lines represent edges between G^1 and G^2 . (b) is the transformed generic graph.

condition (c), the two nodes v_i^1 and v_j^1 in G^1 have an edge, and the two nodes v_i^2 and v_j^2 in G^2 also have an edge. Condition (d) and (e) are situations where only one edge is present. If none of the two pairs of nodes is connected in G^1 or G^2 , then there is no edge between the newly constructed nodes.

Figure 5.4 shows an example using the above definition to transform a coupling graph (Figure 5.4(a)) into a generic graph (Figure 5.4(b)). For ease of presentation, we use superscript 1, 2, or 12 to represent coupling graphs before our transformation and use those with superscript ' to represent generic graphs after the transformation.

Theorem 5.1 *Transformation from H to H' is precise and complete. Preciseness means that all the edges and nodes in H' correspond to some nodes and/or edges in H . Completeness means that all the edges and nodes information in H is contained in H' without information loss.*

Proof: Preciseness is obvious by construction. As to completeness, we show all five components ($V^1, V^2, E^{12}, E^1, E^2$) of H are captured in H' . Wrt V^1 , by definition of coupling graphs, for each v^1 in V^1 , there is a v^2 in V^2 such that the edge (v^1, v^2) is in E^{12} . By the node construction step, there is a node $v^1 v^2$ in V' , thus capturing v^1 . A similar argument shows that every node in V^2 is also captured by a node in V' . Wrt E^{12} , for each edge (v^1, v^2) in E^{12} , by the node construction step, it is captured by the node $v^1 v^2$ in V' . Wrt E^1 , for an edge (v_i^1, v_j^1) in E^1 , by definition of coupling graphs, there are nodes v_i^2 and v_j^2 in E^2 , not necessarily distinct, such that (v_i^1, v_i^2)

and (v_j^1, v_j^2) are in E^{12} . By the edge construction step, there is an edge $(v_i^1 v_i^2, v_j^1 v_j^2)$ in E' with label “10” or “11”. This implies (v_i^1, v_j^1) is captured. Wrt E^2 , a similar argument shows that every edge in E^2 is also captured by an edge in E' . Therefore, the transformation from H to H' is also complete.

Transforming a coupling graph into a generic graph can facilitate mining frequent coupling subgraphs by avoiding generating a huge number of irrelevant graphs. In the following subsection we discuss how to restore a coupling graph from its transformed generic graph, and prove that the transformation is reversible.

5.3.2 Restoration of coupling graph from transformed generic graph

For a transformed generic graph $H'=(V', E')$, we take the following steps to restore its coupling graph $H=((V^1, V^2), (E^1, E^2, E^{12}))$:

- *node restoration*: for each node $v'=v^1 v^2$ of V' , we add v^1 to V_1 and v^2 to V_2 ;
- *edge restoration*: for each node $v'=v^1 v^2$ of V' , we add (v^1, v^2) to E^{12} ; for each edge $e'=(v_i', v_j', l)$ of E' , we add (v_i^1, v_j^1) to E^1 if l is “10” or “11” and add (v_i^2, v_j^2) to E^2 if l is “01” or “11”, where $v_i'=v_i^1 v_i^2$, $v_j'=v_j^1 v_j^2$ and $l \in \{01, 10, 11\}$.

The following result is obvious.

Theorem 5.2 *Transformation from H to H' is reversible, i.e., all the nodes and edges of H can be recovered from H' without introducing additional nodes or edges.*

The above analysis shows that the transformation from H to H' is reversible.

5.3.3 Frequent coupling subgraph mining

For a coupling graph database $\mathbb{H}$, we first transform each coupling graph into a simple graph, then we use subgraph mining methods to obtain frequent subgraphs from the transformed graph database, finally the transformed frequent subgraph are restored to

---

Algorithm 2 Mining frequent coupling graph

Procedure mineCouplingGraph($\mathbb{G}, \delta$)

```

 $\mathbb{G}' := \emptyset, \mathbb{H} := \emptyset, \mathbb{H}' := \emptyset$ 
for  $G := ((V^1, V^2), (E^1, E^2, E^{12}))$  in  $\mathbb{G}$  do
     $\mathbb{G}' \leftarrow \text{transformGraph}(G)$ 
end for
 $\mathbb{H}' := \text{frequentSubgraphMining}(\mathbb{G}', \delta)$ 
for  $H'$  in  $\mathbb{H}'$  do
     $\mathbb{H} \leftarrow \text{restoreGraph}(H')$ 
end for
return  $\mathbb{H}$ 
    
```

Function transformGraph(H)

```

 $V' := \emptyset, E' := \emptyset$ 
for  $e^{12} = (v^1, v^2)$  in  $E^{12}$  do
     $V' \leftarrow v^1 v^2$ 
end for
for  $e^1 = (v_i^1, v_j^1)$  in  $E^1$  do
    for  $e^2 = (v_i^2, v_j^2)$  in  $E^2$  do
        if  $(v_i^1, v_i^2) \in E^{12} \ \&\& \ (v_j^1, v_j^2) \in E^{12}$  then
             $E' \leftarrow (v_i^1 v_i^2, v_j^1 v_j^2, 11)$ 
        else if  $(v_i^1, v_i^2) \in E^{12} \ \&\& \ (v_j^1, v_j^2) \notin E^{12}$  then
             $E' \leftarrow (v_i^1 v_i^2, v_j^1 v_j^2, 10)$ 
        else if  $(v_i^1, v_i^2) \notin E^{12} \ \&\& \ (v_j^1, v_j^2) \in E^{12}$  then
             $E' \leftarrow (v_i^1 v_i^2, v_j^1 v_j^2, 01)$ 
        end if
    end for
end for
return  $H' = (V', E')$ 
    
```

---

obtain the frequent coupling subgraphs. The pseudocode for mining frequent coupling subgraphs is shown in Algorithm 1.

The time complexity of subgraph mining is in proportion to the product between the total number of subgraphs and the complexity of graph isomorphism testing. The main part of the time cost of subgraph mining is for subgraph isomorphism testing, which is NP-complete [181]. The proposed algorithm of coupling graph mining significantly reduces the time cost and memory consumption by using graph transformation which avoids the generation of many irrelevant subgraphs. The time complexity of graph transformation for a data set with n coupling graphs is in proportion to

---

```

Function restoreGraph( $H'$ )
   $V^1 := \emptyset, V^2 := \emptyset, E^1 := \emptyset, E^2 := \emptyset, E^{12} := \emptyset$ 
  for  $v' = v^1 v^2$  in  $V'$  do
     $V^1 \leftarrow v^1, V^2 \leftarrow v^2, E^{12} \leftarrow (v^1, v^2)$ 
  end for
  for  $e' = (v'_i, v'_j, l)$  in  $E'$  do
    if  $l == "11"$  then
       $E^1 \leftarrow (v^1_i, v^1_j), E^2 \leftarrow (v^2_i, v^2_j)$ 
    else if  $l == "10"$  then
       $E^1 \leftarrow (v^1_i, v^1_j)$ 
    else if  $l == "01"$  then
       $E^2 \leftarrow (v^2_i, v^2_j)$ 
    end if
  end for
  return  $H = ((V^1, V^2), (E^1, E^2, E^{12}))$ 
    
```

$\sum_i^n N_1^i \cdot N_2^i$, where N_1^i is the number of edges of graph G_1^i and N_2^i is the number of edges of graph G_2^i .

5.3.4 Transformation and restoration with duplicate node labels

In the above study, we assume that all the node labels in G^1 or in G^2 of a coupling graph H are unique but allowing some identical labels between some nodes in G^1 and G^2 . In practice labels usually have duplicates in V^1 or in V^2 . For example, an interface of protein-protein interacting complex is composed of residues which have twenty types only in nature, hence duplicate residues usually exist in interfaces.

Duplicate labels do not affect coupling graph transformation and transformed generic graph mining, but it does impede graph restoration because whether a new node should be created or not is unknown when a node with a duplicate label is brought in. We take some additional steps to solve coupling graph mining with duplicate labels: (i) map each node in V^1 or in V^2 to a unique label and transform the relabeled coupling graph into generic graph; (ii) mine frequent subgraphs from the transformed generic graph with new labels; (iii) restore each transformed frequent subgraph into a coupling graph and recover the original labels according to the mapping table.

5.4 Efficiency Results on Real Protein Complex Data

In this section, we report the performance of our algorithms in terms of efficiency and memory consumption. We also report the number of irrelevant subgraphs generated by existing subgraph mining algorithms to understand why the high efficiency of our algorithms are achieved by the graph transformation approach. The coupling graphs we used in the evaluation are real data compiled from the Protein Data Bank (PDB) [58]. The purpose is to comprehend to what extent the new algorithm is better than the existing algorithms when dealing with real-world problems.

5.4.1 Coupling graph database compilation

As mentioned in the Introduction, when one protein interacts with another protein, the interacting part of the two proteins can be represented as a coupling graph by using nodes to represent the contacting residues and using edges to represent the close contacting distance. Protein-protein interaction complexes are stored at the widely used PDB database where the 3-dimensional co-ordinates information of atoms of every residues is available.

Protein-protein interaction complexes that satisfy the following criteria are retrieved from PDB: (i) the macromolecular type is protein only without DNA and RNA; (ii) the number of protein chains is larger than two; (iii) the length of each protein (chain) is larger than or equal to 30; and (iv) the X-ray resolution of one complex is less than 3Å. As a result, 29,418 PDB entry with 129,305 protein-protein interaction pairs are obtained. With the removal of those similar chains under BLAST [125] maximum pair-wise sequence similarity threshold of 90%, 9,781 PDB entries containing 10,511 protein-protein interaction (PPI) complexes are left and used for our algorithm efficiency study.

The coupling graph database for the 10,511 protein-protein interaction complexes are built in two steps: (i) determine interfacial residues (i.e., the nodes of a coupling

graph) and connections between the two interfacial surfaces (edges between the two layers of a coupling graph) from a PPI complex by using Euclidian distance of 2.75 Å plus residues' radii [182]; (ii) build connections of residues within each interfacial surface (i.e., edges within each of the two subgraphs of a coupling graph) by using qhull [130]. The average number of nodes and the average number of edges for the coupling graphs in our graph database are 65.3 ± 43.2 and 205.9 ± 155.8 , respectively.

Our experiments were carried out on a platform with Ubuntu 11.04 operating system, 4G physical memory and 8 cores with each of 2.67GHz.

5.4.2 Efficiency results

Frequent subgraphs of the coupling graph database without graph transformation are mined using gSpan [172] which is implemented in the ParMol package [132], while frequent coupling subgraphs with graph transformation are mined by using LCM [134]. As required by LCM, we take a transactional database to represent the coupling graph database. Each transaction represents a transformed coupling graph and the items in this transaction is the entire set of nodes and edges of the transformed graph (duplicate items are preserved and are relabeled in order). Each frequent itemset corresponds to a transformed coupling graph, which can be restored to its equivalent original coupling graph form. The equivalence between a coupling graph and its transformed generic graph has been proved in the above section.

Figure 5.5 shows the running time of mining frequent subgraphs from the database with 10,511 coupling graphs on the original graphs and also on the transformed graphs. It is clear that mining coupling subgraphs from the transformed graphs is remarkably faster than mining subgraphs from original coupling graphs. For example, mining frequent subgraphs from the original coupling graph database costs 3084 seconds at the minimum support of 3%, while the cost is only 147 seconds on the transformed graphs with the same support level. In addition, Figure 5.6 also indicates that using graph transformation consumes significantly less memory.

Figure 5.5: Running time comparison of mining frequent coupling subgraphs from the original coupling graphs and from the transformed coupling graphs.

Figure 5.6: Memory consumption comparison of mining frequent coupling subgraphs from the original coupling graphs and from the transformed coupling graphs.

5.4.3 Irrelevant frequent subgraphs generated by gSpan

We note that the frequent subgraphs mined from the coupling graph database by using gSpan [172] covers a large number of frequent non-coupling subgraphs. For instance as shown in Figure 5.3, the frequent subgraphs generated by gSpan with support of 2 are “1”, “2”, “3”, “1—2”, “1—3”, “2—1—3”; however, only “1—2” is frequent coupling subgraphs. the frequent subgraphs “1”, “2” and “3” are not frequent coupling subgraphs because no two-layered graph exists, while the subgraph “1—3” is not a frequent coupling subgraph because the frequency of the coupling subgraph is only one (in graph H1, and the subgraph in H2 and H3 is not coupling

Figure 5.7: Numbers of connected frequent subgraphs generated from the original graphs and from the transformed graphs.

graphs). Since “1—3” is not frequent coupling subgraph, the frequent subgraph “2—1—3” is also not a frequent coupling sugraph. Therefore, to eliminate these irrelevant frequent subgraphs still takes a plenty of time, especially when extremely huge number of frequent subgraphs are produced. In contrast, every frequent subgraph generated from the transformed graphs is an equivalent of a coupling subgraph thus, no such tremendous cost is needed.

Figure 5.7 shows the number of connected frequent subgraphs generated from the coupling graph database and from its transformed graph database. The averaged number of frequent subgraphs generated by gSpan is about eight times the number of connected frequent subgraphs produced from the transformed graph database. Therefore, about 88% of the frequent subgraphs generated by gSpan are irrelevant frequent subgraphs. It is a very heavy task to remove the irrelevant frequent subgraphs, especially when the minimum support is low.

5.4.4 Statistics on the frequent coupling subgraphs

A coupling graph can be connected or disconnected. For example, the coupling graphs shown in Figure 5.8(a) and Figure 5.8(c) are connected coupling graphs, while the coupling graph shown in Figure 5.8(b) is disconnected. The number of frequent coupling subgraphs of a coupling graph database can be extremely large, partially because some

frequent connected coupling subgraphs can be combined to form new and frequent coupling subgraphs.

Figure 5.8: Examples of frequent coupling patterns shared by antibody-antigen complexes.

Table 5.1 shows the total number of frequent coupling subgraphs and frequent connected coupling subgraphs with respect to different minimum support from our data set containing 10,511 coupling graphs. It can be seen that when the support level is set as minimum 10%, there are still hundreds of connected frequent coupling subgraphs in our graph database. It implies that there are many regular coupling graph patterns in the protein-protein interactions.

Table 5.1: The numbers of frequent coupling subgraphs and frequent connected coupling subgraphs in the database.

min supp	# frequent coupling subgraphs	# frequent connected coupling subgraphs
4%	6,545,268	11,445
5%	1,379,285	3,966
6%	410,476	1,697
7%	153,045	1,040
8%	66,991	593
9%	32,357	455
10%	17,060	383

5.4.5 Scalability

To test the efficiency of our algorithm under different volumes of data, we apply the algorithm to some subsets of our coupling graph databases with varying sizes. We randomly choose 2000, 4000, 6000, 8000, and 10000 coupling graphs from our database to form new databases. Another new database contains 20,000 coupling graphs randomly

taken from our database twice. We report the time cost and memory consumption by our algorithm for each case.

Figure 5.9: Time cost of frequent coupling subgraph mining under different sizes of coupling graph databases.

Figure 5.10: Memory consumption of frequent coupling subgraph mining under different sizes of coupling graph databases.

The time cost and memory consumption of mining frequent coupling subgraphs under different minimum support on various databases of transformed coupling graphs are shown in Figure 5.9 and Figure 5.10. It is clear that the computation cost increases linearly with the increment of database size.

5.4.6 Discussion

Based on the above experimental results, we can see that graph transformation significantly improves the efficiency of coupling subgraph mining on both time cost and memory usage. The improvements are attributed to the following reasons. Firstly, the transformation technique avoids the generation of huge numbers of irrelevant subgraphs. As an example, the number of frequent connected subgraphs is 6 at the support level of 2 in Figure 5.3, while the number of frequent coupling subgraphs is only 1 in the transformed graph of Figure 5.3. Secondly, the transformation technique maps the number of node labels from the order of n to n^2 , which significantly reduces the efforts of isomorphism testing. For example, in our real data the maximal number of labels of graph nodes are 20 (there are only 20 standard residue types) however, the maximal number of node labels are 210 ($=C_2^{20}$) after graph transformation. Last but not least, mining frequent coupling subgraphs on the item sets of the transformed graph can yield disconnected coupling subgraphs, such as the one in Figure 8(b). However, the disconnected coupling subgraphs cannot be obtained by using the well-known subgraph mining tools, such as gSpan.

Coupling subgraph mining by using graph transformation can remarkably improve mining efficiency in solving real problems, such as identifying frequent structural pairing motifs of protein complexes and detecting import regulators in microRNA-gene association networks. However, side effect may also exist when the number of node labels is very small, especially when no node label is available, in which case graph transformation only increases the problem complexity. Therefore, this new algorithm prefers graphs with a number of different node labels, but not solving graphs without node labels.

5.5 Application: Pattern Discovery and Epitope Prediction in Antibody-antigen Complexes

Frequent coupling subgraphs within protein-protein complexes can reveal important patterns shared by multiple complexes. These patterns have potential to discover con-

tact residues or to construct binding partners with the property of “coupling”. In this section, we show an application of using coupling graphs for detecting significant patterns shared by antibody-antigen interacting complexes to identify antibody-specific B-cell epitopes.

5.5.1 Frequent coupling subgraph patterns in antibody-antigen complexes

We collected 156 antibody-antigen structural complexes from the PDB with antigen pair-wise sequence similarity less than 0.5 and the number of mutated antibody residues larger than 30. By using the coupling graph mining algorithm described in this study, we obtained 2,472 frequent coupling subgraphs from the 156 antibody-antigen complexes with the minimum support of 5%. Figure 5.8 shows three examples of significant structural patterns that are common in antibody-antigen complexes. Among these examples, only Figure 5.8(a) and (c) can be found by the existing subgraph mining algorithm, while Figure 5.8(b) cannot be identified by them, but it can be found by our algorithm.

One of our findings from our experiments in coupling subgraph mining is that the residue Tyrosine (Y) in the antibodies is predominantly preferred in partnership with a hydrophilic residue to perform antigen binding. However, in the antigens the favored residues for antibody binding are charged residues (both positively charged or negatively charged), especially residues Arginine (R), Lysine (K), Aspartate (D) and Glutamate (E). Although the preferences of residue contacts within antibodies or within antigens have been explored elsewhere [161, 183], none of them can be used to discover structural patterns between antibodies and antigens.

5.5.2 Epitope prediction using frequent coupling graphs in antibody-antigen complexes

Epitope prediction is an important research for many applications in bioinformatics [184]. However, existing methods for epitope prediction overlook the principle of

context-awareness in antibody-antigen interactions, and thus may not reflect biological reality [30]. Although two-dimensional association model has been designed by us to predict antibody-specific epitopes in line with the principle of context-awareness, it only reveals the interactions between paratope residues and epitope residues but not within paratope/epitope residues [109]. Therefore, we built a model incorporating frequent coupling subgraphs within antibody-antigen complexes to complete the context-awareness representation and thus to predict antibody-specific epitopes.

The main idea is using frequent coupling subgraphs of antibody-antigen complexes from a training data set to identify the candidates of antibody-specific epitope residues of the testing data set, and then the true epitope residues are completely determined by some statistical measures. In particular, we substitute the steps of candidate epitope residues identification from using two-dimensional association to using coupling graph (see the step 4.2.3.3 and step 4.2.3.4 in Chapter 4 for reference). The other steps are exactly the same.

Experimental results conducted on the data set of [109], which is the only existing data set for antibody-specific epitope prediction, show that our coupling graph-based model is much better than the association-based model [109] on epitope prediction. Figure 5.11 shows the performances comparison between the coupling graph-based and the two-dimensional association-based methods for antibody-specific epitope prediction. The t-test p-values between the two models on averaged sensitivity, accuracy and f-score are $3.0e-3$, $4.5e-3$ and $7.8e-4$, respectively. These significant p-values suggest that our method is indeed more accurate on epitope prediction than the association-based model. Based on the comparison between the two models, we can conclude that the improvement comes from the refinement of context-awareness representation. Intuitively, coupling graph is better than two-dimensional association in that the paratope/epitope residues consider the spatial constraint in coupling graph while this is not included in the two-dimensional association.

As an example, the antigen lysozyme C with PDB entry 1P2C, as shown in Figure 5.12, contains 129 residues in which 16 are epitope residues and 113 are non-

Figure 5.11: Performance comparison between the proposed method and ABepar on antibody-specific B-cell epitope prediction.

epitope residues. The coupling graph model can successfully identify 11 epitope residues while only introducing 10 non-epitope residues; however, the association model includes 35 non-epitope residues although 12 epitope residues are correctly predicted. The prediction accuracy of the coupling graph-based mode and association-based mode on this antigen are 0.884 and 0.698, respectively. Frequent connected coupling subgraphs which are used to identify these epitope residues are shown in Figure 5.13. Interestingly, the seed epitope residues are mainly introduced by the frequent coupling subgraphs with paratope residues are D and Y.

5.6 Conclusion

Coupling graph is new and very useful graphical model for representing intrinsic associations between pairs of subgraphs in a complex. In bioinformatics, coupling graphs can be used to reveal the structural interactions of protein-protein interacting complexes, gene-phenotype association networks, microRNA-gene expression regulatory networks, and so on. The frequent coupling subgraphs of these coupling graph databases play an important role in discovering the essential patterns hidden in the coupling graph databases. However, mining the frequent coupling subgraphs from a

Figure 5.12: An antibody-antigen interacting coupling graph extracted from the PDB entry 1P2C, where the paratope and epitope residues are shown. The epitope residues of the antigen are rendered as stick, while paratope residues of the antibody are represented by surface. The inter-edges between paratope and epitope are represented by dash orange lines.

Figure 5.13: Frequent connected coupling subgraphs which are used for identifying antibody-specific epitope residues of the antigen in PDB entry 1P2C.

coupling graph database is very challenging, as existing subgraph mining algorithms perform poorly on coupling subgraph mining. The huge number of irrelevant subgraphs generated by the existing algorithm is the big hurdle to the efficiency. To mine frequent coupling subgraphs with high efficiency, we have introduced a new algorithm by using an novel graph transformation and restoration technique. In this work, a coupling graph is transformed into a generic graph, and then subgraph mining is conducted on the transformed coupling graphs. We have proved that the transformation and restoration are equivalent. Experimental results carried out on a data set containing 10,511 coupling graphs have demonstrated that the proposed algorithm not only

shortens the mining time, but also reduces the memory usage. The usefulness of frequent coupling subgraphs has also been demonstrated on identifying antibody-specific B-cell epitopes. Result has shown that the coupling graph-based model outperforms the association-based mode in antibody-specific epitope prediction.

Chapter 6

Graph-based Antibody-agnostic Epitope Prediction

Antibody-specific epitope prediction has been extensively studied in consonance with the principle of context-awareness, i.e., two-dimensional association model built in Chapter 4 and coupling graph model constructed in Chapter 5. However, antibodies are not always available when an antigen is presented. Therefore, a new graph-based antibody-agnostic model is conceived to predict one or multiple epitopes, which also satisfies the context-awareness principle.

6.1 Background

A B-cell epitope is a set of spatially proximate residues in an antigen that can be recognized by antibodies to activate immune response [2]. B-cell epitopes are of two types: about 10% of them are linear B-cell epitopes and about 90% are conformational B-cell epitopes [16, 17, 18]. Linear epitopes differ from conformational epitopes in the continuity of their residues in primary sequence—residues of a linear-epitope are contiguous in primary sequence while the residues in a conformational-epitope are not. B-cell epitope prediction is a long-studied problem of high complexity which aims to identify those residues in an antigen forming one or multiple epitopes.

This problem has attracted tremendous efforts over the last two decades because of its significance in prophylactic and therapeutic biomedical applications [150]. Various approaches have been proposed to identify conformational epitopes, for example, by

clustering accessible surface area (ASA) [34], by combining residues' ASA and their spatial contact [35], by grouping surface residues under their protrusion index [37], by aggregating epitope-favorable triangular patches [40], or by using naïve Bayesian classifier on residues' physicochemical and geometrical properties [185]. Far more approaches have been developed for predicting linear epitopes. Some of these methods use just a single feature of residues—such as hydrophobicity, polarity, or flexibility only—to detect the crests or troughs of propensity values as epitopes [63, 66]. The other methods take complicated machine learning approaches, including artificial neural network, Bayesian network, and kernel methods, to tackle this problem [12, 13, 41, 42, ?, 43, 33]. With these tremendous efforts, this field of research has been advanced significantly and the best AUC performance has reached to 0.644 [40]. However, there are still many limitations in existing methods, and huge room for performance improvement exists.

A limitation of those methods using geometrical properties [35, 37, 185] is that they only favor epitopes with protrusive shapes, not identifying epitopes in other formations such as planar shapes. In fact, many epitopes are shaped at plain areas of antigens. For example, the surface atoms of the epitope of *paracoccus denitrificans* cytochrome C oxidase is very flat in 3-dimensional space with a root mean square deviation (rmsd, an index of non-planarity) of only 1.08Å (Figure 6.1). The second limitation of the conventional methods is that they do not separate or distinguish between any two epitopes in an antigen when multiple epitopes exist. They only tell which residue of the antigen is antigenic, but not tell to which epitope it belongs to. That is, only a union of all antigenic residues, irrespective to specific epitopes, are just predicted. This is a limitation because multiple epitopes are possible at the same antigen [109]. For instance, there exist two non-overlapping epitopes on the ubiquitin antigen: one of them has a very smooth surface with a non-planarity of 1.04Å, while the other stretches out remarkably with a non-planarity of 3.14Å. See Figure 6.2 for more details of their constituent residues. In this work, we propose a graph-based model to improve the prediction performance by identifying both protrusive and planar epitopes and

by detecting multiple epitopes in an antigen if applicable (i.e., identifying all of the epitopes instead of the union of all epitope residues).

Figure 6.1: A hydrophilic island with a hydrophobic core in the binding site of *paracoccus denitrificans* cytochrome C oxidase interacting with an antibody (PDB complex 1AR1). The hydrophilic residues are rendered by blue and the hydrophobic residues are colored by orange. The shade of colors represents hydrophobic intensity. This figure is produced by using Chimera [186]

The use of graph model is motivated by the following biological observations. First, the tight packing of residues at each protein surface can be effectively represented by a graph. Second, epitope/non-epitope residues form particular patches separately on antigen surfaces, displaying distinct subgraphs of their own characteristics. As shown in Figure 6.1, the binding site shapes like a hydrophilic island (a hydrophilic subgraph) containing a hydrophobic core (a hydrophobic subgraph). It can be also seen that this island subgraph is surrounded by hydrophobic non-epitope residues which form a non-epitope subgraph. Third, the distinction between protrusive and planar epitopes can be manifested by the change of weights in the connections between residues.

Our graph-based prediction method consists of three main steps: construct a weighted graph to represent an antigen surface, cluster the nodes of this weighted graph, and learn a label (epitope or non-epitope) for each cluster. Specifically, we take the idea of Delaunay tessellation and use Qhull [130] in the implementation of Delaunay tessellation to construct a protein surface graph. The weights of the edges in this graph are determined by χ^2 test statistics combined with a log odds ratio of

Figure 6.2: An example of a multiple-epitope antigen. The ten residues with color orange are the residues of the epitope on the ubiquitin antigen (chain X) in PDB complex 3DVG, while the fourteen residues colored in forest green are the residues of the epitope on the ubiquitin antigen (chain Y) in PDB complex 3DVG.

each edge type. An edge type is determined by the amino acid types of the interacting residue pair. Then, a Markov CLustering algorithm (MCL) [187] is used to partition the entire graph into subgraphs based on the weights of the edges and the graph topology. MCL simulates flows in a network with two operations: expansion and inflation. Expansion increases homogeneity of nodes within one subgraph, while inflation evaporates inter-flow between different subgraphs and amplifies flow within subgraphs. These ideas mimic properties of residues connecting within an epitope, within a non-epitope, or between an epitope and a non-epitope. Thus, we can divide the weighted antigen surface graph into a good set of subgraphs for the subsequent learning algorithms to predict these subgraphs as epitopes or non-epitopes accurately.

Experimental results on a set of 92 non-redundant antibody-antigen complexes compiled from the Protein Data Bank (PDB) [58] show that our proposed graph model improves the performance of B-cell epitope prediction significantly and, it is also able to identify multiple epitopes as well as predict epitopes with various geometrical formations. For ease of reference, we refer to the proposed **B-Cell epiTope** prediction model as **BeTop**. Our data and web server for B-cell epitope prediction are available at <http://sunim1.sce.ntu.edu.sg/~s080011/betop/index.php>.

6.2 Materials and Methods

6.2.1 Collection of antigen protein data

Protein complexes satisfying the following criteria were retrieved from the PDB on May 14th, 2011: (i) the macromolecular type is protein only, no DNA, RNA, or their hybrid complexes; (ii) the number of protein chains in an asymmetric unit of one complex is larger than two; (iii) the length of every chain is larger than or equal to 30; (iv) the X-ray resolution of one complex is less than 3Å; and (v) the structure title contains at least one of the following terms: antibody, Fab, Fv, or VHH. We obtained 622 antibody-antigen complexes. As transformed and redundant chains in the raw PDB complexes may cause noise effect on the results, we removed all of the transformed chains and duplicate chains. One antigen chain is considered as a duplicate if there exists one pair-wise chain similarity between this chain and one of the other in the data set larger than 80%, a threshold widely used to remove redundant antigens [53]. Removal of duplicate chains by pair-wise chain similarity may filter out multi-epitope antigens, but it can significantly reduce more noise data because the number of non-epitope residues is extremely larger than the number of epitope residue for an antigen. Asymmetric units in each complex that do not have structural difference were also excluded from our consideration. Finally, a non-redundant data set containing 92 antibody-antigen complexes were collected for our model training and testing. Since the sequence similarity of 80% is a bit high for discriminating antigen sequences, the pair-wise sequence similarities for all the 92 antigen sequences are calculated to illustrate the homogeneity of these sequences; as shown in Figure 6.3. Similarity is defined as the proportion of identically aligned positions against the shorter sequence length. Results show that 99.7% pair-wise sequence similarities are smaller than 0.4. Epitope residues on antigen surfaces were determined by using the Euclidian distance of 4Å for every antigen-antibody PDB complex, following the traditional method for determining epitope residues [35].

Figure 6.3: Distribution of pair-wise antigen sequence similarities.

6.2.2 Construction of epitope prediction model

The training phase of our prediction method has the following steps: (i) antigen surface triangulation, (ii) weight calculation for edges, (iii) clustering on the nodes of the graphs, and (iv) supervised learning for distinguishing between epitope subgraphs and non-epitope subgraphs. The details of each step are presented below.

6.2.2.1 Triangulation of an antigen surface

A surface graph of an antigen structure is built in two steps: determine the surface atoms of the antigen, and then build an atom-level graph for these surface atoms and upgrade into a residue-level graph. To obtain surface atoms of an antigen with 3D coordinates, we compute each atom’s ASA by using NACCESS [188] with the default probe size. Those atoms with $ASA \geq 10\text{\AA}^2$ are defined as surface atoms. A graph of these surface atoms is constructed as per Delaunay triangulation rule which has

been commonly used to construct protein surface graph [128]. To upgrade an atom-level graph into a residue-level graph, we ignore connections of the atoms within the same residue, e.g., ignore connection between C_α and C_β of Alanine; and then merge multiple atom connections between two different residues into one edge, e.g. merge the connection between O_{D1} of Aspartate and C_{G1} of Isoleucine and the connection between O_{D2} of Aspartate and C_{G2} of Isoleucine into one edge. Atom connections that have Euclidian distances larger than $6\text{\AA}$ are also removed. Then, residues are distinguished by their positions—i.e., two residues are considered different if they have different positions even when they are of the same amino acid type. Figure 6.4 shows a graph of an antigen after triangulation, in which nodes are surface residues and edges represent residues' spatial relations.

Figure 6.4: Diagram of an antigen surface graph. Nodes are residues and edges represent residues' spatial relation. Blue nodes are epitope residues, and orange ones are non-epitope residues. Dash lines are boundary edges between two clusters, while solid lines are edges within a cluster.

6.2.2.2 Weight calculation for edges

The weight between two residue types x and y within an epitope subgraph or within a non-epitope subgraph in our graph database is given by

$$W_{xy} = \alpha \cdot \overline{W}_{xy}^{\chi^2} + (1 - \alpha) \cdot \overline{W}_{xy}^L, \quad (\text{Eq. 6.1})$$

where $\overline{W}$ is the normalized value of W , and $\overline{W}_{xy}^{\chi^2}$ and $\overline{W}_{xy}^L$ are the χ^2 test and the log odds ratio of the frequencies of xy (edge between x and y) between epitope clusters

and non-epitope clusters, respectively. $W_{xy}^{\chi^2}$ and W_{xy}^L are calculated by using

$$W_{xy}^{\chi^2} = \sum_c (N_{xy}^c - E_{xy}^c)^2 / E_{xy}^c \quad (\text{Eq. 6.1a})$$

$$W_{xy}^L = \log(P_{xy}/Q_{xy}) \quad (\text{Eq. 6.1b})$$

where $c \in \{\text{epitope}, \text{non-epitope}\}$, N_{xy}^c is the number of edges with type xy and label c in our training data, E_{xy}^c is the number of expected edges with type xy and label c , P_{xy} is the probability that a pair of residues xy in epitopes, and Q_{xy} is the probability that a pair of residues xy in non-epitopes. P_{xy} is given by

$$P_{xy} = N_{xy} / \sum_{x'} \sum_{y'} N_{x'y'},$$

where, N_{xy} is the number of residue pairs xy in a cluster, i.e., the number of edges connecting two nodes with one node labeled as x and the other as y . Q_{xy} is calculated by the same way of computing P_{xy} .

The weight calculation for boundary edges is very innovative. A boundary edge is an edge connecting an epitope residue and a non-epitope residue. We group all of the boundary edges (e.g. dashed black lines in Figure 6.4) in our graph database as a class, and take all epitope edges (e.g. solid blue lines in Figure 6.4) as the other class. Then, we apply Equation (6.1a) and (6.1b) to calculate the weights W'_{xy} for the boundary edges by setting $c \in \{\text{boundary\_class}, \text{epitope\_class}\}$. This process is also applied with regard to the boundary class and non-epitope class (e.g., edges with solid orange lines in Figure 6.4) to determine weights W''_{xy} for the boundary edges. In other words, W'_{xy} and W''_{xy} are determined by using the exactly same equations as computing W_{xy} , with substitution of the relevant class label c . The weight of a boundary edge xy is finally set as W'_{xy} or W''_{xy} whichever is larger. Those boundary edges with heavy weights (larger than a threshold W_0) are *definitely* boundary edges between epitope and non-epitope subgraphs. We remove them to sharpen the distinction in the later clustering step and supervised learning. Boundary edges might change to another set when different computational methods are used to define epitope residues, such as

using accessible surface area larger than $1\text{\AA}^2$ upon binding with an antibody [34, 36] and distance threshold of $4\text{\AA}$ [35, 38], $5\text{\AA}$ [48] or $6\text{\AA}$ [49]. However, Ponomarenko *et. al.* have shown that epitope residues have no significant difference when various criteria are used to capture epitope residues [53].

As a few number of large weights can pull all weight values towards zero after normalization, we further contrast normalized weights W_{xy} to amplify important weights and suppress trifling weights by

$$f(W) = \frac{1}{1 + \left(\theta \frac{W}{1-W}\right)^{-\gamma}}.$$

where θ and γ are optimized as 3 and 3 in this study.

Since there are only 20 different standard residue types, the total number of different weights between two residue types is 210 ($=C_2^{20} + C_1^{20}$).

6.2.2.3 Clustering on nodes in an antigen surface graph

Antigen surface graphs are constructed by Qhull with weights W on edges determined by the procedure above. We then use mcl [187] (an implementation of the MCL algorithm with inflation coefficient r of 1.8) to cluster the nodes and edges of every antigen graph into subgraphs. In the MCL algorithm, the graph of an antigen surface residues is represented by a square matrix M , where each row/column represents a surface residue and the value of each entry is the weight of these two residue types. In the expansion stage of MCL, M is expanded as the normal product of itself; during the inflation stage, the matrix M undertakes Hadamard power with coefficient r followed by normalization. This two steps keep on in iteration until an equilibrium state is reached, i.e., when expansion and inflation do not alter the matrix any more.

The subgraphs of an antigen surface clustered by MCL are not always clean and some subgraphs may contain a mixture of epitope residues and non-epitope residues. To clean up the training data, we consider a subgraph as an epitope subgraph if the number of epitope residues in this subgraph is larger than the number of non-epitope

residues and, as a non-epitope subgraph if no or very few (say, at most two) epitope residues show up. Subgraphs with other situations are considered as noise data which are overlooked during model training. We adopt this strategy because of the small number of epitope residues. We note that this approach is tolerant to false positives, but is sensitive to false negative.

6.2.2.4 Supervised learning for distinguishing epitope subgraphs and non-epitope subgraphs

By using mcl, each antigen surface graph is clustered into a number of subgraphs. To distinguish between epitope subgraphs and non-epitope subgraphs, we design a feature vector to represent all of these subgraphs in our training data. Each subgraph is transformed into a feature vector with 1770 dimensions, which comprises 20 ($=C_1^{20}$) dimensions of single residues, 210 ($=C_1^{20}+C_2^{20}$) dimensions of residue pairs, and 1540 ($=C_1^{20}+C_2^{20} \cdot C_1^2+C_3^{20}$) dimensions of residue triangles. A single-residue feature takes the weighted summation of χ^2 test and log odds ratio on the frequencies of the residue type between epitope clusters and non-epitope clusters, which is similar to the calculation of the weight of a pair of residue types shown in Equation (6.1). A residue-pair feature takes the weight of this edge in the subgraph as its value, and a triplet feature takes the average weight of the three edges in the subgraph as its value.

The number of nodes in a subgraph is very small (15 on average); but the dimension of each vector is very large (1770). Therefore, each vector is very sparse and, some features even have no differentiability between epitope subgraphs and non-epitope subgraphs. Hence, feature selection is conducted to maximize classification performance. The feature selection was done by using the LIBSVM [189] feature-selection module targeting at maximizing classification f-score. As a result, 144 from the 1770 features are chosen for classifying epitope subgraphs from non-epitopes subgraphs.

Due to the extreme imbalance between the epitope residue number and non-epitope residue number for an antigen surface graph (15 and 120 on average in our data set), the number of non-epitope subgraphs far exceeds the number of epitope subgraphs as

generated by mcl. To address this imbalance problem, a two-stage supervised learning, multi-SVM classification and trust-reliable voting, is taken to accomplish the distinction between epitope and non-epitope subgraphs. The number of SVM classifiers is automatically determined by the proportion between non-epitope subgraphs and epitope subgraphs. Based on our data set in this work, the number of SVM classifiers is nine. For each classifier in the first stage, a parameter grid-search is carried out on a balanced training data set to maximize model performance, while in the second stage the final decision is voted and determined by

$$y = \text{sgn} \left(\sum_i w_i \cdot f(x_i) \cdot \delta_i - \theta_0 \right), \quad (\text{Eq. 6.2})$$

where

$$\delta_i = 1 - g(\tau_0 - |p_{x_i}^0 - 0.5|) \cdot h \left(\sum_i \text{sgn}(p_{x_i}^0 - 0.5) \right)$$

$$g(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases} \quad h(x) = \begin{cases} 1, & x \neq 0 \\ 0, & x = 0 \end{cases} \quad \text{sgn}(x) = \begin{cases} 1, & x > 0 \\ -1, & x < 0 \end{cases},$$

and the symbol annotations are as follows:

- y : epitope/non-epitope label for a sample predicted by the model;
- w_i : weight of classifier i computed by its performance;
- $f(x_i)$: label for a sample x determined by classifier i in the first level;
- $p_{x_i}^0$: probability of classifier i that predicts sample x as non-epitope;
- δ_i : determinant of classifier i . δ_i is 0 when the classifier i is dubious and other confident classifiers exist.

θ_0 is a threshold to filter out non-epitope residues, and τ_0 is used to control to what extent we trust the classifier.

6.2.3 Prediction of epitopes for an unknown antigen

Given an antigen with 3D coordinate information, the following steps are taken to identify one or multiple epitope for this antigen: (i) calculate each atom’s ASA by using NACCESS, and filter out those atoms with ASA less than $10\text{\AA}^2$; (ii) construct an atom-level graph by using Qhull and upgrade it to a residue-level graph; (iii) assign weights to all edges of this residue graph, where the weights are those determined during the training; (iv) cluster this undirected and weighted graph into exclusive subgraphs using mcl; and (v) transform every subgraph into a feature vector, and predict its label by the well-trained two-stage classification model. Epitope residues are the residues within those subgraphs which are predicted as epitope. Two epitope subgraphs can be merged together if they are connected in the original surface graph.

6.3 Results and Discussions

Our graph-based method BeTop made remarkable improvement on B-cell epitope prediction in comparison to the state-of-the-art methods. First, BeTop shows significant improvement on overall prediction accuracy. Second, BeTop is capable of predicting epitopes located at both protrusive and planar surface areas. Third, BeTop is able to identify multiple epitopes if an antigen contains them. The detailed results of all these are presented below together with highlighting those features that distinguish epitope subgraphs from non-epitope subgraphs.

6.3.1 Significant improvement of prediction accuracy

Four performance metrics are adopted to evaluate model performance—viz., sensitivity (sen), specificity (spe), f-score, and accuracy (acc). Due to the imbalance nature in the composition of non-epitope residues and epitope residues in an antigen, accuracy is not competent to measure model performance. Instead, f-score is more appropriate to evaluate BeTop’s performance and to compare with other models.

Ten fold cross validation is applied to measure the overall performance of BeTop on the 92 non-redundant antigen-antibody PDB complexes. The f-score comparison

Figure 6.5: B-cell epitope prediction performance by BeTop, DiscoTope, SEPPA, and ElliPro. The optimal parameters α , θ_0 and τ_0 are set to 0.3, 0.3 and 0.05, respectively.

Figure 6.6: Comparison of partial AUC values generated by DiscoTope, SEPPA, ElliPro and BeTop.

between BeTop and the state-of-the-art epitope prediction methods DiscoTope [35], SEPPA [40] and ElliPro [37] are shown in Figure 6.5. We note that ElliPro can produce a short list of candidate epitopes. Its performance reported here is summarized based on its best result among all of the predicted candidates for each antigen. In the case that BeTop identifies multiple epitopes for an antigen, its performance is reported in the same way as ElliPro for a fair comparison. The best performance instead of averaged performance or randomly chosen performance is used for comparison because of the possibility of existing multiple epitopes and the incomplete exploration of epitopes. Obviously, this is a limitation for current study on antibody-agnostic epitope prediction due to the natural shortage of the current data. To comprehensively

evaluate our model performance, we conducted multiple epitope prediction, see later section. From Figure 6.5, it can be seen that BeTop outperforms all existing models significantly. The f-score t -test p -values between BeTop and the other models are shown in Table 6.1 to illustrate the significance level that BeTop is better.

Table 6.1: F-score t -test p -values between BeTop, DiscoTope, SEPPA and ElliPro.

	DiscoTope (0.22±0.14)	SEPPA (0.25±0.16)	ElliPro (0.36±0.20)	BeTop (0.45±0.16)
DiscoTope		1.6e-1	7.9e-8	1.8e-17
SEPPA			2.3e-5	7.3e-15
ElliPro				2.0e-3

Since AUC is mostly recommended to quantify model performance [30], we also plot the distribution of AUC values for the four methods, as shown in Figure 6.6. Not surprisingly, BeTop remarkably outperforms other methods.

The averaged sensitivity, specificity, accuracy and AUC values for DiscoTope, SEPPA, ElliPro and BeTop are shown in Table 6.2. It is clear that BeTop is remarkably better than other models in terms of sensitivity, accuracy and AUC. The specificity of BeTop is slightly lower than that of ElliPro, but this value is much better than the other two models. More detailed results for each antigen in terms of sensitivity, specificity, f-score and accuracy can be found in Table 6.3.

Table 6.2: The averaged performances comparison between BeTop, DiscoTope, SEPPA and ElliPro on sensitivity, specificity, accuracy and AUC.

	DiscoTope	SEPPA	ElliPro	BeTop
sensitivity	0.377±0.278	0.526±0.345	0.501±0.290	0.665±0.239
specificity	0.686±0.168	0.665±0.255	0.849±0.137	0.809±0.162
accuracy	0.631±0.133	0.659±0.193	0.798±0.126	0.802±0.134
AUC	0.531±0.127	0.595±0.157	0.675±0.140	0.737±0.107

One of the novel ideas used in this study is reducing the weight of boundary edges to distinguish epitope from non-epitope. Thus, we further compare the performances of BeTop with suppressing weights of boundary edges and without suppressing weights of boundary edges. Experimental results show that the averaged f-scores are 0.45

Table 6.3: The detail performance of BeTop on 92 antibody-antigen PDB complexes.

PDB*	sen	spe	F1	acc	PDB	sen	spe	F1	acc	PDB	sen	spe	F1	acc
1AR1-CD-B	0.846	0.996	0.880	0.988	1TQB-BC-A	0.833	0.833	0.638	0.833	2XQY-GL-A	0.824	0.966	0.609	0.961
1BGX-HL-T	0.480	0.995	0.615	0.964	1V7M-HL-V	0.118	0.945	0.154	0.848	2XTJ-DB-C	1.000	0.642	0.370	0.676
1CL7-HL-I	0.407	0.818	0.458	0.683	1W72-HL-A	0.667	0.977	0.645	0.960	2XWT-AB-C	0.846	0.880	0.603	0.876
1CZ8-HL-W	1.000	0.850	0.700	0.872	1WEJ-HL-F	1.000	0.777	0.488	0.798	2YC1-AB-C	0.455	0.574	0.256	0.554
1DEE-FE-H	0.917	0.381	0.449	0.500	1YJD-HL-C	0.750	0.736	0.367	0.737	2ZCH-HL-P	0.833	0.973	0.714	0.966
1E6J-HL-P	0.500	0.879	0.286	0.857	1ZTX-HL-E	0.500	0.724	0.311	0.693	3B2U-HL-A	0.789	0.847	0.492	0.841
1EGJ-HL-A	0.727	0.822	0.457	0.812	2ADF-HL-A	0.333	0.864	0.200	0.831	3B9K-HL-B	0.706	0.940	0.686	0.906
1EO8-HL-A	0.231	0.997	0.353	0.966	2AEP-HL-A	0.895	0.959	0.667	0.956	3BN9-DC-B	0.700	0.977	0.718	0.954
1EZV-XY-E	1.000	0.788	0.455	0.805	2B2X-HL-A	0.714	0.862	0.417	0.851	3CVH-HL-A	0.647	0.918	0.449	0.901
1FNS-HL-A	0.700	0.742	0.215	0.740	2BDN-HL-A	0.643	0.722	0.474	0.706	3D85-BA-C	0.667	0.907	0.556	0.880
1FSK-IH-G	0.875	0.888	0.609	0.887	2CMR-HL-A	0.562	0.849	0.346	0.826	3DVG-BA-Y	0.857	0.548	0.444	0.605
1H0D-BA-C	0.529	0.971	0.621	0.910	2DD8-HL-S	0.667	0.874	0.462	0.854	3GI9-HL-C	0.600	0.976	0.529	0.963
1IQD-BA-C	0.733	0.738	0.349	0.737	2FD6-HL-U	0.417	0.911	0.263	0.888	3GRW-HL-A	0.966	0.808	0.615	0.830
1JPS-HL-T	0.500	0.928	0.465	0.885	2H9G-HL-S	0.545	0.792	0.429	0.750	3H42-HL-B	0.619	0.958	0.481	0.943
1JRH-HL-I	0.643	0.765	0.429	0.747	2J4W-HL-D	0.833	0.773	0.741	0.794	3HI6-HL-A	0.667	0.660	0.282	0.661
1LK3-HL-A	1.000	0.587	0.375	0.632	2J5L-CB-A	0.917	0.591	0.687	0.706	3KR3-HL-D	0.882	0.814	0.750	0.833
1MHH-DC-F	0.917	0.540	0.478	0.613	2J88-HL-A	1.000	0.819	0.243	0.824	3KS0-HL-B	0.182	0.975	0.267	0.880
1N8Z-BA-C	0.688	0.989	0.667	0.981	2JEL-HL-P	1.000	0.586	0.508	0.659	3L5X-HL-A	0.400	0.989	0.533	0.931
1NFD-HG-D	0.154	0.978	0.200	0.933	2NY1-DC-A	0.385	0.942	0.286	0.918	3L95-BA-X	0.833	0.689	0.303	0.700
1NL0-HL-G	1.000	0.321	0.345	0.424	2NYY-DC-A	0.526	0.986	0.435	0.979	3LDB-CB-A	0.385	0.931	0.250	0.910
1NMB-HL-N	0.500	0.986	0.563	0.964	2Q8B-HL-A	0.700	0.827	0.346	0.818	3LEV-HL-A	0.111	0.881	0.067	0.842
1NSN-HL-S	0.500	0.667	0.269	0.645	2QQK-HL-A	0.800	0.987	0.640	0.983	3LH2-HL-S	0.545	0.736	0.387	0.703
1OAZ-HL-A	0.357	0.960	0.435	0.887	2QQN-HL-A	0.917	0.803	0.431	0.812	3LHP-HL-S	1.000	0.598	0.406	0.647
1OB1-BA-C	0.833	0.881	0.625	0.875	2R0L-HL-A	0.211	0.995	0.333	0.933	3LIZ-HL-A	0.643	0.927	0.391	0.915
1ORS-BA-C	0.300	0.959	0.333	0.909	2R29-HL-A	0.667	0.988	0.769	0.938	3MJ9-HL-A	0.261	0.981	0.364	0.908
1OSP-HL-O	0.882	0.479	0.195	0.506	2R56-IM-B	0.353	0.739	0.200	0.698	3MXW-HL-A	0.444	0.844	0.340	0.797
1OTS-CD-A	0.556	0.986	0.500	0.977	2UZI-HL-R	0.941	0.651	0.376	0.681	3NGB-HL-G	0.840	0.783	0.362	0.787
1PKQ-BA-E	0.812	0.876	0.619	0.868	2VXQ-HL-A	0.692	0.443	0.273	0.478	3NH7-HL-A	0.850	0.508	0.493	0.588
1QKZ-HL-A	0.917	0.761	0.647	0.793	2VXS-IM-A	0.714	0.582	0.222	0.593	3PNW-HG-I	0.765	0.579	0.565	0.636
1R3J-BA-C	0.308	1.000	0.471	0.913	2VXT-HL-I	0.800	0.745	0.381	0.750	3QWO-HL-P	0.875	0.514	0.583	0.623
1RJL-BA-C	0.417	0.590	0.196	0.568	2XQB-HL-A	0.550	0.871	0.524	0.810					

* “PDB ID”-“antibody heavy chain and antibody light chain”-“antigen chain”

and 0.41 for the two situations, with increment of f-score by 8.9%. The t -test p -value of 0.11 between the two sets of f-scores also demonstrates the improvement of performance by decreasing weights of edges enriched in boundary class.

6.3.2 Locating epitopes with planar formations

Existing conformational epitope prediction methods, such as [35, 37, 185], heavily rely on the spatial structure information and non-planarity properties of antigens. They usually have a good performance on epitopes that have a protrusive surface, otherwise the performance becomes poor. To understand the effect of non-planarity of epitopes on epitope prediction, we divide all of the epitopes in our database into groups based on a non-planarity index. The non-planarity of a residue cluster is measured by the

root-mean-square deviation of all the surface atoms of this cluster of residues. In particular, it is calculated by $\sqrt{E(d(x, \mathcal{P})^2)}$, where $d(\cdot)$ is the Euclidean distance between each exposed epitope atom to the plane $\mathcal{P}$ and $\mathcal{P}$ is the least squares plane. It is expected that those structure-based prediction models favor epitopes with large non-planarity but not flat epitopes.

Our experimental result is shown in Figure 6.7. It is clear that BeTop works very well with an average f-score 0.432 on flat epitopes, namely on those epitopes having a non-planarity less than $2\text{\AA}$. However, DiscoTope, SEPPA and ElliPro all had difficulties to detect such epitopes with f-scores of only 0.214, 0.207, and 0.337 respectively.

Taking PDB entry 1AR1 as example again (Figure 6.1), its epitope consists of 19 residues, and the non-planarity of this epitope is as small as $1.08\text{\AA}$, indicating a very flat surface area. The f-score achieved by BeTop is 0.88 (with 16 true positives and 1 false positive). However, the f-score produced by ElliPro, SEPPA, DiscoTope are 0.273 (with 7 true positives and 22 false positives), 0.000, and 0.000, respectively. As another example, the prediction performance by BeTop, ElliPro, SEPPA and DiscoTope on the epitope residues of PDB entry 1N8Z are 0.667, 0.194, 0.198 and 0.07, respectively. This epitope also has a very planar surface with non-planarity of $1.88\text{\AA}$.

For epitopes having a large non-planarity bigger than or equal to $3\text{\AA}$, BeTop also performs better than the other models. The f-score is improved by 65.6%, 55.7% and 11.8% over DiscoTope, SEPPA and ElliPro, respectively. In particular, in comparison to ElliPro, which detects twisted epitopes based on residues' protrusion index, BeTop still achieved a better performance.

In general, the f-score of the 3 existing methods becomes poor when the non-planarity of epitopes becomes flat, which is mainly caused by the geometrical features that are being used to single out epitope residues. However, BeTop performs equally well under both protrusive and planar conditions, demonstrating that our proposed

BeTop graph model is invariant to the change of epitope non-planarity. The stable-ness, to some extent, ascribes to that that the structural information is transformed into a network and thus, it performs equally well on identifying both protrusive and planar epitopes.

The decrement of performance of existing models in predicting planar epitopes also results from the natural difference between protrusive epitopes and planar epitopes. Fernández finds that the reduction of epitope curvature leads to the decrease of water coordination [190], and Giovambattista reports that the water density and compressibility of flat interfaces is lower comparing with wild types [191].

Figure 6.7: Performances comparison at different levels of epitope non-planarity.

6.3.3 Identifying multiple epitopes from an antigen

Although BeTop is trained on single-epitope antigen-antibody complexes, the framework has no limitation on the number of predicted epitopes. To evaluate BeTop’s capability in identifying multiple epitopes in an antigen, we tested it on a data set of epitopes that are comprehensively explored in [109].

The multiple epitopes of these antigens are determined by the following steps: (i) determine epitope residues for each complex by using the 4Å Euclidian distance criteria between the antigen and antibody; (ii) calculate a pair-wise epitope similarity between two complexes X and Y of the same antigen by using $S_{XY} = |X \cap Y| / \min(|X|, |Y|)$;

(iii) cluster epitopes based on their similarities for each antigen; (iv) select representative epitopes for each cluster with the best resolution, and map all representative epitopes to one of them with the finest resolution. Finally 9 antigens with a total of 20 epitopes are obtained.

BeTop can identify 8 out of the 9 antigens with multiple epitopes; and for all of the 20 epitopes, BeTop can detect 19 of them. However, the conventional approaches would take the union of all the epitope residues on an antigen as a single epitope. The average performance of sensitivity, specificity, f-score and accuracy of applying BeTop to multiple epitope prediction are 0.393, 0.907, 0.321, and 0.858, respectively. As an example, BeTop achieves an averaged f-score as high as of 0.611 in identifying the two epitopes on the prion protein (Figure 6.8). Detailed performance is available at Table 6.4.

Table 6.4: BeTop performance on multiple epitopes prediction.

antigen	PDB ID	antigen chain	sensitivity	specificity	f-score	accuracy
Hen egg white lysozyme	1A2Y	C	0.267	0.939	0.308	0.860
	1JHL	A	0.636	0.915	0.500	0.891
	1P2C	C	0.250	0.982	0.364	0.891
	3D9A	C	0.444	0.802	0.333	0.752
Vascular endothelial growth factor	1CZ8	W	0.812	0.795	0.578	0.798
	2FJG	V	0.000	0.953	0.000	0.863
Ubiquitin	3DVG	Y	0.429	0.823	0.387	0.750
	3DVG	X	0.600	0.415	0.222	0.440
Influenza virus neuraminidase	1NCB	N	0.316	0.957	0.293	0.925
	1NMB	N	0.105	0.984	0.148	0.941
Von willebrand factor	1FE8	C	0.053	0.929	0.062	0.840
	2ADF	A	0.200	0.885	0.158	0.831
Outer surface protein A	1FJ1	F	0.471	0.756	0.195	0.737
	1OSP	O	0.400	0.948	0.400	0.904
Integrin alpha-L	3EOA	I	0.214	0.703	0.091	0.665
	3HI6	A	0.409	0.703	0.231	0.667
Prion protein	1TQB	A	0.824	0.859	0.651	0.853
	2W9E	A	0.400	1.000	0.571	0.909
Tissue factor	1JPS	T	0.316	0.890	0.267	0.835
	1UJ3	C	0.722	0.952	0.650	0.932

As expected, BeTop can identify as many epitopes as possible when they exist on an antigen. For instance, there are four epitopes on the antigen hen egg white lysozyme. BeTop can detect all of the four epitopes with an average f-score and accuracy of 0.376 and 0.849. These experimental results show that multiple epitopes

predicted by BeTop are not false positives, and it does not mix up multiple epitopes either.

Figure 6.8: Two epitopes with 3 overlapping residues on the prion protein. The 17 epitope residues within PDB entry 1TQB are colored in orange, while the 15 epitope residues in PDB entry 2W9E are rendered in blue. The residues with green color are the three overlapping epitope residues.

Figure 6.9: Features that favor the epitope class. Here a feature is represented by two nodes connecting with a solid line. Two nodes with patterns XY and YZ form a complete triangle XYZ, while patterns X and XY form only a contacting residue pair XY. The blue-filled nodes are single residues in favor of the epitopes while the gray-filled nodes are not.

6.3.4 Graphical triplet patterns for epitopes

We are interested in outstanding features that distinguish epitopes from non-epitopes. By transforming epitope and non-epitope subgraphs into feature vectors and selecting distinct features by LIBSVM, we obtained 144 from the 1770 features. See the full details in Table 6.6. Features that favor epitopes are shown in Figure 6.9. Interestingly, residue triangles of the pattern XXY (no order constraint), where X is a polar residue and Y is a hydrophobic or polar residue, are more likely to be epitope residues, but the pattern XX itself has no such differentiability. This type X of residues include Glutamine (Q), Aspartate (D), Tyrosine (Y) and Leucine (L). For example, residue pair Glutamine-Glutamine (QQ) interacting with residue Arginine (R), Tyrosine (Y), Asparagine (N), Lysine (K), Serine (S), Glycine (G), or Proline (P) are rich in epitopes. But Glutamine-Glutamine itself cannot be used to distinguish epitopes from non-epitopes. Furthermore, these meaningful features indicate some general patterns including polar and hydrophilic homogeneous residue pair surrounded by hydrophobic or polar residues as shown in Figure 6.10(a), polar and hydrophobic homogeneous residue pair encircled by polar residues as shown in Figure 6.10(b), and hydrophobic homogenous residue pairs accompanied by hydrophobic residues as shown in Figure 6.10(c). In contrast, such phenomena are not observed in the features enriched in the non-epitope clusters; see Figure 6.11. Surprisingly, hydrophobic residues seem having substantial role in antibody binding. This, to some extent, can be explained by the observation of water coordination reduction which allows other proteins to approach [190].

To test the statistical significance of these features, we calculated their G-test values [192]. The top ten features that are in favor of the epitopes and the top ten features that are enriched in the non-epitopes in terms of G-test are shown in Table 6.5. Intriguingly, the top ten features for the epitope class almost all have the form XXY (no order constraint); this observation consolidates the feature patterns we have identified. However, no similar patterns, such as XXY , can be found in non-epitope preferred features; see Table 6.5 and Table 6.6.

Figure 6.10: Examples of triplet feature patterns rich in the epitopes. The single letters in each colored rectangle is a residue and the double-letter circles stand for contacting residue pairs. Different colors represent distinctive properties of residues. The edges connecting a residue pair and a single residue form interacting residue triangles. For instance, (a) shows a set of residue triangles including QQR, QQN, QQS, QQY, QQQ, QQG and QQP that all favor the epitope class.

Figure 6.11: Negative features of non-epitope clusters that are distinct from epitope clusters.

Table 6.5: Top ten features that are in favor of the epitope class and also those that are enriched in the non-epitopes in terms of G-test.

epitope		non-epitope	
feature	G-test	feature	G-test
SSL	9.12	YF	5.34
GGI	4.92	SAA	4.69
DDW	4.85	KQ	4.19
NGG	4.53	AC	4.04
RRT	4.23	A	3.20
DDF	4.21	EVV	2.74
STT	3.65	EA	2.59
FLL	3.26	FFV	2.54
QQS	3.13	HC	2.35
HF	2.89	N	2.35

Table 6.6: 144 features selected to separate epitope clusters from non-epitope clusters.

	sgl	residue pair	residue triangle
positive	K	RI KI DF QL YS GC	RRQ RTT KQQ NYY DDW QQY QSS ECC YYT STT GGV FLL
	D	KM NN DI EY YT MM	RRH RLL KCC NTT DDS QQS QGG HLL YYI TAA GGI III
	Q	KC NW QY ET YI MF	RRT RII KFF NGG DDF QQG QFF PCC WTT TII GLL
	Y	KF NL QS HF TA CC	RQQ KKY NNQ NFF DPP QEE EEM YYY WLL GGG MFF
	S	KL DH QG YY TM II	RYY KKC NQQ DDP QQP QPP EEL YYS SSL GGF MLL
negative	N	RL DV HT YA SF VV	RRA KKP PPI TFF LVV
	A	KQ EP HC YF GM	RNN QVV YWW GAA
	L	KP EA PW WG GV	RDD EAA YAA AVV
	V	DQ EV PA WV AC	RHH EVV WWA MMI
	I	DM HW PM SM FI	RWW HHP SAA FFV

6.4 Summary

Epitope prediction is an important way to understand the immune basis of antibody-antigen interactions and is beneficial to prophylactic and therapeutic solutions. In this study, we proposed a novel graph-based model (“BeTop”) to predict B-cell epitope by incorporating statistical ideas, graph clustering algorithms and supervised learning approaches. Our experimental results conducted on two data sets of non-redundant antigen-antibody complexes show that BeTop makes great improvements for identifying those planar epitopes and for distinguishing multiple epitopes in an antigen. We have also presented interesting features and triplet feature patterns for the epitopes which will be useful for us to form new hypothesis in the future studies.

Chapter 7

Summary and Future Work

Epitope is a compound including, but not limited to, proteins, lipids, carbohydrates and nucleic acids which can be recognized by the immune system. In the case of proteins, epitope is a cluster of residues that can be linearly connected or are far from each other in the primary sequence but close to each other in 3D space. Upon recognition by the immune system, it could elicit immune responses which, in turn, will depress the infection and eliminate the antigens. However, when there is failure of recognition by the immune system, epitope will cause malfunction and disorder to organisms. Therefore, identification of epitopes is very important [193, 194, 184]. The following is a summary of epitope identification, including what has been done in this research and what will be studied in future.

7.1 Summary

Antibody-antigen interaction follows the context-awareness principle. This extrinsic feature of antigen highlights the existence of multiple epitopes of an antigen and the reconfiguration of epitope residues when the antigen interacts with another antibody. The binary classification of antigen residues as either epitopes or non-epitopes without specifying the corresponding antibody only distinguishes epitope residues from non-epitope residues, while not identifying the epitope itself. Existing discontinuous epitope prediction models, which overlook the context-awareness property, identify a cluster of antigenic residues, rather than epitopes as they claimed. Thus, we study

a new and challenging problem of epitope identification in line with the principle of context-awareness.

We explore this problem comprehensively in the following aspects: (i) systematically verify the context-awareness principle, i.e., we study whether multiple interfaces exist in the same domain, what are the relations between interfaces and whether they associate with different partners; (ii) build antibody-specific epitope prediction models—the two-dimensional association based model and the coupling graph based model—to identify epitopes that are consistent with the context dependence rule; (iii) construct an antibody-agnostic model that predicts one or multiple epitopes if they exist, which overcomes the problem of mixing all the antigenic residues as a single epitope arising from existing discontinuous epitope predictors.

Context-awareness is explored on multi-interface domains that are collectively retrieved from PDB. A multi-interface domain is a domain having more than one distinctive interface, where domain classification is determined based on the SCOP database. The distribution of multi-interface domains are then profiled on different SCOP classification levels, while relations of interfaces within multi-interface domains are analyzed by using graph theory. Results show that about 40% of proteins have multiple interfaces and the multi-interface domains only occupy 1.8% of all the domains. Based on the analyses of fingerprints of interfaces, we observe dominant unique fingerprints of interfaces, which associate with specific interaction partners. These results validate the existence of multiple interfaces and the reconfiguration of binding residues.

The two-dimensional association based model tries to capture the context-awareness that predicts antibody-specific epitopes. The two-dimensional association includes residue-residue pairing preference and dependence between sets of interacting residue pairs. The preference plays the role of bridging paratope and epitope residues, while the dependence is used to infer new interactions. The reciprocal roles of the associations well identify the seed epitope residues, and then they are input into the Hidden Markov Model to complete epitope identification. This model generates good performance on antibody-specific epitope prediction, and its simplified version yields

much better results on antigenic residue detection. Besides, this model is trained on antibody-antigen structures, but it can be applied to antibody-antigen sequence complexes to identify epitopes, which has broader applicability comparing with structure-based models.

The coupling graph based model completes the representation of context-awareness in paratope-epitope interacting complex. The coupling graph model captures the interactions between paratope and epitope, within paratope and within epitope, while only interactions between paratope and epitope can be represented by the two-dimensional association model. The coupling graph is suitable for representing context-awareness, but the frequent coupling graph mining is very challenging. To solve this problem, an innovative coupling graph mining algorithm is designed by using graph transformation. Graph transformation significantly reduces the mining time and memory consumption by avoiding a large number of irrelevant subgraph generation. After applying the coupling graph-based model to antibody-specific epitope prediction, it reveals better performance than the association-based model. In addition, the distinctive coupling graph patterns within paratope-epitope complexes, which are very helpful for understanding the recognition mechanism between antibodies and antigens, are also identified.

Antibody-specific epitopes cannot be directly identified when the antibody is missing; thus, an antibody-agnostic epitope prediction model is built. This model aims at predicting epitopes instead of antigenic residues. To this end, statistical ideas, graph clustering and graph classification approaches are utilized to build a graph-based model. In particular, statistical ideas are used to partially separate candidate epitope graphs from non-epitope graphs by setting different weights between them. Then the graph clustering algorithm is adopted to completely separate the weighted antigen graph into subgraphs. Finally, the classification algorithm is employed to determine the label of each subgraph. This method significantly improves the prediction performance with f-score increment of 24.7% compared to the best existing model. More importantly, this graph-based model detects multiple epitopes whenever

they exist, which overcomes the problem of identifying antigenic residues instead of epitopes caused by existing discontinuous epitope prediction models.

All in all, we have systematically explored the property of multi-interface, designed two-dimensional association-based and coupling graph-based antibody-specific epitope prediction models, and also developed an antibody-agnostic epitope identification model, which are consistent with the principle of context-awareness.

7.2 Future Work

Extensive works have been made on B-cell epitope prediction [34, 35, 37, 38, 39, 40, 161, 93, 94, 109, 195]. However, this problem is far from solved. Firstly, the performances of all the existing methods are still unsatisfactory. Secondly, attention has only been paid to epitopes, which is useful for vaccine design; however, paratopes, which have equal importance in drug development, have been overlooked. Therefore, we propose the following directions to improve epitope identification and paratope construction.

7.2.1 Epitope prediction with low- and high-level features

Current discontinuous epitope prediction models are mainly at the residue level, such as CEP [34], DiscoTope [35], ElliPro [37], PEPITO [38], Eptopia [39], SEPPA [40], EPSVR [93], PREL [94]. These methods can reflect the preference of residues or residue patterns, but they cannot reveal the preference of atoms or atom patterns or even the structural patterns. Thus, we propose two directions to improve epitope identification: drilling down and rolling up. “Drilling down” means exploring antibody-antigen interactions at the atom level. The low-level investigation unveils the binding details within atoms and structural patterns of atoms. As for the “rolling up”, we intend to study the interaction at the systematic level which can show special physicochemical patterns from the system perspective.

Intensive works focusing on protein interaction and assembly have shown that characteristics obtained at the atomic level provides more reliable and explicit bind-

ing information [196, 197, 198, 199]. However, no such work has been done in B-cell epitope prediction. Therefore, it is very promising to borrow the atomic information in protein-protein interaction for identifying antibody-specific B-cell epitopes. Atomic information is very specific and accurate, but it may introduce too much information to be handled. Hence, we also propose to use higher level information, such as the common structural patterns, to narrow down to the informative results. For instance, the hydrophilic island with a hydrophobic core of an epitope that complementarily interacts with a paratope having a hydrophilic core encircled by neutral residues, as shown in Figure 5.2, can be used to filter out irrelevant atomic contacts in antibody-antigen binding. The delicate combination of information at atomic level and systematic level shows encouraging performance in epitope prediction.

Besides taking it from systematic level, important and new features introduced by recent studies should be included as well. Existing epitope prediction models never consider the importance of water. However, the substantial role of water in protein binding has been extensively investigated. In one study, Fernández shows that the water coordination decreases notably relative to the bulk levels upon protein binding. Hence, this feature can be very useful in identifying epitope residues [190]. Gladly, Fernández also develops a method to measure the water coordination, which can be easily included into a model [200]. In another study, Rossky finds that the water surface density is considerably higher at the “dry-core-wet-rim” region of interface than the purely hydrophobic region [201]. This observation, surprisingly in consonance with our findings (see Figure 6.1 and Figure 6.9), indicates that water can be included in a model to accurately predict epitopes.

7.2.2 Paratope construction through context-awareness

Paratope construction, which has equal importance as epitope prediction, is essential for drug development. Figure 7.1 shows a flowchart of using existing epitope prediction model and the property of context-awareness to construct paratope. There are many discontinuous epitope prediction models which can be used to identify epitope

residues from an antigen structure, especially the one developed in this study which identifies one or multiple epitopes when they exist [195]. Moreover, the representation of context-awareness in antibody-antigen interaction has also been explored extensively [109]. The context-awareness plays a bridging role that links paratope and epitope together. By incorporating these two procedures, the paratope which interacts with an epitope could be built based on an antigen. It is noticeable that solving this problem can shed light on solving a lot of other problems, such as drug development and protein docking.

Figure 7.1: Flowchart of using epitope prediction model and context-awareness to construct paratope. (a) is an antigen, the purple residues in (b) is the epitope of (a), and (c) shows the paratope that interacting with the epitope shown in (b).

7.2.3 Vaccine design with the assistance of epitope prediction

Epitope prediction can identify the antigenic residues of an antigen which can be recognized by an antibody, hence the mimic of the antigenic residues of an antigen could be used to elicit the immune response. For example, the study in [202] has shown that 2D1 antibody cross neutralizes the epitopes of 1918 and 2009 influenza viruses, in which the similar critical binding residues are found. Abundant existing epitope prediction models increase the possibility of successfully identifying epitopes. Therefore, it is hopeful to utilize epitope prediction model and peptide engineering

Figure 7.2: An example of full and concise graph transformation. (a) is an coupling graph, (b) is the full transformation of (a), and (c) is the concise and precise transformation of (a).

approaches to design disease-associated vaccine, such as the design of influenza virus oriented vaccine .

7.2.4 Coupling graph mining with concise graph transformation

The context-awareness of antibody-antigen interaction has been completely represented by the coupling graph, and efficient frequent coupling graph mining algorithm has been designed. However, we note that the transformation still introduce duplicate edges which increase the mining complexity. Figure 7.2 shows an example of fully transformed graph of a coupling graph as well as its concise manner. It is clear that the concise transformation contains the complete information of the original coupling graph but only has very small number of edges. One of the future works includes design of more efficient algorithm to mining coupling graphs.

References

- [1] Parham P: *The Immune System*. Garland Science 2004.
- [2] Abbas AK, Lichtman AH, Pillai S: *Cellular and Molecular Immunology*. W.B. Saunders Company, 6th edition 2009.
- [3] YassineMrabet: **Structure générale d'un acide aminé**. <http://en.wikipedia.org/wiki/File:AminoAcidball.svg> 2007.
- [4] Creighton TE: *Proteins: Structures and Molecular Properties*. W. H. Freeman 1992.
- [5] Kyte J, Doolittle RF: **A simple method for displaying the hydropathic character of a protein**. *J Mol Biol* 1982, **157**:105–132.
- [6] **Proteinogenic amino acid** 2010, [http://en.wikipedia.org/wiki/Proteinogenic_amino_acid#Side_chain_properties].
- [7] Eisenberg D: **Three-Dimensional Structure of Membrane and Surface Proteins**. *Annu Rev Biochem* 1984, **53**:595–623.
- [8] **Amino Acid Properties**. <http://www.mcb.ucdavis.edu/courses/bis102/AAProp.html> 2010.
- [9] Hopp T: **Protein surface analysis. Methods for identifying antigenic determinants and other interaction sites**. *J Immunol Methods* 1986, **88**:1–18.
- [10] Parker J, Guo D, Hodges R: **New hydrophilicity scale derived from high-performance liquid chromatography peptide retention data: correlation of predicted surface residues with antigenicity and x-ray-derived accessible sites**. *Biochemistry* 1986, **25**(19):5425–5432.
- [11] Kolaskar A, Tongaonkar PC: **A semi-empirical method for prediction of antigenic determinants on protein antigens**. *FEBS Lett* 1990, **276**:172–174.
- [12] Larsen JEP, Lund O, Nielsen M: **Improved method for predicting linear B-cell epitopes**. *Immunome Res* 2006, **2**(2).

- [13] Söllner J, Mayer B: **Machine learning approaches for prediction of linear B-cell epitopes on proteins.** *J Mol Recognit* 2006, **19**:200–208.
- [14] Rubinstein ND, Mayrose I, Pupko T: **A machine-learning approach for predicting B-cell epitopes.** *Mol Immunol* 2008, **46**(5):840–847.
- [15] Jerne NK: **Immunological speculations.** *Annu Rev Microbiol* 1960, **14**:341–358.
- [16] Atassi M: **Antigenic structure of myoglobin: The complete immunochemical anatomy of a protein and conclusions relating to antigenic structures of proteins.** *Immunochemistry* 1975, **12**(5):423–438.
- [17] Benjamin DC, Berzofsky JA, East IJ, Gurd FRN, Hannum C, Leach SJ, Margoliash E, Michaels JG, Miller A, Prager EM, Reichlin M, Sercarz EE, Smith-Gill SJ, Todd PE, Wilson A: **The antigenic structure of proteins - a reappraisal.** *Annu Rev Immunol* 1984, **2**:67–101.
- [18] Pellequer JL, Westhof E, Van Regenmortel MHV: **Predicting location of continuous epitopes in proteins from their primary structures.** In *Molecular Design and Modeling: Concepts and Applications Part B: Antibodies and Antigens, Nucleic Acids, Polysaccharides, and Drugs, Volume 203 of Methods in Enzymology*. Edited by Langone JJ, Academic Press 1991:176–201.
- [19] Van Regenmortel MHV: **Mapping Epitope Structure and Activity: From One-Dimensional Prediction to Four-Dimensional Description of Antigenic Specificity.** *Methods* 1996, **9**(3):465–472.
- [20] Nicolle M: **Une conception générale des anticorps et de leurs effets.** *C R Soc Biol* 1907, **63**:77.
- [21] Köhler G: **Frequency of precursor cells against the enzyme β -galactosidase: an estimate of the BALB/c strain antibody repertoire.** *Eur J Immunol* 1976, **6**(5):340–346.
- [22] Kabat EA: **Origins of antibody complementarity and specificity—hypervariable regions and minigene hypothesis.** *J Immunol* 1980, **125**(3):961–969.
- [23] Chothia C, Lesk AM: **Canonical structures for the hypervariable regions of immunoglobulins.** *J Mol Biol* 1987, **196**(4):901–917.
- [24] Maccallum RM, Martin AC, Thornton JM: **Antibody-antigen interactions: contact analysis and binding site topography.** 1996.
- [25] Schiffer M, Wu TT, Kabat EA: **Subgroups of variable region genes of beta chains of T-cell receptors for antigen.** *Proc Natl Acad Sci USA* 1986, **83**(12):4461–4463.

- [26] Kim PM, Lu LJ, Xia Y, Gerstein MB: **Relating Three-Dimensional Structures to Protein Networks Provides Evolutionary Insights.** *Science* 2006, **314**(5807):1938–1941.
- [27] Tyagi M, Shoemaker BA, Bryant SH, Panchenko AR: **Exploring functional roles of multibinding protein interfaces.** *Protein Sci* 2009, **18**(8):1674–1683.
- [28] Dasgupta B, Nakamura H, Kinjo AR: **Distinct Roles of Overlapping and Non-overlapping Regions of Hub Protein Interfaces in Recognition of Multiple Partners.** *J Mol Biol* 2011, **411**(3):713–727.
- [29] Nobeli I, Favia AD, Thornton JM: **Protein promiscuity and its implications for biotechnology.** *Nat Biotechnol* 2009, **27**(2):157 – 67.
- [30] Greenbaum JA, Andersen PH, Blythe M, Bui HH, Cachau RE, Crowe J, Davies M, Kolaskar A, Lund O, Morrison S, *et al*: **Towards a consensus on datasets and evaluation metrics for developing B-cell epitope prediction tools.** *J Mol Recognit* 2007, **20**(2):75–82.
- [31] Van Regenmortel MHV: **Structural and functional approaches to the study of protein antigenicity.** *Immunol Today* 1989, **10**(8):266–272.
- [32] Rubinstein ND, Mayrose I, Halperin D, Yekutieli D, Gershoni JM, Pupko T: **Computational characterization of B-cell epitopes.** *Mol Immunol* 2008, **45**(12):3477–3489.
- [33] Sweredoski MJ, Baldi P: **COBEpro: a novel system for predicting continuous B-cell epitopes.** *Protein Eng Des Sel* 2009, **22**(3):113–120.
- [34] Kulkarni-Kale U, Bhosle S, Kolaskar AS: **CEP: a conformational epitope prediction server.** *Nucleic Acids Res* 2005, **33**:168–171.
- [35] Andersen PH, Nielsen M, Lund O: **Prediction of residues in discontinuous B-cell epitopes using protein 3D structures.** *Protein Sci* 2006, **15**(11):2558–2567.
- [36] Rapberger R, Lukas A, Mayer B: **Identification of discontinuous antigenic determinants on proteins based on shape complementarities.** *J Mol Recognit* 2007, **20**(2):113–121.
- [37] Ponomarenko J, Bui HHH, Li W, Fusseder N, Bourne PE, Sette A, Peters B: **ElliPro: a new structure-based tool for the prediction of antibody epitopes.** *BMC bioinformatics* 2008, **9**:514+.
- [38] Sweredoski MJ, Baldi P: **PEPITO: improved discontinuous B-cell epitope prediction using multiple distance thresholds and half sphere exposure.** *Bioinformatics* 2008, **24**(12):1459–1460.

- [39] Rubinstein N, Mayrose I, Martz E, Pupko T: **Epitopia: a web-server for predicting B-cell epitopes**. *BMC Bioinformatics* 2009, **10**:287.
- [40] Sun J, Wu D, Xu T, Wang X, Xu X, Tao L, Li YX, Cao ZW: **SEPPA: a computational server for spatial epitope prediction of protein antigens**. *Nucleic Acids Res* 2009, **37**(suppl 2):W612–W616.
- [41] Saha S, Raghava GPS: **Prediction of continuous B-cell epitopes in an antigen using recurrent neural network**. *Proteins: Struct, Funct, Bioinf* 2006, **65**:40–48.
- [42] El-Manzalawy Y, Dobbs D, Honavar V: **Predicting linear B-cell epitopes using string kernels**. *J Mol Recognit*. 2008, **21**(4):243–55.
- [43] Reimer U: **Prediction of linear B-cell epitopes**. *Methods Mol Biol* 2009, **524**:335–44.
- [44] Schellekens GA, Visser H, De Jong BAW, Van Den Hoogen FHJ, Hazes JMW, Breedveld FC, Van Venrooij WJ: **The diagnostic properties of rheumatoid arthritis antibodies recognizing a cyclic citrullinated peptide**. *Arthritis Rheum* 2000, **43**:155–163.
- [45] Tam JP, Lu YA: **Vaccine engineering: enhancement of immunogenicity of synthetic peptide vaccines related to hepatitis in chemically defined models consisting of T- and B-cell epitopes**. *Proc Natl Acad Sci USA* 1989, **86**(23):9084–9088.
- [46] Hughes EE, Gilleland HE: **Ability of synthetic peptides representing epitopes of outer membrane protein F of Pseudomonas aeruginosa to afford protection against P. aeruginosa infection in a murine acute pneumonia model**. *Vaccine* 1995, **13**(18):1750–1753.
- [47] Chirino AJ, Ary ML, Marshall SA: **Minimizing the immunogenicity of protein therapeutics**. *Drug Discov Today* 2004, **9**(2):82–90.
- [48] Chen H, Zhou HX: **Prediction of interface residues in protein-protein complexes by a consensus neural network method: Test against NMR data**. *Proteins: Struct, Funct, Bioinf* 2005, **61**:21–35.
- [49] Schlessinger A, Ofra Y, Yachdav G, Rost B: **Epitome: database of structure-inferred antigenic epitopes**. *Nucleic Acids Res* 2006, **34**:777–780.
- [50] Beaver JE, Bourne PE, Ponomarenko JV: **EpitopeViewer: a Java application for the visualization and analysis of immune epitopes in the Immune Epitope Database and Analysis Resource (IEDB)**. *Immunome Res* 2007, **3**:3.

- [51] Lee B, Richards F: **The interpretation of protein structures: Estimation of static accessibility.** *J Mol Biol* 1971, **55**(3):379–400.
- [52] Shrake A, Rupley JA: **Environment and exposure to solvent of protein atoms. Lysozyme and insulin.** *J Mol Biol* 1973, **79**(2):351–371.
- [53] Ponomarenko JV, Bourne PE: **Antibody-protein interactions: benchmark datasets and prediction tools evaluation.** *BMC Struct Biol* 2007, **7**:64.
- [54] Pellequer JL, Westhof E, Van Regenmortel MH: **Correlation between the location of antigenic sites and the prediction of turns in proteins.** *Immunol Lett* 1993, **36**:83–99.
- [55] Alix AJP: **Predictive estimation of protein linear epitopes by using the program PEOPLE.** *Vaccine* 1999, **18**(3-4):311–314.
- [56] Toseland C, Clayton D, McSparron H, Hemsley S, Blythe M, Paine K, Doytchinova I, Guan P, Hattotuwigama C, Flower D: **AntiJen: a quantitative immunology database integrating functional, thermodynamic, kinetic, biophysical, and cellular data.** *Immunome Res* 2005, **1**:4.
- [57] Saha S, Bhasin M, Raghava G: **Bcipep: A database of B-cell epitopes.** *BMC Genomics* 2005, **6**:79.
- [58] Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE: **The Protein Data Bank.** *Nucleic Acids Res* 2000, **28**:235–242.
- [59] Murzin AG, Brenner SE, Hubbard T, Chothia C: **SCOP: a structural classification of proteins database for the investigation of sequences and structures.** *J Mol Biol* 1995, **247**(4):536–540.
- [60] Peters B, Sidney J, Bourne P, Bui HH, Buus S, Doh G, Fleri W, Kronenberg M, Kubo R, Lund O, Nemazee D, Ponomarenko JV, Sathiamurthy M, Schoenberger S, Stewart S, Surko P, Way S, Wilson S, Sette A: **The Immune Epitope Database and Analysis Resource: From Vision to Blueprint.** *PLoS Biol* 2005, **3**(3):e91.
- [61] Ansari HR, Flower DR, Raghava GPS: **AntigenDB: an immunoinformatics database of pathogen antigens.** *Nucleic Acids Res* 2010, **38**(suppl 1):D847–D853.
- [62] Van Regenmortel MHV: **Immunoinformatics may lead to a reappraisal of the nature of B cell epitopes and of the feasibility of synthetic peptide vaccines.** *J Mol Recognit.* 2006, **19**(3):183–187.

- [63] Hopp TP, Woods KR: **Prediction of protein antigenic determinants from amino acid sequences.** *Proc Natl Acad Sci USA* 1981, **78**(6):3824–3828.
- [64] Jameson B, Wolf H: **The antigenic index: a novel algorithm for predicting antigenic determinants.** *Comput Appl Biosci* 1988, **4**:181–186.
- [65] Emini EA, Hughes JV, Perlow DS, Boger J: **Induction of hepatitis A virus-neutralizing antibody by a virus-specific synthetic peptide.** *J Virol* 1985, **55**(3):836–839.
- [66] Karplus PA, Schulz GE: **Prediction of chain flexibility in proteins: a tool for the selection of peptide antigen.** *Naturwissenschaften* 1985, **72**(4):212–213.
- [67] Vihinen M, Torkkila E, Riikonen P: **Accuracy of protein flexibility predictions.** *Proteins: Struct, Funct, Genetics* 1994, **19**(2):141–149.
- [68] Garnier J, JF G, B R: **GOR method for predicting protein secondary structure from amino acid sequence.** *Methods Enzymol* 1996, **266**:540–53.
- [69] Efremov RG, Alix AJ: **Environmental characteristics of residues in proteins: three-dimensional molecular hydrophobicity potential approach.** *J Biomol Struct Dyn* 1993, **11**(3):483–507.
- [70] Alix AJP: **Molecular modeling of globular proteins: strategy 1D to 3D: secondary structures and epitopes.** *Biomolecular structures and dynamics: experimental and theoretical advances* 1997, :121–150.
- [71] Huang J, Honda W, Kanehisa M: **Predicting B cell epitope residues with network topology based amino acid indices.** *Genome Inform* 2007, **19**:40–49.
- [72] Huang J, Kawashima S, Kanehisa M: **New amino acid indices based on residue network topology.** *Genome Inform* 2007, **18**:152–61.
- [73] Blythe MJ, Flower DR: **Benchmarking B cell epitope prediction: under-performance of existing methods.** *Protein Sci* 2005, **14**:246–8.
- [74] Korber BTM, Brander C, Haynes BF, Koup R, Moore JP, Walker BD, Watkins DI: **HIV Immunology and HIV/SIV Vaccine Databases 2003.** Los Alamos National Laboratory, Theoretical Biology and Biophysics, Los Alamos, New Mexico 2003.
- [75] Bairoch A, Apweiler R: **The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000.** *Nucleic Acids Res* 2000, **28**:45–48.
- [76] El-Manzalawy Y, Dobbs D, Honavar V: **Predicting flexible length linear B-cell epitopes.** *Comput Syst Bioinformatics Conf* 2008, **7**:121–32.

- [77] Söllner J, Grohmann R, Rapberger R, Perco P, Lukas A, Mayer B: **Analysis and prediction of protective continuous B-cell epitopes on pathogen proteins.** *Immunome Res* 2008, **4**.
- [78] Forrest LR, Tang CL, Honig B: **On the accuracy of homology modeling and sequence alignment methods applied to membrane proteins.** *Biophys J* 2006, **91**(2):508–17.
- [79] Wee LJ, Simarmata D, Kam YW, Ng LF, Tong JC: **SVM-based prediction of linear B-cell epitopes using Bayes Feature Extraction.** *BMC Genomics* 2010, **11**(Suppl 4):S21.
- [80] Su CH, Pal NR, Lin KL, Chung IF: **Identification of Amino Acid Propensities That Are Strong Determinants of Linear B-cell Epitope Using Neural Networks.** *PLoS ONE* 2012, **7**(2):e30617.
- [81] Janin J, Wodak S: **Conformation of amino acid side-chains in proteins.** *J Mol Biol* 1978, **125**(3):357–86.
- [82] Ponnuswamy PK, Prabhakaran M, Manavalan P: **Hydrophobic packing and spatial arrangement of amino acid residues in globular proteins.** *Biochim Biophys Acta* 1980, **623**(2):301–16.
- [83] Chou PY, Fasman GD: **Prediction of the secondary structure of proteins from their amino acid sequence.** *Adv Enzymol Relat Areas Mol Biol* 1978, **47**:145–148.
- [84] Gao J, Faraggi E, Zhou Y, Ruan J, Kurgan L: **BEST: Improved Prediction of B-Cell Epitopes from Antigen Sequences.** *PLoS ONE* 2012, **7**(6):e40104.
- [85] Wang L, Liu J, Zhu S, Gao Y: **Prediction of Linear B-Cell Epitopes Using AAT Scale.** In *3rd International Conference on Bioinformatics and Biomedical Engineering* 2009:1–4.
- [86] Wang HW, Lin YC, Pai TW, Chang HT: **Prediction of B-cell Linear Epitopes with a Combination of Support Vector Machine Classification and Amino Acid Propensity Identification.** *J Biomed Biotechnol* 2011, **2011**.
- [87] Chang HT, Liu CH, Pai TW: **Estimation and extraction of B-cell linear epitopes predicted by mathematical morphology approaches.** *J Mol Recognit* 2008, **21**(6):431–441.
- [88] Eswar N, Webb B, Marti-Renom MA, Madhusudhan MS, Eramian D, Shen MY, Pieper U, Sali A: **Comparative protein structure modeling using Modeller.** *Curr Protoc Bioinformatics* 2006, **Chapter 5**.

- [89] Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ: **Basic local alignment search tool.** *J Mol Biol* 1990, **215**(3):403–410.
- [90] Taylor WR, Thornton JM, Turnell WG: **An ellipsoidal approximation of protein shape.** *J Mol Graph* 1983, **1**(2):30–38.
- [91] Thornton JM, Edwards MS, R TW, J BD: **Location of 'continuous' antigenic determinants in the protruding regions of proteins.** *EMBO J* 1986, **5**(2):409–13.
- [92] Liang S, Zheng D, Zhang C, Zacharias M: **Prediction of antigenic epitopes on protein surfaces by consensus scoring.** *BMC Bioinformatics* 2009, **10**:302+.
- [93] Liang S, Zheng D, Standley DM, Yao B, Zacharias M, Zhang C: **EPSVR and EPMeta: prediction of antigenic epitopes using support vector regression and multiple server results.** *BMC Bioinformatics* 2010, **11**:381.
- [94] Ambroise J, Giard J, Gala JL, Macq B: **Identification of Relevant Properties for Epitopes Detection Using a Regression Model.** *IEEE/ACM Trans Comput Biol Bioinformatics* 2011, **8**(6):1700–1707.
- [95] Ansari HR, Raghava GP: **Identification of conformational B-cell Epitopes in an antigen from its primary sequence.** *Immunome Res* 2010, **6**:6.
- [96] Liu R, Hu J: **Prediction of Discontinuous B-Cell Epitopes Using Logistic Regression and Structural Information.** *J Proteomics Bioinform* 2011, **4**:010–015.
- [97] Mitchell JC, Kerr R, Ten Eyck LF: **Rapid atomic density methods for molecular shape characterization.** *J Mol Graph Model* 2001, **19**(3-4):325–330.
- [98] Camacho CJ, Zhang C: **FastContact: rapid estimate of contact and binding free energies.** *Bioinformatics* 2005, **21**(10):2534–2536.
- [99] Lundegaard C, Lund O, Kesmir C, Brunak S, Nielsen M: **Modeling the adaptive immune system: predictions and simulations.** *Bioinformatics* 2007, **23**(24):3265–3275.
- [100] Swets JA: **Measuring the accuracy of diagnostic systems.** *Science* 1988, **240**(4857):1285–1293.
- [101] The Gene Ontology Consortium: **Gene ontology: tool for the unification of biology.** *Nat Genet* 2000, **25**:25–29.

- [102] Narayan V, Halada P, Hernychova L, Chong YP, Zakova J, Hupp TR, Vojtesek B, Ball KL: **A multi-protein binding interface in an intrinsically disordered region of the tumour suppressor protein interferon regulatory factor-1.** *J Biol Chem* 2011.
- [103] Keskin O, Gursoy A, Ma B, Nussinov R: **Principles of Protein-protein Interactions: What are the Preferred Ways For Proteins To Interact?** *Chem Rev* 2008, **108**(4):1225–1244.
- [104] Qi Y, Balem F, Faloutsos C, Klein-Seetharaman J, Bar-Joseph Z: **Protein complex identification by supervised graph local clustering.** *Bioinformatics* 2008, **24**(13):i250–i268.
- [105] Björkholm P, Sonnhammer ELL: **Comparative analysis and unification of domain-domain interaction networks.** *Bioinformatics* 2009, **25**(22):3020–3025.
- [106] Itzhaki Z, Akiva E, Margalit H: **Preferential use of protein domain pairs as interaction mediators: order and transitivity.** *Bioinformatics* 2010, **26**(20):2564–2570.
- [107] Zhao L, Li J: **Sequence-based B-cell epitope prediction by using associations in antibody-antigen structural complexes.** In *IEEE International Conference on Bioinformatics and Biomedicine Workshop, 2009.* 2009:165–172.
- [108] Kortemme T, Baker D: **A simple physical model for binding energy hot spots in protein-protein complexes.** *Proc Natl Acad Sci USA* 2002, **99**(22):14116–14121.
- [109] Zhao L, Wong L, Li J: **Antibody-Specified B-Cell Epitope Prediction in Line with the Principle of Context-Awareness.** *IEEE/ACM Trans Comput Biol Bioinf* 2011, **8**(6):1483–1494.
- [110] Shulman-Peleg A, Shatsky M, Nussinov R, Wolfson HJ: **MultiBind and MAPPIS: webserver for multiple alignment of protein 3D-binding sites and their interactions.** *Nucleic Acids Res* 2008, **36**(suppl 2):W260–W264.
- [111] Jambon M, Imberty A, Deléage G, Geourjon C: **A new bioinformatic approach to detect common 3D sites in protein structures.** *Proteins: Structure, Function, and Genetics* 2003, **52**(2):137–145.
- [112] Wangikar PP, Tendulkar AV, Ramya S, Mali DN, Sarawagi S: **Functional sites in protein families uncovered via an objective and automated graph theoretic approach.** *J Mol Bio* 2003, **326**(3):955–978.

- [113] Kinjo AR, Nakamura H: **Geometric similarities of protein-protein interfaces at atomic resolution are only observed within homologous families: an exhaustive structural classification study.** *J Mol Bio* 2010, **399**(3):526–540.
- [114] Kinjo AR, Nakamura H: **Composite structural motifs of binding sites for delineating biological functions of proteins.** *PloS ONE* 2012, **7**(2).
- [115] Cho Ki, Lee K, Lee KH, Kim D, Lee D: **Specificity of molecular interactions in transient protein-protein interaction interfaces.** *Proteins: Struct, Funct, Bioinf* 2006, **65**(3):593–606.
- [116] Yan C, Wu F, Jernigan RL, Dobbs D, Honavar V: **Characterization of protein-protein interfaces.** *Protein J* 2008, **27**:59–70.
- [117] Hu J, Yan C: **A Comparative Analysis of Protein Interfaces.** *Protein Pept Lett* 2010, **17**(11):1450–1458.
- [118] Zhou HX, Qin S: **Interaction-site prediction for protein complexes: a critical assessment.** *Bioinformatics* 2007, **23**(17):2203–2209.
- [119] Huang J, Koide A, Makabe K, Koide S: **Design of protein function leaps by directed domain interface evolution.** *Proc Natl Acad Sci USA* 2008, **105**(18):6578–6583.
- [120] Crowley PB, Matias PM, Mi H, Firbank SJ, Banfield MJ, Dennison C: **Regulation of Protein Function: Crystal Packing Interfaces and Conformational Dimerization.** *Biochemistry* 2008, **47**(25):6583–6589.
- [121] Lee J, Natarajan M, Nashine VC, Socolich M, Vo T, Russ WP, Benkovic SJ, Ranganathan R: **Surface Sites for Engineering Allosteric Control in Proteins.** *Science* 2008, **322**(5900):438–442.
- [122] Guntas G, Purbeck C, Kuhlman B: **Engineering a protein-rotein interface using a computationally designed library.** *Proc Natl Acad Sci USA* 2010.
- [123] Schneidman-Duhovny D, Dror O, Inbar Y, Nussinov R, Wolfson HJ: **PharmaGist: a webserver for ligand-based pharmacophore detection.** *Nucleic Acids Res* 2008, **36**(suppl 2):W223–W228.
- [124] Meireles LMC, Döling AS, Camacho CJ: **ANCHOR: a web server and database for analysis of protein-rotein interaction binding pockets for drug discovery.** *Nucleic Acids Res* 2010, **38**(suppl 2):W407–W411.
- [125] Johnson M, Zaretskaya I, Raytselis Y, Merezhuk Y, McGinnis S, Madden TL: **NCBI BLAST: a better web interface.** *Nucleic Acids Res* 2008, **36**(suppl 2):W5–W9.

- [126] Larkin MA, Blackshields G, Brown NP, Chenna R, McGettigan PA, McWilliam H, Valentin F, Wallace IM, Wilm A, Lopez R, Thompson JD, Gibson TJ, Higgins DG: **Clustal W and Clustal X version 2.0**. *Bioinformatics* 2007, **23**(21):2947–2948.
- [127] Krissinel E, Henrick K: **Secondary-structure matching (PDBeFold), a new tool for fast protein structure alignment in three dimensions**. *Acta Cryst, Sect D: Biol Crystallogr* 2004, **D60**(12 Part 1):2256–2268.
- [128] Huan J, Wang W, Bandyopadhyay D, Snoeyink J, Prins J, Tropsha A: **Mining Protein Family Specific Residue Packing Patterns from Protein Structure**. In *Eighth Annual International Conference on Research in Computational Molecular Biology (RECOMB)* 2004:308–315.
- [129] Tropsha A, Carter CW, Cammer S, Vaisman II: **Simplicial neighborhood analysis of protein packing (SNAPP): a computational geometry approach to studying proteins**. *Methods Enzymol* 2003, **374**(1999):509–44.
- [130] Barber CB, Dobkin DP, Huhdanpaa H: **The Quickhull algorithm for convex hulls**. *ACM T Math Software* 1996, **22**(4):469–483.
- [131] Shindyalov IN, Bourne PE: **Protein structure alignment by incremental combinatorial extension (CE) of the optimal path**. *Protein Eng* 1998, **11**(9):739–747.
- [132] Meinel T, Wölein M, Urzova O, Fischer I, Philippsen M: **The ParMol Package for Frequent Subgraph Mining**. *ECEASST* 2006, **1**.
- [133] Huan J, Wang W, Prins J: **Efficient Mining of Frequent Subgraphs in the Presence of Isomorphism**. In *Proceedings of the Third IEEE International Conference on Data Mining* 2003:549–552.
- [134] Uno T, Kiyomi M, Arimura H: **LCM ver.3: collaboration of array, bitmap and prefix tree for frequent itemset mining**. In *OSDM '05: Proceedings of the 1st international workshop on open source data mining* 2005:77–86.
- [135] Keskin O, Nussinov R: **Similar Binding Sites and Different Partners: Implications to Shared Proteins in Cellular Pathways**. *Structure* 2007, **15**(3):341–354.
- [136] Kar G, Gursoy A, Keskin O: **Human Cancer Protein-Protein Interaction Network: A Structural Perspective**. *PLoS Comput Biol* 2009, **5**(12):e1000601.
- [137] Felsenstein J: **PHYLIP (Phylogeny Inference Package) version 3.6**. *Distributed by the author. Department of Genome Sciences, University of Washington, Seattle*. 2005.

- [138] Hahn MW, Kern AD: **Comparative Genomics of Centrality and Essentiality in Three Eukaryotic Protein-Interaction Networks.** *Mol Biol Evol* 2005, **22**(4):803–806.
- [139] Nooren IMA, Thornton JM: **Diversity of protein-protein interactions.** *EMBO J* 2003, **22**(14):3486–3492.
- [140] Ofra Y, Rost B: **Analysing six types of protein-protein interfaces.** *J Mol Bio* 2003, **325**(2):377–387.
- [141] Vishveshwara S, Brinda KV, Kannan N: **Protein structure: insights from graph theory.** *J Theor Comput Chem* 2002, **1**:187–212.
- [142] Li Y, Huang Y, Swaminathan CP, Smith-Gill SJ, Mariuzza RA: **Magnitude of the Hydrophobic Effect at Central versus Peripheral Sites in Protein-Protein Interfaces.** *Structure* 2005, **13**(2):297–307.
- [143] de Araújo AFP: **Protein Folding Cooperativity: Basic Insights from Minimalist Models.** *Protein Pept Lett* 2005, **12**(3):223–228.
- [144] Ferrante A, Gorski J: **Cooperativity of Hydrophobic Anchor Interactions: Evidence for Epitope Selection by MHC Class II as a Folding Process.** *J Immunol* 2007, **178**(11):7181–7189.
- [145] Aric AH, Daniel AS, Pieter JS: **Exploring Network Structure, Dynamics, and Function using NetworkX.** In *Proceedings of the 7th Python in Science Conference*, Pasadena, CA USA 2008:11–15.
- [146] Pietrosevoli N, Crespo A, Fernández A: **Dehydration Propensity of OrderDisorder Intermediate Regions in Soluble Proteins.** *J Proteome Res* 2007, **6**(9):3519–3526.
- [147] Crespo A, Fernández A: **Induced Disorder in ProteinLigand Complexes as a Drug-Design Strategy.** *Mol Pharm* 2008, **5**(3):430–437.
- [148] Zhang X, Crespo A, Fernández A: **Turning promiscuous kinase inhibitors into safer drugs.** *Trends Biotechnol* 2008, **26**(6):295 – 301.
- [149] Fernández A, Crespo A, Tiwari A: **Is there a case for selectively promiscuous anticancer drugs?** *Drug Discov Today* 2009, **14**(1-2):1 – 5.
- [150] Irving MB, Pan O, Scott JK: **Random-peptide libraries and antigen-fragment libraries for epitope mapping and the development of vaccines and diagnostics.** *Curr Opin Chem Biol* 2001, **5**(3):314–324.
- [151] Wu TT, Kabat EA: **An analysis of the sequences of the variable regions of bence jones proteins and myeloma light chains and their implications for antibody complementarity.** *J Exp Med* 1970, **132**:211–250.

- [152] Bublil EM, Freund NT, Mayrose I, Penn O, Roitburd-Berman A, Rubinstein ND, Pupko T, Gershoni JM: **Stepwise prediction of conformational discontinuous B-cell epitopes using the Mapitope algorithm.** *Proteins: Struct, Funct, Bioinf* 2007, **68**:294–304.
- [153] Singh SP, Tyagi S, Feroz K, BN M: **Benchmarking the Propensity Scales for the Prediction of Linear B-Cell Epitopes.** *J Comput Intell Bioinf* 2008, **1**:45–53.
- [154] Li J, Liu Q: **'Double water exclusion': a hypothesis refining the O-ring theory for the hot spots at protein interfaces.** *Bioinformatics* 2009, **25**(6):743–750.
- [155] Li J, Liu G, Li H, Wong L: **Maximal Biclique Subgraphs and Closed Pattern Pairs of the Adjacency Matrix: A One-to-one Correspondence and Mining Algorithms.** *IEEE Trans Knowl Data Eng* 2007, **19**(12):1625–1637.
- [156] Coenen F, Goulbourne G, Leng P: **Tree Structures for Mining Association Rules.** *Data Min Knowl Discov* 2004, **8**:25–51.
- [157] Abhinandan K, Martin AC: **Analysis and improvements to Kabat and structurally correct numbering of antibody variable domains.** *Mol Immunol* 2008, **45**(14):3832–3839.
- [158] Forney GD: **The viterbi algorithm.** *Proceedings of The IEEE* 1973, **61**(3):268–278.
- [159] Almagro JC: **Identification of differences in the specificity-determining residues of antibodies that recognize antigens of different size: implications for the rational design of antibody repertoires.** *J Mol Recognit* 2004, **17**(2):132–143.
- [160] Delano WL: **The PyMOL Molecular Graphics System.** DeLano Scientific, Palo Alto, CA, USA 2002, [<http://www.pymol.org>].
- [161] Zhao L, Li J: **Mining for the antibody-antigen interacting associations that predict the B cell epitopes.** *BMC Struct Biol* 2010, **10**(Suppl 1):S6.
- [162] Pellegrini M, Haynor D, Johnson JM: **Protein Interaction Networks.** *Expert Rev Proteomics* 2004, **1**(2):239–249.
- [163] Davidson E, Levin M: **Gene regulatory networks.** *Proc Natl Acad Sci USA* 2005, **102**(14):4935.
- [164] Korbel JO, Doerks T, Jensen LJ, Perez-Iratxeta C, Kaczanowski S, Hooper SD, Andrade MA, Bork P: **Systematic Association of Genes to Phenotypes by Genome and Literature Mining.** *PLoS Biol* 2005, **3**(5):e134.

- [165] Gennarino VA, D'Angelo G, Dharmalingam G, Fernandez S, Russolillo G, Sanges R, Mutarelli M, Belcastro V, Ballabio A, Verde P, Sardiello M, Banfi S: **Identification of microRNA-regulated gene networks by expression analysis of target genes.** *Genome Research* 2012, **22**(6):1163–1172.
- [166] Davies DR, Padlan EA: **Antibody-Antigen Complexes.** *Annu Rev Biochem* 1990, **59**:439–473.
- [167] Jackson RJ, Standart N: **How Do MicroRNAs Regulate Gene Expression?** *Sci STKE* 2007, **2007**(367):re1.
- [168] He L, Vandin F, Pandurangan G, Bailey-Kellogg C: **Ballast: A Ball-Based Algorithm for Structural Motifs.** In *RECOMB* 2012:79–93.
- [169] Inokuchi A, Washio T, Motoda H: **An Apriori-Based Algorithm for Mining Frequent Substructures from Graph Data.** In *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery, PKDD '00*, London, UK, UK: Springer-Verlag 2000:13–23.
- [170] Kuramochi M, Karypis G: **Frequent Subgraph Discovery.** In *Proceedings of the 2001 IEEE International Conference on Data Mining, ICDM '01*, Washington, DC, USA: IEEE Computer Society 2001:313–320.
- [171] Borgelt C, Berthold MR: **Mining Molecular Fragments: Finding Relevant Substructures of Molecules.** In *ICDM '02: Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02)*, IEEE Computer Society 2002.
- [172] Yan X, Han J: **gSpan: Graph-Based Substructure Pattern Mining.** In *Proceedings of the 2002 IEEE International Conference on Data Mining, ICDM '02*, IEEE Computer Society 2002.
- [173] Nijssen S, Kok JN: **Frequent graph mining and its application to molecular databases.** In *Proceedings of the IEEE International Conference on Systems Man and Cybernetics, Volume 5*, IEEE 2004:4571–4577.
- [174] Kuramochi M, Karypis G: **Discovering Frequent Geometric Subgraphs.** In *Proceedings of the 2002 IEEE International Conference on Data Mining, ICDM '02*, Washington, DC, USA: IEEE Computer Society 2002:258–.
- [175] Vanetik N, Gudes E, Shimony SE: **Computing Frequent Graph Patterns from Semistructured Data.** In *Proceedings of the 2002 IEEE International Conference on Data Mining, ICDM '02*, Washington, DC, USA: IEEE Computer Society 2002:458–.
- [176] Ke Y, Cheng J, Ng W: **Correlation search in graph databases.** In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '07*, New York, NY, USA: ACM 2007:390–399.

- [177] Ke Y, Cheng J, Yu JX: **Efficient Discovery of Frequent Correlated Subgraph Pairs**. In *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*, ICDM '09, IEEE Computer Society 2009:239–248.
- [178] Ozaki T, Ohkawa T: **Mining correlated subgraphs in graph databases**. In *Proceedings of the 12th Pacific-Asia conference on Advances in knowledge discovery and data mining*, PAKDD'08, Springer-Verlag 2008:272–283.
- [179] Xiong H, Tan PN, Kumar V: **Mining Strong Affinity Association Patterns in Data Sets with Skewed Support Distribution**. In *Proceedings of the Third IEEE International Conference on Data Mining*, ICDM '03, Washington, DC, USA: IEEE Computer Society 2003:387–.
- [180] Takigawa I, Tsuda K, Mamitsuka H: **Mining Significant Substructure Pairs for Interpreting Polypharmacology in Drug-Target Network**. *PLoS ONE* 2011, **6**(2):e16999.
- [181] Cook SA: **The complexity of theorem-proving procedures**. In *Proceedings of the third annual ACM symposium on Theory of computing*, STOC '71, New York, NY, USA: ACM 1971:151–158.
- [182] Li Z, He Y, Wong L, Li J: **Progressive dry-core-wet-rim hydration trend in a nested-ring topology of protein binding interfaces**. *BMC Bioinformatics* 2012, **13**:51.
- [183] Kringelum JV, Nielsen M, Padkjær SB, Lund O: **Structural analysis of B-cell epitopes in antibody:protein complexes**. *Mol Immunol* 2012, **53**(1-2):24–34.
- [184] Caoili SEC: **B-cell epitope prediction for peptide-based vaccine design: towards a paradigm of biological outcomes for global health**. *Immunome Res* 2011, **7**(2):2.
- [185] Rubinstein N, Mayrose I, Martz E, Pupko T: **Epitopia: a web-server for predicting B-cell epitopes**. *BMC Bioinformatics* 2009, **10**:287+.
- [186] Pettersen EF, Goddard TD, Huang CC, Couch GS, Greenblatt DM, Meng EC, Ferrin TE: **UCSF Chimera—a visualization system for exploratory research and analysis**. *J Comput Chem* 2004, **25**(13):1605–12.
- [187] van Dongen S: **Graph Clustering by Flow Simulation**. *PhD thesis*, University of Utrecht 2000.
- [188] Hubbard SJ, Thornton JM: **Naccess V2.1.1 - Solvent accessible area calculations**. <http://www.bioinf.manchester.ac.uk/naccess/> 1992.

- [189] Chang CC, Lin CJ: **LIBSVM: A library for support vector machines.** *ACM Transactions on Intelligent Systems and Technology* 2011, **2**:27:1–27:27.
- [190] Fernández A: **Epistructural Tension Promotes Protein Associations.** *Phys Rev Lett* 2012, **108**:188102.
- [191] Giovambattista N, Lopez CF, Rossky PJ, Debenedetti PG: **Hydrophobicity of protein surfaces: Separating geometry from chemistry.** *Proc Natl Acad Sci USA* 2008, **105**(7):2274–2279.
- [192] Sokal RR, Rohlf FJ: *Biometry: The Principles and Practices of Statistics in Biological Research.* W. H. Freeman, third edition edition 1994.
- [193] Florea L, Halldórsson B, Kohlbacher O, Schwartz R, Hoffman S, Istrail S: **Epi-tope Prediction Algorithms for Peptide based Vaccine Design.** In *Proceedings of the IEEE Computer Society Conference on Bioinformatics, CSB '03*, Washington, DC, USA: IEEE Computer Society 2003:17–.
- [194] Caoili SEC: **Benchmarking B-cell epitope prediction for the design of peptide-based vaccines: problems and prospects.** *J Biomed Biotechnol* 2010, **2010**(2010):910524.
- [195] Zhao L, Wong L, Hoi SC, Lu L, Li J: **B-cell epitope prediction through a graph model.** *BMC Bioinformatics* 2012.
- [196] Bickerton G, Higuero A, Blundell T: **Comprehensive, atomic-level characterization of structurally characterized protein-protein interactions: the PICCOLO database.** *BMC Bioinformatics* 2011, **12**:313.
- [197] Chen K, Kurgan L: **Investigation of Atomic Level Patterns in Protein-small Ligand Interactions.** *PLoS ONE* 2009, **4**(2):e4473.
- [198] Shaw DE, Maragakis P, Lindorff-Larsen K, Piana S, Dror RO, Eastwood MP, Bank JA, Jumper JM, Salmon JK, Shan Y, Wriggers W: **Atomic-Level Characterization of the Structural Dynamics of Proteins.** *Science* 2010, **330**(6002):341–346.
- [199] Kuhlman B, Dantas G, Ireton GC, Varani G, Stoddard BL, Baker D: **Design of a Novel Globular Protein Fold with Atomic-Level Accuracy.** *Science* 2003, **302**(5649):1364–1368.
- [200] Fernández A: **Variational mechanics of water at biological interfaces.** *J Phys A-Math Theor* 2011, **44**(29):292001.
- [201] Giovambattista N, Debenedetti PG, Rossky PJ: **Hydration Behavior under Confinement by Nanoscale Surfaces with Patterned Hydrophobicity and Hydrophilicity.** *J Phys Chem C* 2007, **111**(3):1323–1332.

- [202] Liu Q, Hoi SCH, Chinh STT, Li Z, Kwoh CK, Wong L, Li J: **Structural analysis of the hot spots in the binding between H1N1 HA and the 2D1 antibody: do mutations of H1N1 from 1918 to 2009 affect much on this binding?** *Bioinformatics* 2011.