Enhanced prediction of several protein structural attributes with machine learning algorithms

Yang, Jianyi

2012

Yang, J. (2012). Enhanced prediction of several protein structural attributes with machine learning algorithms. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/48058>

<https://doi.org/10.32657/10356/48058>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

**ENHANCED PREDICTION OF SEVERAL
PROTEIN STRUCTURAL ATTRIBUTES WITH
MACHINE LEARNING ALGORITHMS**

YANG JIANYI

SCHOOL OF PHYSICAL AND MATHEMATICAL SCIENCES

2012

ENHANCED PREDICTION OF SEVERAL PROTEIN STRUCTURAL
ATTRIBUTES WITH MACHINE LEARNING ALGORITHMS

YANG JIANYI

2012

**ENHANCED PREDICTION OF SEVERAL
PROTEIN STRUCTURAL ATTRIBUTES WITH
MACHINE LEARNING ALGORITHMS**

YANG JIANYI

School of Physical and Mathematical Sciences

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2012

Acknowledgments

First and foremost, I would like to thank my supervisor, Dr. Chen Xin, who has supported me throughout my Ph.D. study with his patience and knowledge whilst allowing me the room to work in my own way. His tireless pursuit of excellence in research is contagious and motivational for me. His remarkable trust and support are essential for the success of my Ph.D. pursuit.

Second, I would like to thank the following professors for their kind support and help during my Ph.D. study: Tai Xue-Cheng, Wang Desheng, Wang Li-Lian, Xiang Liming, Yu Zu-Guo, Wu Guohua, Lian Heng, Pan Guangmin, and Pang Zhen.

Third, I wish to express my gratitude to all the co-authors of my publications for their hard work and contributions, especially to Peng Zhen-Ling.

Fourth, I would like to thank the thesis examiners, whose insightful comments and suggestions help me to improve the quality of this dissertation.

Last but not least, I offer my best regards and blessings to my family and all of those who supported me in any respect during the period of my doctoral study.

Table of Contents

Summary	7
1 Introduction	9
1.1 Sequence-based <i>In-silico</i> Prediction of Protein Structure	9
1.2 Protein Structural Class, Fold Type, GPCR Family and Contact Map	12
1.2.1 Protein Structural Class	12
1.2.2 Protein Fold Type	15
1.2.3 G-Protein-Coupled Receptor Family	18
1.2.4 Protein Contact Map	21
1.3 Thesis Organization	22
2 Prediction of Protein Structural Class by Recurrence Quantifica-	
tion Analysis and Chaos Game Representation	27
2.1 Introduction	27
2.2 Materials and Methods	30
2.2.1 Datasets	30
2.2.2 Reverse Encoding for Amino Acids	31
2.2.3 Chaos Game Representation of Proteins	31
2.2.4 Recurrence Plot	34
2.2.5 Recurrence Quantification Analysis	36
2.2.6 Selection of Parameters	39
2.2.7 Fisher’s Discriminant Algorithm	39
2.2.8 Performance Evaluation	42
2.3 Results and Discussions	43
2.3.1 Accuracy of the Proposed Method	43

2.3.2	Comparisons with Other Methods	43
2.3.3	Effect of Reverse Encoding	46
2.3.4	Results of Other CGRs	47
2.4	Conclusions	47
3	Improving Protein Structural Class Prediction with Predicted Sec-	
	ondary Structure	49
3.1	Introduction	49
3.2	Materials and Methods	50
3.2.1	Datasets	50
3.2.2	Secondary Structure Prediction	51
3.2.3	Chaos Game Representation of Predicted Secondary Structure	52
3.2.4	Recurrence Plot and Recurrence Quantification Analysis . .	54
3.2.5	K -string Based Information Entropy	58
3.2.6	Segment-based Analysis	59
3.2.7	Fisher’s Discriminant Algorithm	60
3.2.8	Prediction Assessment	61
3.2.9	Selection of ε and K	61
3.3	Results and Discussions	62
3.3.1	Prediction Accuracies for Three Benchmark Datasets	62
3.3.2	Comparison with Existing Methods	63
3.3.3	Contribution of Features	65
3.3.4	Differentiating Between the $\alpha + \beta$ and α/β Classes	67
3.4	Conclusions	68
4	Improving Taxonomy-based Protein Fold Recognition by Using	
	Global and Local Features	71
4.1	Introduction	71
4.2	Materials and Methods	73
4.2.1	Datasets	73
4.2.2	Feature Extraction Methods	75
4.2.3	Support Vector Machines	82
4.2.4	The Proposed Method	83

4.2.5	Performance Evaluation	85
4.3	Results and Discussions	86
4.3.1	Optimal Values of λ and $l_{\max}$	86
4.3.2	Analysis of Feature Contribution	88
4.3.3	Accuracies for Four Tested Datasets	90
4.3.4	Fold Recognition for 710 Folds	92
4.3.5	Comparisons with Taxonomy-based Methods	94
4.3.6	Comparisons with Template-based Methods	99
4.4	Conclusions	101
5	An Improved Classification of G-Protein-Coupled Receptors Using Sequence-Derived Features	103
5.1	Introduction	103
5.2	Materials and Methods	107
5.2.1	Datasets	107
5.2.2	Physicochemical Properties	111
5.2.3	Sequence-Derived Features	112
5.2.4	Principal Component Analysis	119
5.2.5	Intimate Sorting Algorithm	120
5.2.6	Prediction Assessment	121
5.3	Results and Discussions	122
5.3.1	Selection of m	122
5.3.2	Predicting GPCRs at Five Levels	122
5.3.3	Contribution of Features	125
5.3.4	Comparison with BLAST-based Classification	129
5.3.5	Comparison with Previous Methods	130
5.4	Conclusions	133
6	A Consensus Approach to Predicting Protein Contact Map via Logistic Regression	135
6.1	Introduction	135
6.2	Materials and Methods	136
6.2.1	Datasets	136

6.2.2	Contact Definition	137
6.2.3	Performance Evaluation	138
6.2.4	Consensus Prediction via Logistic Regression	139
6.2.5	Overall Architecture	140
6.2.6	Selection of Residue Pairs	142
6.3	Results	142
6.3.1	Results on the CASP9 Dataset	143
6.3.2	A Good CASP9 Contact Prediction Example	146
6.3.3	Results on the D856 Dataset	146
6.4	Discussions	149
6.4.1	The Impact of Residue Pair Selection	150
6.4.2	The Impact of Individual Predictors	150
6.4.3	The Impact of Classification Algorithm	151
6.5	Conclusions	154
7	Conclusion	155
7.1	Summary	155
7.2	Technical Contributions	157
7.3	Limitations and Future Work	159
	Bibliography	163
	List of Publications	183
	Appendix A	185

Summary

The classical *sequence-structure-function* paradigm for proteins illustrates that the amino acid sequence of a protein determines its three-dimensional (3D) structure and function. With the great success of genome sequencing projects, the gap between the number of sequence-known proteins and the number of structure-known proteins is widening rapidly. *In-silico* prediction of protein structure from amino acid sequence has the potential to bridge this gap.

This thesis presents the machine learning-based computational methods that we developed to predict four protein structural attributes: (1) protein structural class, (2) protein fold, (3) G-protein-coupled receptors, and (4) protein contact map.

First, for protein structural class prediction, we propose to use the *chaos game representation* and *recurrence quantification analysis* to extract a set of features directly from the amino acid sequences. *Fisher's discriminant algorithm* is adopted as the classification algorithm, and about 65% overall accuracy is achieved for proteins from low-similarity datasets. Comparisons with other methods (that use the same kind of input information) show that the proposed method has higher or comparable accuracy depending on different datasets tested. When the similar idea is applied using the predicted protein secondary structure to predict the class, the resulting prediction accuracy could exceed 80%.

Second, for taxonomy-based protein fold recognition, a new method named

TAXFOLD is proposed by extracting a comprehensive set of global and local features from the PSI-BLAST and PSIPRED profiles. These features are then fed into *support vector machine* to make fold recognition. Experimental tests on seven datasets demonstrate that TAXFOLD makes an average 6.9% improvement over the best available taxonomic method and performs comparably well with the best conventional template-based fold recognition methods.

Third, for hierarchical classification of GPCRs, we develop a new method named PCA-GPCR that could classify GPCRs at all the *five levels* of the GPCR classification hierarchy. It relies on a comprehensive set of 1497 sequence-derived features. Because the number of dimensions of the feature space is very high, the *principal component analysis* is employed to reduce the dimensionality to 32. Jackknife tests on a large dataset show that the overall accuracies of PCA-GPCR at five levels (from the first to the fifth) are 99.5%, 88.8%, 80.47%, 80.3%, and 92.34%, respectively. Experimental comparisons show that PCA-GPCR consistently outperforms the BLAST-based classification and other competing predictors.

At last, for protein contact map prediction, a consensus approach named LRcon is proposed to improve the performance of existing predictors. Our new approach combines the prediction results from several complementary predictors by using a logistic regression model. Tests on the targets from the recent CASP9 experiment and a large dataset consisting of 856 protein chains show that LRcon not only outperforms its component predictors but also simple averaging and voting schemes.

Chapter 1

Introduction

1.1 Sequence-based *In-silico* Prediction of Protein Structure

Proteins are macromolecules performing numerous functions in biological organisms. The ‘building blocks’ of proteins are the 20 naturally occurring amino acids (residues). Typically, a protein may consist of 50 to hundreds of amino acids, which are bonded together by peptide bonds. The linear sequence of the amino acids in a protein is called the *primary sequence* of the protein. The composition of amino acids and their order of appearing in a protein sequence are believed to play an important role in determining the protein’s 3D structure and function, as indicated in the classical *sequence-structure-function* paradigm for proteins [1].

With the accomplishment of various genome sequencing projects, the gap between the number of proteins with known sequences and the number of proteins with known structures is becoming larger and larger. For example, there are 12,167,392 protein sequences in the release 46 of the NCBI’s RefSeq database (March 10, 2011). However, only 68,812 protein structures are deposited in the RCSB Protein Data Bank (PDB) [2] (July 05, 2011), which account for only

0.57% of protein sequences. Figure 1.1 shows the growing gap in the last 9 years. Sequence-based *in-silico* prediction of protein structure is probably the only way to bridge this large gap, so it has received intensive attention over the past five decades [3].

Figure 1.1: The numbers of protein sequences in the NCBI’s RefSeq database (left y-coordinate) and the experimentally solved protein structures in PDB (right y-coordinate) in the last 9 years.

In fact, *in-silico* prediction of the 3D structure of a protein from its amino acid sequence is one of the *grand challenges* in computational biology which remains unsolved after five decades of efforts [4]. It has been tackled from different aspects and levels. For example, protein structures have been predicted at 1D, 2D, and 3D levels, where the 1D and 2D predictions largely serve as intermediate steps for the

3D prediction. At the 1D level, each amino acid of a protein is predicted into certain structural features, such as *secondary structure* [5, 6] and *solvent accessibility* [7]. That is to say, the primary sequence of a protein is translated into a sequence of structural features as the result of the 1D prediction. Currently, about 82% accuracy has been achieved for the three-state secondary structure prediction [8]. At the 2D level, one of the most intensively studied problems is the prediction of *contact map* [9–14], which depicts the topological relationship between the residues of a protein. Even though the accuracy of contact map prediction is quite low at present, it is still valuable information that can be used to enhance the 3D structure prediction [15]. At the 3D level, it is desired to predict the coordinates of all atoms or residues of a protein [16]. The 3D structure prediction methods can be broadly divided into three categories: comparative modeling, threading, and free modeling (or Ab initio modeling) [17]. In comparative modeling, the homologous sequences (have $> 35\%$ sequence similarity with the target sequence) with known 3D structure are identified and used to infer the 3D structure of the target sequence. The most popular comparative modeling methods are MODELLER [18] and SWISS-MODEL [19]. The threading approach [20, 21] is also termed *fold recognition*, which aims to detect proteins sharing the same fold (please see Section 1.2.2 for the definition of protein fold) with the target sequence and use them to build the model of the target protein. The last category, free modeling, builds the 3D model of the target proteins from scratch based on physical principle or fragment assembly. The most representative method in this category may be attributed to the pioneer work (ROSETTA) by David Baker [22]. A highly influential platform for the assessment of various protein structure prediction methods is CASP (Critical Assessment of Techniques for Protein Structure Prediction). It is a community-wide, worldwide experiment for protein structure prediction and held every two years since 1994. According to the last three CASP experiments (CASP7, CASP8, CASP9), the I-

TASSER server [16] developed by the Zhang lab (<http://zhanglab.ccmb.med.umich.edu/>) represents the state-of-the-art method for 3D structure prediction [23–25].

Besides the above 1D, 2D and 3D predictions of protein structure, significant efforts have been made to predict many other protein structural attributes (for a comprehensive review of these attributes, please see [26]). In this thesis, we will consider sequence-based *in-silico* predictions of the following four protein structural attributes: (1) structural class, (2) fold type, (3) G-protein-coupled receptor (GPCR) family, and (4) contact map. When developing predictors for these structural attributes, we bear in mind three aims to be achieved whenever possible:

- Make the methods simple and fast.
- Improve the predictive quality over the state-of-the-art methods.
- Provide web servers for public use.

1.2 Protein Structural Class, Fold Type, GPCR Family and Contact Map

In this section, the backgrounds about protein structural class, fold type, GPCR family and contact map are briefly presented.

1.2.1 Protein Structural Class

There are two important databases in which protein structures are classified hierarchically: SCOP (Structural Classification of Proteins) [27] and CATH (Class, Architecture, Topology and Homologous superfamily) [28]. The SCOP classification of proteins has been constructed manually by visual inspection and structure

comparison, whereas the CATH classification applies a combination of automated and manual procedures. SCOP provides a detailed and comprehensive description of the structural and evolutionary relationships of protein structures. It organizes the protein structures hierarchically at different levels (from top to bottom): class, fold, superfamily, family, and species. In comparison, CATH clusters proteins at four levels only: class, architecture, topology, and homologous superfamily.

Both SCOP and CATH define structural class but their definitions are different. At the class level of SCOP, the tertiary structures of proteins are categorized into 11 structural classes: (1) α , (2) β , (3) $\alpha + \beta$, (4) α/β , (5) multi-domain proteins, (6) membrane and cell surface proteins, (7) small proteins, (8) coiled coil proteins, (9) low resolution proteins, (10) peptides, and (11) designed proteins. According to the secondary structure composition and packing within the structure, CATH defines four structural classes: mainly- α , mainly- β , α - β , and protein domains which have low secondary structure content. Because the structural classes defined in CATH do not differentiate between $\alpha + \beta$ and α/β (as defined by SCOP), we do not adopt its definition here. In the remaining of this thesis, the definition of structural class is based on the database SCOP by default.

The knowledge of the structural class is beneficial to study protein function, regulation and interactions. For instance, the information about structural class has been used when developing methods to tackle problems such as contact map prediction [29], conformation search [30], DNA-binding sites prediction [31], secondary structure content prediction [32], discrimination of outer membrane proteins [33], protein folding rates prediction [34], and so on. Therefore, if we can predict the structural class of a protein from its amino acid sequence, it would provide substantial help in other related researches on proteins.

Sequence-based prediction of the first four major structural classes (α , β , $\alpha + \beta$, α/β) has been intensively studied [35–61]. The classification of these classes is

mainly based on the types and arrangements of their secondary structural elements (i.e., α helix, β sheet, and random coil). Proteins in the α (resp. β) class contain mainly helices (resp. strands), and proteins in the $\alpha + \beta$ and α/β classes contain a mixture of α helices and β strands —the α helices and β strands are largely segregated in the $\alpha + \beta$ class, but instead largely interspersed in the α/β class [62]. Figure 1.2 shows representative structures for the four major SCOP classes.

Figure 1.2: Cartoon structures of proteins in the four structural classes. (a) α class, PDB ID: 2QB0; (b) β class, PDB ID: 2K57; (c) $\alpha + \beta$ class, PDB ID: 3BH8; (d) α/β class, PDB ID: 3DEI.

1.2.2 Protein Fold Type

The arrangement type and topological connection of secondary structure elements in space are commonly defined as the fold of a protein. For example, TIM barrel is a typical protein fold consisting of eight α helices and eight parallel β strands that alternate along the peptide backbone, which is shown in Figure 1.3. The fold level in SCOP is a level below the class level, and thus a detailed classification of the class. For example, the α class is further classified into many different folds: *globin-like*, *long α -hairpin*, *type I dockerin domain*, *LEM/SAP HeH motif*, *DNA/RNA-binding 3-helical bundle*, and so on. Like the structural class, fold can be used to measure the overall structural similarity between different proteins.

How many types of protein folds are there? Taking into account the huge number of amino acid sequences, one would expect a high number of different folds. But in fact it is not the case. The number of folds is instead very limited, for example, the total number of folds in globular, water-soluble proteins was estimated to be about 1,000 [63, 64]. According to the PDB Statistics (http://www.pdb.org/pdb/static.do?p=general_information/pdb_statistics/index.html), the total number of folds available in the PDB (based on SCOP, V1.75) is 1,393. The growth in the number of folds available in the PDB in the last 21 years is illustrated in Figure 1.4. As seen from the figure, the number of folds seems to become stable in recent years, because no new folds were found in the PDB library during the years 2009-2011. A study [65] already claimed that the current PDB library of the solved protein structures is complete at the level of single-domain proteins, which brought better opportunities to solve the protein structure prediction problem. With the advance from PSI:Biology (Protein Structure Initiative, <http://sbkb.org/about/psi.html>), more protein 3D structures become available, which however would change the above conclusion.

Figure 1.3: Structure illustration of the TIM barrel fold with a typical protein (PDB ID: 2PDG).

Figure 1.4: The number of folds available in the PDB based on SCOP from 1990 to 2011.

A protein’s fold is more evolutionarily conserved than its amino acid sequence. Thus, one of the key steps in the template-based modeling of protein structure prediction is to identify the fold of a target sequence [4]. Through *fold recognition* we are able to identify proteins with known structures that share common folds but have very low sequence similarity with the target sequence. The folds of the identified structures can then be used as templates to model the structure of the target sequence. The conventional methods for fold recognition is *threading*, in which the target sequence is aligned against a library of potential fold templates to find the *best* template [21]. Another category of fold recognition approach is *taxonomy-based fold recognition* [66, 67] (also called protein fold type prediction [68, 69]), which classifies a query protein into one of the known folds and has been shown very promising for protein fold recognition.

1.2.3 G-Protein-Coupled Receptor Family

G-protein-coupled receptors (GPCRs) comprise the largest family of integral membrane proteins and act as cell surface receptors responsible for the transduction of extracellular signals into intracellular reactions [70]. The structure of a GPCR generally comprises seven α -helical transmembrane domains and thus GPCRs are also known as 7TM receptors [71]. Figure 1.5 shows the structure of a typical GPCR protein, β_2 -adrenergic receptor [72].

Many diseases associated with differentiation, proliferation, angiogenesis, cancer, development, and cell survival involve the malfunction of these receptors, which makes them the most important drug targets [73, 74]. For example, among the 1,357 unique components from the 21,000 FDA approved drug products (FDA, December 2005), more than a quarter of drugs target the Class A Rhodopsin-like GPCRs [75].

Although GPCRs play an important role in drug market, their 3D experimen-

Figure 1.5: The structure (shown in green cartoon) of β_2 -adrenergic receptor (taken from http://en.wikipedia.org/wiki/Beta-2_adrenergic_receptor).

tal structures are largely unavailable [76]. Relative to globular proteins, the overall low success rate for GPCR structure determination is mainly due to the difficulties in the expression of recombinant receptors because of the inefficient transport and insertion of receptors in the plasma membrane and the toxic effect on host cells [77]. In addition, GPCR purification requires detergents that have a negative effect on receptor yields and stability. Finally, the crystallization process is hampered by the presence of detergents, inherent GPCR flexibility, and the relatively small hydrophilic loops that provide few potential crystal contacts [77, 78]. Until now, there are only four human GPCRs that have their structures solved, i.e. β_2 -adrenergic [72], adenosine A2a [79], CXCR4 chemokine [80], and dopamine D3 receptors [81]. Therefore, sequence-based analysis of GPCRs [71] and GPCR modeling [82] are of great value, especially for the purpose of modern drug discovery.

The GPCRDB (G-Protein-Coupled Receptor Data Base) is a molecular-class

information system that collects, combines, validates and stores large amounts of heterogenous data on GPCRs [71] (<http://www.gpcr.org/7tm/>). Based on pharmacological classification and/or binding ligand types of GPCRs, GPCRDB hierarchically organizes GPCRs into families, sub-families, *etc.* (Figure 1.6). For example, the current version (as of 4 Mar 2011) of the GPCRDB classifies GPCRs into six families, four of which are further classified into sub-families, sub-sub-families and so forth. One interesting question arises: could the GPCRs be accurately classified by using the sequence information alone? In recent years, some methods for sequence-based classification of GPCR families were already developed to answer this question [54, 83–87].

The screenshot shows the GPCRDB website interface. On the left, a tree view titled 'GPCR families' shows a hierarchy starting with 'Class A Rhodopsin like (19)', which includes 'Amine (7)'. The 'Amine' sub-family is expanded, showing various receptors like 'Peptide (32)', 'Hormone protein (5)', etc. On the right, the 'Amine family' page is displayed, showing a list of proteins. The table below shows the first few entries from this list.

ID	Description	Species	Family
5HT1AA_TAKRU	5-hydroxytryptamine receptor 1A-alpha	Takifugu rubripes	Serotonin type 1a
5HT1AB_TAKRU	5-hydroxytryptamine receptor 1A-beta	Takifugu rubripes	Serotonin type 1a
5HT1A_CANFA	5-hydroxytryptamine receptor 1A	Canis lupus familiaris	Serotonin type 1a
5HT1A_GORGO	5-hydroxytryptamine receptor 1A	Gorilla gorilla gorilla	Serotonin type 1a
5HT1A_HORSE	5-hydroxytryptamine receptor 1A	Equus caballus	Serotonin type 1a
5HT1A_HUMAN	5-hydroxytryptamine receptor 1A	Homo sapiens	Serotonin type 1a

Figure 1.6: An illustration of the GPCR family page in the GPCRDB system [88]. The GPCR family tree is shown on the left and GPCRs in the *Amine* sub-family are shown on the right.

1.2.4 Protein Contact Map

Contact map is a 2D description of protein structure. It is represented by a symmetrical matrix M with elements being 0 or 1. The element M_{ij} in M is 1 if the distance between the i th and the j th residue of the protein in 3D space is less than a preassigned threshold, and 0 otherwise. A graph can be used to depict the matrix by plotting the matrix on a 2D space (each element with value 1 is represented by a dot). One of the clear advantages of this representation is that a contact map is invariant to rotations and translations of the 3D structures. Figure 1.7 shows an example of contact map.

Figure 1.7: An example of contact map. The contacts are omitted for those residue pairs being very close along the sequence ($|j - i| < 6$).

The information embedded in the contact map is useful in a variety of researches

on proteins. As shown in [89], it is possible to reconstruct the 3D coordinates of a protein using its contact map under certain circumstances. Contact maps are also used for protein superimposition and to describe similarity between protein structures [90]. In addition, the predicted contact maps can be used to evaluate protein models [91], select the models generated by structure prediction programs [92], improve the predictions of 3D structure [15], helix-helix interactions [93], transmembrane helix packing arrangements [94], protein folding rates [95], and so on. Therefore, sequence-based prediction of contact map has attracted much attention in the past years and has been a part of the CASP since CASP2 [96]. However, the prediction accuracy of contact map is notoriously low [92].

1.3 Thesis Organization

A statistical machine learning approach is typically concerned with the development of algorithms and techniques to construct models for the observed data and then to make predictions for the query data. It has been widely employed to solve various kinds of problems in computational biology: protein structure prediction [16, 97], RNA-binding sites prediction [98], gene finding [99], promoter prediction [100, 101], RNA secondary structure prediction [102], and so on. This thesis presents several machine learning algorithms that we developed for solving the following four problems: (1) protein structural class prediction, (2) taxonomy-based fold recognition, (3) hierarchical classification of GPCRs, and (4) protein contact map prediction. The thesis is organized as follows.

Chapter 2 aims at predicting protein structural class for amino acid sequences from a low-similarity dataset. We propose to decompose the *chaos game representation* (CGR) of an amino acid sequence into two time series. Then, a non-linear analysis technique, called *recurrence quantification analysis* (RQA), is applied to

analyze these time series. For a given protein sequence, a total of 16 features are calculated with RQA, which are fed into Fisher’s discriminant algorithm to make predictions. Two widely used low-similarity datasets were used to evaluate the proposed method: *1189* (containing 1092 proteins) and *25PDB* (containing 1673 proteins). Their pairwise sequence identities are below 40% and 25%, respectively. It was demonstrated that the overall accuracies of the *step-by-step* procedure (evaluated by *jackknife* test) are 65.8% and 64.2% for *1189* and *25PDB* datasets, respectively. Comparisons based on the two datasets show that the overall accuracies of our method are higher than or comparable with the other methods (excluding those methods that use the predicted secondary structure information).

Chapter 3 further improves the prediction of protein structural class by utilizing the predicted secondary structure information. In particular, PSIPRED is first used to predict the secondary structure for each protein sequence. A comprehensive set of 24 features are then generated from the predicted secondary structure by using CGR, RQA, *K-string based information entropy* and *segment-based analysis*. These feature vectors are finally fed into Fisher’s discriminant algorithm for the prediction of protein structural class. We tested the proposed method on three benchmark datasets in low similarity and achieved the overall prediction accuracies of 82.9%, 83.1% and 81.3%, respectively. Comparisons with existing methods showed that our method performs better or comparably well for all the tested datasets. A web server that implements the proposed method is freely available at http://www1.spms.ntu.edu.sg/~chenxin/RKS_PPSC/. The high prediction accuracy achieved by our proposed method is attributed to the design of a comprehensive feature set on the predicted secondary structure sequences, which is capable of characterizing the sequence order information, local interactions of the secondary structural elements, and spacial arrangements of α helices and β strands. Thus, it is a valuable method to predict protein structural class particularly for

low-similarity amino acid sequences.

Chapter 4 proposes to improve taxonomy-based fold recognition with global and local features. A new taxonomy-based protein fold recognition method called TAXFOLD is developed. It extensively exploits the sequence evolution information from PSI-BLAST profiles and the secondary structure information from PSIPRED profiles. A comprehensive set of 137 features is constructed, which allows for the depiction of both global and local characteristics of PSI-BLAST and PSIPRED profiles. We tested TAXFOLD on six datasets and compared it with several major existing taxonomic methods for fold recognition. Its recognition accuracies range from 79.6% to 90% for 27, 95 and 194 folds, achieving an average 6.9% improvement over the best available taxonomic method. Further test on the Lindahl benchmark dataset shows that TAXFOLD is comparable to the best conventional template-based threading method at the SCOP fold level. These experimental results demonstrate that the proposed set of features is highly beneficial to protein fold recognition. A web server that implements the proposed method is freely available at <http://www1.spms.ntu.edu.sg/~chenxin/TAXFOLD/>.

Chapter 5 presents a new method called PCA-GPCR, to classify GPCRs hierarchically at *five* levels using a comprehensive set of 1497 sequence-derived features. The *principal component analysis* is first employed to reduce the dimension of the feature space to 32. Then, the resulting 32-dimensional feature vectors are fed into a simple yet powerful classification algorithm, called intimate sorting, to predict GPCRs at *five* levels. The prediction at the first level determines whether a protein is a GPCR or a non-GPCR. If it is predicted to be a GPCR, then it will be further predicted into certain *family*, *subfamily*, *sub-subfamily* and *subtype* by the classifiers at the second, third, fourth, and fifth levels, respectively. To train the classifiers applied at five levels, a non-redundant dataset is carefully constructed, which contains 3178, 1589, 4772, 4924, and 2741 protein sequences at the respective

levels. Jackknife tests on this training dataset show that the overall accuracies of PCA-GPCR at five levels (from the first to the fifth) are 99.5%, 88.8%, 80.47%, 80.3%, and 92.34%, respectively. We further perform predictions on a dataset of 1238 GPCRs at the second level, and on another two datasets of 167 and 566 GPCRs at the fourth level. The overall prediction accuracies of our method are consistently higher than those of the existing methods compared. The comprehensive set of 1497 features are believed to be capable of capturing information about amino acid composition, sequence order as well as various physicochemical properties of proteins. Therefore, high accuracies are achieved when predicting GPCRs at all the five levels with our proposed method. For the convenience of public use, a web server was developed, which is freely available at http://www1.spms.ntu.edu.sg/~chenxin/PCA_GPCR/.

Chapter 6 presents a consensus approach to improve the prediction of protein contact map. In this chapter, a consensus contact map prediction method called LRcon is developed, which combines the prediction results from several complementary predictors by using a logistic regression model. Tests on the targets from the recent CASP9 experiment and a large dataset D856 consisting of 856 protein chains show that LRcon not only outperforms its component predictors but also the simple averaging and voting schemes. For example, LRcon achieves 41.5% accuracy on the D856 dataset for the top $L/10$ long-range contact predictions, which is about 5% higher than its best-performing component predictor. The improvements made by LRcon are mainly attributed to the application of a consensus approach to complementary predictors and the logistic regression analysis under the machine learning framework.

Chapter 7 finally summarizes the thesis and provides some suggestions for the future research direction.

Chapter 2

Prediction of Protein Structural Class by Recurrence Quantification Analysis and Chaos Game Representation [†]

2.1 Introduction

In this chapter, we aim to predict the four main protein structural classes (α , β , $\alpha + \beta$, and α/β) from the amino acid sequences. We list in Table 2.1 the major protein structural class prediction methods developed in last decade. As we can see from the table, these methods differ in two aspects: one is the input information and the other is the classification algorithm, which are reviewed as follows.

[†] This chapter is mainly based on an article published in *Journal of Theoretical Biology* as: Jian-Yi Yang, Zhen-Ling Peng, Zu-Guo Yu, Rui-Jie Zhang, Vo Anh, and Desheng Wang, Prediction of protein structural classes by recurrence quantification analysis based on chaos game representation, *Journal of Theoretical Biology*, 257:618-626, 2009.

Table 2.1: List of the input information and classifiers used by existing structural class predictors.

Reference	Input information						Classifier	Year
	AAC	CUSTOM	PSC	PSS	PSSM	FunD		
[35]	X						LMD	1995
[36]	X						LMD	1995
[37]	X						Bayes	2000
[38]	X						NN	2000
[39]	X						SVM	2001
[40]	X	X					STD	2002
[41]	X						SVM	2003
[42]						X	IS	2004
[43]	X						FC	2005
[44]	X	X					MULTI	2006
[45]	X	X					Ensemble	2006
[46]		X					SVM	2006
[47]	X						LogitBoost	2006
[48]	X	X					Rough sets	2006
[49]	X	X	X				LR	2007
[50]	X	X					IDQD	2007
[51]	X	X	X	X			SVM	2008
[52]				X			SVM	2008
[53]		X					FKNN	2008
[54]	X	X					CD	2008
[55]	X	X	X	X	X		MULTI	2008
[56]	X	X					SVM	2009
[60]		X					FD	2009
[57]	X	X	X	X	X		SVM	2009
[61]				X			FD	2010
[58]					X		SVM	2010
[59]					X		SVM	2011

Meaning of abbreviations: AAC: amino acid composition; CUSTOM: other custom-designed features using solely the amino acid sequence, such as pseudo-amino acid composition, dipeptide composition, physicochemical measure, sequence-order descriptor, and so on; PSC: predicted secondary structure; PSS: predicted secondary structure; PSSM: position-specific score matrix; FunD: functional domain; MULTI: multiple classifiers; LMD: least Mahalanobis distance, STD: stepwise discriminant analysis, IDQD: increment of diversity combined with quadratic discriminant, CD: covariant-discriminant, LR: logistic regression, FD: Fisher's discriminant; IS: intimate sorting; FC: fuzzy clustering; FKNN: Fuzzy k-nearest neighbor; NN: neural network; SVM: support vector machine.

The major types of input information include the amino acid sequence, predicted secondary structure content, predicted secondary structure, functional domain [42], and PSI-BLAST profile (i.e., PSSM matrix) [103]. The most widely used are the amino acid composition and other features computed from individual amino acid sequences [35–41, 43, 47, 53, 54, 56, 60]. The amino acid composition represents a protein as a twenty-dimensional vector corresponding to the frequencies of 20 amino acids in a given protein’s amino acid sequence [35–39, 43]. However, these features ignored the important sequence order information which has been shown beneficial to the predictions. To overcome this limitation to some extent, various new feature sets were designed, e.g., dipeptide composition and *pseudo amino acid (PseAA) composition* [104]. These custom-designed set of features have been shown very successful in the prediction of protein structural class [46, 53, 54, 56, 60], especially for high-similarity protein datasets. However, when low-similarity datasets with pairwise sequence similarity below 40% were tested, these methods were not so effective any more. For instance, for the widely used dataset *25PDB* whose sequence similarity is about 25%, the reported overall accuracies with these methods were below 70% [44, 45, 53, 54, 56, 60]. To address this challenge, Kurgan *et al.* [49, 55] proposed to incorporate the predicted secondary structure *content* (i.e., composition of α -helix, β -strand, and random coil) and evolutionary information embedded in the PSI-BLAST profile. Thus, higher prediction accuracy was reported. In addition, with more protein structural class related features extracted from the predicted secondary structure (e.g., PSIPRED), over 80% prediction accuracy could be achieved [51, 57–59, 61] for low-similarity protein datasets.

A variety of classification algorithms have been used in protein structural class prediction, for example, covariant-discriminant analysis [54], fuzzy clustering [43], LogitBoost [47], Bayesian decision theory [37], neural network [38], and support

vector machine (SVM) [39, 41, 46, 51, 57]. From the table, we can see that SVM [105] is probably the most commonly used algorithm in protein structural class prediction.

In this chapter, we would like to introduce a new comprehensive feature set that was extracted from the primary sequence based on chaos game representation [106] and recurrence quantification analysis [107]. We demonstrate by experiments on two benchmark datasets that the prediction of protein structural class can be improved for low-similarity amino acid sequences.

2.2 Materials and Methods

2.2.1 Datasets

Two widely used low-similarity protein datasets, *25PDB* and *1189*, are used to evaluate the proposed method and provide a comprehensive and unbiased comparison with the existing prediction methods. They are downloaded from RCSB Protein Data Bank [2] (<http://www.rcsb.org/pdb/home/home.do>) with the PDB IDs listed in [44]. The dataset *25PDB* contains 1673 proteins of pairwise sequence similarity being about 25%, while the dataset *1189* contains 1092 proteins of 40% sequence similarity. Table 2.2 shows for the above two datasets the numbers of proteins belonging to each structural class, where protein structural classifications are retrieved from the database SCOP [27] and considered as *true* for prediction evaluation.

Table 2.2: The number of proteins in the datasets.

Dataset	α	β	α/β	$\alpha + \beta$	Total
<i>25PDB</i>	443	443	346	441	1673
<i>1189</i>	223	294	334	241	1092

2.2.2 Reverse Encoding for Amino Acids

Several different DNA codons may code the same amino acid. For example, the amino acid *Leucine* (L) can be coded by six DNA codons: TTA, TTG, CTT, CTA, CTC, and CTG. As a result, there are many *candidate DNA sequences* that can be translated into a given protein sequence. Here we use the *reverse* encoding method listed in Table 2.3, which has been used in [108]. Deschavanne and Tufféry [108] explained that “the rationale for the choice of this fixed code was to keep a balance in base composition so as to maximize the difference between the amino acid codes”. Two more selected reverse encoding methods will be discussed in Section 2.3.

Table 2.3: The reverse encoding of amino acids in Ref. [108].

A=GCT	G=GGT	M=ATG	S=TCA	C=TGC	H=CAC	N=AAC	T=ACT	D=GAC	I=ATT
P=CCA	V=GTG	E=GAG	K=AAG	Q=CAG	W=TGG	F=TTC	L=CTA	R=CGA	Y=TAC

2.2.3 Chaos Game Representation of Proteins

Chaos game representation (CGR) of protein structures was first proposed by Fiser *et al.* [106]. We denote this CGR by 20-CGR because 20 letters were used in this method to represent a protein sequence. Later Basu *et al.* [109] and Yu *et al.* [110] proposed other kinds of CGRs for proteins, in which 12 and 4 letters were used to represent protein sequences, respectively. We denote them by 12-CGR and 4-CGR, respectively. We previously applied a 6-letter model to cluster 49 large proteins in [111], so we will discuss below this model (denoted by 6-CGR) as well.

From previous section, a protein sequence could be transformed back into DNA sequences with a fixed reverse encoding of amino acids [108]. It thus allows the CGR of DNA sequences [112] to be applied to protein sequences as well. For the sake of convenience, we denote this representation method by AAD-CGR (Amino

Acids to DNA). Our subsequent discussions in this chapter are mainly based on this representation.

After a protein sequence is transformed into a DNA sequence, we can use AAD-CGR [112] to analyze it. We recapture the concept of CGR briefly here. CGR on a DNA sequence is defined in a square $[0, 1] \times [0, 1]$, where the four corner vertices correspond to the four letters A, C, G and T. The first point of the plot is placed half way between the center of the square and the vertex corresponding to the first letter of the DNA sequence; the i -th point of the plot is then placed half way between the $(i - 1)$ -th point and the vertex corresponding to the i -th letter. The obtained plot is then called CGR of the DNA sequence, or equivalently the AAD-CGR of the protein sequence. The AAD-CGR of a typical protein is shown in Figure 2.1 as an example.

It is not easy to analyze the resulting 2-dimension plot directly. Noting that the AAD-CGR of proteins is determined by the (x, y) coordinates, we proposed in our previous work [113] to decompose the AAD-CGR plot into two time series and then analyze them by *multifractal* analysis. Any point in the AAD-CGR plot is determined by two coordinates, namely, x and y coordinates. Thus, two time series can be generated from the AAD-CGR plot: the first time series comprises the x -coordinates of the plotted points and the second comprises their corresponding y -coordinates. We denote them as CGRx and CGRy, respectively. Figure 2.2 shows the two time series generated from the AAD-CGR plot depicted in Figure 2.1.

Because the AAD-CGR plot can be uniquely reconstructed from these two time series, all the information presented in the AAD-CGR plot is included in the time series. In addition, the information in the AAD-CGR plot is derived from the primary sequence of proteins. Therefore, analysis of the two time series is equivalent to analysis of the primary sequence of a protein. It is anticipated that such analysis could provide better results than direct analysis on the primary

Figure 2.1: AAD-CGR of a protein (PDB ID: 1AAB). One point in the figure represents one corresponding amino acid in the protein sequence. The order for the points (corresponding to the order in the protein sequence) is saved (not shown in the figure). See text for more details about how to obtain such plots.

Figure 2.2: Two time series generated from the plots of Figure 2.1. The first time series (left panel) comprises the x-coordinates of the points in Figure 2.1 and the second (right panel) comprises their respective y-coordinates.

sequences of proteins.

Previously, we applied the 20-CGR to predict the structural class for 100 large proteins based on multifractal analysis [113]. A shortcoming of this approach is that it requires the protein length to be long enough (often > 300 , i.e. large proteins). However, many proteins (especially the domain sequences) in the PDB library, can not meet this requirement. In order to overcome this shortcoming, we adopt another analysis technique called *recurrence quantification analysis* (RQA) [107], which is described below.

2.2.4 Recurrence Plot

Recurrence plot (RP) is a purely graphical tool originally proposed by Eckmann *et al.* [114] to detect patterns of recurrence in the data. It is able to describe the natural time correlation information in a time series. Given a time series $z_1 z_2 \cdots z_L$ of length L , we first embed it into the space R^m of dimension m using a time delay τ . Let us define

$$Z_i = (z_i, z_{i+\tau}, z_{i+2\tau}, \cdots, z_{i+(m-1)\tau}), \quad i = 1, 2, \cdots, N_m, \quad (2.1)$$

where $N_m = L - (m - 1)\tau$. Hence, we obtain N_m vectors (*i.e.*, points) in the embedding space R^m . The values of m and τ have to be chosen appropriately based on nonlinear dynamical theory [115] and it will be discussed in Section 2.2.6. We further construct a *distance matrix* (DM=($D_{i,j}$)) of size $N_m \times N_m$ from the N_m points. Its elements are the (Euclidean) distances between all pairs of points after being scaled down by the maximum distance. As a result, all the element values of DM are located in the interval between 0 and 1, one advantage of which is to allow the recurrence plots in different scales to be statistically compared [115]. Finally, we define a *recurrence matrix* (RM) by applying a *threshold* ε (namely radius) on

the element values of DM. Formally, let $\text{RM}=(R_{i,j}(\varepsilon))_{N_m \times N_m}$ and

$$R_{i,j}(\varepsilon) = H(\varepsilon - D_{i,j}), \quad i, j = 1, 2, \dots, N_m \quad (2.2)$$

where H is the *Heaviside function*; that is,

$$H(x) = \begin{cases} 0, & \text{if } x < 0, \\ 1, & \text{if } x \geq 0. \end{cases} \quad (2.3)$$

Recurrence plot is simply a visualization of RM by plotting points on i - j plane for those elements in RM with values equal to 1. If $R_{i,j}(\varepsilon) = 1$, we say the j -th point recurs with reference to the i -th point. For any $\varepsilon > 0$, the RP has always a black line along main diagonal since $R_{i,i}(\varepsilon) \equiv 1$. Furthermore, the RP is symmetric with respect to the main diagonal as $R_{i,j}(\varepsilon) = R_{j,i}(\varepsilon)$.

For example, the RPs for the two time series shown in Figure 2.2 are given in Figure 2.3. ε is a crucial parameter of RP. If ε is chosen too small, then there might leave only a few of recurrence points so that we can not learn any recurrence structure of the underlying time series. But if ε is too large, almost all the points will be enclosed in the neighbor of a point, thereby introducing a lot of structure artifacts. Because the elements $D_{i,j}$ in the matrix DM is between 0 and 1, the allowed range of ε is set to be $[0, 1]$. For the sake of discussion convenience, we represent ε as a percentage between 0% and 100% in the rest of this chapter. We will discuss the selection of ε value in Section 2.2.6.

Figure 2.3: The corresponding RPs for the two time series in Figure 2.2. The parameters used are $m = 8$, $\tau = 2$, and $\varepsilon = 30\%$. Note that there is a black line along the main diagonal in the plots since a point always recurs with itself. Moreover, the points in the RP are symmetric with respect to the main diagonal line.

2.2.5 Recurrence Quantification Analysis

Recurrence quantification analysis (RQA) is a nonlinear technique used to quantify the information supplied by a recurrence plot [116, 117]. RQA has been successfully applied to many different fields [118, 119]. The ability of RQA to deal with protein sequences was investigated in [120–124]. The works using signal analysis methods in elucidation of protein sequence-structure relationships were reviewed in [107]. In analyzing time series, unlike the multifractal analysis [111, 113, 125], RQA does not have a strict requirement for the length of time series. Therefore, it is possible to apply RQA in the prediction of structural class of small proteins. For convenience, we briefly describe RQA as follows. There are 8 recurrence variables used to quantify each RP [115, 126]. We refer the readers to Refs. [115, 126] as there are detailed descriptions of these variables. Because the RP is symmetric with respect to the main diagonal, the recurrence points considered in the following definitions will only comprise those in the upper triangle of a RP (excluding the main diagonal line as well).

The first recurrence variable is called *recurrence* (*%REC*). It is a measure of the density of recurrence points in a RP, taking a value ranging from 0 (when there is no recurrent point) to 1 (when all points are recurrent). That is,

$$\%REC = \frac{\# \text{ recurrent points in upper triangle}}{N_m(N_m - 1)/2}, \quad (2.4)$$

where $\#$ stands for counting the number of points.

The second recurrence variable is called *determinism* (*DET*). It measures the proportion of recurrent points that form diagonal line structures. Before evaluating this variable, we need to set the minimum number of recurrent points that a diagonal line segment requires. The commonly used number is 2, which is used in this study as well. Formally, we define determinism as

$$DET = \frac{\# \text{ recurrent points in diagonal lines}}{\# \text{ recurrent points}}. \quad (2.5)$$

The third recurrence variable is called *linemax* and denoted by L_{max} . It simply represents the length of the longest diagonal line segment in RP, and essentially inversely scales with the largest positive Lyapunov exponent [114]. Note that in general, the longer a time series, the longer diagonal line segments as well. In order to cancel the length influence of the time series (equal to the length of the corresponding protein amino acid sequence), we normalize the length of the longest diagonal line segment by dividing N_m . That is,

$$L_{max} = \frac{\text{length of the longest diagonal line in RP}}{N_m}. \quad (2.6)$$

The fourth recurrence variable is *entropy* (*ENT*), which is the Shannon information entropy of the distribution probability of the length of the diagonal lines.

That is,

$$ENT = - \sum_{k=L_{min}, p(k) \neq 0}^{L_{max}} p(k) \log_2(p(k)), \quad (2.7)$$

where L_{min} is the minimum length of diagonal lines in RP and

$$p(k) = \frac{\# \text{ diagonal lines of length } k \text{ in RP}}{\# \text{ diagonal lines in RP}}. \quad (2.8)$$

The fifth recurrence variable is called *trend* (TND), which quantifies the stationarity degree of time series. It is calculated as the level that the *local recurrences* of diagonal lines fits their displacements from the main diagonal by least squares regression, where the *local recurrence* of a diagonal line refers to the proportion of points on the diagonal line that are the recurrence points. We would like to emphasize that the variable $\%REC$ is defined on the whole upper triangle of RP while the *local recurrence* is instead defined only on a certain diagonal line of RP.

The remaining three variables are defined based on the vertical line structure. The sixth recurrence variable is called *laminarity* (LAM). It is analogous to DET , but calculated using recurrence points forming vertical line structures. That is,

$$LAM = \frac{\# \text{ recurrent points in vertical lines}}{\# \text{ recurrent points}}. \quad (2.9)$$

The seventh variable, called *trapping time* (TT), is the *normalized* average length of vertical line structures (i.e., average length of vertical line structures divided by N_m). The eighth recurrence variable is the *maximum normalized length of the vertical lines* in RP, which is analogous to the definition of L_{max} and denoted by V_{max} .

2.2.6 Selection of Parameters

As mentioned in Section 2.2.4, we have to select the values of three parameters in RP: embedding dimension m , time delay τ , and radius ε . Because RP is analyzed by RQA, we use the variable $\%REC$ to select these parameters. Similar to that done in [115], we examine the values of $\%REC$ for embedding dimensions m between 6 and 9, and time delay τ from 1 to 4. A general guideline is that ε should be selected such that $\%REC$ remains low (often less than 5%) [115]. We are looking for small (or smooth) changes in parameter settings yielding small (or smooth) changes in output measures.

The time series used here is the x -coordinate sequence (CGRx) in AAD-CGR of protein 1AAB. In order to decide the values of the parameters in RQA, we present the surface plots for embedding dimensions m from 6 to 9, delays τ from 1 to 4 and radius ε from 10% to 39% in Figure 2.4. We finally select $m = 8$ and $\tau = 2$, because we observed from the figure that the changes for $\%REC$ are relatively small for $m = 7, 8, 9$, that the changes are smooth for $m = 8$ with the increase of radius ε and τ , and that the values of $\%REC$ are relatively stable when $\tau \geq 2$. The value of ε is select to be 30% to make sure $\%REC$ is smaller than 5% [115]. Note that these parameters are kept the same for all proteins in all datasets used in this chapter.

2.2.7 Fisher’s Discriminant Algorithm

For each protein, a set of 16 features (2×8 , 2 means two RPs, and 8 means 8 variables used to quantize each RP) can be computed as described above, and these features will be fed into a classification algorithm to predict the protein’s structural class. In this chapter, a simple yet powerful algorithm Fisher’s linear discriminant algorithm [127] is used as the classification algorithm.

Figure 2.4: Surface plots of $\%REC$ (%) with embedding dimensions varying between 6 and 9, and delays between 1 and 4. The plots are for the time series CGRx of the protein shown in Figure 2.2. In the calculation, the parameter ε varies between 10% and 49%.

Fisher's discriminant algorithm works as follows. It first builds a classifier with a training set. A training set $H = \{x_1, x_2, \dots, x_n\}$ is given to contain training vectors from two classes. There are n_1 training vectors from one class forming a subset H_1 and n_2 training vectors from another class forming a subset H_2 . Hence, $n_1 + n_2 = n$ and $H = H_1 \cup H_2$. Assume that each x_i is a m -dimension vector. Then, a parameter vector $\mathbf{w} = (w_1, w_2, \dots, w_m)^T$ is estimated such that it allows as many training vectors as possible to be accurately predicted. Specifically,

$$\mathbf{m}_j = \frac{1}{n_j} \sum_{\mathbf{x}_i \in H_j} \mathbf{x}_i, \quad j = 1, 2, \quad (2.10)$$

$$\mathbf{S}_j = \sum_{\mathbf{x}_i \in H_j} (\mathbf{x}_i - \mathbf{m}_j)(\mathbf{x}_i - \mathbf{m}_j)^T, \quad j = 1, 2, \quad (2.11)$$

$$\mathbf{S}_w = \mathbf{S}_1 + \mathbf{S}_2, \quad (2.12)$$

and

$$\mathbf{w} = \mathbf{S}_w^{-1}(\mathbf{m}_1 - \mathbf{m}_2), \quad (2.13)$$

By Fisher's discriminant rule, $\mathbf{x}$ is hence assigned to the class of H_1 if $\text{dist} = (\mathbf{m}_1 - \mathbf{m}_2)^T \mathbf{S}_w^{-1}[\mathbf{x} - \frac{1}{2}(\mathbf{m}_1 + \mathbf{m}_2)] > 0$ and to the class of H_2 otherwise.

For protein structural class prediction, we proposed a *step-by-step* procedure to cluster protein structures [111, 125]. It works by discriminating proteins in a class from proteins in other classes using three steps. One predictor is trained in each step. After testing all the possible procedures (altogether 12 combinations), we find that the following procedure results in higher overall accuracy:

- Step 1: discriminate proteins in the β class from proteins in the $\{\alpha, \alpha + \beta, \alpha/\beta\}$ classes;
- Step 2: discriminate proteins in the α/β class from proteins in the $\{\alpha, \alpha + \beta\}$ classes;

- Step 3: discriminate proteins in the α class from the proteins in the $\alpha + \beta$ class and the remaining proteins are assigned to the $\alpha + \beta$ class.

2.2.8 Performance Evaluation

In statistical prediction, the following three cross-validation tests are often used to examine the power of a predictor: independent dataset test, sub-sampling (such as 5-fold or 10-fold sub-sampling) test, and jackknife test [35]. Of these three, the jackknife test can always yield a unique result for a given benchmark dataset [55, 128, 128–130]. Hence, in this chapter, we use the jackknife test to evaluate our method. In the jackknife test, each of the proteins in the dataset is in turn singled out as a tested protein, and the predictor is trained with the remaining proteins. Therefore, jackknife is also called *leave-one-out* test.

The prediction accuracies are then measured by the following formulae.

$$\text{Acc}_\alpha = \frac{a}{N_a}, \quad \text{Acc}_\beta = \frac{b}{N_b}, \quad \text{Acc}_{\alpha+\beta} = \frac{c}{N_c}, \quad \text{Acc}_{\alpha/\beta} = \frac{d}{N_d}, \quad (2.14)$$

and

$$\text{Acc}_{\text{overall}} = \frac{o}{N_o}. \quad (2.15)$$

In the above, a, b, c and d denote the number of correctly predicted proteins in $\alpha, \beta, \alpha + \beta$ and α / β classes, respectively. N_a, N_b, N_c and N_d denote the total number of proteins in $\alpha, \beta, \alpha + \beta$ and α / β classes, respectively. Furthermore, $o = a + b + c + d$ and $N_o = N_a + N_b + N_c + N_d$.

2.3 Results and Discussions

2.3.1 Accuracy of the Proposed Method

The prediction accuracies of the proposed method evaluated on the datasets *25PDB* and *1189* are given in Table 2.4. We can see the overall accuracy for the dataset *25PDB* is 1.6% lower than that for the dataset *1189*, which might be due to the fact that the sequence similarity of the former dataset is lower than that of the latter.

Table 2.4: Accuracies from AAD-CGR with step-by-step algorithm.

Dataset	α	β	$\alpha + \beta$	α/β	Overall
<i>1189</i>	60.5	67.7	61.4	71.0	65.8
<i>25PDB</i>	58.0	65.0	65.1	69.9	64.2

2.3.2 Comparisons with Other Methods

For the four-class prediction problem, the *one-against-all* algorithm [46, 131, 132] is widely used to transfer it into two two-class problems. Therefore, we calculate the prediction accuracies of our method based on the one-against-all algorithm as well, for a fair comparison with other methods. The accuracies of our method and other representative methods that do not make use of predicted secondary structure information are listed in Table 2.5. Comparison with those methods that utilize the predicted secondary structure information will be made in Chapter 3.

On the *1189* dataset, note that the method in [55] made use of the PSI-BLAST profile [103] and the predicted secondary structure content, which contains more information than the single amino acid sequence used by our method and other methods listed in the table. This may help explain the higher overall accuracy of this method over other methods.

Table 2.5: Comparison of our method with other methods. The accuracies of other methods are taken directly from the corresponding references. The best results are highlighted in bold.

Dataset	Reference	# Features	α	β	$\alpha + \beta$	α/β	Overall
<i>1189</i>	[37]	19	NA	NA	NA	NA	53.8
	[44]	66	57.0	62.9	25.3	64.6	53.9
	[45]	34	NA	NA	NA	NA	58.9
	[53]	27	48.9	59.5	26.6	81.7	56.9
	[133]	50	NA	NA	NA	NA	54.7
	This chapter	16	62.3	67.7	63.1	66.5	65.2
	[55]	50	75.8	75.2	31.8	82.6	67.6
<i>25PDB</i>	[44]	66	69.1	61.6	38.3	60.1	57.1
	[45]	34	NA	NA	NA	NA	59.9
	[49]	58	77.4	66.4	45.4	61.3	62.7
	[56]	56	76.5	67.3	45.8	66.8	64.0
	This chapter	16	64.3	65.0	61.7	65.0	64.0

Because the remaining methods in the table make use of the same kind of input information, the following comparison excludes the method in [55]. From the table, we can see that the overall prediction accuracy of our method is higher than others (from 6.3% to 11.4%). Although the accuracy in predicting α/β class proteins in [53] is higher than ours, our prediction accuracies are higher for all the other classes. Overall, the improvement made by our method ranges between 4.8% and 37.8%.

On the *25PDB* dataset, our method is comparable with the methods in [49, 56] and better than those in [44, 45]. Specifically, the overall prediction accuracy of our method is 4.1% and 6.9% higher than those in [45] and [44], respectively. Another remarkable feature is for the prediction of the $\alpha + \beta$ class, where our method makes more than 15% improvements over other methods compared. Unfortunately it comes with almost the same amount of accuracy decrease for predicting the α class compared to [49, 56]. In addition, our method used only 16 features, which is less than those used by other methods, suggesting the effectiveness of the proposed

method.

We look into the reasons why the proposed method achieves better predictions than others. It should be noted that most existing methods [37, 44, 45] are based on amino acid composition. As a result, the important information from sequence order is ignored in these methods. On the other method, for the methods based on PseAA [104, 134], sequence order information is partially incorporated. Experiments also demonstrated that the better prediction results can be obtained with PseAA [53] than with the simple amino acid composition [37, 44]. The impact of coupling effect among different amino acid components on the performance of protein structural class prediction were made in [135–137]. When it comes to the method proposed here, the protein sequence was converted into time series based on CGR, so that the order information would not be lost in this transform. By analyzing the resulting two time series with RQA, the prediction would take the sequence order information into consideration. Consequently, better prediction results are anticipated from the proposed method.

In order to predict protein structural class more accurately, Chou and Cai [42] proposed a method based on functional domain. In this method, each protein is presented by a 7785-dimensional vector. It was used to predict protein structural classification among 7 classes: α , β , α/β , $\alpha + \beta$, multi-domain, small protein, and peptide. A very high success rate (98%) was obtained on 2230 proteins in which none of protein has more than 20% pairwise sequence similarity to any others. This suggests that a feature representation containing more useful information of a protein is helpful in predicting protein structural class. Unfortunately, this method [42] did not test on the two datasets that we used here and, moreover, it is difficult to implement due to the excessive number of features used [57], which prevents from further comparisons with our method.

2.3.3 Effect of Reverse Encoding

Besides the reverse encoding method listed in Table 2.3, we test two other encoding methods to see whether they have any significant impact on the final prediction results. We compute the corresponding results for two randomly selected encoding methods (as seen in Table 2.6). The encoding method in Table 2.3 is denoted by En1, and those in Table 2.6 by En2 and En3, respectively. Based on the one-against-all procedure, the prediction results are listed in Table 2.7. From the table, the overall accuracy from En1 is the highest, which is consistent with the conclusion in [108] (i.e., the encoding method in Table 2.3 generally resulted in a better result).

As mentioned in [108], En1 is selected to “keep a balance in base composition so as to maximize the difference between the amino acid codes”. We simply measure the difference between a pair of amino acid codes by the number of mismatching bases. For example, the difference between the codes for A=GCT and for G=GGT in En1 is 1. The total difference between the amino acid codes in each setting is thus defined as the sum over the 190 ($=20 \times 19/2$) pairs of amino acid codes. We find that the total difference value in En1 (455) is higher than those in En2 (448) and En3 (442). This might explain to some extent why En1 has better result than En2 and En3.

Table 2.6: Two more reverse encoding methods for amino acids (denoted as En2 and En3, respectively).

A=GCA P=CCC	G=GGA V=GTA	M=ATG E=GAA	S=TCG K=AAA	C=TGT Q=CAA	H=CAT W=TGG	N=AAT F=TTT	T=ACA L=CTT	D=GAT R=CGC	I=ATA Y=TAT
A=GCC P=CCT	G=GGG V=GTC	M=ATG E=GAG	S=TCC K=AAG	C=TGT Q=CAG	H=CAT W=TGG	N=AAT F=TTC	T=ACC L=TTG	D=GAC R=CGT	I=ATC Y=TAT

Table 2.7: Prediction accuracies from En1, En2 and En3 based on the one-against-all procedure. The best results are highlighted in bold.

Dataset	Encoding	α	β	$\alpha + \beta$	α/β	Overall
<i>1189</i>	En1	62.3	67.7	63.1	66.5	65.2
	En2	46.6	54.1	59.0	65.9	57.2
	En3	53.1	57.5	51.5	63.5	56.7
<i>25PDB</i>	En1	64.3	65.0	61.7	65.0	64.0
	En2	49.2	58.2	58.3	72.5	58.8
	En3	52.6	59.6	57.4	69.4	59.2

2.3.4 Results of Other CGRs

We compare the results from AAD-CGR (En1) and the other four CGRs: 20-CGR, 12-CGR, 6-CGR, and 4-CGR in this subsection. A similar procedure introduced in Section 2.2.6 is used to determine the values of the parameters in RQA. The values of the parameters for each CGR are listed in Table 2.8. Based on the one-against-all procedure, the prediction accuracies with these methods are shown in Table 2.8, which indicates that the results vary with different CGRs. In terms of overall accuracy, the results from AAD-CGR are relatively better than other CGRs. Therefore, we recommend to choose AAD-CGR as the CGR of proteins to predict protein structural class.

2.4 Conclusions

In this chapter, we proposed a method to improve the protein structural class prediction by using chaos game representation (CGR) and recurrence quantification analysis (RQA) to extract features directly from the amino acid sequences. With the reverse encoding of amino acids in [108], AAD-CGR was introduced in this chapter. In order to analyze AAD-CGR of proteins more effectively, we decomposed

Table 2.8: Accuracies obtained with five different CGRs. The numbers in parentheses after each method are the values of the parameters (m, τ, ε) used in RQA. The best results are highlighted in bold.

Dataset	Method	α	β	$\alpha + \beta$	α/β	Overall
<i>1189</i>	4-CGR (6, 1, 25)	46.2	57.5	51.0	63.5	55.6
	6-CGR (7, 1, 25)	44.0	60.2	56.4	64.1	57.2
	12-CGR (7, 1, 25)	39.9	49.3	59.3	63.8	54.0
	20-CGR (7, 1, 30)	44.4	54.4	51.5	63.2	54.4
	AAD-CGR (8, 2, 30)	62.3	67.7	63.1	66.5	65.2
<i>25PDB</i>	4-CGR (6, 1, 25)	51.2	60.0	57.1	68.2	58.6
	6-CGR (7, 1, 25)	51.5	57.3	50.8	65.6	55.8
	12-CGR (7, 1, 25)	46.5	58.0	57.6	68.8	57.1
	20-CGR (7, 1, 30)	51.7	58.9	51.0	69.1	57.0
	AAD-CGR (8, 2, 30)	64.3	65.0	61.7	65.0	64.0

the AAD-CGR of proteins into two time series. RQA was then used to analyze the time series and 16 (2×8) features were extracted. To predict protein structural class with Fisher’s linear discriminant algorithm, we applied a step-by-step procedure to these features. The resulted overall accuracies are 65.8% and 64.2% for two low-similarity datasets *1189* and *25PDB* (sequence similarity being 40% and 25%, respectively), respectively. The other four different CGRs of proteins were also tested. The accuracies from AAD-CGR are comparatively higher, which suggests to better use AAD-CGR to predict protein structural class.

We compared our results with several existing methods. The prediction accuracy of the α class of the methods in [49, 55, 56] is higher than ours. Nevertheless, the overall accuracies of our method are higher than or comparable with those of the other methods (that use the same kind of input information), depending on different datasets tested. However, we can see that the accuracy is still far from satisfactory. Chapter 3 will present an approach that uses the predicted secondary structure to further improve the prediction accuracy.

Chapter 3

Improving Protein Structural Class Prediction with Predicted Secondary Structure [†]

3.1 Introduction

In this chapter, we will improve the prediction of protein structural class further by using predicted secondary structures. As we can see from Chapter 2, the accuracy of existing methods for protein structural class prediction are far from satisfactory for low-similarity datasets. Recently, Kurgan *et al.* [49, 51, 55, 57] proposed to extract features from the predicted secondary content and/or the predicted secondary structure rather than directly from the protein’s amino acid sequence, and reported that higher prediction accuracy can be consequently achieved. Inspired by this, we

[†] This chapter is mainly based on an article published in *BMC Bioinformatics* as: Jian-Yi Yang, Zhen-Ling Peng, and Xin Chen, Prediction of protein structural classes for low-homology sequences based on predicted secondary structure, *BMC Bioinformatics*, 11:S9, 2010.

intend to improve the protein structural class prediction further for low-similarity proteins by extracting novel features from predicted secondary structure.

We would like to introduce a new comprehensive feature set that was constructed from the predicted secondary structure. The method for extracting features are similar with that in Chapter 2. First, chaos game representation is used to transform the predicted secondary structure into two time series. Second, the time series are analyzed by recurrence quantification analysis. The difference from the method in Chapter 2 is the analysis in this chapter is for predicted secondary structures rather than the amino acid sequences. We demonstrate by experiments on three benchmark datasets that the prediction of protein structural class can be further improved for low-similarity proteins.

3.2 Materials and Methods

3.2.1 Datasets

The proposed method is tested on three low-similarity protein datasets that are widely used in the literature, and compared to a variety of existing methods [37, 44, 45, 49, 51, 53, 55, 60, 133, 138]. The first two datasets have been mentioned in Chapter 2 and they are referred to as *25PDB* and *1189*. The third protein dataset, referred to as *640*, was first studied in [55]. It contains 640 proteins of 25% sequence similarity, and freely available from the web server at <http://biomine.ece.ualberta.ca>. The number of proteins in this dataset in the α , β , α/β and $\alpha + \beta$ classes are 138, 154, 177, and 171, respectively. The distribution of proteins in the four classes for the first two datasets is listed in Table 2.2 of Chapter 2.

It is worth noting that protein sequences of the *25PDB* dataset are also provided at <http://biomine.ece.ualberta.ca/SCPRED/SCPRED.htm> by the study of [51]. However, some of them are different from those in our test dataset that were instead

downloaded from the PDB library, which would not allow for a fair performance comparison if one uses the prediction accuracy values given in the paper [51]. We looked into these sequence differences, and found that our test dataset is indeed the latest version in the PDB library. Therefore, we re-implemented the approach SCPRED by following the details presented in the paper [51] and tested on our test dataset. Experimental results showed that the prediction accuracies of SCPRED got further improved by 0.6% over those reported in [51].

3.2.2 Secondary Structure Prediction

Every amino acid in a protein sequence can be predicted into one of the three secondary structural elements, H (helix), E (strand), and C (coil). It is a problem known as protein secondary structure prediction, and many computational approaches have been developed in the past decades to predict the 3-state secondary structure from protein sequences. In this study we chose PSIPRED [6], which predicts protein secondary structure based on the position specific scoring matrices generated by PSI-BLAST [103] and was shown to outperform any other competing prediction methods [139]. For example, the protein 1E0G has a domain with the amino acid sequence DSITYRVRKGDSSLSSIAKRHGVNIKDVMRWNSDTANLQPGDKLTLFVK. If we submit this sequence to the PSIPRED 2.6 web server (<http://bioinf.cs.ucl.ac.uk/psipred/>), the predicted secondary structure to be returned will be CCEEEEECCCCCHHHHHHHHCCCHHHHHHHCCCCCCC-CCCCEEEEEEC.

3.2.3 Chaos Game Representation of Predicted Secondary Structure

To use the predicted secondary structural sequences to predict protein structural class, Kurgan and colleagues [49, 51] extract a set of features that contains the predicted secondary structure content, aiming to quantify the amount of residues in the amino acid sequence that assume helical and strand conformation. However, we can easily see that this approach did not make full use of the predicted sequence of protein secondary structure but disregard its high-level information such as sequence order. For example, if a predicted secondary structure sequence is randomly permuted, we could obtain a new secondary structure sequence belonging to a different structural class. However, these two sequences have the same content such that the aforementioned approach will erroneously predict them into the same protein structural class. Hence, a set of new features are needed to describe the predicted protein secondary structure in order to achieve the more accurate prediction of protein structure class. To this end, Kurgan and colleagues [55, 57] proposed a large set of features from the predicted secondary structure and/or the PSI-BLAST profile, which is followed by feature selection to select those class-related features. About 80% accuracy was obtained with these new methods.

We propose here a way to build a new set of features based on the CGR of secondary structure sequences. Noting that the CGR introduced in Chapter 2 is applied on the amino acid sequence rather than on the secondary structure sequence, we thus briefly introduce it below. Given a secondary structure sequence, we start with an equilateral triangle with the unit length of sides and each vertex associated with a distinct letter of H, E and C. For each letter of the given secondary structure sequence, we then plot a point inside the triangle in the following way. The first point is placed half way between the center of the triangle and the

vertex corresponding to the first letter of the secondary structure sequence, and the i -th point is then placed half way between the $(i - 1)$ -th point and the vertex corresponding to the i -th letter. The obtained plot is then called the CGR of the secondary structure sequence. Figure 3.1 depicts the CGRs for four proteins, each belonging to a different structural class. It is very interesting to see that for proteins in α and β classes, the plotted points tend to be distributed around the sides HC and EC, respectively. For proteins in $\alpha + \beta$ and α/β classes, however, the points lie around both sides HC and EC without preference.

Figure 3.1: The CGRs of predicted secondary structure for proteins from four structural classes. The blue edges represents the sides of equilateral triangles and the black points represent the CGR points. The order of the black points (corresponding to the order in the predicted secondary structure) is saved, but not shown in the figure. The PDB IDs for four proteins are 1A6M (α class), 1AJW (β class), 1GQOV (α/β class), and 1DEF ($\alpha + \beta$ class).

Observe that each secondary structure sequence gives rise to a distinct (x, y) -coordinate sequence of the plotted points. Hence we can faithfully model a CGR plot as a combination of two time series, one composed of the x -coordinates and the other of the y -coordinates. For simplicity, we call them x -time series and y -time series, respectively. Because a CGR plot can be fully reconstructed from the corresponding x and y time series, no information present in the CGR plot would be lost in the combination of two time series. For example, the eight time series corresponding to the four CGR plots of Figure 3.1 are depicted in Figure 3.2.

The average values of x - and y -time series points are calculated respectively as

$$\bar{x} = \frac{1}{L} \sum_{i=1}^L x_i \quad \text{and} \quad \bar{y} = \frac{1}{L} \sum_{i=1}^L y_i, \quad (3.1)$$

where L denotes the length of the time series and x_i and y_i are the coordinate values of the i -th point in CGR. As the average values for proteins in the α -class tend to be smaller than those for proteins in the other classes (see Figure 3.2), these two quantities will be used as the first two features in our proposed approach for protein structural class prediction.

3.2.4 Recurrence Plot and Recurrence Quantification Analysis

Similar to that done in Chapter 2, the *recurrence plots* of the time series obtained from CGR can be drawn after the parameters (m , τ , and ε) are fixed. While the values of m and τ have to be chosen appropriately based on nonlinear dynamical theory [115], τ is often set to be 1 in practical. Because an α -helix segment generally comprises at least three residues, we set m to be 3 in this study. We will discuss the selection of the ε value in Section 3.2.9. The RPs for the four x -time series (and y -time series) of Figure 3.2 are shown in Figure 3.3 (and Figure 3.4, respectively).

Figure 3.2: Eight time series that represent the four CGRs in Figure 3.1. Each panel in Figure 3.1 gives rise to two time series (x - and y -coordinates, respectively). As a result, we obtain eight time series for four CGRs.

Figure 3.3: The corresponding RPs for the four x -time series in Figure 3.2. The parameters used are $m = 3$, $\tau = 1$, and $\varepsilon = 20\%$. Note that there is a black line along the main diagonal in the plots since a point always recurs with itself. Moreover, the points in the RP are symmetric with respect to the main diagonal line.

Figure 3.4: The corresponding RPs for the four y -time series in Figure 3.2. The parameters used are $m = 3$, $\tau = 1$, and $\varepsilon = 20\%$. Some interesting patterns can be seen to emerge from the plots, but it is not so easy to characterize them. In this study we chose RQA.

RQA has been introduced in Chapter 2 to analyze the primary sequences of proteins. Here we use it instead to analyze the predicted secondary structure sequences. Compared to 20 states (i.e., bases) of amino acid sequence, the predicted secondary structure sequences have only three states (i.e., H, E and C). By applying RQA on each time series obtained from CGR in Section 3.2.3, we obtain eight recurrence variables to characterize a predicted secondary structure sequences. These eight recurrence variables are computed in a similar way as described in Chapter 2 and their definitions are thus omitted here to avoid repetition. These variables will be included into our set of features for the protein structural class prediction.

3.2.5 K -string Based Information Entropy

Given a predicted secondary structure sequence of length L , we call any substring $s_1s_2\cdots s_K$ of length K a K -string, where each s_i represents a letter in the set $\{H, E, C\}$. There are totally 3^K distinct K -strings for any K . We denote the occurrence frequency of the K -string $s_1s_2\cdots s_K$ by $f(s_1s_2\cdots s_K)$. The probability of this K -string occurring in the given predicted secondary structure sequence are then estimated as

$$p(s_1s_2\cdots s_K) = \frac{f(s_1s_2\cdots s_K)}{L - K + 1}. \quad (3.2)$$

When $K = 1$, $p(H)$, $p(E)$, and $p(C)$ are simply the corresponding probabilities of H, E and C occurring in the given predicted secondary structure sequence.

The K -th order information entropy I_K is hence calculated as

$$I_K = - \sum_s \sum_{s_1s_2\cdots s_{K-1}} p(s)p(s|s_1s_2\cdots s_{K-1}) \log_2(p(s|s_1s_2\cdots s_{K-1})), \quad K > 1,$$

where the first sum is over the set $\{H, E, C\}$ and the second one is over all possible $(K - 1)$ -strings. $p(s|s_1s_2\cdots s_{K-1})$ is the conditional probability of the letter s

occurring after the $(K - 1)$ -string $s_1 s_2 \cdots s_{K-1}$ in the given predicted secondary structure sequence. It can be calculated by Bayes' rule as

$$p(s|s_1 s_2 \cdots s_{K-1}) = \frac{f(s_1 s_2 \cdots s_{K-1} s)}{f(s_1 s_2 \cdots s_{K-1})}, \quad K > 1. \quad (3.3)$$

The two quantities, $p(H)$ and $p(E)$, are included in our feature set. They are equivalent to the predicted secondary structure content and have been shown to be helpful in improving the accuracies of protein structural class prediction [51]. $p(C)$ is not included as its value depends on $p(H)$ and $p(E)$ due to $p(H) + p(E) + p(C) = 1$. In addition, we include $I_2, I_3, \cdots, I_K$ in our feature set; however, the value of K remains to be determined. We will discuss this issue later in Section 3.2.9.

3.2.6 Segment-based Analysis

While proteins in the $\alpha + \beta$ class and α / β class contain both α helices and β strands, they differ in the distributions of α helices and β strands. The α helices and β strands are largely segregated in the $\alpha + \beta$ class, but instead largely interspersed in the α / β class. While the exact distributions can only be known from the spatial arrangement of secondary structure segments, it is still reasonable to expect that they could be more or less inferred from their secondary structure sequences. To this end, we construct below three features from the secondary structure sequences characterizing the distributions of α helices and β strands, and hope that they can be used to differentiate between the $\alpha + \beta$ class from α / β class.

As the first step of feature construction, we reduce a secondary structure sequence into a *segment* sequence, which is composed of *helix segments* and *strand segments* (denoted by α and β , respectively). Here, a helix segment refers to a continuous segment of all H symbols in the secondary structure sequence, and a similar definition is also applied to a strand or coil segment. Since at least three

and two residues are generally required to form an α helix segment and an β strand segment respectively, we will ignore those helix and strand segments that do not meet this size requirement. Moreover, in order to focus on the arrangement of α helix and β strand segments, the coil segments are ignored as well. For example, given a secondary structure sequence, CCEECCCHHCCHHHHEEEHHHHCCCC-CCECCEECCHHHCCEEEEEEEC, its reduced segment sequence is $\beta\alpha\beta\alpha\beta\alpha\beta$, in which the α helices and β strands are largely interspersed, suggesting that the corresponding protein more likely belongs to the α/β class rather than $\alpha + \beta$ class.

Let p_t denote the probability of transitions between α and β segments in a segment sequence, which is essentially the relative frequency of the substring $\alpha\beta$ or $\beta\alpha$ occurring in the segment sequence. Let p_{c_α} (respectively, p_{c_β}) denote the probability of two consecutive α (respectively, β) segments. Note that $p_t + p_{c_\alpha} + p_{c_\beta} = 1$; therefore, any probability can be deduced from the other two. In order to measure the degree of segment aggregation, we chose two of the above three probabilities to be included into our feature set. In our experimental study, p_t and p_{c_α} are used. The third feature represents the probability of helix (or strand) segments occurring in a segment sequence, denoted by $p(\alpha)$ (or $p(\beta)$). Clearly, $p(\alpha) + p(\beta) = 1$. $p(\beta)$ is used in this study.

3.2.7 Fisher's Discriminant Algorithm

As discussed above, we extract a set of $23+K-1$ features from the predicted secondary structure sequences. Therefore, each protein is represented by a $(23+K-1)$ -D vector. The value of K is chosen in order to maximize the overall prediction accuracy. For classification algorithms, we use the same algorithm as in Chapter 2, i.e., Fisher's discriminant algorithm [127], whose details have been presented in Chapter 2.

Fisher's discriminant algorithm is designed for a two-class problem, i.e., it pro-

vides a binary classifier. We could transform a four-class problem of protein structural class prediction into six two-class problems, namely, α -vs- β , α -vs- $(\alpha + \beta)$, α -vs- (α/β) , β -vs- $(\alpha + \beta)$, β -vs- (α/β) and $(\alpha + \beta)$ -vs- (α/β) . That is to say, six binary classifiers are trained and each classifier gives one vote for its preferred class. The final prediction will be the class with the most votes. Once a tie happens, we calculate a weight as the sum of the absolute value of *dist* over all votes in the tied cases and the final prediction is hence the class with the largest weight.

3.2.8 Prediction Assessment

We extract a set of $23+K-1$ features from the predicted secondary structure sequences. These feature vectors are fed into Fisher’s discriminant algorithm [127] for the prediction of protein structural class. The prediction accuracies are measured by the number of correctly predicted proteins divided by the number of proteins observed. They are computed by the same formulae as 2.14 and 2.15 in Chapter 2. The jackknife test is used to evaluate our method.

3.2.9 Selection of ε and K

As mentioned above, the value ε in RQA and the number K for K -string remain to be determined. Here, we choose their values by aiming to achieve the highest overall prediction accuracy as possible. For this purpose, a simple grid search strategy is adopted, where ε is allowed to take a value only between 1% to 50% and K only between 2 to 15. We use the dataset *25PDB* to compute the overall prediction accuracies (i.e., $\text{Acc}_{\text{overall}}$) for different combinations of ε and K based on the jackknife test. Once selected, the values of ε and K are kept the same for all tested sets. For example, when $K = 2$, $\text{Acc}_{\text{overall}}$ for different values of ε are shown in the left panel of Figure 3.5. When $\varepsilon = 39\%$, $\text{Acc}_{\text{overall}}$ for different values

of K are shown in the right panel of Figure 3.5.

Figure 3.5: The overall prediction accuracies of the dataset *25PDB* with varying values of ϵ and K . When $K = 2$, ϵ ranges from 1% to 50% (left panel). When $\epsilon = 39\%$, K ranges between 2 and 15 (right panel).

By the above grid search, we found that the highest accuracy (83.0%) is obtained with the combination of $\epsilon = 48\%$ and $K = 14$ (giving rise to 36 features) and the second highest accuracy (82.9%) is given by the combination of $\epsilon = 39\%$ and $K = 2$ (giving rise to 24 features), which is only 0.1% lower. We also notice that, to gain such a negligible accuracy improvement, the first combination requires a larger amount of computer time and memory to calculate feature vectors of higher dimension (i.e., 36-dimension v.s. 24-dimension). Based on this observation, we chose $\epsilon = 39\%$ and $K = 2$ in our experiments.

3.3 Results and Discussions

3.3.1 Prediction Accuracies for Three Benchmark Datasets

The proposed method is tested on three benchmark datasets in low similarity: *25PDB*, *640*, and *1189*. The resulting prediction accuracies are listed in Table 3.1. It can be seen that the overall accuracies for the three datasets are all above 80%.

To be specific, the overall accuracies of 82.9%, 83.1% and 81.3% are achieved for the datasets *25PDB*, *640* and *1189*, respectively. If comparing the four structural classes to each other, the predictions of proteins in the α classes are always the best (with accuracies about 90% for all the datasets). Satisfactory prediction accuracies (about 85%) are also achieved for proteins in the β and α/β classes. However, it seems very challenging to predict proteins in $\alpha + \beta$ classes as their prediction accuracies are relatively low (ranging between 65.6% and 71.4%) when compared with the other classes. It is still consistent with the previous observation [55] that the low prediction accuracy might be due to the non-negligible overlaps between $\alpha + \beta$ class and the other classes. A web server that implements the proposed method is freely available at http://www1.spms.ntu.edu.sg/~chenxin/RKS_PPSC/.

3.3.2 Comparison with Existing Methods

The proposed method were compared with some representative methods [37, 44, 45, 49, 51, 53, 55, 57, 60, 133, 138], and the results are listed in Table 3.1. Except for the method *SCPRED* [51], the listed accuracy values are taken directly from their references. Because some inconsistencies were found between our test dataset *25PDB* and the one used in [51] to test *SCPRED* (see Section 3.2.1), the direct comparison with the accuracy values reported in [51] would not be fair. Therefore, we re-implemented the *SCPRED* method following the details presented in the reference paper and applied it to our test dataset. It turns out that, the obtained accuracies (by jackknife test) for predicting proteins in the α , β , $\alpha + \beta$ and α/β classes are 93.7%, 81.3%, 71.7% and 73.1%, respectively, giving rise to the improvements of 1.1% 1.2% and 0.7% for the first three classes over those given in the reference paper. The overall prediction accuracy hence increases by 0.6% with our test dataset as well. These new accuracy values are listed in Table 3.1, and we used them as the performance measurements of the method *SCPRED* for

comparison.

Table 3.1: Prediction accuracies of our method for three datasets and comparison with other reported results. The accuracies are evaluated by jackknife test and measured by the percentage of correctly predicted proteins. The best results are highlighted in bold face.

Dataset	Reference	α	β	$\alpha + \beta$	α/β	Overall
<i>25PDB</i>	[44]	69.1	61.6	60.1	38.3	57.1
	[138]	60.6	60.7	44.3	67.9	58.6
	[45]	NA	NA	NA	NA	59.9
	[49]	77.4	66.4	45.4	61.3	62.7
	[56]	76.5	67.3	45.8	66.8	64.0
	Chapter 2	64.3	65.0	61.7	65.0	64.0
	[51]	93.7	81.3	71.7	73.1	80.3
	[57]	92.3	83.7	68.3	81.2	81.4
	This chapter	92.8	83.3	70.1	85.8	82.9
<i>640</i>	[55]	73.9	61.0	33.9	81.9	62.3
	[51]	90.6	81.8	66.7	85.9	80.8
	This chapter	89.1	85.1	71.4	88.1	83.1
<i>1189</i>	[37]	NA	NA	NA	NA	53.8
	[44]	57.0	62.9	25.3	64.6	53.9
	[133]	NA	NA	NA	NA	54.7
	[53]	48.9	59.5	26.6	81.7	56.9
	[45]	NA	NA	NA	NA	58.9
	Chapter 2	62.3	67.7	63.1	66.5	65.2
	[55]	75.8	75.2	31.8	82.6	67.6
	[51]	87.4	84.7	53.1	84.7	78.3
	[57]	92.3	87.1	65.4	87.9	83.5
	This chapter	89.2	86.7	65.6	82.6	81.3

Note that at the time of performing experiments in [61], the method MODAS [57] had not been published and thus originally we did not compare our method with MODAS in [61]. For the sake of completion, we also compare our method with it here. From Table 3.1, we can see that the proposed method performs best on the *1189* dataset and comparably well with the method MODAS on the *25PDB* dataset. In particular, the overall accuracy of our method is 1.5% higher

than MODAS on the *25PDB* dataset, while 2.2% lower than MODAS on the *1189* dataset. These suggest that two methods are complementary to each other and better prediction could be achieved by combining them. A possible reason why MODAS performs better than our method on the *25PDB* dataset is that MODAS made use of additional information, i.e., the evolutionary information obtained from PSI-BLAST [103].

We compare our method with *SCPRED* below in detail, as both methods used the same kind of information, i.e., predicted secondary structure. Compared to the overall accuracy values that were obtained with the method *SCPRED*, our method makes improvements of 2.6%, 2.3%, and 3% for the three tested datasets, respectively. We also notice that significant improvements were made in particular for the $\alpha + \beta$ class and the α/β class. For example, the proposed method obtained the 85.8% accuracy for predicting proteins of the α/β class from the dataset *25PDB*, which is 12.7% higher than that given by the method *SCPRED*. When the dataset *1189* is tested, the accuracy for predicting proteins of the $\alpha + \beta$ class is 12.5% higher than that given by the method *SCPRED*. Bear in mind that both *SCPRED* and our proposed method use features that are extracted from the secondary structure predicted with PSIPRED [6]. The prediction improvements hence clearly indicate that our features are more comprehensive and informative than those used by *SCPRED*.

3.3.3 Contribution of Features

To represent a protein, we used three different approaches to extract features from the predicted secondary structure sequences — recurrence quantification analysis, K -string based information entropy, and segment-based analysis. For brevity, let R , K and S denote the feature subsets generated by the corresponding approaches. Below, we investigate how these feature subsets contribute to the prediction results.

Table 3.3 lists the overall prediction accuracies that were obtained with all the possible combinations of feature subsets. For weighted random assignment, the overall prediction accuracy would be 25.3%, 25.6%, 25.2% for *25PDB*, *1189* and *640*, respectively [140]. It can be seen that when the feature subsets are used individually, the resulting overall prediction accuracies for three datasets are all well above 26%. It means that these predictions are not random at all. In other words, every feature subset makes its own positive contributions to the predictions. On the other hand, as more features are involved in the prediction, the overall accuracy values are shown to increase steadily (The only exceptional case is when the feature subset K is combined with the feature subset R , the accuracy value decreases slightly from 81.4% to 81.2%). For example, when tested on the dataset *640*, the prediction accuracy with the feature subset R is 80.5%. If the feature subset K is included, the accuracy value increases to 81.1%. If the feature subset S is further included, i.e., all the extracted features are used, the accuracy value increases by another 2.0% up to 83.1%. Therefore, we can conclude that these three feature subsets can make complementary contributions to each other to the predictions of protein structural class.

Table 3.2: Overall accuracies obtained with different combinations of feature subsets. See text for the notations of R , K and S . The best results are highlighted in bold face.

Dataset	R	K	S	$R + K$	$R + S$	$K + S$	$R + K + S$
<i>25PDB</i>	81.4	76.0	72.7	81.2	82.3	78.2	82.9
<i>640</i>	80.5	77.2	73.9	81.1	82.5	78.9	83.1
<i>1189</i>	79.6	75.8	73.0	80.3	81.0	79.7	81.3

3.3.4 Differentiating Between the $\alpha + \beta$ and α/β Classes

Because the segment-based features (i.e., the feature subset S) are aimed to differentiate between the $\alpha + \beta$ and α/β classes, it is very interesting to know how effective they are. To avoid any potential outside effects, we would like to make tests on (pure) datasets that comprise proteins only in the $\alpha + \beta$ and α/β classes. For this purpose, we generate a subset for each benchmark dataset by removing all the proteins in the α class or the β class, and then train the classifier (i.e., Fisher’s discriminant algorithm in our study) on these reduced subsets instead of the whole datasets.

Table 3.3: The accuracies of differentiating between the $\alpha + \beta$ and α/β classes. The datasets comprise only proteins in the $\alpha + \beta$ and α/β classes. The best results are highlighted in bold face.

Dataset	Ref. [51]			$R + K$			$R + K + S$		
	$\alpha + \beta$	α/β	Overall	$\alpha + \beta$	α/β	Overall	$\alpha + \beta$	α/β	Overall
<i>25PDB</i>	83.2	76.0	80.1	79.1	84.4	81.4	82.8	86.4	84.4
<i>640</i>	77.2	89.3	83.3	78.4	85.9	82.2	83.6	88.1	85.9
<i>1189</i>	63.1	88.6	77.9	76.8	83.2	80.5	81.3	83.8	82.8

Table 3.3 lists the prediction accuracy values obtained with the reduced subsets using different combinations of feature subsets. As we can see from the table, the combination $R + K$ provides the overall prediction accuracies that are only comparable to those given by the method *SCPRED*. In particular for the dataset *640*, it even gives a lower accuracy value (82.2% v.s. 83.3%). With the addition of the feature subset S , the overall prediction accuracies got improved by about 3.0%, and most importantly, all exceed those given by the method *SCPRED*. Specifically, there are the accuracy improvements of 4.3%, 2.6%, and 4.9% for the three tested datasets, respectively. These experiments further demonstrate that the segment-based features are very valuable for differentiating between the $\alpha + \beta$ and α/β

classes.

3.4 Conclusions

To predict structural class for low-similarity protein sequences (i.e., pairwise sequence similarity being between 20% and 40%), existing methods work very poorly, with only relatively low accuracies obtained. In this chapter, we aim to develop a new method so as to improve the prediction accuracy. To do so, we first use PSIPRED to predict the secondary structure sequence from a given amino acid sequence. Then, the chaos game representation (CGR) is employed to represent the predicted secondary structure as two time series, from which a comprehensive set of 24 features are generated by three different approaches — that is, the recurrence quantification analysis, K -string based information entropy, and segment-based analysis. The resulting feature vectors, each representing one protein, are fed into Fisher’s discriminant algorithm for the final prediction of protein structural class. Experimental results showed that all these features can make their own positive and complementary contributions so that higher prediction accuracies are achieved. For example, to predict proteins in the dataset *25PDB*, our method achieved the accuracies of 92.8%, 83.3%, 70.1% and 85.8% for the α , β , $\alpha + \beta$ and α/β classes, respectively, and the overall accuracy of 82.9%, which is 2.6% higher than that given by the method *SCPRED*. In addition, our method is shown to be comparable to the method MODAS, which made use of both the predicted secondary structure and the evolutionary information.

By comparisons with existing methods, we may attribute the high prediction accuracy of the proposed method to the superior performance of PSIPRED predicting secondary structures and the comprehensive set of features constructed. The first attribution can be seen from the comparison with the method proposed in

[49], which used the secondary structure prediction tool developed in [32] instead of PSIPRED. A previous study [139] showed that PSIPRED is superior to any other competing secondary structure prediction method. The second attribution can be seen from the comparison with the method *SCPRED*, which differs from our proposed method mainly in the selection of features. We used three different approaches to extract a comprehensive set of features from the predicted secondary structures, where the recurrence quantification analysis is used to capture the sequence order information of the time series, the K -string based information entropy to reflect certain short-term (local) interaction along the secondary structure, and the segment-based features to characterize the spacial arrangements of α helices and β strands. Thus, our proposed method may provide a promising tool for the accurate prediction of protein structural class, in particular for low-similarity amino acid sequences. A web server that implements the proposed method is freely available at http://www1.spms.ntu.edu.sg/~chenxin/RKS_PPSC/.

Chapter 4

Improving Taxonomy-based Protein Fold Recognition by Using Global and Local Features [†]

4.1 Introduction

Protein fold recognition from amino acid sequences is one of the fundamental problems in structural bioinformatics, as fold information could facilitate the identification of a protein's tertiary structure and function. In the last two decades, a substantial amount of research effort has been devoted to developing efficient and effective computational methods for protein fold recognition. These computational methods can be broadly classified into two categories, i.e., template-based [20, 21, 97, 141–144] and taxonomy-based [66, 68, 132, 145–148]. In recent years,

[†] This chapter is mainly based on an article published in *Proteins: Structure, Function, and Bioinformatics* as:

Jian-Yi Yang and Xin Chen, Improving taxonomy-based protein fold recognition by using global and local features, *Proteins: Structure, Function, and Bioinformatics*, 79:2053-2064, 2011.

the taxonomy-based method has attracted great attention due to its encouraging performance.

The taxonomy-based method for protein fold recognition was first proposed in 1995 by Dubchak *et al.* [145, 146]. It follows a popular belief that there are only a limited number of different protein folds in nature. Consequently, the problem of protein fold recognition can be viewed as a *classification* problem so that it can be tackled by using the methods in machine learning. Most implementations of the taxonomy-based method, if not all, have adopted the SCOP protein structural classification architecture [27], with which a query protein is classified into one of the known folds. To implement a classification task, two major procedures are generally required — feature extraction and a machine learning classifier. Below we briefly review the existing taxonomy-based methods from these two aspects.

Feature extraction refers to a procedure by which we extract features from a query amino acid sequence so as to represent the underlying protein as a fixed-length numerical vector. Dubchak *et al.* [145, 146] first proposed a way to extract features using *global description* of amino acid sequence. Since then, many new features have been developed to improve the recognition accuracy, such as those based on *pseudo-amino acid composition* [104, 147], structural properties of amino acid residues and amino acid residue pairs [149], *autocross-covariance transformation* [66], and hidden Markov model structural alphabet [150]. Besides the features extracted directly from amino acid sequences, some features are constructed through exploiting information such as predicted secondary structure [68], sequence evolution [66, 68], functional domain [151], and predicted solvent accessibility [149]. These features were reported capable of achieving satisfactory fold recognition accuracies, especially when they are utilized in combination [152, 153].

A machine learning classifier is basically an algorithmic procedure that assigns each fixed-numerical vector a pre-defined class label. For protein fold recognition, a

number of classifiers have been applied, such as Neural Networks (NNs) [132, 145, 146], Support Vector Machines (SVMs) [66, 132, 149, 153], probabilistic multi-class multi-kernel classifier [152] and various ensemble classifiers [68, 147, 148, 151, 154].

Note that feature extraction is a key step towards the success of classification. For protein fold recognition, however, it is not yet clear what features are the most discriminative. In this study, we look into this challenging problem and explore various ways to extract features from PSI-BLAST profiles [103] and also from PSIPRED profiles [6]. These two profiles are believed to contain rich information about protein sequence evolution and secondary structure, respectively. To depict their global and local characteristics, we extract 82 features from PSI-BLAST profiles and 55 features from PSIPRED profiles, resulting in a comprehensive set of 137 features. With these features we have developed a new taxonomy-based fold recognition method called TAXFOLD, which additionally uses an SVM-based classifier. As our experimental tests on seven datasets demonstrate, TAXFOLD can achieve an average 6.9% accuracy improvement over the best available taxonomic method, indicating that the proposed set of features has the enhanced power to discriminate between different folds.

4.2 Materials and Methods

4.2.1 Datasets

We used seven datasets in this study to evaluate our proposed method: DD, RDD, EDD, F95, F194, F710 and Lindahl. The first dataset was originally created by Ding and Dubchak[132], and later revised by Shen and Chou [147]. It comprises 311 protein domain sequences for the training purpose and 383 sequences for testing, each of which is classified into one of 27 folds. It was reported that none of the testing sequences shared more than 35% sequence sim-

ilarity to any of the training sequences [132]. However, Chen and Kurgan [68] later found seven duplicate pairs between the training and testing sequences. Our close inspection further revealed that 11 training sequences and 76 testing sequences were already updated in the latest release of SCOP database (release 1.75, June 2009, <http://astral.berkeley.edu/>). See details in Supporting Information A (available at <http://onlinelibrary.wiley.com/doi/10.1002/prot.23025/supinfo>). Moreover, there exists a domain (1BUCA1) which is no longer classified into any of the above 27 folds. Based on these observations, we updated the dataset accordingly, so that the latest domain sequences are experimented and the domain 1BUCA1 is excluded from further consideration. The resulting dataset is thus called the revised DD dataset (RDD), to distinguish from its original dataset (called DD).

The other four datasets (EDD, F95, F194, and F710) are constructed with the same procedure as done in [66], except that the latest release of SCOP (release 1.75, June 2009) is used. The domain sequences that have less than 40% pairwise similarity are first extracted from the Astral SCOP 1.75 release (<http://astral.berkeley.edu/>) and then those shorter than 31 residues are further removed, resulting in a total of 10493 domain sequences. Of these sequences, 3397 are classified into one of the above-mentioned 27 folds, so we use them to construct an extended DD dataset (EDD). In order to cover more folds, we construct another three datasets by selecting the folds that contain at least 26, 11, and 2 sequences, respectively. Consequently, three datasets comprises 6364, 8026, and 10011 sequences from 95, 194, and 710 folds, respectively. We call them F95, F194, and F710, respectively.

The above four datasets, RDD, EDD, F95 and F194, are available at <http://www1.spms.ntu.edu.sg/~chenxin/TAXFOLD/>. The dataset F710 is used to test the performance of the proposed method in real-world application. The last dataset is

Lindahl benchmark dataset [155], which can be downloaded from <http://www.bioinfo.se/protein-id/>. This dataset contains 976 proteins in three SCOP levels: family, superfamily and fold. The pair-wise sequence similarity in this dataset is smaller than 40%. This dataset is used to compare our method with template-based threading methods.

Note that, in this study, we make fold recognition for domain sequences rather than the whole protein sequences. For multiple-domain proteins, we will make fold recognition for each individual domain sequence separately.

4.2.2 Feature Extraction Methods

In this study, we extract features from both profiles of PSI-BLAST [103] and PSIPRED [6], where the rich sequence evolution information and secondary structure information are present. The features are carefully developed so that they can depict both *global* and *local* characteristics of profiles. Global characteristics refer to the patterns that the whole profile is held by, while local characteristics refer to the patterns particular to some profile fragment. Accordingly, the features thus extracted are called global and local features, respectively. We further divide the global features into two categories. If a global feature does not depend on sequence order, we call it a *globalA* feature (e.g., the first-order entropy of sequences); otherwise, a *globalB* feature. The whole procedure of feature extraction is depicted in Figures 4.1, 4.3 and 4.4.

4.2.2.1 PSI-BLAST Profile-based Features

The PSI-BLAST profile is represented as a so-called *position-specific score matrix* (PSSM), which is obtained through aligning a query amino acid sequence to the NCBI's Non Redundant (NR) database by using PSI-BLAST [103] with three iterations and a cutoff E-value of 0.001. The PSSM is a log-odds matrix of size $L \times 20$,

where L is the length of the query sequence and 20 is due to the 20 amino acids (see Figure 4.1 for an example of PSSM). Its elements are the log-odds ratios between the observed base frequencies and the background base frequencies, followed by scaling by 10 and rounding down to the nearest integer. Therefore, the positive (resp., negative) element values mean that the corresponding amino acids appear more (resp., less) often than expected from the background.

Figure 4.1: Illustration of the procedure to extract features from PSI-BLAST profiles. A PSSM is first transformed to a FM, from which a consensus sequence is then constructed. In order to extract local features, the FM is further divided into three fragments of equal length.

It should be noted that PSSMs have been considered in many taxonomy-based fold recognition methods. For example, Chen and Kurgan [68] extracted from PSSM a 20-D *profile-based composition vector* (PCV) in a way by which the negative elements of PSSM are first replaced by zero and then each column is averaged. Although replacing negative elements by zero can ensure that the elements of PCV are all nonnegative, it would definitely lose valuable evolutionary information that might be beneficial to fold recognition. In order to avoid this disadvantage, we propose an alternative way to extract features from PSSMs, as detailed below.

Our feature extraction method starts by transforming each element s_{ij} of the

PSSM into s'_{ij} using

$$s'_{ij} = 2^{0.1 \times s_{ij}}. \quad (4.1)$$

Note that this transformation is the inverse of the algorithmic operation that PSI-BLAST used to compute the PSSM log-odds ratios [103]. The resulting value s'_{ij} thus represents a ratio between the observed base frequency and the corresponding background based frequency, and is guaranteed to be nonnegative even when s_{ij} is negative. We further apply the normalization to the values s'_{ij} such that each row would sum to one. Let f_{ij} denote the normalized value of s'_{ij} . All the values f_{ij} form a matrix, which we called the frequency matrix (FM).

In order to extract globalA features (i.e., sequence order-free features), a so-called *consensus sequence* (CS) [156] is first constructed from the FM as follows:

$$\mu(i) = \arg \max \{f_{ij} : 1 \leq j \leq 20\}, \quad 1 \leq i \leq L \quad (4.2)$$

where “arg” stands for the argument of the maximum. The i -th base CS(i) of the consensus sequence is then set to be the $\mu(i)$ -th amino acid in the amino acid alphabet. It can be seen that a consensus sequence retains the most valuable evolutionary information from the PSSM. Then, we compute

$$AACCS(j) = \frac{n(j)}{L}, \quad 1 \leq j \leq 20 \quad (4.3)$$

where $n(j)$ is the number of the amino acid j occurring in the CS. It will give 20 features corresponding to the *amino acid composition* of the CS. Moreover, we also include the *entropy* into our feature set, that is,

$$ECS = - \sum_{j=1}^{20} AACCS(j) \ln AACCS(j) \quad (4.4)$$

where the base of the logarithm is Euler’s number e .

Note that the above features can be computed with the original protein sequences as well. Our experimental results in Section 4.3.2 show that the features extracted from CSs are more fold-specific discriminative than those from the original protein sequences. Another entropy-based feature is directly computed from FM to reflect the global characteristic of the PSSM.

$$EFM = -\frac{1}{L} \sum_{i=1}^L \sum_{j=1}^{20} f_{ij} \ln f_{ij} \quad (4.5)$$

In order to extract local features, we first divide FM into λ non-overlapping fragments of equal length (see Figure 4.1)¹. Then, for each fragment s , by applying a similar procedure in [151], the following 20 features are computed:

$$AOF_s(j) = \frac{1}{\text{len}_s} \sum_i f_{ij}, \quad 1 \leq s \leq \lambda, \quad 1 \leq j \leq 20 \quad (4.6)$$

where the summation is done over the fragment s and len_s is the length of the fragment s . $AOF_s(j)$ represents the *average occurrence frequency* of the amino acid j in the fragment s during the evolution process. The features $AOF_1(j)$ and $AOF_\lambda(j)$ in the first and last fragments may reflect the sequence characteristics at the N-terminus and C-terminus, respectively. How to determine the optimal value of λ will be discussed in Section 4.3.1.

In summary, for each query domain sequence, a total of $(22 + 20 \times \lambda)$ features are extracted from its PSI-BLAST profile, among which 22 are globalA features and $20 \times \lambda$ are local features.

¹The last fragment may be longer because L is not always divisible by λ .

4.2.2.2 PSIPRED Profile-based Features

The PSIPRED profile of a query protein contains the secondary structure information predicted with PSIPRED [6]. It comprises a *state* sequence and three *probability* sequences. The state sequence is a sequence of three possible symbols H, E and C, representing states of helix, strand and coil, respectively. The three probability sequences, each for one state, are the sequences of probability values with which the states occur along the query amino acid sequence. See Figure 4.2 for an example. We make use of all these sequences to extract features.

First, we extract several global features from a state sequence. These features are defined similarly as those in Chapter 3 and we recapture the basic steps in extracting them from the state sequence.

The first two features ($p(H)$ and $p(E)$ in Section 3.2.5 of Chapter 3) describe the helix and strand contents of the state sequence. In addition, the *entropy* of the state sequence and the *length* L of the query domain sequence are also included into our feature set because our experiments show that they can improve prediction accuracies. Note that these are four globalA features. Another three globalA features are extracted with segment-based analysis: p_t , p_{c_α} , and $p(\beta)$ (please refer to Section 3.2.6 of Chapter 3)).

In addition, we extract 18 globalB features using CGR and RQA. The procedure of extracting features from a state sequence based on CGR and RQA is illustrated in Figure 4.3. For the details about CGR and RQA and the features, please refer to Chapters 2 and 3.

Now, we show how to extract features from the probability sequences. Denote the three probability sequences by $ph = (ph_1, ph_2, \dots, ph_L)$, $pe = (pe_1, pe_2, \dots, pe_L)$ and $pc = (pc_1, pc_2, \dots, pc_L)$, respectively. Three globalA features are computed as

Figure 4.2: An example of PSIPRED profile.

Figure 4.3: Illustration of the procedure to extract features from the state sequences of PSIPRED profiles by applying CGR and RQA.

follows:

$$\overline{ph} = \frac{1}{L} \sum_{i=1}^L ph_i, \quad \overline{pe} = \frac{1}{L} \sum_{i=1}^L pe_i, \quad \overline{pc} = \frac{1}{L} \sum_{i=1}^L pc_i. \quad (4.7)$$

To depict the sequence order information present in the probability sequences, we apply autocovariance (AC) transformation on the probability sequences. AC is the covariance of the sequence against a time-shifted version of itself. That is, for the sequence $t = (t_1, t_2, \dots, t_L)$ of length L , the AC transformation will return

$$AC_{l,t} = \frac{1}{L-l} \sum_{i=1}^{L-l} (t_i - \bar{t})(t_{i+l} - \bar{t}), \quad l = 1, 2, \dots, l_{\max} \quad (4.8)$$

where $\bar{t}$ is the average over all t_i , l the lag (distance) between two positions along the sequence and $l_{\max}$ the maximum of l . All these $AC_{l,t}$ will be used as features. For the three probability sequences, we thus obtain $3 \times l_{\max}$ globalB features. The selection of the optimal value of $l_{\max}$ will be discussed in Section 4.3.1.

Note that AC transformation was previously applied to PSSM for fold recognition [66], resulting a huge number of features, which makes it difficult to train SVM-based classifiers. We have tried to incorporate these features into our feature set but the results were not improved and thus we do not use these features in this

study.

In summary, a total of $(28 + 3 \times l_{\max})$ features have been extracted from a PSIPRED profile. Among these, 4 globalA features and 21 globalB features are extracted from the state sequence, while 3 globalA features and $3 \times l_{\max}$ globalB features are from the probability sequences. Combining these features with those extracted from PSI-BLAST, we obtain $(50 + 20 \times \lambda + 3 \times l_{\max})$ features in total, which will be fed into a SVM-based classifier to perform protein fold recognition.

4.2.3 Support Vector Machines

SVMs are one of the state-of-the-art machine learning algorithms for binary classification introduced by Vapnik [105]. SVMs have been extensively applied in various fields and theoretical descriptions about SVMs abound in the literature. Protein fold recognition is a multi-class classification problem, which can be converted into binary classification problems by using either *one-against-one* or *one-against-all* strategy. For the implementation of SVMs, we use the LIBSVM package [157] with the one-against-one classification strategy.

There are four basic kernel functions commonly used by SVMs; that is, linear, polynomial, radial basis function (RBF) and sigmoid. Here, we choose the RBF kernel because it produces higher prediction accuracy than other kernel functions (see e.g., [51, 66, 68]). It is formally defined as

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2} \quad (4.9)$$

where γ is a kernel parameter, $\mathbf{x}_i$ and $\mathbf{x}_j$ are the feature vectors of proteins i and j , respectively. The values of γ and the cost parameter C (used to control the trade-off between allowing training errors and forcing rigid margins) are optimized as follows.

In our experiments, grid search is used to select the optimal values of C and γ . The grid is set to be $C = [2^0, 2^1, \dots, 2^{10}]$ and $\gamma = [2^{-1}, 2^{-2}, \dots, 2^{-10}]$. For the original DD dataset and the RDD datasets, the C and γ in SVM are optimized based on standard 5-fold cross-validation on the training datasets. Then the independent testing datasets are used to evaluate the accuracies of TAXFOLD. For the EDD, F95 and F194 datasets, the following adjusted n -fold cross-validation is adopted to assess TAXFOLD. First, a dataset (e.g., EDD) is randomly partitioned into n subsets of equal size. Second, 80% of $(n - 1)$ subsets are used to train SVMs, the remaining 20% of the $(n - 1)$ subsets are used to find the optimal C and γ , and the remaining subset (called *validation set*) is used to evaluate the prediction accuracies. The second step is repeated n times with each of the n subsets used exactly once as the validation set.

We want to mention that the above test procedure has some limitations. Because it uses different parameters in each of the n runs, this test is not the conventional cross-validation but n independent tests. It may not provide the best model (i.e., a fixed set of SVM parameter values that works the best for all the datasets as a whole) due to the different parameter values used for different tests on different datasets. These are the limitations of the performed tests that we shall emphasize in this chapter.

4.2.4 The Proposed Method

Figure 4.4 illustrates the overall architecture of our proposed method called TAXFOLD. The query amino acid sequence submitted to TAXFOLD is first input into PSI-BLAST and PSIPRED to obtain the sequence evolution information and predicted secondary structure. Then, a comprehensive set of features are extracted from the output profiles of PSI-BLAST and PSIPRED, as described above. These features are finally fed into an SVM-based classifier for fold recognition. Com-

Figure 4.4: The overall architecture of TAXFOLD. The proposed set of features are enclosed inside a red box. The detailed procedures of ① and ② in the figure are already illustrated in Figure 4.1 and Figure 4.3, respectively.

pared to the existing taxonomy-based methods such as PFRES [68], TAXFOLD differs mainly by a novel set of features which are so carefully designed that they are capable of capturing both global and local characteristics of PSI-BLAST and PSIPRED profiles. For the experiments in this study, we train SVM-based classifiers in TAXFOLD for three different numbers (27, 95, and 194) of folds.

For public use, we have developed a web server for TAXFOLD at <http://www1.spms.ntu.edu.sg/~chenxin/TAXFOLD/>. The top five predictions and their corresponding probability estimates are provided for users to determine the reliability of predictions.

4.2.5 Performance Evaluation

Two metrics, *precision* and *recall*, are used to evaluate the performance of TAXFOLD on each individual fold. They are defined respectively as

$$p(i) = \frac{TP(i)}{TP(i) + FP(i)}, \quad i = 1, 2, \dots, \mu \quad (4.10)$$

and

$$r(i) = \frac{TP(i)}{TP(i) + FN(i)}, \quad i = 1, 2, \dots, \mu \quad (4.11)$$

where $TP(i)$, $FP(i)$ and $FN(i)$ represent the numbers of true positives, false positives and false negatives, respectively, and μ the total number of folds under consideration.

Note that the using only the above two metrics might not be able to reflect the performance of a classifier robustly. For example, if a predictor simply predicts any protein to be the same fold type, then the recall value for this fold will be 1 and 0 for other folds. Therefore, a more robust metric called *F-measure* is often used,

which is basically a harmonic mean of precision and recall as define below.

$$Fm(i) = \frac{2 \times p(i) \times r(i)}{p(i) + r(i)}, \quad i = 1, 2, \dots, \mu \quad (4.12)$$

The overall accuracy Q of a classifier is defined as the ratio of correctly predicted instances against all the tested instances [66, 132, 149]. It can be calculated as

$$Q = \frac{\sum_i^{\mu} TP(i)}{\sum_i^{\mu} [TP(i) + FN(i)]}. \quad (4.13)$$

It will be used to evaluate the overall performance of TAXFOLD.

4.3 Results and Discussions

4.3.1 Optimal Values of λ and $l_{\max}$

As mentioned earlier, the values of λ and $l_{\max}$ remain to be determined. In this study, we choose their values by aiming to achieve the overall prediction accuracy as high as possible. To this end, we run TAXFOLD on the training sequences of the dataset RDD and compute the overall accuracies Q s with varying values of λ and $l_{\max}$ based on 5-fold cross-validation. Figure 4.5 depicts all Q s obtained when λ ranges from 1 to 5 and $l_{\max}$ from 1 to 15. The highest accuracy of 77.2% is thus achieved when $\lambda = 3$ and $l_{\max} = 9$. For the sake of simplicity and generality, we set the default values of λ and $l_{\max}$ in TAXFOLD to be 3 and 9, respectively, and all the rest experiments are carried out with these default values. This setting hence gives rise to a total of 137 ($= 50 + 20 \times 3 + 3 \times 9$) features among which 29, 48 and 60 are globalA, globalB and local features, respectively (see Figure 4.4).

Figure 4.5: The overall accuracies for the RDD training dataset obtained by varying the values of λ and $l_{\max}$ from 1 to 5 and 1 to 15, respectively

4.3.2 Analysis of Feature Contribution

In order to investigate the contributions of features to the overall prediction accuracy, we divide the 137 features into three subsets: S1 (contains 29 globalA features), S2 (contains 48 globalB features), and S3 (contains the remaining 60 local features). In addition, to assess the contributions of predicted secondary structure and evolutionary information, we separate features generated from the PSIPRED profile and PSI-BLAST profile and denote them as PSIPRED and PSI-BLAST, respectively. The feature of protein length can be incorporated into both subsets.

Table 4.1: Overall accuracies (%) on four datasets obtained with different combinations of feature subsets. For the dataset RDD, the overall accuracies are obtained from the independent testing sequences. For the other datasets, the overall accuracies are obtained instead by applying 10-fold cross-validation. Enclosed in the parentheses are the number of features of the corresponding feature subset. The accuracies from feature subsets PSIPRED and PSI-BLAST are obtained with/without the feature of protein length.

Dataset	S1 (29)	S2 (48)	S3 (60)	PSIPRED (54/55)	PSI-BLAST (82/83)	S1+S2 (77)	S1+S3 (89)	S2+S3 (108)	S1+S2+S3 (137)
RDD	72.8	66.8	72.3	70.4/71.2	70.9/74.9	79.1	78.3	78.8	83.2
EDD	76	77.5	80	78.3/78.8	81.7/82.5	86.5	86.3	88.9	90
F95	62.2	61.2	72.3	62.2/63.1	74.1/75.5	75.3	78.3	81.5	82.4
F194	58.6	57	70.5	57.7/58.8	72.2/73	71.3	76.2	78.8	79.6

Table 4.1 lists the overall prediction accuracies obtained with all the possible combinations of feature subsets S1, S2, and S3. It can be seen that when the feature subsets are used individually, the resulting overall prediction accuracies range from 57.7% to 82.5% for the four tested datasets. The feature subsets S1 and S2 perform comparatively well for all datasets, while the feature subset S3 performs the best for three datasets (EDD, F95 and F194). As more features are used, the overall accuracy values increase steadily. For instance, the combination of the feature subsets S2 and S3 achieves the accuracies of 78.8%, 88.9%, 81.5%, and 78.8% for the four tested datasets, making at least 6.5% improvement over those obtained with any individual feature subset S2 or S3. If all these feature

subsets are used together, the accuracy values increase to the maximum possible for all datasets (83.2%, 90%, 82.4%, and 79.6%, respectively). Therefore, these three feature subsets could make complementary contributions to each other for improving protein fold recognition.

From Table 4.1, we can see that the PSI-BLAST features seem to make higher contributions than the PSIPRED features for all tested datasets. This is especially obvious for the F95 and F194 datasets, where the PSI-BLAST features achieve more than 10% improvement compared with the PSIPRED features. However, just using the PSI-BLAST features can not obtain the best prediction accuracy. As can be seen from Table 4.1, when the PSIPRED features are combined with the PSI-BLAST features, the accuracies are improved by 8.3%, 6.5%, 6.9%, and 6.6% for the four tested datasets, respectively. This suggests that the PSI-BLAST features and PSIPRED features make complementary contributions to each other and it is important to use both kinds of features for enhanced protein fold recognition. In addition, it is interesting to see that the feature of protein length improves the accuracy by about 1%, suggesting the necessariness of using this feature.

In order to demonstrate the advantage of features extracted from the consensus sequences rather than directly from the corresponding amino acid sequences (please refer to Section 4.2.2.1), we compared their respective prediction accuracies. When the 21 globalA features are extracted from the original amino acid sequences (see 4.3 and 4.4), the overall accuracies are only 51.3%, 42.9%, 27.5%, and 22% for the four tested datasets. When these features are instead extracted from the consensus sequences, the prediction accuracies increase to 63.1%, 62.7%, 47.8%, and 43.7%, respectively, making an average 18.4% improvement. These indicate that the consensus sequences contain much more evolution information than the amino acid sequences, thereby leading to more accurate protein fold recognition.

4.3.3 Accuracies for Four Tested Datasets

TAXFOLD is evaluated on four datasets, RDD, EDD, F95 and F194, and the resulting overall accuracies are listed in the last column of Table 4.1. The precision, recall and F-measures of individual folds and the corresponding values of the parameters C and γ in SVMs are presented in Supporting Information C (available at <http://onlinelibrary.wiley.com/doi/10.1002/prot.23025/supinfo>).

We can see that the overall accuracies for the first three datasets are all above 80% (i.e., 83.2%, 90% and 82.4%, respectively). It is commonly believed that, the larger number of training sequences, the more reliable a classifier to be trained. Although both RDD and EDD are made of sequences from the same 27 folds, the latter contains more sequences than the former. Therefore, it is not surprising to see that the overall accuracy for EDD is higher than that for the dataset RDD. Moreover, as revealed in [66], the prediction becomes very challenging as more folds are considered. This can also be observed from Table 4.1, where the accuracy value decreases from 90% to 82.4% and further to 79.6% when the number of folds increase from 27 to 95 and 194, respectively.

We also performed histogram analysis on the F-measure values over all the folds. We count the number of folds that have F-measure values in one of the following five intervals: $[0, 0.6)$, $[0.6, 0.7)$, $[0.7, 0.8)$, $[0.8, 0.9)$, $[0.9, 1]$. The resulting histograms are shown in Figure 4.6. Note that, among the 27 folds used for the dataset RDD, 22 obtained the F-measures over 70%. When more sequences of these 27 folds are experimented in the dataset EDD, the F-measures become over 70% for all the folds. Indeed, all the folds achieve a F-measure value over 80%, except for the fold *OB-fold*. These F-measure results further indicate that TAXFOLD can achieve superior performance for this 27-class fold recognition problem.

TAXFOLD can still achieve satisfactory prediction for 95 and 194 folds, al-

Figure 4.6: Histogram of the F-measure values for four tested datasets.

though the F-measures get relatively lower values. For the five folds in the dataset F95, their F-measures are all below 60%. For the dataset F194, there are 21 folds with F-measures less than 60%. Among these 21 folds, 11 have no more than 20 sequences. This might partially explain their low F-measures because it is generally very hard to train a reliable classifier on a small number of instances. It is thus anticipated that the performance of TAXFOLD on a large number of folds would be improved as more amino acid sequences are accumulated.

4.3.4 Fold Recognition for 710 Folds

The current version (1.75) of SCOP database has 1195 folds, which is significantly more than those (27, 94, and 194 folds) we discussed. This fact makes it difficult for real-world use of TAXFOLD. In order to make prediction for as many folds as possible, we further test the performance of TAXFOLD with the dataset F710. By using this large dataset, the likelihood of a newly sequenced protein to be the prediction target of TAXFOLD should be about 60% (710/1195).

With the dataset F710, we are able to perform fold recognition for 710 folds, which is an approximation of real-world situation. We assess the prediction accuracy of TAXFOLD on this dataset based on 2-fold cross-validation. The overall accuracy of 68.1% is achieved and the distribution of F-measures for individual folds is shown in Figure 4.7. There are 417 folds that can be predicted with F-measures higher than 60%. On the other hand, the F-measures for 185 folds are lower than 10%. Among these folds, there are 183 folds containing less than 10 sequences, which is a major reason for the low accuracies. In general, it is hard for taxonomic methods to make accurate predictions for folds that contain only small number of sequences. It is anticipated that when more sequences are accumulated in the SCOP database, we are able to improve the prediction and at the same time make predictions for more folds.

Figure 4.7: Histogram of the F-measure values for the dataset F710.

4.3.5 Comparisons with Taxonomy-based Methods

In order to demonstrate the effectiveness of the proposed method, we compare TAXFOLD with several major existing taxonomy-based methods on four datasets.

The original DD dataset has been widely used to evaluate various methods. For a fair comparison, we tested TAXFOLD on both the original DD dataset and its revised version RDD. Both the accuracies that were reported in the literature and those obtained from our experiments are listed in Table 4.2. It should be noticed that the reported accuracies of ACCFold_AC [66], ACCFold_ACC [66] and Shamim [149] were measured by applying 2-fold cross-validation on the training sequences of the original DD dataset, while the reported accuracies of the other methods were obtained with the independent testing sequences. So, we ran ACCFold_ACC on the testing sequences of the original DD dataset², and obtained its prediction accuracy (66.6%) which turns out to be slightly lower than the accuracy (70.1%) measured via 2-fold cross-validation. Besides, we ran four major existing methods, PFRES [68], ACCFold_AC [66], ACCFold_ACC [66] and Shamim [149] on the revised DD dataset RDD, and collected their overall prediction accuracies in Table 4.2. Their corresponding precision, recall and F-measures can be found in Supporting Information C. It is evident from Table 4.2 that, when tested on the original DD dataset, TAXFOLD achieves an accuracy of 71.5%, which is at least 1% higher than that of any other method. When tested on the revised DD dataset RDD, all the tested methods had their accuracies increased. However, TAXFOLD still achieves the highest overall accuracy of 83.2%, which is 3.1-17% higher than those of the other tested methods.

We further tested the above methods on the datasets EDD, F95, and F194. However, we found that the running of ACCFold_ACC and PFRES on these large-

²The PSSMs were acquired from the authors.

Table 4.2: Comparisons of prediction accuracies (%) obtained from the original DD dataset and its revised dataset RDD. Note that the prediction accuracies for the original DD dataset are taken directly from their corresponding references, except for the methods ACCFold_AC and ACCFold_ACC.

Methods	References	Q^a	Q^b
D-D	[132]	56	NA
PFP-Pred	[147]	62.1	NA
GAOEC	[148]	64.7	NA
Multi-kernel	[152]	68.1	NA
PFP-FunDSeqE	[151]	70.5	NA
SWPSSM ^c	[158]	67.8	NA
Shamim	[149]	60.5	66.2 ^d
ACCFold_AC	[66]	65.3	73.6 ^e
ACCFold_ACC	[66]	66.6	73.8 ^f
PFRES	[68]	68.4	80.1
TAXFOLD	This chapter	71.5	83.2

^a Overall accuracies for the original DD dataset.

^b Overall accuracies for the revised dataset RDD.

^c Cited from [66].

^d The accuracy obtained with the feature set Feature4.

^e The accuracy obtained with the parameter LG being 8.

^f The accuracy obtained with the parameter LG being 10.

sized datasets with 10-fold cross-validation seems no way to complete in a reasonable amount of time (executed on a workstation with 8 CPUs of 2.8GHz each and 24GB RAM). Therefore, to make the comparisons possible, we adopt 2-fold cross-validation rather than 10-fold cross-validation for them. Even with 2-fold cross-validation, it is still too challenging to complete the test of PFRES on the largest dataset F194. Note that PFRES relies on an ensemble classifier which consists of six individual classifiers (SVMs, multiple logistic regression, instance learning-based Kstar, IB1 algorithms, Navie Bayes, and random forest), and it was reported that this ensemble classifier made only a slight accuracy improvement over the individual classifier random forest (68.4% vs. 66.8%) when tested on the original DD dataset [68]. Based on these observations, to run PFRES on the dataset F194, we activate the classifier random forest only. Although the resulting accuracy might be biased against PFRES, we include it for the sake of a complete

comparison. On the other hand, for the method Shamim, the feature set Feature4 as described in [149] can yield the best prediction on the original DD dataset. So, this feature set is applied to the other three datasets as well. Table 4.3 lists the overall accuracies obtained with these methods. The corresponding precision, recall and F-measures are presented in Supporting Information C.

Table 4.3: Comparisons of overall accuracies (%) on three large-sized datasets. These accuracies are obtained via 2-fold cross-validation.

Datasets	Shamim	ACCFold_AC	ACCFold_ACC	PFRES	TAXFOLD
EDD	61	73.9	77.3	81.1	86.9
F95	41.6	62.5	71.8	68	76.5
F194	35.4	58.6	68.8	60 ^a	72.6

^a The accuracy obtained with the classifier random forest alone.

In general, 10-fold cross-validation shall yield a higher accuracy value than 2-fold cross-validation because more sequences are used to train a classifier in the former case. It is true for our method TAXFOLD as we can see from Tables 4.1 and 4.3, and so is for the methods Shamim, ACCFold_AC, ACCFold_ACC, as the accuracy values obtained with 2-fold cross-validation are lower than those reported in the literature which were instead obtained with 5-fold cross-validation.

It is evident from Table 4.3 that TAXFOLD outperforms any other method for all the tested datasets. Shamim performs the worst among these methods, which we believe is due to the fact that it does not exploit PSI-BLAST profiles. Therefore, we conclude that the sequence evolution information is a key factor for successful protein fold recognition. PFRES performs better than ACCFold_ACC for the dataset EDD but worse for two datasets F95 and F194. It might be explained in part by the fact that PFRES was developed (trained) particularly for the recognition of 27 folds. Compared with other methods, TAXFOLD achieves 5.8-25.9%, 4.7-34.9% and 3.8-37.2% improvements for the datasets EDD, F95 and F194, respectively. We believe that the superior performance of TAXFOLD shall

be attributed to a comprehensive set of features developed, which is capable of capturing both global and local characteristics of PSI-BLAST and PSIPRED profiles of a query protein domain.

Further comparisons are made on F-measures of the individual folds. We count the number of folds for which TAXFOLD has a lower (resp., equal or higher) F-measure value than any other methods. It implies that, for the folds to be counted, TAXFOLD performs worse (resp., comparatively or better) than the method under comparison. The detailed tally is shown in Table 4.4, where we can see that TAXFOLD performs better for a majority of folds in all cases.

Moreover, statistical test is applied to the F-measure values of TAXFOLD and other methods to assess their statistical significance. The average F-measure values on the corresponding test folds are tested as follows. Shapiro-Wilk test is first used to determine if the samples are normally distributed (at 0.05 significance level). If they follow normal distribution, the paired t -test is then used for statistical test on the differences between TAXFOLD and other methods; otherwise, the non-parametric Wilcoxon rank sum test is used. The resulting p -values are also reported in Table 4.4. We can see that the p -values are smaller than 0.05 for all the tested datasets and methods except the case of ACCFold_Acc on the F194 dataset, indicating that TAXFOLD has made statistically significant improvements over any other method for fold recognition.

Table 4.4: Comparisons with other methods on the four benchmark datasets. RDD represents revised DD dataset. The columns marked by W, E, and B mean respectively the number of folds that TAXFOLD has worse, equal, and better performance compared with the corresponding method. The p -values are used to measure the statistical significance.

Methods	RDD				EDD				F95				F194			
	W	E	B	p -values	W	E	B	p -values	W	E	B	p -values	W	E	B	p -values
Shamim	0	1	26	3.75e-9	0	0	27	1.42e-10	0	0	95	5.41e-30	0	3	191	1.21e-48
ACCFold_AC	2	3	22	4.36e-5	0	0	27	1.01e-8	3	0	92	8.32e-15	25	13	156	2.38e-7
ACCFold_ACC	5	4	18	0.003	0	0	27	7.41e-4	11	1	83	7.36e-6	79	21	94	0.541
PFRES	6	6	15	0.013	3	1	23	2.12e-5	2	0	93	9.99e-13	15	7	172	9.98e-12

4.3.6 Comparisons with Template-based Methods

TAXFOLD is further compared with the conventional template-based threading methods: RAPTOR [141], HHPred [142], FOLDPro [97], SP5 [143], DescFold [144], and BoostThreader [159]. To this end, the Lindahl benchmark dataset [155] is used and 2-fold cross-validation is adopted to assess the accuracy. As done in [66], before performing predictions, the dataset is preprocessed so that the number of proteins in each category is larger than or equal to a threshold N_{min} . For details about this preprocessing, please refer to [66].

The accuracies of TAXFOLD, ACCFold_AC, ACCFold_ACC, and eight template-based methods are listed in Table 4.5. First, we can see that TAXFOLD performs consistently better than ACCFold_AC and ACCFold_ACC at each SCOP level. Second, for the family and superfamily levels, when the number of samples in each category is very small, the taxonomic methods including TAXFOLD perform worse than the template-based threading methods. This is not surprising because a few samples in general do not allow us to train an accurate classifier. When the number of samples increases, the performance of taxonomic methods is improved, which can be seen from the accuracy increase from Subset0 to Subset1 and Subset2. Nevertheless, at the fold level, TAXFOLD achieved an accuracy comparable to the best template-based threading method (BoostThreader; for Subset0), still showing that TAXFOLD is very promising for protein fold recognition.

Table 4.5: Comparisons with template-based methods on the Lindahl benchmark dataset. Note that the only the top 1 accuracies (%) are reported for all methods.

	Family			Superfamily			Fold		
	Subset0	Subset1	Subset2	Subset0	Subset1	Subset2	Subset0	Subset1	Subset2
	$N_{min} = 1$	$N_{min} = 3$	$N_{min} = 4$	$N_{min} = 1$	$N_{min} = 3$	$N_{min} = 5$	$N_{min} = 1$	$N_{min} = 3$	$N_{min} = 5$
No. of sequences	555	97	47	434	225 ^a	91 ^a	321	239 ^a	177
No. of categories	176	13	5	86	23 ^a	6 ^a	38	16 ^a	8
TAXFOLD	68.6	90.7	100	39.3	61.7	84.5	40.6	56.9	67.7
ACCFold_ACC ^b	53.9	79.6	95.7	23.1	55.4	78.3	29.9	41.4	51.9
ACCFold_AC ^b	53.1	79.5	93.6	20	47.7	64	28	41.3	50.9
RAPTOR ^b	86.6			56.3			38.2		
Fugue ^c	82.2			41.9			12.5		
HHPred ^c	82.9			58.8			25.2		
SPARKS ^c	81.6			52.5			24.3		
SP5 ^c	82.4			59.8			37.9		
FOLDPro ^c	85			55.5			26.5		
DescFold_I ^c	80.7			57.8			24.9		
DescFold_II ^c	81.1			60.6			32.4		
BoostThreader ^d	86.5			66.1			42.6		

^a Slightly different from those in [66] because of random partition of the dataset into two subsets for cross-validation. In our partition, we tried to make the number of samples in each subset of the same category as even as possible.

^b The results were cited from [66].

^c The results were cited from [144].

^d The results were cited from [159].

4.4 Conclusions

In this study, we have developed a new taxonomy-based method called TAXFOLD for protein fold recognition. It extensively exploits the sequence evolution information from PSI-BLAST profiles [103] and the secondary structure information from PSIPRED profiles [6]. A comprehensive set of 137 features is thus constructed, which has been demonstrated capable of depicting both global and local characteristics of profiles. We notice that some computational methods already proposed to exploit PSI-BLAST and PSIPRED profiles for feature extraction (e.g., [68]). However, we carried out this task in quite a few different ways, as briefly summarized below.

- (1) We deal with the negative elements of PSSMs by applying an inverse algorithmic operation rather than simply replacing all of them by zero. The later would lose sequence evolution information to some extent.
- (2) We extract features from the consensus sequences constructed from PSSMs rather than from their respective amino acid sequences. The former retains richer sequence evolution information.
- (3) We divide PSI-BLAST profiles into several non-overlapping fragments which allows for the depiction of local characteristics (e.g., at N-terminus and C-terminus).
- (4) We reduce the state sequences of PSIPRED profiles into the so-called segment sequences, which allows us to characterize the spatial arrangements of α helices and β strands.
- (5) We apply AC transformation to the probability sequences of PSIPRED profiles for feature extraction. To our best knowledge, this is the first attempt to

extract features from the probability sequences.

Four datasets (RDD, EDD, F95 and F194) are used to test and compare the proposed method TAXFOLD with the major existing methods. The first two datasets contain protein domain sequences from 27 folds, and the third and fourth contain sequences from 95 and 194 folds, respectively, representing different levels of challenges that we might face for the task of fold recognition. Our experiments show that, among all the tested methods, TAXFOLD achieves the highest overall accuracies of 83.2%, 90%, 82.4% and 79.6% respectively for the four tested datasets, making an average accuracy improvement of 6.9% over the best available method. We further tested TAXFOLD with a dataset containing a large number of folds (710 folds). The resulting overall accuracy is relatively low (68.1%), because for as many as 185 fold types there are less than ten sequences. Comparisons on the Lindahl benchmark dataset shows that TAXFOLD performs comparably with the best conventional template-based threading methods at the SCOP fold level. These together clearly indicate that the proposed set of features is highly beneficial to protein fold recognition. Therefore, we believe that TAXFOLD is a promising and practical tool for protein fold recognition. For public use, a web server for TAXFOLD is freely accessible at <http://www1.spms.ntu.edu.sg/~chenxin/TAXFOLD/>.

Chapter 5

An Improved Classification of G-Protein-Coupled Receptors Using Sequence-Derived Features [†]

5.1 Introduction

GPCRs play a key role in diverse physiological processes such as neurotransmission, secretion, cellular differentiation, cellular metabolism, and so forth [70]. It has been estimated that almost two-thirds of drugs on the market interact with GPCRs [160], which indicates that GPCRs are pharmacologically important. Therefore, both academic and industrial researchers are very interested in the studies on GPCRs to understand their structures and functions. As mentioned in Chapter 1, the 3D protein structures of GPCRs are largely unavailable [76]. Although some advanced

[†] This chapter is mainly based on an article published in *BMC Bioinformatics* as: Zhen-Ling Peng*, Jian-Yi Yang*, and Xin Chen, An improved classification of G-protein-coupled receptors using sequence-derived features, *BMC Bioinformatics*, 11:420, 2010. (***Contributed equally**)

technologies such as NMR spectroscopy allow to detect the 3D protein structures, their experiments are generally very time-consuming and costly. In contrast, a large number of GPCR primary sequences are known [161]. To facilitate the identification and characterization of novel receptors [161], it is therefore very valuable to develop a computational method to predict GPCRs from the protein primary sequences.

Based on their binding ligand types, GPCRs are often classified into different groups, some of which are further divided into subgroups, sub-subgroups, etc. The GPCRDB database [71] (<http://www.gpcr.org/7tm/>) is one of the most popular database for GPCRs, which organizes GPCRs using a hierarchical structure. As in [162, 163], we call each layer of this hierarchical structure a *level*. The top layer is then referred to as the second level¹, and the second layer is referred to as the third level, etc. According to the latest version of the GPCRDB database (Version 9.9.1, September 2009), GPCRs in the second level are classified into five *families* or *classes*²; that is, (1) *Class A Rhodopsin like*, (2) *Class B Secretin like*, (3) *Class C Metabotropic glutamate/pheromone*, (4) *Vomeronasal receptors (V1R & V2R)*, and (5) *Taste receptors T2R*. For the first four families above³, each is further divided into *subfamilies* located at the third level. Furthermore, located on the fourth and fifth levels of the hierarchical structure are the *sub-subfamilies* and *subtypes*, respectively. On the other hand, given a new protein, the first step is to determine whether it is a GPCR or a non-GPCR. Therefore, we add one more level on the top of the hierarchical structure of the above classification system. It is referred to as the *first* level. The complete hierarchical structure of five levels is illustrated in Figure 5.1.

In this chapter we will look into the following classification problem, which

¹We will add one more layer on the top of the hierarchical structure later.

²In the older versions of the GPCRDB database, e.g., June 2006 release, GPCRs are classified into six families in this level.

³The fifth family in the second level has no subfamily.

Figure 5.1: The hierarchical structure for GPCRs. The organization of GPCR sequences in the GPCRDB database does not include the first level in this figure. We add it in this study because we performed prediction at this level.

is referred to as a *five-level* classification problem. Given a protein sequence, we need to determine whether it is a GPCR or a non-GPCR. If it is predicted into a GPCR, we need to further determine which family, subfamily, sub-subfamily, and subtype it belongs to. To tackle this problem, a set of distinct classifiers is generally needed for each level depicted in Figure 5.1. In the literature, many computational methods have been proposed to predict GPCRs. However, to our best knowledge, there are no methods that can deal with the five-level problem completely, (i.e., allow to make predictions at all the five levels). For example, the methods presented in [164–167] predict GPCRs just at a single level (the second, third or fourth level), and the methods in [168] predict GPCRs only at the third and fourth levels. The prediction methods in [163] and [162] instead considered three and four levels, respectively.

Today’s academic and industrial researchers are both interested in the func-

tional roles of GPCRs at the finest *subtype* level. This is mainly because each subtype demonstrates its own characteristic ligand binding property, coupling partners of trimeric G-proteins, and interaction partners of oligomerization [169]. Therefore, discrimination of functions of a GPCR subtype from the others (i.e., prediction of GPCRs at the fifth level as shown in Figure 5.1) becomes very important in the effort to decipher GPCRs. However, we can expect that it is a challenging task that shall not be easier than the prediction of GPCRs at any of the first four levels. Fortunately, more and more GPCR sequences are now being accumulated into the GPCRDB database, which makes it possible to accurately predict GPCRs at all the five levels. This is the main goal of our present study.

A lot of related work has been done previously. In general, there are two important components in a classification task — one is feature extraction and the other is a classification algorithm. Feature extraction means how to extract features from protein sequences so that each protein is represented as a fixed-length numerical vector. Various methods have been proposed to extract information from protein sequence in the past decades (See e.g., [54, 83–86]). The commonly used feature extraction methods are based on amino acid composition [164–166] and dipeptide composition [162, 167, 168, 170, 171], and more complicated ones include Chou’s pseudo amino acid composition [83], the cellular automaton image approach [84], profile hidden Markov models [172], fast Fourier transform [173], wavelet-based time series analysis [174], and Fisher Score Vectors [175]. Once protein sequences are represented by numerical vectors, any general-purpose classification algorithms can be used for classification, for instance, covariant discriminant [84, 164–166], nearest neighbor [162], bagging classification tree [168], and support vector machines [167, 170, 171, 173–175].

In this chapter, we focus on predicting GPCRs at the five levels. Five groups of descriptors are used to extract information from the amino acid sequences. These

five groups are (1) *amino acid composition and dipeptide composition*, (2) *autocorrelation descriptors*, (3) *global descriptors*, (4) *sequence-order descriptors*, and (5) *Chou’s pseudo amino acid composition descriptors*. These descriptors reflect various physicochemical properties of proteins and have been adopted to predict many other protein attributes, such as protein subcellular localization [86, 176], outer membrane protein [177], nuclear receptors [178], and protein structural classes [54, 85]. By combining these descriptors, a comprehensive set of 1497 features are calculated for each amino acid sequence. By applying the *principal component analysis* on a dataset, we then reduce them to a set of 32 features that could retain as much of the data variability as possible. Finally, a simple yet powerful algorithm called *intimate sorting* is employed to predict GPCRs, and the experimental tests on the benchmark datasets show that the classifications can be improved. Jack-knife test shows that the overall accuracies of the proposed method at the first, second, third, fourth, and fifth levels are 99.5%, 88.8%, 80.47%, 80.3%, and 92.34%, respectively. Comparisons with several existing methods show that the proposed method consistently achieves higher prediction performance.

5.2 Materials and Methods

5.2.1 Datasets

We construct a collection of non-redundant datasets from the latest release of the GPCRDB database (Version 9.9.1, September 2009, <http://www.gpcr.org/7tm/>) to evaluate and train the classifiers for the GPCRs prediction. As mentioned in the Section 5.1, the sequences in the GPCRDB database are organized in four levels: family or class, subfamily, sub-subfamily, and subtype. We download the GPCR sequences from the GPCRDB database and then filter out the high-similarity sequences using the program CD-HIT [179]. In order to ensure that there are enough

sequences to train the classifiers, we apply different thresholds in CD-HIT for sequences at different levels. They are 0.4, 0.7, 0.8, and 0.9 for the family, subfamily, sub-subfamily, and subtype levels, respectively. After filtering, only families (sub-families, sub-subfamilies, and subtypes) with more than 10 sequences are retained for training classifiers. Because the fifth family (*Taste receptors T2R*) has no sub-family and there are only 14 sequences remaining after filtering by CD-HIT, it is therefore ignored in subsequent analysis. At the end, we obtained 1589, 4772, 4924, and 2741 GPCRs at the family, subfamily, sub-subfamily and subtype levels, respectively. The name of families, subfamilies, sub-subfamilies, and subtype, together with the number of GPCR proteins retained at each level are listed in the **Additional file 1** (available at <http://www.biomedcentral.com/content/supplementary/1471-2105-11-420-s1.pdf>).

The GPCR protein sequences retained at the family level are used to construct a positive dataset for training and evaluation. A negative dataset of non-GPCRs is then constructed in a almost the same way as in Ref. [175], except the latest version of ASTRAL SCOP (Version 1.75, <http://astral.berkeley.edu/>) is used. Firstly, we download the sequences that have less than 40% similarity to each other (i.e., the file with the name “seq.75;item=seqs;cut=40”). Then, remove those sequences of length less than 30, and those having similarity above 40% using CD-HIT. Finally, a total of 10325 sequences remain, from which 1589 sequences are randomly selected to form a negative dataset. Because these selected proteins are organized into five levels, for the sake of convenience, we call them the datasets *GDFL* (GPCR Datasets in Five Levels). They are available at the web server provided in this chapter.

In addition, in order to perform comparison with other existing methods directly, four benchmark datasets from previous studies are experimented in this study as well. For the sake of simplicity, they are referred to as *D167*, *D566*, *D1238*

and *D365*, respectively. We know that all of them were constructed based on the older version of the GPCRDB database. The proteins in the dataset *D167* [165] (belonging to the fourth level) are classified into four sub-subfamilies: (1) *acetylcholine*, (2) *adrenoceptor*, (3) *dopamine*, and (4) *serotonin*. The dataset *D566* [166] (belonging to the fourth level) instead includes proteins in seven sub-subfamilies: (1) *adrenoceptor*, (2) *chemokine*, (3) *dopamine*, (4) *neuropeptide*, (5) *olfactory type*, (6) *rhodopsin*, and (7) *serotonin*. *D1238* [164] (belonging to the second level) comprises proteins from three families: (1) *rhodopsin like*, (2) *secretin like*, and (3) *metabotropic/glutamate/pheromone*. The last dataset *D365* [84] (belonging to the second level) comprises proteins in the six families: (1) *rhodopsin-like*, (2) *secretin-like*, (3) *metabotropic/glutamate/pheromone*; (4) *fungus pheromone*, (5) *cAMP receptor* and (6) *frizzled/smoothed family*. The numbers of proteins in the above four datasets are given in Table 5.1. Furthermore, 365 non-GPCR sequences are taken from the Swiss-Prot database to serve as a negative dataset against *D365* [84].

The sequence similarity level is an important factor that affects the effectiveness of a classification method. Therefore, it is worthwhile to take a look at the sequence similarity levels of proteins in these datasets before performing any evaluation test. For simplicity, we analyze the similarity level of the whole dataset rather than the subsets in the dataset. Chou and Elrod [164–166] reported that all the receptor sequences in the aforementioned datasets were generally lower than 40%, according to their definition of the *average sequence similarity percentage* between two protein sequences. Here, we run a protein sequence clustering program called CD-HIT [179] on each dataset with the varying thresholds of sequence similarity. For example, if a threshold of 0.9 is used, the proteins having pairwise residue identities of 90% or above would be placed into a same cluster. In general, the fewer resulting clusters imply the higher overall sequence similarities. The test results are shown in Table

Table 5.1: The number of proteins in the four datasets and the corresponding prediction accuracies. $Tot(i)$ is the number of sequences observed in class i , $c(i)$ is the number of correctly predicted sequences of class i , and ACC is the prediction accuracy.

Dataset	Family/sub-subfamily	$Tot(i)$	$c(i)$	ACC(%)
<i>D167</i>	Acetylcholine	31	31	100
	Adrenoceptor	44	44	100
	Dopamine	38	36	94.74
	Serotonin	54	53	98.15
	<i>Overall</i>	167	164	98.2
<i>D566</i>	Adrenoceptor	66	65	98.48
	Chemokine	92	90	97.83
	Dopamine	43	40	93.02
	Neuropeptide	31	30	96.77
	Olfactory	84	84	100
	Rhodopsin	183	180	98.36
	Serotonin	67	65	97.01
	<i>Overall</i>	566	554	97.88
<i>D1238</i>	Rhodopsin-like	1103	1102	99.91
	Secretin-like	84	83	98.81
	Metabotropic/glutamate/pheromone	51	50	98.04
	<i>Overall</i>	1238	1235	99.76
<i>D365</i>	Rhodopsin-like	232	222	95.69
	Secretin-like	39	34	87.18
	Metabotropic/glutamate/pheromone	44	39	88.64
	Fungal pheromone	23	22	95.65
	CAMP receptor	10	10	100
	Frizzled/smoothened	17	11	64.71
	<i>Overall</i>	365	338	92.6

5.2, where the proteins are clustered with the thresholds of 0.9, 0.8, 0.7, 0.6, 0.5 and 0.4, respectively. In particular, 100 clusters are obtained for 167 proteins in the dataset *D167* with the threshold of 0.9. It indicates that there are high-similarity protein pairs, but they only take up a small proportion of the total number (i.e., $12861 = 167 \times 166/2$) of distinct protein pairs. The use of the threshold of 0.4 further reduces the number of clusters to 30, which could suggest that the *average* sequence similarity of proteins is quite low. However, to avoid the overestimation of prediction accuracy, it would be better if those high-similarity sequences are filtered out with CD-HIT. For instance, the dataset *D365* does not contain any protein pairs having $\geq 40\%$ pairwise sequence similarity except in the E-cAMP receptor family, which contains too few (only 10) GPCRs to apply filtering.

Table 5.2: The CD-HIT clustering results for the four benchmark datasets. γ denotes the threshold for the sequence similarity percentage. The row of $\gamma = 1.0$ gives the total number of proteins in each dataset.

γ	Dataset			
	<i>D167</i>	<i>D566</i>	<i>D1238</i>	<i>D365</i>
1.0	167	566	1238	365
0.9	100	346	777	361
0.8	73	226	540	361
0.7	61	169	421	361
0.6	52	142	358	359
0.5	38	106	281	357
0.4	30	69	207	356

5.2.2 Physicochemical Properties

In order to capture as much information of protein sequences as possible, a variety of physicochemical properties [180] are used in the procedure of feature extraction. These physicochemical properties are listed in Table 5.3, of which the first eighteen are used to measure the physicochemical properties of individual amino acids and

the last two to measure the physicochemical distances between two amino acids.

Table 5.3: The physicochemical properties of the amino acids and distances between two amino acids.

Order	Physicochemical property	Range of property	Reference
1	Hydrophobicity scales	$[-1.14, 1.81]$	[180]
2	Average flexibility indices	$[0.295, 0.544]$	[180]
3	Polarizability parameter	$[0, 0.409]$	[180]
4	Free energy of solution in water	$[-2.24, 4.91]$	[180]
5	Residue accessible surface area in tripeptide	$[75, 255]$	[180]
6	Residue volume	$[36.3, 135.4]$	[180]
7	Steric parameter	$[0, 1.02]$	[180]
8	Relative mutability	$[18, 134]$	[180]
9	Hydrophobicity	$[-2.53, 1.38]$	[104]
10	Hydrophilicity	$[-3.4, 3]$	[104]
11	Side-chain mass	$[1, 130]$	[104]
12	Normalized van der Waals volume	$[0, 8.08]$	[181]
13	Polarity	$[4.9, 13.0]$	[181]
14	Polarizability	$[0, 0.409]$	[181]
15	Charge	Positive, Neutral, Negative	[181]
16	Secondary structure	Helix, Strand, Coil	[181]
17	Solvent accessibility	Buried, Exposed, Intermediate	[181]
18	Relative hydrophobicity	Polar, Neutral, Hydrophobic	[181]
19	Grantham chemical distance	$[0, 215]$	[181]
20	Schneider-Wrede physicochemical distance	$[0, 1]$	[86]

5.2.3 Sequence-Derived Features

As mentioned in the introduction, amino acid composition was widely used to transform GPCR sequences into 20-dimension numerical vectors [164–166]. However, the sequence order information would be completely lost. In order to address this issue, dipeptide composition was proposed to represent GPCR sequences by 400-dimension vectors, which captures local-order information and has been reported to improve classifications [162, 167, 168, 170, 171]. Recently, GPCR-CA [84] utilized the conception of *Chou’s pseudo amino acid composition* [104] to represent each protein sequence by 24 features. The first 20 features corresponded to the amino acid composition and the remaining four features were calculated from a so-called *cellular automation image*. These four features were shown capable of

reflecting a protein's overall sequence pattern. Inspired by this study, we seek a new set of features that can comprehend as much information as possible from GPCR sequences. To this end, we investigate the following five groups of features, where the parameters are set to the same values as in [181].

5.2.3.1 Amino Acid Composition (AAC) and Dipeptide Composition (DC)

Amino acid composition is defined as the occurrence frequencies of 20 amino acids in a protein sequence. That is,

$$f_A(i) = \frac{n_A(i)}{L}, \quad (5.1)$$

where each $i = 1, 2, \dots, 20$ corresponds to a distinct amino acid and $n_A(i)$ is the number of amino acid i occurring in the protein sequence of length L .

Similarly, dipeptide composition is defined as the occurrence frequencies of the 400 dipeptides (i.e., 400 amino acid pairs). That is,

$$f_D(i) = \frac{n_D(i)}{L - 1}, \quad (5.2)$$

where each $i = 1, 2, \dots, 400$ corresponds to one of the 400 dipeptides and $n_D(i)$ is the number of dipeptide i occurring in the sequence.

5.2.3.2 Autocorrelation Descriptors (AD)

We use three autocorrelation descriptors — *normalized Moreau-Broto autocorrelation descriptors*, *Moran autocorrelation descriptors* and *Geary autocorrelation descriptors*. They are all defined based on the value distributions of the first eight physicochemical properties of amino acids along a protein sequence (see Table 5.3). The measurement values of these properties are first standardized before we pro-

ceed to calculate the three autocorrelation descriptors. The standardization is performed as follows.

$$P(i) = \frac{P_0(i) - \bar{P}_0}{\sigma}, \quad i = 1, 2, \dots, 20, \quad (5.3)$$

where $P_0(i)$ are the property value of the amino acid i , $\bar{P}_0 = \frac{1}{20} \sum_{i=1}^{20} P_0(i)$, and

$$\sigma = \sqrt{\frac{1}{20} \sum_{i=1}^{20} (P_0(i) - \bar{P}_0)^2}.$$

Normalized Moreau-Broto autocorrelation descriptors are defined as:

$$NMBA(d) = \frac{MBA(d)}{L - d}, \quad d = 1, 2, \dots, 30, \quad (5.4)$$

where $MBA(d) = \sum_{i=1}^{L-d} P(R_i)P(R_{i+d})$, R_i and R_{i+d} are the amino acids at position i and $i + d$ along the protein sequence, respectively. As mentioned earlier, we use the same parameter values as in [181], so the maximum value of d is 30.

Moran autocorrelation descriptors are defined as:

$$MA(d) = \frac{\frac{1}{L-d} \sum_{i=1}^{L-d} (P(R_i) - \tilde{P})(P(R_{i+d}) - \tilde{P})}{\frac{1}{L} \sum_{i=1}^L (P(R_i) - \tilde{P})^2}, \quad d = 1, 2, \dots, 30, \quad (5.5)$$

where $\tilde{P} = \frac{1}{L} \sum_{i=1}^L P(R_i)$ is the average value of the property of interest along the sequence.

Geary autocorrelation descriptors are defined as:

$$GA(d) = \frac{\frac{1}{2(L-d)} \sum_{i=1}^{L-d} (P(R_i) - P(R_{i+d}))^2}{\frac{1}{L-1} \sum_{i=1}^L (P(R_i) - \tilde{P})^2}, \quad d = 1, 2, \dots, 30. \quad (5.6)$$

For each autocorrelation descriptor, we obtain 240 ($= 30 \times 8$) features. In total,

720 ($= 240 \times 3$) features will be obtained to describe a protein sequence.

5.2.3.3 Global Descriptors (GD)

These descriptors were firstly proposed by Dubchak *et al.* [145] to predict protein folding classes, and later applied to predict human Pol II promoter sequences [101]. They are constructed as follows. Firstly, given each of the following seven amino acid properties: *normalized van der Waals volume*, *polarity*, *polarizability*, *charge*, *secondary structure*, *solvent accessibility* and *relative hydrophobicity* (i.e., properties 12–18 listed in Table 5.3), the 20 amino acids are divided into three groups according to their property values. Then, for a given amino acid sequence, we may obtain a new sequence of three symbols, each corresponding to one group of amino acids. Finally, three group of quantities are defined on the new sequence; that is, composition (*Comp*), transition (*Tran*) and distribution (*Dist*), as demonstrated below.

For the sake of simplicity, suppose that a sequence is made of only two letters (A and B). *Comp* is defined as the occurrence frequency of each letter in the sequence. For example, we have a sequence BABBABABBABBAABABABBAAAB-BABABABA, in which there are 14 As and 16 Bs. Therefore, the occurrence frequencies of A and B are $14/(14+16) \times 100.00 = 46.67$ and $16/(14+16) \times 100.00 = 53.33$, respectively. *Tran* is used to represent the occurrence frequency of pairs AB or BA. In the above sequence, there are 21 transitions from one letter to another; so *Tran* is computed as $(21/29) \times 100.00 = 72.14$. On the other hand, *Dist* calculates the relative positions of the first, 25%, 50%, 75% and 100% of the total amount of a particular letter in the sequence. In the above sequence, for example, the first, 25%, 50%, 75% and 100% of the total amount of the letter B are located at the first, 6th, 12th, 20th and 29th positions, respectively. The quantities *Dist* for the letter B are hence $1/30 \times 100.00 = 3.33$, $6/30 \times 100.00 = 20.00$, $12/30 \times 100.00 = 40.00$,

$20/30 \times 100.00 = 66.67$ and $29/30 \times 100.00 = 96.67$. Similarly, we can find the *Dist* values for the letter A; they are 6.67, 23.33, 53.33, 73.33 and 100.00. At the end, the global descriptors of the above sequence become $(Comp; Tran; Dist) = (46.67, 53.33; 72.14; 6.67, 23.33, 53.33, 73.33, 100.00, 3.33, 20.00, 40.00, 66.67, 96.67)$.

Suppose there are n distinct symbols in a sequence, then the number of features in *Comp*, *Tran*, and *Dist* are $\binom{n}{1}$, $\binom{n}{2}$, and $5 \times n$, respectively. Recall that the 20 amino acids are divided into three groups by each amino acid property, which leads to a new sequence of three symbols ($n = 3$). Following the similar procedure demonstrated above, we will obtain 21 ($=\binom{3}{1} + \binom{3}{2} + 5 \times 3$) features to describe the new sequence (of three symbols). Combining all the features to be extracted based on the seven amino acid properties, we will obtain a total of 147 ($=21 \times 7$) features for each input protein sequence from the global descriptors.

5.2.3.4 Sequence-Order Descriptors (SD)

In order to derive sequence-order descriptors, we rely on two distance measures for amino acid pairs. One is called the *Grantham chemical distance matrix* [181], and the other called the *Schneider-Wrede physicochemical distance matrix* [86]. Then, the j th-rank sequence-order-coupling number is defined as:

$$\tau(j) = \sum_{i=1}^{L-j} (d(R_i, R_{i+j}))^2, \quad j = 1, 2, \dots, 30, \quad (5.7)$$

where $d(R_i, R_{i+j})$ is one of the above distances between the two amino acids R_i and R_{i+j} located at position i and $i + j$, respectively.

The *quasi-sequence-order descriptors* are defined as:

$$QSO(i) = \begin{cases} \frac{f_A(i)}{\sum_{j=1}^{20} f_A(j) + \omega \sum_{j=1}^{30} \tau(j)}, & (1 \leq i \leq 20) \\ \frac{\omega \cdot \tau(i)}{\sum_{j=1}^{20} f_A(j) + \omega \sum_{j=1}^{30} \tau(j)}, & (21 \leq i \leq 50) \end{cases} \quad (5.8)$$

where ω is a weighting factor (default $\omega = 0.1$).

We end up with 60 ($= 30 \times 2$) sequence-order-coupling numbers and 100 ($= 50 \times 2$) quasi-sequence-order descriptors. In total, there are 160 features extracted from the sequence-order descriptors.

5.2.3.5 Chou's Pseudo Amino Acid Composition Descriptors (PseAAC)

This set of features were originally developed by Chou [104] and have been used widely to predict various attributes of proteins, such as outer membrane protein [177], nuclear receptors [178], and protein structural classes [54, 85]. The Chou's pseudo amino acid composition descriptors are defined similarly as the quasi-sequence-order descriptors. The difference lies in the coupling number $\tau(j)$, which is modified to:

$$\theta(d) = \frac{1}{L-d} \sum_{i=1}^{L-d} \Theta(R_i, R_{i+d}), \quad d = 1, 2, \dots, 30, \quad (5.9)$$

where $\Theta(R_i, R_{i+d})$ is the d th-tier correlation factor that reflects the sequence order correlation between all the most contiguous residues along a protein chain. It is defined as:

$$\Theta(R_i, R_{i+d}) = \frac{1}{3} \{ [H_1(R_i) - H_1(R_{i+d})]^2 + [H_2(R_i) - H_2(R_{i+d})]^2 + [M(R_i) - M(R_{i+d})]^2 \}, \quad (5.10)$$

where $H_1(R_i)$, $H_2(R_i)$ and $M(R_i)$ are the *hydrophobicity*, *hydrophilicity*, and *side-chain mass of amino acid*, respectively [104]. The original values of them are

standardized similarly as we have done in the definition of autocorrelation descriptors (i.e., eq. (3)). Finally, the Chou's pseudo amino acid composition descriptors are defined as:

$$PseAAC(i) = \begin{cases} \frac{f_A(i)}{\sum_{j=1}^{20} f_A(j) + \omega \sum_{d=1}^{30} \theta(d)}, & (1 \leq i \leq 20) \\ \frac{\omega \cdot \theta(i-20)}{\sum_{j=1}^{20} f_A(j) + \omega \sum_{d=1}^{30} \theta(d)}, & (21 \leq i \leq 50) \end{cases} \quad (5.11)$$

where ω is a weighting factor (default $\omega = 0.1$). It will generate 50 features from the Chou's pseudo amino acid composition descriptors.

In summary, a comprehensive set of 1497 features, which measure the protein sequences from different aspects, will be generated from the above five descriptors. The number of features in each group of descriptors is listed in Table 5.4. These features are used to represent every protein sequence, and may be directly fed into a classification algorithm. Note that, however, there are some correlations or redundancies among these features such as the first twenty features in the fourth and fifth groups of features. On the other hand, the dimension of the features is too large, which might make it difficult to work with many machine learning algorithms for classification. Therefore, it is necessary to reduce the dimension. In this study, we adopt one of the most popular and powerful techniques, namely, *principal component analysis*, for the purpose of dimensionality reduction.

Table 5.4: The number of features in each group of descriptors.

Order	Name	Number of features
(i)	Amino acid composition (AAC) and dipeptide composition (DC)	420
(ii)	Autocorrelation descriptors (AD)	720
(iii)	Global descriptors (GD)	147
(iv)	Sequence-order descriptors (SD)	160
(v)	Chou's pseudo amino acid composition descriptors (PseAAC)	50
(vi)	All features	1497

5.2.4 Principal Component Analysis

Principal Component Analysis (PCA) is a classical statistical method which is still widely used in modern data analysis. PCA involves a mathematical procedure that transforms a large number of (possibly) correlated variables into a smaller number of uncorrelated variables, called *principal components* (PCs), that retain as much variability of the data as possible [182]. Given a data matrix denoted by $X = (X_1, X_2, \dots, X_p)$, where X_i is a column vector of size n which is equal to the number of proteins of interest and p denotes the number of protein sequence features, a typical PCA is performed as follows. First, we shall standardize every X_i by

$$Y_i = \frac{X_i - \bar{X}_i}{\sqrt{\text{Var}(X_i)}}, \quad i = 1, 2, \dots, p, \quad (5.12)$$

where $\bar{X}_i$ and $\text{Var}(X_i)$ are the mean and variance of the vector components of X_i , respectively. Then, the covariance matrix of $Y = (Y_1, Y_2, \dots, Y_p)$ is obtained as

$$\text{Cov}(Y) = \frac{1}{n-1} Y^T Y. \quad (5.13)$$

For the covariance matrix $\text{Cov}(Y)$, we find all its eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p$ and the corresponding eigenvectors, $E_1, E_2, \dots, E_p$. Note that each $E_i = (E_{i,1}, E_{i,2}, \dots, E_{i,p})^T$ is a column vector of size p and $E_1, E_2, \dots, E_p$ are linearly uncorrelated according to the basic knowledge of linear algebra. Finally, we construct the i -th PC $PC(i)$ as the linear combination of $Y_1, Y_2, \dots, Y_p$ with the coefficients being the elements of the i -th eigenvector E_i , i.e.,

$$PC(i) = E_{i,1}Y_1 + E_{i,2}Y_2 + \dots + E_{i,p}Y_p, \quad i = 1, 2, \dots, p. \quad (5.14)$$

We can see that each $PC(i)$ is a column vector with size n and the j -th element

in $PC(i)$ represents the i -th PC of protein j . Thereafter, a total of p uncorrelated PCs are obtained.

In order to reduce the dimension of the feature space, only the first m PCs are used to represent each protein sequence ($m \leq p$). It is generally hard to determine the optimal value of m . In this study, we aim to find a value of m that could make the overall prediction accuracy of GPCRs as high as possible, which we will further discuss later.

We want to mention that when applying PCA to a dataset with n proteins based on the jackknife test, a total of n different runs would be performed. In each run, the proteins in the training set are used to find the eigenvectors $E_1, E_2, \dots, E_p$, in order to reduce the dimension of the remaining proteins. If all proteins in the dataset are used to find the eigenvectors in PCA, it would lead to over-estimation of the performance because the information in the test instance is used during training in such case.

5.2.5 Intimate Sorting Algorithm

Many classification algorithms in the literature have been used to predict GPCRs, for instance, covariant discriminant [84, 164–166], nearest neighbor [162], bagging classification tree [168], and support vector machines [167, 170, 171, 173–175]. In this study, we use a simple yet powerful algorithm called *intimate sorting* [176]. This algorithm is easy to implement and does not need to set any parameters as some other algorithms (e.g., support vector machines).

Suppose that a training set consists of N proteins $\{\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_N\}$, each of which $\mathbf{P}_i$ is a λ -dimension vector, $\mathbf{P}_i = (p_{i,1}, p_{i,2}, \dots, p_{i,\lambda})^T$. The GPCR classes of these proteins are already known, and each protein belongs to exactly one of the μ classes. The intimate sorting algorithm aims to place a query protein $\mathbf{P} = (p_1, p_2, \dots, p_\lambda)^T$ into one of the μ classes based on the information from the N

proteins in the training set. To this end, a measure of *similarity score* between $\mathbf{P}$ and $\mathbf{P}_i$ is defined as

$$\Phi(\mathbf{P}, \mathbf{P}_i) = \frac{\mathbf{P} \cdot \mathbf{P}_i}{\|\mathbf{P}\| \|\mathbf{P}_i\|}, \quad i = 1, 2, \dots, N, \quad (5.15)$$

where $\mathbf{P} \cdot \mathbf{P}_i = \sum_{j=1}^{\lambda} p_j p_{i,j}$, $\|\mathbf{P}\| = \sqrt{\sum_{j=1}^{\lambda} p_j^2}$ and $\|\mathbf{P}_i\| = \sqrt{\sum_{j=1}^{\lambda} p_{i,j}^2}$. When $\mathbf{P} \equiv \mathbf{P}_i$, it can be easily seen that $\Phi(\mathbf{P}, \mathbf{P}_i) = 1$, suggesting that they are most likely to belong to a same class. In general, we have $-1 \leq \Phi(\mathbf{P}, \mathbf{P}_i) \leq 1$. The higher the $\Phi(\mathbf{P}, \mathbf{P}_i)$ value, the more likely two proteins belong to a same class. Among the N proteins in the training set, the one with the highest score with the query protein $\mathbf{P}$ is picked out, which we denote by $\mathbf{P}_k$, $k \in [1, N]$. If there is a tie, we would randomly select one of them. In the final step, the intimate sorting algorithm simply assigns $\mathbf{P}$ into the same GPCR class as $\mathbf{P}_k$.

5.2.6 Prediction Assessment

In this chapter, we also use the jackknife test [183] to evaluate our method. The prediction accuracies (ACC) and overall accuracy ($OACC$) are measured by the following formulae:

$$ACC(i) = \frac{C(i)}{\text{Tot}(i)}, \quad i = 1, 2, \dots, \mu, \quad \text{and} \quad OACC = \frac{\sum_i^{\mu} C(i)}{\sum_i^{\mu} \text{Tot}(i)}, \quad (5.16)$$

where $\text{Tot}(i)$ is the total number of sequences in class i , $C(i)$ the number of correctly predicted sequences of class i , and μ the total number of classes under consideration. Note that this prediction assessment method was already adopted in several previous studies, e.g., [83, 84, 162, 174].

5.3 Results and Discussions

5.3.1 Selection of m

As we mentioned earlier, the number of PCs in PCA, i.e., m , remains to be determined. Here, we choose its value by aiming to achieve the overall prediction accuracy as high as possible. To this end, we use the dataset *D365* to compute the overall prediction accuracies *OACCs* of GPCR families for varying values of m . When m ranges from 1 to 80, *OACCs* thus obtained are plotted in Figure 5.2. We found that the highest accuracy (92.6%) is achieved with $m = 32$. Based on this observation, we chose $m = 32$ in our experiments.

5.3.2 Predicting GPCRs at Five Levels

For simplicity, we call the proposed method PCA-GPCR. PCA-GPCR preforms the prediction at five levels, and its flowchart structure is depicted in Figure 5.3. By the first-level classifier a new protein sequence is predicted to be either a GPCR or a non-GPCR. If it is predicted to be a GPCR, it will be further classified into one of the four families, which is done by the second-level classifier. The third-level classifiers hence determine which subfamily the protein belongs to. For some subfamilies (see **Additional file 1**), the fourth-level classifiers determine the sub-subfamily of the protein. Finally, the fifth-level classifiers determine the subtypes of the protein, if any (see **Additional file 1**). We carried out the experiments on the collection of datasets *GDFL* (Please see Section 5.2.1 for the details of datasets). Jackknife tests show that the overall accuracies of PCA-GPCR are 99.5%, 88.8%, 80.47%, 80.3%, and 92.34% for the five levels, respectively. The details of experimental results are presented in the **Additional file 1**. It is commonly believed that, the smaller number of training sequences, the less reliable a classifier to be

Figure 5.2: Selection of m . The overall prediction accuracies of GPCR families for the *D365* dataset obtained by varying the number m of principle components. The highest overall accuracy is achieved when $m = 32$, which is marked by the dotted lines.

Figure 5.3: The structure of PCA-GPCR. For the name of the families, subfamilies, sub-subfamilies, and subtypes, please refer to the **Additional file 1**. The fourth and fifth levels are only applicable for some subfamilies and subtypes, which are also listed in the **Additional file 1**.

trained. Therefore, it is not surprising to see that the prediction accuracies are higher at the first and second levels and relatively lower at the third and fourth levels. On the other hand, to filter out high-similarity sequences, we used CD-HIT with a less stringent threshold (0.9) for the fifth level than the one for any other levels, which results in a larger number of training sequences for the fifth level. This might partly explain why the accuracy achieved for the fifth level (subtype) is higher than those of the second, third and fourth levels. For the convenience of public use, a web server was already developed, which is freely available at http://www1.spms.ntu.edu.sg/~chenxin/PCA_GPCR/.

5.3.3 Contribution of Features

Inspired by the PCA-based feature selection method described in [182, 184], we use the following procedure to assess the contributions of the 1497 features to prediction accuracy. Recall that in the previous principle component analysis on the dataset *D365* we obtained 32 eigenvectors $E_1, E_2, \dots, E_{32}$, and each eigenvector comprises 1497 components. Let us denote $E_i = (E_{ij})$, where $1 \leq i \leq 32$ and $1 \leq j \leq 1497$. To find the i -th PC in PCA, E_{ij} is used to weight the j -th feature. In this sense, the value E_{ij} can be viewed as the weight of contributions that the j -th feature makes to the i -th PC. To combine the contributions to all the PCs, we may compute $w_j = \sqrt{\sum_{i=1}^{32} E_{ij}^2}$. Then, w_j can be naturally viewed as the weight of contributions that the j -th feature makes to the final prediction accuracy because our method is based on these 32 PCs. In general, the higher the weight w_j , the more contributions the j -th feature makes.

The contribution of each of the 1497 features is computed and depicted in Figure 5.4(A), where we can see that the contributions of the amino acid composition (AAC) in the first group of descriptors are much higher than those of the dipeptide composition (DC). Therefore, we separate the AAC from the DC in the first group

Figure 5.4: Contribution of features. The meanings of the notations AAC, DC, AD, GD, SD, and PseAAC can be found in Table 5.4. The divisions of these six subsets are marked by vertical lines.

of descriptors in the following discussions. In addition, we find that the features from the autocorrelation descriptors (AD) made the highest contributions among all the features. Because there are 1497 features, it is not convenient to discuss the contributions of all individuals one by one. Instead, we compute the average contributions of the features in the following six subsets: AAC, DC, AD, GD, SD, and PseAAC. Their results are shown in Figure 5.4(B). It is evident from the figure that the highest average contribution is obtained with the features in the PseAAC subset (0.1657). The slightly lower contributions are provided by the AD and AAC subsets (0.1595 and 0.1579, respectively). On the contrary, the features in the GD and SD subsets achieve the average contributions only slightly higher than 0.14. The features in the DC subset instead achieve the least average contribution (0.1092). In summary, if we arrange the features in the six subsets in a decreasing order according to their average contributions, then we obtain PseAAC, AD, AAC, GD, SD, and DC.

In particular, among the AD subset, some features made quite high contributions while the others made relatively low contributions, as we can see in Figure 5.4(A). For a thorough investigation, we plot the contributions of all the features in the AD subset again in Figure 5.5. These features are divided into eight groups according to the physicochemical properties used to compute them. In Figure 5.5, the eight groups of features are separated by vertical longer lines and indicated by P1, P2, ..., P8, respectively. Note that P_i represents the i -th physicochemical property listed in Table 5.3. It is evident from Figure 5.5 that the highest contributions are due to the features computed with the physicochemical properties P3, P5 and P6; they are the polarizability parameter, residue accessible surface area in tripeptide and residue volume, respectively. For the group P3, we can further divide its 90 features into three subgroups according to three different autocorrelation descriptors (normalized Moreau-Broto, Moran, and Geary autocorrelation

Figure 5.5: Contribution of features in the AD subset. The divisions of these eight groups are marked by longer vertical lines. Among groups P3, P5 and P6, divisions of three subgroups are marked by shorter vertical lines.

descriptors). These three subgroups are separated by shorter vertical lines in Figure 5.5, and indicated by P3.1, P3.2 and P3.3, respectively. In each subgroup, the feature contributions are computed with the values of d varying from 1 to 30 (from left to right on the horizontal axis). Observe that Moran and Geary autocorrelation descriptors (P3.2 and P3.3) made much higher contributions than the normalized Moreau-Broto descriptors (P3.1). Furthermore, for Moran and Geary autocorrelation descriptors, the features that are computed with a value of d in the range from 20 to 30 generally give rise to a fairly high contribution, while the maximum is attained when $d = 26$. The similar characteristics can also be observed for the groups P5 and P6 from Figure 5.5.

5.3.4 Comparison with BLAST-based Classification

The most straightforward method for predicting GPCRs might be based on homology search by sequence alignment tools such as BLAST and PSI-BLAST [103]. A given GPCR sequence is hence predicted into the class to which its most similar GPCR sequence belongs. However, as the pairwise sequence similarities get lower, such an alignment-based method would rarely yield satisfactory predictions. For instance, when applied to the dataset *GDFL* for the prediction at the first level, the BLAST-based method achieved the overall accuracy of 74.58%, which is 14.92% lower than that from PCA-GPCR. Note that PCA-GPCR is instead an alignment-free method. The above experimental results therefore show that an alignment-free method is very promising in the high accurate prediction of GPCR classes.

5.3.5 Comparison with Previous Methods

In order to demonstrate the superior performance of PCA-GPCR, we make comparisons with a number of previous methods. Depending on the predictive capability of previous methods, the comparisons are made at a single level and at the first two levels, as follows.

5.3.5.1 Comparison at a Single Level

Because many previous methods predicted GPCR at a single level [164–167], we also predict GPCR at just one level in order to compare with them fairly. Three benchmark datasets that contain a proportion of high-similarity sequence pairs, *D167*, *D566* and *D1238*, are used here (Please see Section 5.2.1 for the details of these datasets). The first two datasets comprise GPCRs from the fourth level, and the last one is composed of GPCRs from the second level. The resulting prediction accuracies for these datasets are listed in Table 5.1. We can see that the overall accuracies for three datasets are all above 97%. To be specific, the overall accuracies of 98.2%, 97.88%, and 99.76% are achieved for the datasets *D167*, *D566*, and *D1238*, respectively. They are slightly higher than the accuracies reported in Refs. [162, 164–168, 171]. Indeed, these prediction accuracies are all very high and, in some cases, has reached 100% or nearly 100%.

Because the dataset *D167* has been widely used to test various methods, it is adopted here for further detailed comparison with the other five methods [162, 165, 167, 168, 171]. The experimental results are presented in Table 5.5. It is evident from the table that our method achieved the highest overall prediction accuracy. Our method performs better than any other tested methods in the predictions of the GPCR sub-subfamilies except for the sub-subfamily *Serotonin*.

Table 5.5: Comparison with other methods at the fourth level based on the *D167* dataset. The results of other methods are taken directly from the corresponding references. The best results are highlighted in bold.

Reference	Acetylcholine	Adrenoceptor	Dopamine	Serotonin	Overall
[165]	67.74	88.64	81.58	88.89	83.23
[168]	90.3	86.4	78.9	79.6	83.2
[167]	93.6	100	92.1	98.2	96.4
[162]	93.3	100	94.7	100	97.6
[171]	96.7	100	92.1	100	97.6
This chapter	100	100	94.74	98.15	98.2

5.3.5.2 Comparison with GPCR-CA at the First Two Levels

We further compare our method with GPCR-CA [84] on the dataset *D365*, which comprises GPCRs from the second level. Unlike the datasets tested in the above subsection, *D365* contains almost no high-similarity sequence pairs. Note that the GPCR-CA is able to predict GPCRs at the first two levels.

The prediction accuracies of both GPCR-CA and PCA-GPCR at the first and second levels are listed in Table 5.6 and Table 5.7, respectively. At the first level, to distinguish GPCRs from non-GPCRs, our method achieves the overall accuracy of 95.21%, which is 3.57% higher than that of GPCR-CA. At the second level, the overall accuracy of our method improves over GPCR-CA by 9.04%. Meanwhile, according to the prediction accuracies of individual families, our method performs much better than GPCR-CA except for the *rhodopsin-like* family. It is also noticeable that a substantial improvement of 86.95% ($= 95.65\% - 8.70\%$) has been made for the prediction of the *fungus pheromone* family. We check this in detail and find that the reason for this is because the high similarity scores (in the intimate sorting algorithm) between the 32D vectors (formed by the 32 principal components) that are used to represent the GPCRs in this family.

GPCR-CA extracts 24 features, including 20 features from amino acid compo-

Table 5.6: Comparison with GPCR-CA in identifying the GPCRs and non-GPCRs. The results of GPCR-CA are directly taken from the Ref.[84]. The best results are highlighted in bold.

Protein type	GPCR-CA[84]	This chapter
GPCR	92.33	96.99
Non-GPCR	90.96	93.42
<i>Overall</i>	91.64	95.21

Table 5.7: Comparison with GPCR-CA for the dataset *D365* in predicting GPCR families. The results of GPCR-CA are directly taken from the Ref.[84]. The best results are highlighted in bold.

Family	GPCR-CA	This chapter
Rhodopsin-like	96.55	95.69
Secretin-like	74.36	87.18
Metabotropic/glutamate/pheromone	81.82	88.64
Fungal pheromone	8.70	95.65
CAMP receptor	60	100
Frizzled/smoothened	47.06	64.71
<i>Overall</i>	83.56	92.60

sition and four features from cellular automaton image [84]. While the last four features were reported to be able to reveal the protein’s overall sequence patterns, only four features might not suffice to reveal overall sequence patterns completely. On the contrary, our method explores the amino acid sequences comprehensively to gain as much information from the protein primary sequences as possible. Both the amino acid composition and the dipeptide composition are utilized in our method and, moreover, the important sequence-order information and a variety of physicochemical properties of amino acids are carefully explored as well. We believe that it is this comprehensive set of features that lead our method to a higher prediction accuracy.

5.4 Conclusions

In this chapter, we have proposed a new method called PCA-GPCR to predict GPCRs at five levels. In this method, a comprehensive set of 1497 sequence-derived features are generated from five groups of descriptors — that is, *amino acid composition and dipeptide composition*, *autocorrelation descriptors*, *global descriptors*, *sequence-order descriptors*, and *Chou’s pseudo amino acid composition descriptors*. These features are able to capture the information about the amino acid composition, sequence order as well as various physicochemical properties of proteins. Because of the high dimensionality of the feature space, the principal component analysis is hence used to reduce the dimension from 1497 to 32. The resulting 32-dimensional feature vectors are finally fed into a simple yet powerful intimate sorting algorithm for the prediction of GPCRs at five levels.

Evaluating on the datasets constructed from the latest version of the GPCRDB database, the overall accuracies of our method from the first level to the fifth level are 99.5%, 88.8%, 80.47%, 80.3%, and 92.34%, respectively. We further test and

compare our method with other methods based on four benchmark datasets widely used in the literature. At the second level, for a dataset containing 1238 GPCRs, the overall accuracy of our method reaches 99.76%. At the fourth level, for two different datasets that contain 167 and 566 GPCRs, the overall accuracies of our method reach up to 98.2% and 97.88%, respectively. They are all higher than those of other methods under comparison. At the first two levels, we further test our method on a low-similarity dataset (with only a few sequence pairs of more than 40% sequence similarity). The overall accuracies thus achieved at the first level and second level are 95.21%, 92.6%, respectively, which are 3.57% and 9.04% higher than those of the other compared method GPCR-CA.

We conclude that the high prediction accuracy of the proposed method is attributed to the comprehensive set of features that we constructed from five groups of descriptors. It is anticipated that our method could contribute more to the characterization of novel proteins and gain more insights into their functions, thereby facilitating drug discovery. A web server that predicts GPCRs at five levels with our proposed method is freely available at http://www1.spms.ntu.edu.sg/~chenxin/PKA_GPCR/.

Chapter 6

A Consensus Approach to Predicting Protein Contact Map via Logistic Regression [†]

6.1 Introduction

As mentioned in Chapter 1, sequence-based prediction of contact map has attracted great attention in the past years. Many computational methods are already proposed to predict protein contact map. These methods can be classified into two major categories: (i) methods based on correlated mutations [9, 10, 185–187], and (ii) methods based on machine learning [11, 12, 14, 29, 188–191]. There also exist some other methods, e.g., based on template-threading [190, 192] and integer linear optimization [193]. However, the accuracy of contact prediction, especially for

[†] This chapter is mainly based on an article published in *Lecture Notes in Bioinformatics* as: Jian-Yi Yang and Xin Chen, A consensus approach to predicting protein contact map via logistic regression, *Lecture Notes in Bioinformatics*, Springer Verlag, 6674:136-147, 2011.

long-range contact prediction, is still rather low [92, 194].

In this chapter, we intend to improve the accuracy of contact map prediction by using a consensus approach, which means that the prediction results from several existing predictors will be consolidated. To our best knowledge, not much effort has been made to develop a consensus contact prediction method except the following two approaches. Confuzz is a consensus approach based on the weighted average of the probability estimates from individual predictors (please refer to the website of CASP9). The other approach is based on integer linear programming [195]. We instead choose to tackle this problem in a different way. We consolidate the prediction results from individual predictors by using a logistic regression analysis under the machine learning framework. Tests on the CASP9 dataset as well as on another large dataset show that the proposed method not only outperforms its component predictors but also the simple averaging and voting schemes.

6.2 Materials and Methods

6.2.1 Datasets

In this chapter, two datasets are used to test the proposed method, which are downloadable at <http://www3.ntu.edu.sg/home2008/YANG0241/LRcon/>. The first one was collected from the targets in the recent CASP9 experiment. In CASP9, there are 28 participating groups in the contact prediction category. As one group might have several contact prediction models for the same target, here we selected the results only from the “model 1” of each predictor. In addition, we removed from further consideration those groups that made predictions for just a few targets and those targets that were predicted by just a few participating groups. As a result, we obtained 80 targets and 23 predictors. For the sake of convenience, we denote this dataset by D80. Finally, the true contact map for each target was derived from

its 3D structure provided on the CASP9 website.

The second dataset was harvested from the PDB library [2] using the selected protein chains from the latest (May 2010) PDB_select 25% list [196]. Originally, there are 4869 protein chains in this list. A subset was extracted as follows. First, those chains with length less than 50 and/or coordinates information missing for some amino acids were removed. Second, those with pair-wise sequence similarity higher than 25% and those with sequence similarity to the NNcon training set [191] higher than 25% were further removed. This filtering process ends up with a total of 856 chains. We denote this dataset by D856.

6.2.2 Contact Definition

Two residues are defined to be in contact if the Euclidean distance between the 3D coordinates of their C_α atoms is less than or equal to 8 Å [12, 190, 191]. The CASP experiments [194], however, used C_β atoms instead of C_α atoms in determining two residues in contact. In this study, we choose the former definition because (i) it is a definition close to the one used in 3D structural modeling [197] and (ii) it was already used by two methods (i.e., [190] and [191]) that will be included in our consensus predictor.

For a protein with length L , the (true) contact information for all pairs of residues can be represented by a matrix $C = (c_{ij})_{L \times L}$, where $c_{ij} = 1$ if the residues i, j are in contact and $c_{ij} = 0$ otherwise. This matrix is often called a *contact map*. It is in fact a 2D description of protein structure, and a specific example of contact map is given in Figure 6.1.

Depending on the separation of two residues along the sequence, the residue-residue contact is classified into three classes: short-range contact (separation $6 \leq s < 12$), medium-range contact ($12 \leq s < 24$) and long-range contact ($s \geq 24$). Contacts for those residues too close along the sequence ($s < 6$) are omitted.

Figure 6.1: An example of contact map at sequence separation $s \geq 6$. The left panel is a cartoon visualization of the 3D structure of the protein (PDB ID: 2NWF). The right panel is the contact map of this protein. A blue point in the figure indicates that the pair of residues are in contact. Note that the map is symmetrical with respect to the black main diagonal line.

6.2.3 Performance Evaluation

The predicted contact map $PC = (pc_{ij})_{L \times L}$ is a matrix of probability estimates. The element pc_{ij} is the estimate for the contact probability of the residues i and j . In general, the top λL predictions (sorted by the probability estimates) are selected, which are then compared with the true contact map for evaluation. In the literature [13, 14, 29, 191, 194], the value of λ is usually set to be 0.1 or 0.2 and two metrics are used to evaluate the predictions: accuracy (Acc) and coverage (Cov).

$$\text{Acc} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{Cov} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad (6.1)$$

where TP, FP, TN and FN, are true positive, false positive, true negative and false negative predictions, respectively. A residue pair is said to be a positive (resp., negative) pair if the two residues are (resp., are not) in contact.

In addition, a more robust metric called *F-measure* (Fm) is also used, which is

basically a harmonic mean of precision and recall as defined below:

$$F_m = 2 \cdot \frac{\text{Acc} \times \text{Cov}}{\text{Acc} + \text{Cov}} \quad (6.2)$$

6.2.4 Consensus Prediction via Logistic Regression

Suppose there are p predictors, then we have p predicted contact maps for each protein. We attempt to combine these p maps to make a consensus prediction. The first difficulty appears that the output of some predictors (e.g., FragHMMent [11]) is not the whole map but part of the map. To overcome it, the probability estimates for those missing predictions are simply set to be 0 (i.e., not in contact).

A direct and simple way to combine the p predicted contact maps is to average over the p probability estimates for each residue pair and then select the top λL predictions. We call this method the *averaging* scheme. Another way is to first select the top λL predictions from each predicted map and then use these selected predictions to vote. The residue pairs with votes in the top λL positions are then output to be the top λL predictions. We call this method the *voting* scheme.

In this chapter, we propose to combine the p predicted contact maps via a logistic regression analysis. Logistic regression (LR) is a non-linear regression model in particular for a binary response variable [198]. It estimates the posterior probabilities by using the following formula:

$$P(Y_i = 1|P_i) = \frac{\exp(\alpha + \sum_{j=1}^p \beta_j p_{ij})}{1 + \exp(\alpha + \sum_{j=1}^p \beta_j p_{ij})} \quad (6.3)$$

where $P(Y_i = 1|P_i)$ is the posterior probability of the i -th residue pair being in contact given P_i . $P_i = (p_{i1}, p_{i2}, \dots, p_{ip})$ is a probability vector for the i -th residue pair, of which each component p_{ij} is the probability estimate of the component

predictor j on the i -th residue pair. The constants α and β_j ($j = 1, 2, \dots, p$) are the regression coefficients whose values can be estimated with a training set through Quasi-Newton optimization [198]. We used the implementation of LR in the software package Weka [199] (with default parameters) for our experiments.

6.2.5 Overall Architecture

Figure 6.2 depicts the overall architecture of our proposed method named LRcon. It comprises two major procedures: training and testing. In the training procedure, a logistic regression model (LR-Predictor) is built up with a training set of protein chains. In the testing procedure, a query amino acid sequence is first input into p individual predictors and, for each predictor, the top λL predictions are selected. Then, we take the union of all the selected residue pairs for further consideration (Please refer to Section 6.2.6 for more details). For each selected pair, the probability estimates of the p predictors are used to form a feature vector, which is then fed into the LR-Predictor for consensus contact prediction. Finally, the top λL contact predictions are selected as our consensus predictions.

For the D80 datasets, we use 23 predictors from CASP9 to build our consensus predictor. For the D856 dataset, only three predictors (SVM-SEQ (RR204) [190], NNcon (RR119) [191] and FragHMMent (RR158) [11]) are instead used, because there is no software available for the other predictors except SVMcon (RR002) [12]. SVMcon is excluded because it was developed based on the same classification algorithm as SVM-SEQ, so their predicted results are expected to have a large overlap.

Figure 6.2: The overall architecture of LRcon. In the training procedure, the consensus predictor is built upon p individual predictors $M_1, M_2, \dots, M_p$. The set TD is used to store the training feature vectors of the selected residue pairs. During the testing, the prediction result is stored in $rr_i^{LR} = (r_{i,1}^{LR}, r_{i,2}^{LR}, p_i^{LR})$, where $1 \leq r_{i,1}^{LR}, r_{i,2}^{LR} \leq L$ are the indices of the predicted residue pair i , and p_i^{LR} is its probability estimate. $k_1, \dots, k_{\lambda L}$ are the indices of the top λL predictions.

6.2.6 Selection of Residue Pairs

Given a protein of length L , the total number of residue pairs is $(L+1-s) \times (L-s)/2$ for sequence separation at least s . If all these residue pairs are used, we would not be able to obtain a reliable LR-Predictor, due to at least two factors: (1) A large number of training samples does not allow to estimate the regression coefficients in a reasonable amount of computing time; (2) Most of the residue pairs belong to the negative class, so that a small proportion of positive samples make the predictions be severely biased against the positive class. This would inevitably discount the performance of LRcon if we choose to work this way.

Here we propose to use the *union* of the residue pairs corresponding to the top λL predictions from each of the p component predictors. For a protein of length L , we denote the set of the top λL residue pairs returned by the k -th predictor by $R^k = \{r_1^k, r_2^k, \dots, r_{\lambda L}^k\}$, where $r_i^k = (r_{i,1}^k, r_{i,2}^k)$ represents a residue pair with $1 \leq r_{i,1}^k, r_{i,2}^k \leq L$. The residue pairs selected for this protein to train and test our LR-Predictor are then taken from the set

$$R^{LR} = \bigcup_{k=1}^p R^k \quad (6.4)$$

6.3 Results

In the following, the experimental results are evaluated on the top $0.1L$ and $0.2L$ predictions at sequence separations $6 \leq s < 12$, $12 \leq s < 24$ and $s \geq 24$.

6.3.1 Results on the CASP9 Dataset

Figure 6.3: Histogram of the accuracy, coverage and F-measure for the top $0.1L$ predictions of LRcon and other predictors on the CASP9 dataset D80. The predictor codes for the component predictors are directly taken from CASP9. Ave and Vot represent the averaging and voting schemes, respectively.

In order to estimate the regression coefficients of LR and to assess the performance of LRcon, we applied 10-fold cross-validation to the CASP9 dataset D80. For the top $0.1L$ predictions, Figure 6.3 shows the average accuracy, coverage, and F-measure of LRcon, its component predictors, and the averaging and voting schemes (refer to Section 6.2.4) for a comprehensive comparison. It is evident from the figure that the averaging and voting schemes could perform better than most component predictors, but never all in any cases. On the contrary, the LRcon is able to outperform all the component predictors and the averaging and voting schemes as well. In addition, we can see from the F-measures that the prediction

of long-range contact is much more challenging than the prediction of short-range and medium-range contacts. For the top $0.2L$ predictions, LRcon also outperforms all the component predictors and the averaging and voting schemes. The detailed results are presented in Figure 6.4.

Figure 6.4: Histogram of the accuracy, coverage and F-measure for the top $0.2L$ predictions of LRcon and other predictors on the CASP9 dataset D80.

We assess the statistical significance of the prediction differences between LRcon and each other predictor as follows. The test method is similar to what used in Chapter 4. First, 80% protein chains are selected at random from the original dataset to construct a (sub)dataset. This is repeated 100 times so as to obtain 100 different datasets. Then, we collected the prediction results of all the tested predictors from these 100 datasets. Finally, Shapiro-Wilk test is used to determine if the F-measure values of the 100 datasets are normally distributed. If they follow

normal distribution, the paired t -test is applied to assess their statistical significance on the F-measure differences; otherwise, the non-parametric Wilcoxon rank sum test is used. We summarized in Table 6.1 the experimental results tested on the CASP9 dataset.

Table 6.1: The results of the statistical significance tests for the F-measures of LRcon and other predictors on the D80 dataset. The ‘+’ / ‘-’ indicates that the method in a given column is significantly better/worse than the method in a given row with p -value < 0.001 , and ‘=’ means that the results are not shown statistically different. ‘A’, ‘V’, and ‘L’ in the table represent the averaging scheme, voting scheme, and LRcon, respectively.

Predictor	Top $0.1L$ predictions									Top $0.2L$ predictions								
	$6 \leq s < 12$			$12 \leq s < 24$			$s \geq 24$			$6 \leq s < 12$			$12 \leq s < 24$			$s \geq 24$		
	A	V	L	A	V	L	A	V	L	A	V	L	A	V	L	A	V	L
RR002	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
RR043	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
RR080	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
RR081	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
RR103	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
RR114	+	+	+	+	-	+	+	-	+	+	+	+	+	-	+	+	-	+
RR119	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
RR138	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
RR147	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
RR158	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
RR202	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
RR204	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
RR214	-	-	+	-	-	+	=	-	+	-	-	+	-	-	+	-	-	+
RR215	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
RR236	+	+	+	+	-	+	+	-	+	+	+	+	+	-	+	+	-	+
RR244	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
RR262	+	+	+	+	-	+	+	-	+	+	+	+	+	-	+	+	-	+
RR357	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
RR375	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
RR391	-	-	+	-	-	+	-	-	+	-	-	+	-	-	+	-	-	+
RR422	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
RR490	-	-	+	-	-	+	-	-	+	-	-	+	-	-	+	-	-	+
RR051	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
A	=	-	+	=	-	+	=	-	+	=	-	+	=	-	+	=	-	+
V	+	=	+	+	=	+	+	=	+	+	=	+	+	=	+	+	=	+

We can observe from the above tests that: (1) The averaging scheme appears to perform better than the voting scheme, (2) Neither the averaging nor voting scheme achieve a better prediction than all the component predictors (in particular, e.g.,

RR391 and RR490), and (3) LRcon outperforms all the other predictors, including the averaging and voting schemes.

6.3.2 A Good CASP9 Contact Prediction Example

A good CASP9 example that where LRcon outperforms all the other predictors is depicted in Figure 6.5. The true contact map and the predicted contact maps (top $0.2L$ predictions) at sequence separation $s \geq 24$ of five best CASP9 predictors and LRcon are shown in Figure 6.5 as well. From the figure, we can see that the five best CASP9 predictors correctly identify most of the contacts (at top $0.2L$ predictors) for this target. LRcon further improves the predictions of these predictors, yielding the highest accuracy (0.904), coverage (0.155), and F-measure (0.265).

6.3.3 Results on the D856 Dataset

Because the size of the D856 dataset is significantly larger than that of the D80 dataset, we adopt here a different way, rather than using 10-fold cross-validation, to evaluate the prediction results of LRcon as follows. First, the 856 protein chains in the D856 dataset are partitioned at random into a training and a test dataset of equal size (i.e., 428 protein chains in each). Note that the training and test datasets are independent each other, so no sequence in the test dataset has over 25% sequence similarity with any sequences in the training dataset. Second, three predictors, SVM-SEQ[190], NNcon[191], and FragHMMent[11], are used to make predictions on both the training and test datasets. Third, we use the predictions of these three predictors on the training dataset to estimate the regression coefficients of LR. Finally, the performance of LRcon is assessed on the test dataset.

The experimental results of LRcon and other predictors on the independent test dataset are listed in Tables 6.2 and 6.3. We can see from the tables that

Figure 6.5: A good contact prediction example. The top panel is a cartoon visualization of the 3D structure of the CASP9 target T0532. The lower panels are the true contact map and predicted contact maps at sequence separation $s \geq 24$. The blue points in the upper triangles represent the true contact map. The red points in the lower triangles represent the top $0.2L$ predictions of five CASP9 predictors and LRcon.

Table 6.2: Comparison of accuracies, coverage and F-measures on the independent test (sub)dataset of the D856 dataset for the top $0.1L$ predictions. The best results are shown in bold.

Predictor	$6 \leq s < 12$			$12 \leq s < 24$			$s \geq 24$		
	Acc	Cov	Fm	Acc	Cov	Fm	Acc	Cov	Fm
FragHMMent	.365	.116	.176	.306	.079	.126	.275	.026	.047
NNcon	.591	.187	.284	.455	.118	.187	.283	.026	.048
SVM-SEQ	.610	.193	.293	.483	.125	.199	.366	.034	.062
Ave	.529	.167	.254	.415	.108	.171	.365	.034	.062
Vot	.563	.178	.271	.443	.115	.182	.314	.029	.053
LRcon	.650	.206	.313	.531	.138	.218	.415	.039	.071

Table 6.3: Comparison of accuracies, coverage and F-measures on the independent test (sub)dataset of the D856 dataset for the top $0.2L$ predictions. The best results are shown in bold.

Predictor	$6 \leq s < 12$			$12 \leq s < 24$			$s \geq 24$		
	Acc	Cov	Fm	Acc	Cov	Fm	Acc	Cov	Fm
FragHMMent	.340	.219	.267	.272	.143	.188	.275	.052	.088
NNcon	.478	.308	.374	.376	.198	.260	.235	.045	.075
SVM-SEQ	.508	.327	.103	.407	.215	.281	.398	.323	.061
Ave	.377	.243	.295	.316	.167	.218	.288	.055	.092
Vot	.474	.305	.371	.371	.195	.256	.289	.055	.092
LRcon	.538	.346	.421	.443	.234	.306	.355	.067	.113

LRcon outperforms all the other predictors in terms of accuracy, coverage and F-measure. For example, LRcon achieves an average accuracy of 41.5% for the top $L/10$ long-range contact predictions, which is about 5% higher than its best-performed component predictor (i.e., SVM-SEQ). We also conduct a statistical significance test in the same way as we did earlier on the CASP9 dataset, and the results are shown in Table 6.4. It can be seen that the simple averaging and voting schemes perform better than NNcon and FragHMMent, but worse than SVM-SEQ. On the other hand, LRcon once again consistently outperforms all the other predictors, including the simple averaging and voting schemes.

Table 6.4: The results of the statistical significance tests for the F-measures of LRcon and other predictors on the independent test (sub)dataset of the D856 dataset. ‘A’, ‘V’, and ‘L’ in the table represent the averaging scheme, voting scheme, and LRcon, respectively.

Predictor	Top $0.1L$ predictions									Top $0.2L$ predictions								
	$6 \leq s < 12$			$12 \leq s < 24$			$s \geq 24$			$6 \leq s < 12$			$12 \leq s < 24$			$s \geq 24$		
	A	V	L	A	V	L	A	V	L	A	V	L	A	V	L	A	V	L
FragHMMent	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
NNcon	-	-	+	-	-	+	+	+	+	-	-	+	-	-	+	+	+	+
SVM-SEQ	-	-	+	-	-	+	-	-	+	-	-	+	-	-	+	-	-	+
A	=	+	+	=	+	+	=	-	+	=	+	+	=	+	+	=	+	+
V	-	=	+	-	=	+	+	=	+	-	=	+	-	=	+	-	=	+

6.4 Discussions

In this section, we discuss how the performance of LRcon is affected by the following three factors: the residue pair selection, the component predictors and the classification algorithm.

6.4.1 The Impact of Residue Pair Selection

For each protein chain, we have used formula (6.4) to select the residue pairs for training and testing LRcon. In order to demonstrate the effectiveness of this filtering process, we tested the performance of LRcon when all the residue pairs satisfying the sequence separation condition were used. Because it takes too much computing time and computer memory, we just tested the results of LRcon for the top $0.1L$ predictions at sequence separation $s \geq 24$ using the CASP9 dataset D80. In this case, the resulting average accuracy of LRcon decreases to 0.806, which is 0.026 lower than that obtained with the filtering process employed (see Figure 6.3). Therefore, it is necessary to train LRcon with a properly selected subset of residue pairs in order to achieve more accurate contact map prediction.

6.4.2 The Impact of Individual Predictors

The major reason of LRcon’s superior performance is believed that its component predictors can make complementary predictions to each other. We say two predictors M_1 and M_2 are complementary if their correct predictions (denoted respectively by TP_1 and TP_2) among the top λL predictions are not completely the same or one is not a subset of the other. We have the following two observations. First, the sizes of TP_1 and TP_2 should be as large as possible ($\leq \lambda L$) for LRcon to be accurate enough. Second, in order to improve over M_1 and M_2 by combining them, TP_1 and TP_2 should not be the same. Otherwise, we would not be able to make any improvement by combining them; instead, the predictions might even become worse. However, we have not developed a metric to quantitatively measure the complementarity between two predictors in this chapter.

We conduct the experiments on the CASP9 dataset D80 to further confirm the above observations as follows. As mentioned in Section 6.2.1, when we selected

component predictors, only “model 1” of each predictor was used in our previous experiments. For some participating groups there are several prediction models. For instance, the participating group RR114 in CASP9 have five prediction models for each target. We looked into these five models and found that their prediction results were in fact very similar, indicating that these models are not complementary to each other. When all models were used for each participating group in CASP9, we obtained 28 predictors in total. Using these predictors as component predictors of LRcon, we found that the predictions became worse than before. For example, the accuracy value decreased slightly from 0.832 to 0.827 for the case of the top $0.1L$ predictions at sequence separation $s \geq 24$. Therefore, one possible way to further improve the performance of LRcon is to select just those accurate while complementary predictors as component predictors. To this end, some metrics might be needed to quantify the complementary property among individual predictors.

6.4.3 The Impact of Classification Algorithm

Besides the logistic regression, the following four classification algorithms are experimented to explore the impact of a classification algorithm on the performance of LRcon. They are random forest (RF), k -nearest neighbor (k -NN), Naive Bayes (NB), and J48. The details about these algorithms can be obtained from Weka [199]. Because the running time of the SVM algorithm is much more than others and we can not get the results from it in a reasonable time, we do not discuss this algorithm here. We chose the algorithm implementations in Weka in our subsequent experiments. Except for k -NN, where k was set to be 10 to produce probability estimates, the parameters for RF, NB, and J48 were all set to be their respective default values.

The experimental results of LR and the other four algorithms on the datasets

D80 and D856 are presented in Tables 6.5, 6.6, 6.7, and 6.8. We can see that the accuracies, coverage and F-measures of LR are consistently higher than those of any other algorithm on the D80 dataset. When tested on the the D856 dataset, LR and NB achieved comparable results and better than the other three algorithms. These observations lead to our selection of LR as the classification algorithm in this chapter.

Table 6.5: Comparison of accuracies, coverage and F-measures of five classification algorithms on the CASP9 dataset D80 for the top $0.1L$ predictions. The best results are shown in bold.

Algorithm	$6 \leq s < 12$			$12 \leq s < 24$			$s \geq 24$		
	Acc	Cov	Fm	Acc	Cov	Fm	Acc	Cov	Fm
RF	.824	.283	.422	.810	.219	.345	.816	.069	.127
NB	.769	.264	.393	.756	.204	.322	.780	.066	.122
J48	.736	.253	.377	.724	.196	.308	.741	.061	.113
KNN	.797	.274	.407	.777	.210	.330	.799	.067	.124
LR	.839	.288	.429	.822	.222	.350	.832	.070	.129

Table 6.6: Comparison of accuracies, coverage and F-measures of five classification algorithms on the CASP9 dataset D80 for the top $0.2L$ predictions. The best results are shown in bold.

Algorithm	$6 \leq s < 12$			$12 \leq s < 24$			$s \geq 24$		
	Acc	Cov	Fm	Acc	Cov	Fm	Acc	Cov	Fm
RF	.689	.479	.565	.721	.394	.510	.786	.134	.228
NB	.654	.454	.536	.656	.359	.464	.744	.126	.216
J48	.658	.458	.540	.680	.372	.481	.741	.126	.215
KNN	.685	.476	.562	.707	.386	.500	.773	.131	.225
LR	.710	.493	.582	.727	.398	.514	.799	.136	.232

Table 6.7: Comparison of accuracies, coverage and F-measures of five classification algorithms on the independent test (sub)dataset of the D856 dataset for the top $0.1L$ predictions. The best results are shown in bold.

Algorithm	$6 \leq s < 12$			$12 \leq s < 24$			$s \geq 24$		
	Acc	Cov	Fm	Acc	Cov	Fm	Acc	Cov	Fm
RF	.594	.188	.286	.464	.120	.191	.352	.033	.060
NB	.650	.206	.313	.531	.138	.218	.406	.038	.069
J48	.621	.197	.299	.492	.127	.202	.394	.037	.067
KNN	.623	.197	.299	.488	.126	.201	.383	.036	.065
LR	.650	.206	.313	.531	.138	.218	.415	.039	.071

Table 6.8: Comparison of accuracies, coverage and F-measures of five classification algorithms on the independent test (sub)dataset of the D856 dataset for the top $0.2L$ predictions. The best results are shown in bold.

Algorithm	$6 \leq s < 12$			$12 \leq s < 24$			$s \geq 24$		
	Acc	Cov	Fm	Acc	Cov	Fm	Acc	Cov	Fm
RF	.485	.313	.380	.388	.205	.268	.312	.059	.099
NB	.540	.348	.423	.440	.232	.304	.349	.066	.111
J48	.506	.326	.396	.414	.218	.286	.323	.061	.103
KNN	.508	.327	.398	.404	.213	.279	.325	.062	.104
LR	.538	.346	.421	.443	.234	.306	.355	.067	.113

6.5 Conclusions

Prediction of protein contact map plays an important role in the prediction of protein 3D structure. However, the accuracy of current computational methods is rather low. In this chapter, we explored the possibility of improving the accuracy of an individual protein contact predictors by using a consensus approach.

Under the machine learning framework, an improved sequence-based protein contact map prediction method, named LRcon, has been developed based on logistic regression. LRcon is built upon the prediction results from its component contact map predictors. For each residue pair, the probability estimates of the component predictors are used to form a feature vector, which is then fed into the logistic regression-based algorithm to make a consensus prediction. Logistic regression models are trained and assessed under the machine learning framework by using independent training and test datasets. Experimental results on the CASP9 dataset and another large-sized dataset containing 856 protein chains show that LRcon can make statistically significant improvements over its component predictors and the simple averaging and voting schemes as well. We believe that these improvements made by LRcon are mainly attributed to the application of a consensus approach to the complementary predictors and the logistic regression analysis under the machine learning framework.

Chapter 7

Conclusion

7.1 Summary

In the post-genomic era, the gap between the number of sequence-known proteins and the number of structure-known proteins is widening rapidly. Developing efficient computer-based algorithms for predicting 3D structures from sequences promises to bridge this gap to a large extent. In this dissertation, we have presented several machine learning algorithms that aimed at solving the following four problems: (1) protein structural class prediction, (2) taxonomy-based fold recognition, (3) hierarchical classification of GPCRs, and (4) protein contact map prediction. In the following, I summarize for each problem how the three aims mentioned in Chapter 1 are fulfilled.

- (1) For protein structural class prediction, we have used CGR and RQA to extract features directly from the amino acid sequences and the predicted secondary structure. They are relatively simple and very fast. We compared the proposed method with the state-of-the-art methods SCPRED and MODAS, showing that our method has higher or comparable predictive accuracy. For public use, a web server was provided at http://www1.spms.ntu.edu.sg/~chenxin/RKS_

PPSC/.

- (2) For taxonomy-based fold recognition, we extract a comprehensive set of 137 features from the PSI-BLAST and PSIPRED profiles. Because this method relies on the alignment against the NCBI's Non Redundant database, it runs relatively slow. The majority of running time is spent in the generation of the profiles. We have comprehensively compared our method with ten taxonomic fold prediction methods and nine threading methods, showing our method performs better than other taxonomic methods and is comparable with threading methods. For public use, a web server was provided at <http://www1.spms.ntu.edu.sg/~chenxin/TAXFOLD/>.
- (3) For hierarchical classification of GPCRs, a total of 1497 features are generated directly from the amino acid sequence, followed by PCA to reduce the dimensionality. Because our method is an alignment-free method and a very simple yet powerful classifier (intimate sorting) is used, it runs very fast and can make predictions for one protein in less than one second. Comparison with other methods shows the proposed method perform consistently well on all the tested datasets. For public use, a web server was provided at http://www1.spms.ntu.edu.sg/~chenxin/PCA_GPCR/.
- (4) For protein contact map prediction, because our consensus method explores the outputs from several individual predictors with logistic regression, its running time mainly depends on its component predictors. Once the results from other predictors are in hand, our method can make consensus prediction for one protein in a few seconds. Comparison shows that our method is statistically better than its component predictors. Because some component predictors (e.g., SVM-SEQ) often take several hours or even longer to process a protein of about 300 amino acids, no web server is provided for our consensus method.

7.2 Technical Contributions

The technical contributions of this dissertation include:

- Representation of each amino acid sequence by two time series based on CGR.
- Application of RQA to the time series for feature extraction, which can avoid loss of sequence-order information to some extent.
- Application of CGR and RQA techniques to the predicted secondary structure.
- Development of the K -string based information entropy of the predicted secondary structure.
- Segment-based analysis of the predicted secondary structure to characterize the spatial arrangements of α helices and β strands and differentiate between the $\alpha + \beta$ and α/β classes.
- Development of a web server, RKS-PPSC, for protein structural class prediction (http://www1.spms.ntu.edu.sg/~chenxin/RKS_PPSC/).
- Creation of five datasets (RDD, EDD, F95, F194, F710) for the assessment of fold prediction (see Appendix A).
- Dealing with the negative elements of PSSMs by applying an inverse algorithmic operation rather than simply replacing all of them by zero.
- Extraction of features from the consensus sequences (which retains richer sequence evolution information) constructed from PSSMs rather than from their respective amino acid sequences.

- Division of the PSI-BLAST profiles into several non-overlapping fragments which allows for the depiction of local characteristics (e.g., at N-terminus and C-terminus).
- Application of the autocovariance transformation to the probability sequences of PSIPRED profiles for feature extraction.
- Development of a web server, TAXFOLD, for protein fold prediction (<http://www1.spms.ntu.edu.sg/~chenxin/TAXFOLD/>).
- Comprehensive comparison of TAXFOLD with ten taxonomic fold prediction methods and nine threading methods.
- Creation of a dataset (*GDFL*) at five levels for assessment of hierarchical classification of GPCRs (see Appendix A).
- Development of a hierarchical classifier to comprehensively predict GPCRs at five levels, till the subtype level of the GPCRDB information system.
- Development of a web server, PCA-GPCR, for GPCRs classification (http://www1.spms.ntu.edu.sg/~chenxin/PCA_GPCR/).
- Creation of a large non-redundant dataset (D856) for the assessment of protein contact map prediction (see Appendix A).
- Development of a logistic regression based consensus predictor for protein contact map prediction.

7.3 Limitations and Future Work

The studies performed in this dissertation contribute to the improved prediction of several structural attributes of proteins. The limitations and several other related research topics are summarized as follows.

1. For protein structural class prediction, only four major structural classes (α , β , $\alpha+\beta$, α/β) were considered in this thesis. As mentioned in Chapter 1, there are other classes in SCOP but the majority of existing methods make predictions for only four major structural classes. To the best of our knowledge, there are only three methods working for seven classes [42, 57, 200]. However, the prediction accuracies for some classes are relatively low. For example, only 58% prediction accuracy is reported for the membrane proteins class [57], which is an important class of proteins acting as the targets of many drugs [160]. The major reason shall be due to the very small number of samples in these classes. With the PSI:Biology phase and the technical advance in structure determination, such as X-ray crystallography and NMR spectroscopy, more and more protein structures will become available in the PDB library and the SCOP database. It then allows us to test whether the prediction methods proposed in this dissertation can still work equally well on a large number of structural classes, or instead a totally new method is desired.
2. For taxonomy-based protein fold recognition, there are more than 1000 different folds in SCOP (V1.75) and the number of folds considered in this thesis is at most 710. Satisfactory results (about 80% overall accuracy) were achieved when the number of folds considered is less than 200. However when we consider 710 folds, the accuracy of our method decreases to about 68.1%, and even worse results are expected if all the folds in SCOP are considered. Therefore, it

is very challenging to develop a taxonomic fold recognition method that can make highly accurate prediction for a large number of folds being considered. In addition, it is interesting to see if fold recognition can be enhanced by a mixed approach of taxonomic fold recognition and template-based threading. Some related research is in progress.

3. For hierarchical classification of GPCRs, we can see that the accuracies of PCA-GPCR in classifying some individual categories are still very low, which is largely due to the small number of samples in the respective categories. It is anticipated that these accuracies could be improved once more data from the updated version of the GPCRDB database is used to train a reliable classifier. Besides the classification, it is very interesting to investigate other GPCR-related research, such as improving GPCR modeling [82] and studying the characteristics of those pharmaceutically important human GPCRs (e.g., Adenosine, Adrenergic, Chemokine, Dopamine, Histamine, and Muscarinic acetylcholine) [75], in particular considering the pharmaceutical value of GPCRs.
4. For protein contact map prediction, the work done in this thesis may serve as a good starting point to construct consensus predictors. To do so, some advanced method may be needed to quantitatively measure the complementarity of individual predictors in order to choose those highly complementary predictors for making a consensus prediction. On the other hand, a more advanced classification algorithms than logistic regression, such as support vector machine and neural network, may be employed to further improve the accuracy in contact map prediction.
5. Lastly, it is interesting to see how well the ideas and methodologies in this thesis can be applied to predict many other protein attributes: protein subcellular localization [86], membrane protein types [177], nuclear receptors [178], residue

depth [201], protease types [202], and so on.

Bibliography

- [1] C. Anfinsen. Principles that govern the folding of protein chains. *Science*, 181:223–230, 1973.
- [2] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, and P.E. Bourne. The Protein Data Bank. *Nucleic Acids Res.*, 28:235–242, 2000.
- [3] D.J. Rigden. *From Protein Structure to Function with Bioinformatics*. Springer Verlag, 2009.
- [4] L.J. McGuffin. *Protein fold recognition and threading, in Computational Structural Biology: Methods and Applications*. Schwede & Peitsch, Eds, World Scientific, Singapore, 2008.
- [5] B. Rost and C. Sander. Improved prediction of protein secondary structure by use of sequence profiles and neural networks. *Proc. Natl. Acad. Sci. USA*, 90:7558–7562, 1993.
- [6] D.T. Jones. Protein secondary structure prediction based on position specific scoring matrices. *J. Mol. Biol.*, 292:195–202, 1999.
- [7] B. Rost and C. Sander. Conservation and prediction of solvent accessibility in protein families. *Proteins*, 20:216–226, 1994.
- [8] H. Zhang, T. Zhang, K. Chen, K.D. Kedarisetti, M.J. Mizianty, W. Bao, Q. Stach, and L. Kurgan. Critical assessment of high-throughput standalone methods for secondary structure prediction. *Brief. Bioinform.*, 12:672–688, 2011.

- [9] O. Olmea and A. Valencia. Improving contact predictions by the combination of correlated mutations and other sources of sequence information. *Fold. Des.*, 2:S25–S32, 1997.
- [10] P.J. Kundrotas and E.G. Alexov. Predicting residue contacts using pragmatic correlated mutations method: reducing the false positives. *BMC Bioinformatics*, 7:503, 2006.
- [11] P. Björkholm, P. Daniluk, A. Kryshchuk, K. Fidelis, R. Andersson, and T.R. Hvidsten. Using multi-data hidden Markov models trained on local neighborhoods of protein structure to predict residue-residue contacts. *Bioinformatics*, 25:1264–1270, 2009.
- [12] J. Cheng and P. Baldi. Improved residue contact prediction using support vector machines and a large feature set. *BMC Bioinformatics*, 8:113, 2007.
- [13] I. Ezkurdia, O. Graña, J.M.G. Izarzugaza, and M.L. Tress. Assessment of domain boundary predictions and the prediction of intramolecular contacts in CASP8. *Proteins*, 77:196–209, 2009.
- [14] M. Punta and B. Rost. PROFcon: novel prediction of long-range contacts. *Bioinformatics*, 21:2960–2968, 2005.
- [15] S. Lee and J. Skolnick. Benchmarking of TASSER_2.0: an improved protein structure prediction algorithm with more accurate predicted contact restraints. *Biophys. J.*, 95:1956–1964, 2008.
- [16] A. Roy, A. Kucukural, and Y. Zhang. I-TASSER: a unified platform for automated protein structure and function prediction. *Nat. Protoc.*, 5:725–738, 2010.
- [17] Y. Zhang. Protein structure prediction: when is it useful? *Curr. Opin. Struct. Biol.*, 19:145–155, 2009.
- [18] A. Sali and T.L. Blundell. Comparative protein modelling by satisfaction of spatial restraints. *J. Mol. Biol.*, 234:779–815, 1993.

- [19] T. Schwede, J. Kopp, N. Guex, and M.C. Peitsch. Swiss-model: an automated protein homology-modeling server. *Nucleic Acids Res.*, 31:3381–3385, 2003.
- [20] J.U. Bowie, R. Luthy, and D. Eisenberg. A method to identify protein sequences that fold into a known three-dimensional structure. *Science*, 253:164–170, 1991.
- [21] D.T. Jones, W.R. Taylor, and J.M. Thornton. A new approach to protein fold recognition. *Nature*, 358:86–89, 1992.
- [22] K.T. Simons, C. Kooperberg, E. Huang, and D. Baker. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and bayesian scoring functions1. *J. Mol. Biol.*, 268:209–225, 1997.
- [23] J.N.D. Battey, J. Kopp, L. Bordoli, R.J. Read, N.D. Clarke, and T. Schwede. Automated server predictions in CASP7. *Proteins*, 69:68–82, 2007.
- [24] D. Cozzetto, A. Kryzhtafovych, K. Fidelis, J. Moult, B. Rost, and A. Tramontano. Evaluation of template-based models in CASP8 with standard measures. *Proteins*, 77:18–28, 2009.
- [25] L. Kinch, S.Y. Shi, Q. Cong, H. Cheng, Y. Liao, and N.V. Grishin. CASP9 assessment of free modeling target predictions. *Proteins*, 79:59–73, 2011.
- [26] K.C. Chou. Some remarks on protein attribute prediction and pseudo amino acid composition. *J. Theor. Biol.*, 273:236–247, 2011.
- [27] A. Murzin, S. Brenner, T. Hubbard, and C. Chothia. SCOP: a structural classification of protein database for the investigation of sequence and structures. *J. Mol. Biol.*, 357:536–540, 1995.
- [28] C. Orengo, A. Michie, D. Jones, M. Swindells, and J. Thornton. CATH: a hierarchic classification of protein domain structures. *Structure*, 5:1093–1108, 1997.
- [29] B. Xue, E. Faraggi, and Y. Zhou. Predicting residue-residue contact maps by a two-layer, integrated neural-network method. *Proteins*, 76:176–183, 2009.

- [30] I. Bahar, A.R. Atilgan, R.L. Jernigan, and B. Erman. Understanding the recognition of protein structural classes by amino acid composition. *Proteins*, 29:1172–1185, 1997.
- [31] I.B. Kuznetsov, Z. Gou, R. Li, and S. Hwang. Using evolutionary and structural information to predict DNA-binding sites on DNA-binding proteins. *Proteins*, 64:19–27, 2006.
- [32] Z.D. Zhang, Z.R. Sun, and C.T. Zhang. A new approach to predict the helix/strand content of globular proteins. *J. Theor. Biol.*, 208:65–78, 2001.
- [33] M.M. Gromiha and M. Suwa. A simple statistical method for discriminating outer membrane proteins with better accuracy. *Bioinformatics*, 21:961–968, 2005.
- [34] M.M. Gromiha. A statistical model for predicting protein folding rates from amino acid sequence with structural class information. *J. Chem. Inf. Model.*, 45:494–501, 2005.
- [35] K.C. Chou and C.T. Zhang. Prediction of protein structural class. *Crit. Rev. Biochem. Mol. Biol.*, 30:275–349, 1995.
- [36] K.C. Chou. A novel approach to predicting protein structural classes in a (20-1)-D amino acid composition space. *Proteins*, 21:319–344, 1995.
- [37] Z.X. Wang and Z. Yuan. How good is the prediction of protein structural class by the component-coupled method? *Proteins*, 38:165–175, 2000.
- [38] Y.D. Cai and G.P. Zhou. Prediction of protein structural classes by neural network. *Biochimie*, 82:783–785, 2000.
- [39] Y.D. Cai, X.J. Liu, X. Xu, and G.P. Zhou. Support vector machines for predicting protein structural class. *BMC Bioinformatics*, 2:3, 2001.
- [40] R. Luo, Z. Feng, and J. Liu. Prediction of protein structural class by amino acid and polypeptide composition. *Eur. J. Biochem.*, 269:4219–4225, 2002.

- [41] Y.D. Cai, X.J. Liu, X.B. Xu, and K.C. Chou. Support vector machines for prediction of protein domain structural class. *J. Theor. Biol.*, 221:115–120, 2003.
- [42] K.C. Chou and Y.D. Cai. Predicting protein structural class by functional domain composition. *Biochem. Biophys. Res. Commun.*, 321:1007–1009, 2004.
- [43] H.B. Shen, J. Yang, X.J. Liu, and K.C. Chou. Using supervised fuzzy clustering to predict protein structural classes. *Biochem. Biophys. Res. Commun.*, 334:577–581, 2005.
- [44] L.A. Kurgan and L. Homaeian. Prediction of structural classes for protein sequences and domains—impact of prediction algorithms, sequence representation and homology, and test procedures on accuracy. *Pattern Recog.*, 39:2323–2343, 2006.
- [45] K.D. Kedarisetti, L.A. Kurgan, and S. Dick. Classifier ensembles for protein structural class prediction with varying homology. *Biochem. Biophys. Res. Commun.*, 348:981–988, 2006.
- [46] C. Chen, Y.X. Tian, X.Y. Zou, P.X. Cai, and J.Y. Mo. Using pseudo-amino acid composition and support vector machine to predict protein structural class. *J. Theor. Biol.*, 243:444–448, 2006.
- [47] Y.D. Cai, K.Y. Feng, W.C. Lu, and K.C. Chou. Using LogitBoost classifier to predict protein structural classes. *J. Theor. Biol.*, 238:172–176, 2006.
- [48] Y. Cao, S. Liu, L. Zhang, J. Qin, J. Wang, and K. Tang. Prediction of protein structural class with Rough Sets. *BMC Bioinformatics*, 7:20, 2006.
- [49] L.A. Kurgan and K. Chen. Prediction of protein structural class for the twilight zone sequences. *Biochem. Biophys. Res. Commun.*, 357:453–460, 2007.
- [50] H. Lin and Q.Z. Li. Using pseudo amino acid composition to predict protein structural class: approached by incorporating 400 dipeptide components. *J. Comput. Chem.*, 28:1463–1466, 2007.

- [51] L.A. Kurgan, K. Cios, and K. Chen. SCPRED: Accurate prediction of protein structural class for sequences of twilight-zone similarity with predicting sequences. *BMC Bioinformatics*, 9:226, 2008.
- [52] L.A. Kurgan, T. Zhang, H. Zhang, S. Shen, and J. Ruan. Secondary structure-based assignment of the protein structural classes. *Amino Acids*, 35:551–564, 2008.
- [53] T.L. Zhang, Y.S. Ding, and K.C. Chou. Prediction protein structural classes with pseudo-amino acid composition: approximate entropy and hydrophobicity pattern. *J. Theor. Biol.*, 250:186–193, 2008.
- [54] X. Xiao, P. Wang, and K.C. Chou. Predicting protein structural classes with pseudo amino acid composition: an approach using geometric moments of cellular automaton image. *J. Theor. Biol.*, 254:691–696, 2008.
- [55] K. Chen, L.A. Kurgan, and J. Ruan. Prediction of protein structural class using novel evolutionary collocation-based sequence representation. *J. Comput. Chem.*, 29:1596–1604, 2008.
- [56] Z.C. Li, X.B. Zhou, Z. Dai, and X.Y. Zou. Prediction of protein structural classes by chou’s pseudo amino acid composition: approached using continuous wavelet transform and principal component analysis. *Amino Acids*, 37:415–425, 2009.
- [57] M.J. Mizianty and L.A. Kurgan. Modular prediction of protein structural classes from sequences of twilight-zone identity with predicting sequences. *BMC Bioinformatics*, 10:414, 2009.
- [58] T. Liu, X. Zheng, and J. Wang. Prediction of protein structural class for low-similarity sequences using support vector machine and PSI-BLAST profile. *Biochimie*, 92:1330–1334, 2010.
- [59] T. Liu, X. Geng, X. Zheng, R. Li, and J. Wang. Accurate prediction of protein structural class using auto covariance transformation of PSI-BLAST profiles. *Amino Acids*, In press, 2011.

- [60] J.Y. Yang, Z.L. Peng, Z.G. Yu, R.J. Zhang, V. Anh, and D. Wang. Prediction of protein structural classes by recurrence quantification analysis based on chaos game representation. *J. Theor. Biol.*, 257:618–626, 2009.
- [61] J.Y. Yang, Z.L. Peng, and X. Chen. Prediction of protein structural classes for low-homology sequences based on predicted secondary structure. *BMC Bioinformatics*, 11:S9, 2010.
- [62] M. Levitt and C. Chothia. Structural patterns in globular proteins. *Nature*, 261:552–558, 1976.
- [63] C. Chothia. One thousand families for the molecular biologist. *Nature*, 357:543–544, 1992.
- [64] Y.I. Wolf, N.V. Grishin, and E.V. Koonin. Estimating the number of protein folds and families from complete genome data. *J. Mol. Biol.*, 299:897–905, 2000.
- [65] Y. Zhang and J. Skolnick. The protein structure prediction problem could be solved using the current PDB library. *Proc. Natl. Acad. Sci. USA*, 102:1029–1034, 2005.
- [66] Q. Dong, S. Zhou, and J. Guan. A new taxonomy-based protein fold recognition approach based on autocross-covariance transformation. *Bioinformatics*, 25:2655–2662, 2009.
- [67] J.Y. Yang and X. Chen. Improving taxonomy-based protein fold recognition by using global and local features. *Proteins*, 79:2053–2064, 2011.
- [68] K. Chen and L.A. Kurgan. PFRES: protein fold classification by using evolutionary information and predicted secondary structure. *Bioinformatics*, 23:2843–2850, 2007.
- [69] K. Chen, W. Stach, L. Homaeian, and L. Kurgan. iFC²: an integrated web-server for improved prediction of protein structural class, fold type, and secondary structure content. *Amino Acids*, 40:963–973, 2011.

- [70] T.E. Hébert and M. Bouvier. Structural and functional aspects of G-protein-coupled receptor oligomerization. *Biochem. Cell Biol.*, 76:1–11, 1998.
- [71] F. Horn, J. Weare, M.W. Beukers, S. Hörsch, A. Bairoch, W. Chen, O. Edwardsen, F. Campagne, and G. Vriend. GPCRDB: an information system for G-protein-coupled receptors. *Nucleic Acids Res.*, 26:275–279, 1998.
- [72] V. Cherezov, D.M. Rosenbaum, M.A. Hanson, S.G. Rasmussen, F.S. Thian, T.S. Kobilka, H.J. Choi, P. Kuhn, W.I. Weis, B.K. Kobilka, and R.C. Stevens. High-resolution crystal structure of an engineered human β_2 -adrenergic G protein-coupled receptor. *Science*, 318:1258–1265, 2007.
- [73] J. Drews. Drug discovery: a historical perspective. *Science*, 287:1960–1964, 2000.
- [74] D.R. Flower. Modelling G-protein-coupled receptors for drug design. *Biochim. Biophys. Acta*, 1422:207–234, 1999.
- [75] J.P. Overington, B. Al-Lazikani, and A.L. Hopkins. How many drug targets are there? *Nat. Rev. Drug Discov.*, 5:993–996, 2006.
- [76] K. Palczewski, T. Kumasaka, T. Hori, C.A. Behnke, H. Motoshima, B.A. Fox, I. Le Trong, D.C. Teller, T. Okada, R.E. Stenkamp, M. Yamamoto, and M. Miyano. Crystal structure of rhodopsin: a G-protein-coupled receptor. *Science*, 289:739–745, 2000.
- [77] K. Lundstrom. Structural biology of G-protein-coupled receptors. *Bioorg. Med. Chem. Lett.*, 15:3654–3657, 2005.
- [78] K. Lundstrom and M. Chiu. *G-Protein-Coupled Receptors in Drug Discovery*. Taylor & Francis, 2005.
- [79] V.P. Jaakola, M.T. Griffith, M.A. Hanson, V. Cherezov, E.Y. Chien, J.R. Lane, A.P. Ijzerman, and R.C. Stevens. The 2.6 angstrom crystal structure of a human A2A adenosine receptor bound to an antagonist. *Science*, 322:1211–1217, 2008.

- [80] B. Wu, E.Y. Chien, C.D. Mol, G. Fenalti, W. Liu, V. Katritch, R. Abagyan, A. Brooun, P. Wells, F.C. Bi, D.J. Hamel, P. Kuhn, T.M. Handel, V. Cherezov, and R.C. Stevens. Structures of the CXCR4 chemokine GPCR with small-molecule and cyclic peptide antagonists. *Science*, 330:1066–1071, 2010.
- [81] E.Y. Chien, W. Liu, Q. Zhao, V. Katritch, G.W. Han, M.A. Hanson, L. Shi, A.H. Newman, J.A. Javitch, V. Cherezov, and R.C. Stevens. Structure of the human dopamine D3 receptor in complex with a D2/D3 selective antagonist. *Science*, 330:1091–1095, 2010.
- [82] Y. Zhang, M.E. Devries, and J. Skolnick. Structure modeling of all identified G protein-coupled receptors in the human genome. *PLoS Comput. Biol.*, 2:e13, 2006.
- [83] W.Z. Lin, X. Xiao, and K.C. Chou. GPCR-GIA: a web-server for identifying G-protein-coupled receptors and their families with grey incidence analysis. *Protein Eng.*, 22:699–705, 2009.
- [84] X. Xiao, P. Wang, and K.C. Chou. GPCR-CA: A cellular automaton image approach for predicting G-protein-coupled receptor functional classes. *J. Comput. Chem.*, 30:1413–1423, 2009.
- [85] X. Xiao, S.H. Shao, Z.D. Huang, and K.C. Chou. Using pseudo amino acid composition to predict protein structural classes: approached with complexity measure factor. *J. Comput. Chem.*, 27:478–482, 2006.
- [86] K.C. Chou. Prediction of protein subcellar locations by incorporating quasi-sequence-order effect. *Biochem. Biophys. Res. Commun.*, 278:477–483, 2000.
- [87] Z.L. Peng, J.Y. Yang, and X. Chen. An improved classification of G-protein-coupled receptors using sequence-derived features. *BMC Bioinformatics*, 11:420, 2010.
- [88] B. Vroiling, M. Sanders, C. Baakman, A. Borrmann, S. Verhoeven, J. Klomp, L. Oliveira, J. de Vlieg, and G. Vriend. GPCRDB: information system for G protein-coupled receptors. *Nucleic Acids Res.*, 39:D309–D319, 2011.

- [89] M. Vassura, L. Margara, P. Di Lena, F. Medri, P. Fariselli, and R. Casadio. Reconstruction of 3D structures from protein contact maps. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, 5:357–367, 2008.
- [90] L. Holm and C. Sander. Mapping the protein universe. *Science*, 273:595–603, 1996.
- [91] Z. Wang, A.N. Tegge, and J. Cheng. Evaluating the absolute quality of a single protein model using structural features and support vector machines. *Proteins*, 75:638–647, 2009.
- [92] M.L. Tress and A. Valencia. Predicted residue-residue contacts can help the scoring of 3D models. *Proteins*, 78:1980–1991, 2010.
- [93] A. Lo, Y.Y. Chiu, E.A. Rødland, P.C. Lyu, T.Y. Sung, and W.L. Hsu. Predicting helix-helix interactions from residue contacts in membrane proteins. *Bioinformatics*, 25:996–1003, 2009.
- [94] T. Nugent and D.T. Jones. Predicting transmembrane helix packing arrangements using residue contacts and a force-directed algorithm. *PLoS Comput. Biol.*, 6:e1000714, 2010.
- [95] G.N. Lin, Z. Wang, D. Xu, and J. Cheng. SeqRate: sequence-based protein folding type classification and rates prediction. *BMC Bioinformatics*, 11:S1, 2010.
- [96] A.M. Lesk. CASP2: report on ab initio predictions. *Proteins*, Suppl. 1:151–166, 1997.
- [97] J. Cheng and P. Baldi. A machine learning information retrieval approach to protein fold recognition. *Bioinformatics*, 22:1456–1463, 2006.
- [98] C.W. Cheng, E. Su, J.K. Hwang, T.Y. Sung, and W.L. Hsu. Predicting RNA-binding sites of proteins using support vector machines and evolutionary information. *BMC Bioinformatics*, 9:S6, 2008.

- [99] G. Schweikert, A. Zien, G. Zeller, J. Behr, C. Dieterich, C.S. Ong, P. Philips, F. De Bona, L. Hartmann, A. Bohlen, N. Krüger, S. Sonnenburg, and G. Rätsch. mGene: accurate SVM-based gene finding with an application to nematode genomes. *Genome Res.*, 19:2133–2143, 2009.
- [100] T. Abeel, Y. Saeys, E. Bonnet, P. Rouzé, and Y. Van de Peer. Generic eukaryotic core promoter prediction using structural features of DNA. *Genome Res.*, 18:310–323, 2008.
- [101] J.Y. Yang, Y. Zhou, Z.G. Yu, V. Anh, and L.Q. Zhou. Human Pol II promoter recognition based on primary sequences and free energy of dinucleotides. *BMC Bioinformatics*, 9:11, 2008.
- [102] E. Bindewald and B.A. Shapiro. RNA secondary structure prediction from sequence alignments using a network of k-nearest neighbor classifiers. *RNA*, 11:342–352, 2006.
- [103] S.F. Altschul, T.L. Madden, A.A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D.J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, 25:3389–3402, 1997.
- [104] K.C. Chou. Prediction of protein cellular attributes using pseudo amino acid composition. *Proteins*, 43:246–255, 2001.
- [105] V.N. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, New York, 1995.
- [106] A. Fiser, G.E. Tusnády, and I. Simon. Chaos game representation of protein structures. *J. Mol. Graph.*, 12:302–304, 1994.
- [107] A. Giuliani, R. Benigni, J.P. Zbilut, C.L.Jr Webber, P. Sirabella, and A. Colosimo. Nonlinear signal analysis methods in the elucidation of protein sequence-structure relationships. *Chem. Rev.*, 102:1471–1491, 2002.
- [108] P. Deschavanne and P. Tufféry. Exploring an alignment free approach for protein classification and structural class prediction. *Biochimie*, 90:615–625, 2008.

- [109] S. Basu, A. Pan, C. Dutta, and J. Das. Chaos game representation of proteins. *J. Mol. Graphics*, 15:279–289, 1997.
- [110] Z.G. Yu, V. Anh, and K.S. Lau. Chaos game representation of protein sequences based on the detailed HP model and their multifractal and correlation analyses. *J. Theor. Biol.*, 226:341–348, 2004.
- [111] J.Y. Yang, Z.G. Yu, and V. Anh. Clustering structures of large proteins using multifractal analyses based on a 6-letter model and hydrophobicity scale of amino acids. *Chaos, Solitons and Fractals*, 40:607–620, 2009.
- [112] H.J. Jeffrey. Chaos game representation of gene structure. *Nucleic Acids Res.*, 18:2163–2170, 1990.
- [113] J.Y. Yang, Z.G. Yu, and V. Anh. Protein structure classification based on chaos game representation and multifractal analysis. In *Proceeding of the 4th International Conference on Natural Computation*, pages 665–669, 2008.
- [114] J.P. Eckmann, S.O. Kamphorst, and D. Ruelle. Recurrence plots of dynamical systems. *Europhys. Lett.*, 4:973–977, 1987.
- [115] M.A. Riley and G.C. van Orden. Tutorials in contemporary nonlinear methods for the behavioral sciences, Retrieved March 1, 2005, from <http://www.nsf.gov/sbe/bcs/pac/nmbs/nmbs.jsp>.
- [116] J.P. Zbilut and C.L.Jr Webber. Embeddings and delays as derived from quantification of recurrence plots. *Phys. Lett. A*, 171:199–203, 1992.
- [117] C.L.Jr Webber and J.P. Zbilut. Dynamical assessment of physiological systems and states using recurrence plot strategies. *J. Appl. Physiol.*, 76:965–973, 1994.
- [118] A. Giuliani and M. Tomasi. Recurrence quantification analysis reveals interaction partners in paramyxoviridae envelope glycoproteins. *Proteins*, 46:171–176, 2002.
- [119] J.M. Zaldívar, F. Strozzi, S. Dueri, D. Marinov, and J.P. Zbilut. Characterization of regime shifts in environmental time series with recurrence quantification analysis. *Ecol. Model.*, 210:58–70, 2008.

- [120] A. Giuliani, P. Sirabella, R. Benigni, and A. Colosimo. Mapping protein sequence spaces by recurrence: a case study on chimeric structures. *Protein Eng.*, 13:671–678, 2000.
- [121] C. Manetti, M.A. Ceruso, A. Giuliani, C.L.Jr Webber, and J.P. Zbilut. Recurrence quantification analysis as a tool for the characterization of molecular dynamics simulations. *Phys. Rev. E*, 59:992–998, 1999.
- [122] C.L.Jr Webber, A. Giuliani, J.P. Zbilut, and A. Colosimo. Elucidating protein secondary structures using alpha-carbon recurrence quantifications. *Proteins*, 3:292–303, 2001.
- [123] J.P. Zbilut, J.C. Mitchell, A. Giuliani, A. Colosimo, N. Marwan, and C.L.Jr Webber. Singular hydrophobicity patterns and net charge: a mesoscopic principle for protein aggregation/folding. *Physica A*, 343:348–358, 2004.
- [124] Y. Zhou, Z.G. Yu, and V. Anh. Cluster protein structures using recurrence quantification analysis on coordinates of alpha-carbon atoms of proteins. *Phys. Lett. A*, 368:314–319, 2007.
- [125] Z.G. Yu, V. Anh, K.S. Lau, and L.Q. Zhou. Clustering of protein structures using hydrophobic free energy and solvent accessibility of proteins. *Phys. Rev. E*, 73:031920, 2006.
- [126] N. Marwan, M.C. Romano, M. Thiel, and J. Kurths. Recurrence plots for the analysis of complex systems. *Phys. Rep.*, 138:237–329, 2007.
- [127] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley & Sons, New York, 2001.
- [128] C. Chen, L.X. Chen, X.Y. Zou, and P.X. Cai. Predicting protein structural class based on multi-features fusion. *J. Theor. Biol.*, 253:388–392, 2008.
- [129] K.C. Chou and H.B. Shen. Recent progress in protein subcellular location prediction. *Anal. Biochem.*, 370:1–16, 2007.
- [130] X.B. Zhou, C. Chen, Z.C. Li, and X.Y. Zou. Using Chou’s amphiphilic pseudo-amino acid composition and support vector machine for prediction of enzyme subfamily classes. *J. Theor. Biol.*, 248:546–551, 2007.

- [131] M.P.S. Brown, W.N. Grundy, D. Lin, N. Cristianini, C. Sugnet, J.M. Ares, and D. Haussler. Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc. Natl. Acad. Sci. USA*, 97:262–267, 2000.
- [132] C.H.Q. Ding and I. Dubchak. Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics*, 17:349–358, 2001.
- [133] A. Anand, G. Pugalenth, and P.N. Suganthan. Predicting protein structural class by SVM with class-wise optimized features and decision probabilities. *J. Theor. Biol.*, 253:375–380, 2008.
- [134] K.C. Chou. Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes. *Bioinformatics*, 21:10–19, 2005.
- [135] Y.D. Cai. Is it a paradox or misinterpretation? *Proteins*, 43:336–338, 2001.
- [136] K.C. Chou, W.M. Liu, G.M. Maggiora, and C.T. Zhang. Prediction and classification of domain structural classes. *Proteins*, 31:97–130, 1998.
- [137] F. Eisenhaber, C. Frömmel, and P. Argos. Prediction of secondary structural content of proteins from their amino acid composition alone. *Proteins*, 25:169–179, 1998.
- [138] S. Costantini and A.M. Facchiano. Prediction of the protein structural class by specific peptide frequencies. *Biochimie*, 91:226–229, 2009.
- [139] F. Birzele and S. Kramer. A new representation for protein secondary structure prediction based on frequent patterns. *Bioinformatics*, 22:2628–2634, 2006.
- [140] K.C. Chou and D.W. Elrod. Protein subcellular location prediction. *Protein Eng.*, 12:107–118, 1999.
- [141] J. Xu, M. Li, D. Kim, and Y. Xu. RAPTOR: optimal protein threading by linear programming. *J. Bioinform. Comput. Biol.*, 1:95–118, 2003.

- [142] J. Söding, A. Biegert, and A.N. Lupas. The HHpred interactive server for protein homology detection and structure prediction. *Nucleic Acids Res.*, 33:W244–W248, 2005.
- [143] W. Zhang, S. Liu, and Y. Zhou. SP⁵: improving protein fold recognition by using torsion angle profiles and profile-based gap penalty model. *PLoS One*, 3:e2325, 2008.
- [144] R.X. Yan, J.N. Si, C. Wang, and Z. Zhang. DescFold: A web server for protein fold recognition. *BMC Bioinformatics*, 10:416, 2009.
- [145] I. Dubchak, I. Muchink, S.R. Holbrook, and S.H. Kim. Prediction of protein folding class using global description of amino acid sequence. *Proc. Natl. Acad. Sci. USA*, 92:8700–8704, 1995.
- [146] I. Dubchak, I. Muchnik, C. Mayor, I. Dralyuk, and S.H. Kim. Recognition of a protein fold in the context of the SCOP classification. *Proteins*, 35:401–407, 1999.
- [147] H.B. Shen and K.C. Chou. Ensemble classifier for protein fold pattern recognition. *Bioinformatics*, 22:1717–1722, 2006.
- [148] X. Guo and X. Gao. A novel hierarchical ensemble classifier for protein fold recognition. *Protein Eng.*, 21:659–664, 2008.
- [149] M.T.A. Shamim, M. Anwaruddin, and H.A. Nagarajaram. Support vector machine-based classification of protein folds using the structural properties of amino acid residues and amino acid residue pairs. *Bioinformatics*, 23:3320–3327, 2007.
- [150] P. Deschavanne and P. Tufféry. Enhanced protein fold recognition using a structural alphabet. *Proteins*, 76:129–137, 2009.
- [151] H.B. Shen and K.C. Chou. Predicting protein fold pattern with functional domain and sequential evolution information. *J. Theor. Biol.*, 256:441–446, 2009.

- [152] T. Damoulas and M.A. Girolami. Probabilistic multi-class multi-kernel learning: on protein fold recognition and remote homology detection. *Bioinformatics*, 24:1264–1270, 2008.
- [153] Y. Ying, K. Huang, and C. Campbell. Enhanced protein fold recognition through a novel data integration approach. *BMC Bioinformatics*, 10:267, 2009.
- [154] G. Bologna and R.D. Appel. A comparison study on protein fold recognition. In *Proceedings of the 9th International Conference on Neural Information Processing*, pages 2492–2496, 2002.
- [155] E. Lindahl and A. Elofsson. Identification of related proteins on family, superfamily and fold level. *J. Mol. Biol.*, 295:613–625, 2000.
- [156] L. Patthy. Detecting homology of distantly related proteins with consensus sequences. *J. Mol. Biol.*, 198:567–577, 1987.
- [157] C.C. Chang and C.J. Lin. LIBSVM: a library for support vector machines, software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>, 2001.
- [158] H. Rangwala and G. Karypis. Profile-based direct kernels for remote homology detection and fold recognition. *Bioinformatics*, 21:4239–4247, 2005.
- [159] J. Peng and J. Xu. Boosting protein threading accuracy. *Lect. Notes Comput. Sci.*, 5541:31–45, 2009.
- [160] C. Ellis. The state of GPCR research in 2004. *Nat. Rev. Drug Discov.*, 3:577–626, 2004.
- [161] A. Gaulton and T.K. Attwood. Bioinformatics approaches for the classification of G-protein-coupled receptors. *Curr. Opin. Pharm.*, 3:114–120, 2003.
- [162] Q.B. Gao and Z.Z. Wang. Classification of G-protein-coupled receptors at four levels. *Protein Eng.*, 19:511–516, 2006.
- [163] M.N. Davies, A. Secker, A.A. Freitas, M. Mendao, J. Timmis, and D.R. Flower. On the hierarchical classification of G-protein-coupled receptors. *Bioinformatics*, 23:3113–3118, 2007.

- [164] K.C. Chou. Prediction of G-protein-coupled receptor classes. *J. Proteome. Res.*, 4:1413–1418, 2005.
- [165] D.W. Elrod and K.C. Chou. A study on the correlation of G-protein-coupled receptor types with amino acid composition. *Protein Eng.*, 15:713–715, 2002.
- [166] K.C. Chou and D.W. Elrod. Bioinformatical analysis of G-protein-coupled receptors. *J. Proteome. Res.*, 1:429–433, 2002.
- [167] M. Bhasin and G.P.S. Raghava. GPCRsclass: a web tool for the classification of amine type of G-protein-coupled receptors. *Nucleic Acids Res.*, 33:W143–W147, 2005.
- [168] Y. Huang, J. Cai, L. Ji, and Y. Li. Classifying G-protein-coupled-receptors with bagging classification tree. *Comput. Biol. Chem.*, 28:275–280, 2004.
- [169] K. Kristiansen. Molecular mechanisms of ligand binding, signaling, and regulation within the superfamily of G-protein-coupled receptors: molecular modeling and mutagenesis approaches to receptor structure and function. *Pharmacol. Ther.*, 103:21–80, 2004.
- [170] M. Bhasin and G.P.S. Raghava. GPCRpred: an SVM-based method for prediction of families and subfamilies of G-protein-coupled receptors. *Nucleic Acids Res.*, 32:W383–W389, 2004.
- [171] Q.B. Gao, C. Wu, X.Q. Ma, J. Lu, and J. He. Classification of amine type G-protein-coupled receptors with feature selection. *Protein Pept. Lett.*, 15:834–842, 2008.
- [172] P.K. Papasaikas, P.G. Bagos, Z.I. Litou, and S.J. Hamodrakas. A novel method for GPCR recognition and family classification from sequence alone using signatures derived from profile hidden Markov models. *SAR QSAR Environ. Res.*, 14:413–420, 2003.
- [173] Y.Z. Guo, M. Li, M. Lu, Z. Wen, K. Wang, G. Li, and J. Wu. Classifying G-protein-coupled receptors and nuclear receptors on the basis of protein power spectrum from fast Fourier transform. *Amino Acids*, 30:397–402, 2006.

- [174] R. Gupta, A. Mittal, and K. Singh. A novel and efficient technique for identification and classification of GPCRs. *IEEE Trans. Inf. Technol. Biomed.*, 12:541–548, 2008.
- [175] R. Karchin, K. Karplus, and D. Haussler. Classifying G-protein-coupled receptors with support vector machines. *Bioinformatics*, 18:147–159, 2002.
- [176] K.C. Chou and Y.D. Cai. A new hybrid approach to predict subcellular localization of proteins by incorporating gene ontology. *Biochem. Biophys. Res. Commun.*, 311:743–747, 2003.
- [177] Y.D. Cai and K.C. Chou. Predicting membrane protein type by functional domain composition and pseudo-amino acid composition. *J. Theor. Biol.*, 238:395–400, 2006.
- [178] Q.B. Gao, Z.C. Jin, X.F. Ye, C. Wu, and J. He. Prediction of nuclear receptors with optimal pseudo amino acid composition. *Anal. Biochem.*, 387:54–59, 2009.
- [179] W. Li and A. Godzik. CD-HIT: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, 22:1658–1659, 2006.
- [180] AAindex database: <http://www.genome.ad.jp/dbget/aaindex.html>.
- [181] Z.R. Li, H.H. Lin, L.Y. Han, L. Jiang, X. Chen, and Y.Z. Chen. PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence. *Nucleic Acids Res.*, 34:W32–W37, 2008.
- [182] I.T. Jolliffe. *Principal Component Analysis*. Springer, New York, 2002.
- [183] K.C. Chou and H.B. Shen. Cell-PLoc: a package of web-servers for predicting subcellular localization of proteins in various organisms. *Nat. Protoc.*, 3:153–162, 2008.
- [184] I. Cohen, Q. Tian, X.S. Zhou, and T.S. Huang. Feature selection using principal feature analysis. *Univ. of Illinois at Urbana-Champaign*, 2002.

- [185] D.J. Thomas, G. Casari, and C. Sander. The prediction of protein contacts from multiple sequence alignments. *Protein Eng.*, 9:941–948, 1996.
- [186] N. Hamilton, L. Burrage, M.A. Ragan, and T. Huber. Protein contact prediction using patterns of correlation. *Proteins*, 7:679–684, 2004.
- [187] G. Shackelford and K. Karplus. Contact prediction using mutual information and neural nets. *Proteins*, 69:159–164, 2007.
- [188] G. Pollastri and P. Baldi. Prediction of contact maps by GIOHMMs and recurrent neural networks using lateral propagation from all four cardinal corners. *Bioinformatics*, 18:S62–S70, 2002.
- [189] A. Vullo, I. Walsh, and G. Pollastri. A two-stage approach for improved prediction of residue contact maps. *BMC Bioinformatics*, 7:180, 2006.
- [190] O. Grana, D. Baker, R.M. MacCallum, J. Meiler, M. Punta, B. Rost, M.L. Tress, and A. Valencia. A comprehensive assessment of sequence-based and template-based methods for protein contact prediction. *Bioinformatics*, 24:924–931, 2008.
- [191] A.N. Tegge, Z. Wang, J. Eickholt, and J. Cheng. NNcon: improved protein contact map prediction using 2D-recursive neural networks. *Nucleic Acids Res.*, 37:W515–W518, 2009.
- [192] Y. Shao and C. Bystroff. Predicting interresidue contacts using templates and pathways. *Proteins*, 53:497–502, 2003.
- [193] R. Rajgaria, Y. Wei, and C.A. Floudas. Contact prediction for beta and alpha-beta proteins using integer linear optimization and its impact on the first principles 3D structure prediction method ASTRO-FOLD. *Proteins*, 78:1825–1846, 2010.
- [194] J.M.G. Izarzugaza, O. Graña, M.L. Tress, A. Valencia, and N.D. Clarke. Assessment of intramolecular contact predictions for CASP7. *Proteins*, 69:152–158, 2007.

- [195] X. Gao, D. Bu, J. Xu, and M. Li. Improving consensus contact prediction via server correlation reduction. *BMC Struct. Biol.*, 9:28, 2009.
- [196] S. Griep and U. Hobohm. PDBselect 1992–2009 and PDBfilter-select. *Nucleic Acids Res.*, 38:D318–D319, 2009.
- [197] Y. Zhang, A. Kolinski, and J. Skolnick. TOUCHSTONE II: a new approach to ab initio protein structure prediction. *Biophys. J.*, 85:1145–1164, 2003.
- [198] L.S. Cessie and J.C. van Houwelingen. Ridge estimators in logistic regression. *Appl. Statist.*, 41:191–201, 1992.
- [199] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten. The WEKA data mining software: an update. *SIGKDD Explorations*, 11:1018, 2009.
- [200] W. Chen, S.W. Zhang, H.F. Yang, K. Zhao, and K.C. Chou. Prediction of seven protein structural classes by fusing multi-feature information including protein evolutionary conservation information. In *Proceedings of the Second International Conference on Bioinformatics and Biomedical Engineering*, pages 17–20, 2008.
- [201] H. Zhang, T. Zhang, K. Chen, S. Shen, J. Ruan, and Kurgan L. Sequence based residue depth prediction using evolutionary information and predicted secondary structure. *BMC Bioinformatics*, 9:388, 2008.
- [202] K.C. Chou and Y.D. Cai. Prediction of protease types in a hybridization space. *Biochem. Biophys. Res. Commun.*, 339:1015–1020, 2006.

List of Publications

The following list enumerates the author's publications pertinent to this dissertation.

- **Jian-Yi Yang** and Xin Chen, Improving taxonomy-based protein fold recognition by using global and local features, *Proteins: Structure, Function, and Bioinformatics*, 79:2053-2064, 2011.
- **Jian-Yi Yang** and Xin Chen, A consensus approach to predicting protein contact map via logistic regression, *ISBRA'11, Lecture Notes in Bioinformatics*, Springer Verlag, 6674:136-147, 2011.
- Zhen-Ling Peng*, **Jian-Yi Yang***, and Xin Chen, An improved classification of G-protein-coupled receptors using sequence-derived features, *BMC Bioinformatics*, 11:420, 2010. (***Contributed equally**)
- **Jian-Yi Yang**, Zhen-Ling Peng, and Xin Chen, Prediction of protein structural classes for low-homology sequences based on predicted secondary structure, *BMC Bioinformatics*, 11:S9, 2010.
- **Jian-Yi Yang**, Zhen-Ling Peng, Zu-Guo Yu, Rui-Jie Zhang, Vo Anh, and Desheng Wang, Prediction of protein structural classes by recurrence quantification analysis based on chaos game representation, *Journal of Theoretical Biology*, 257:618-626, 2009.

The following list enumerates the author's other publications at NTU.

- Xin Chen and **Jian-Yi Yang**, Constructing consensus genetic maps in comparative analysis, *Journal of Computational Biology*, 17:1561-1573, 2010.
- Heng Lian, Xin Chen, and **Jian-Yi Yang**, Doubly SCAD penalized additive model regression with an application to expression prediction from sequences, *Biometrics*, in press, 2011.

Appendix A

Appendix A provides the links of the datasets used in this dissertation.

1. Datasets used in Chapters 2 and 3. The updated datasets *25PDB*, *1189*, *640*, and the predicted secondary structure for each dataset are available at: http://www1.spms.ntu.edu.sg/~chenxin/RKS_PPSC/.
2. Datasets used in Chapter 4. The five datasets RDD, EDD, F95, F194, and F710 used are available at <http://www1.spms.ntu.edu.sg/~chenxin/TAXFOLD/>.
3. Datasets used in Chapter 5. The *GDFL* dataset at five levels is available at http://www1.spms.ntu.edu.sg/~chenxin/PCA_GPCR/.
4. Datasets used in Chapter 6. The datasets D80 and D856 are available at <http://www3.ntu.edu.sg/home2008/YANG0241/LRcon/>.