國立交通大學

生物資訊所

碩士論文

MuLiSA: 多重配體結構比對為基礎之蛋白質功能片段及重要氨基酸之預測分析

MuLiSA: Analysis and Identification of Functional Motifs and Residues in Proteins by Multiple Ligand-bound Structure Alignments

研 究 生：林建宏

指導教授：楊進木　教授

中 華 民 國 九 十 三 年 六 月

MuLiSA: 多重配體結構比對為基礎之蛋白質功能片段及重要氨基酸之預測分析

MuLiSA: Analysis and Identification of Functional Motifs and Residues of Proteins by Multiple Ligand-bound Structure Alignments

研 究 生：林建宏　　Student：Chien-Hung Lin

指導教授：楊進木　　Advisor：Jinn-Moon Yang

國 立 交 通 大 學

生 物 資 訊 所

碩 士 論 文

A Thesis Submitted to Institute of Bioinformatics
National Chiao Tung University in partial Fulfillment of the Requirements
for the Degree of Master in
Bioinformatics

June 2004

Hsinchu, Taiwan, Republic of China

中華民國九十三年六月

MuLiSA: 多重配體結構比對為基礎之蛋白質功能片段及重要氨基酸之預測分析

學生：林建宏　　指導教授：楊進木

國立交通大學生物資訊所碩士班

摘　　要

由於快速大量增加的蛋白質序列相關資訊及蛋白質多樣性，僅利用蛋白質序列來預測並鑑定蛋白質功能是一件相當重要且急迫的任務。在這篇論文中，我們發展了一個新的方法來鑑定與配體結合的蛋白質高度保留氨基酸及 motifs。在 MuLiSA（多重配體結構比對）這個新方法中，我們首先將多個與配體結合蛋白質的配體重疊，使位在配體結合區域的氨基酸自然而然地疊合在一起。接著我們利用氨基酸位置及氨基酸序列片段亂度計算的 z-score 來鑑定重要的氨基酸位置及典型的序列片段。當我們鑑定出新的典型序列片段後，我們會建立該典型序列片段的側寫並用來對預測只擁有蛋白質序列資訊的蛋白質功能。我們已將此方法應用在三種與配體結合的蛋白質上：ATP-binding proteins，ADP-binding proteins 和 HEM-binding proteins。實驗的結果顯示由我們鑑定出的高度保留氨基酸及典型片段與配體結合的功能有相當程度的關係，並已鑑定出一些文獻上證實的重要氨基酸位置。儘管目前所鑑定出的重要片段對擁有特定功能蛋白質的覆蓋度不高，例如在 ATP-binding proteins，motor proteins 及 HEM-binding proteins 的覆蓋率為 23.51%，47.64% 及 13.60%。然而在 kinesin 的功能預測下準確率高達 86.49%。因此我們相信當我們加大與配體結合之蛋白三級結構資訊後，我們能增加蛋白質功能預測的準確度並且挖掘出更多新的資訊供科學家們做更深入的研究。我們發現多重配體結構比對能鑑定出高度保留的典型序列片段並且在部分的與配體結合蛋白質中比一些傳統蛋白質結構或序列比對工具，如 CE 及 CLUSTALW 表現更佳。我們認為此多重配體結構比對技術能幫助科學家們發現與配體結有高度合專一性的氨基酸及重要的典型片段。

MuLiSA: Analysis and Identification of Functional Motifs and Residues in Proteins by Multiple Ligand-bound Structure Alignments

Student: Chien-Hung Lin Advisor: Jinn-Moon Yang

Institute of Bioinformatics
National Chiao Tung University

ABSTRACT

To predict and identify details regarding function from protein sequences is an emergency task since the growing number and diversity of protein sequence. Here, we develop a novel approach for identifying conservation residues and motifs of ligand-binding proteins. In this method, called MuLiSA (Multiple Ligand-bound Structure Alignment), we first superimpose the ligands of ligand-binding proteins and then the residues of ligand-binding sites are naturally aligned. We identify important residues and patterns based on the z scores of the residue entropy and residue-segment entropy. After identifying new pattern candidates, the profiles of patterns are generated to predict the protein function from only protein sequences. We tested our approach on three kinds of ligand-binding proteins: ATP-binding proteins, ADP-binding proteins and HEM-binding proteins. The experiments show that the conservation residues and novel patterns we identified are really correlated with protein functions of certain ligand-binding proteins and we have also identified conservation residues that were verified by previous studies. Although the coverage is not good, such as the coverage rate of ATP-binding proteins, motor proteins and HEM-binding proteins are 23.51%, 47.64 and 13.60%, we also observed that perdition accuracy of kinesin is 86.49%. We believe if we broaden the training dataset, we can improve the prediction accuracy and mining more new information for researchers to do further research. We found that multiple ligand-bound structure alignments can identify conservation patterns and is better than traditional alignments such as CE and CLUSTALW in some ligand-binding proteins. We think that this multiple ligand-bound structure alignment technique can help researchers to discover ligand-binding specificity-determining residues and functional important patterns.

Acknowledgments

My foremost thank goes to my thesis adviser Dr. Jinn-Moon Yang. Dr. Yang teaches us how to think deeply, solve problems, and present conceptions. Thanks for his patience to gives us chances to try errors, spends times on discussion, and teaches us how to write program.

I must thank to all members in Yang's lab. We write programs together, work together, and of course, have funs together. Thanks for everyone who help me to fix my computer, for I was nearly a computer idiot. Thanks for guys who help me to debug when I made stupid mistakes in programming. Of course, thanks for all of you who play balls, poker, and games with me after a long, depressed day. Anyway, thanks for every body in Yang's lab. Especially, I must thank for C.C. Chen's patience for he always pick up the pieces for us.

And also I must thank for Dr. E.S. Lin for his guidance and advice in early phases of research.

Finally, thanks for my whole family, for their support during past twenty-fifth years.

CONTENTS

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Problem Formulation

Human genome have been sequenced and led to a flood of sequence information. On the other hand, recent developments in X-ray crystallography and NMR have made it faster in solving protein structures. These data contains a lot of information that can be extracted by techniques which were used to visualize the sequence conservation information.

The residues most related to the functions of a protein are often the most conserved [1]. Many studies have demonstrated that most protein domains of same protein families, such as PROSITE [2] and Pfam [3], share conserved peptide patterns, called motifs, and some critical residues. For example, the phenylalanine and histidine residues are both conserved in the aligned sequences of all known functional myoglobins including α- and β-globins, the globins of invertebrates, and plant leghemoglobins. The fundamental problems in proteomics include both identifying and understanding the role of the essential sites that determine that structure and proper function of the proteins. After solving these problems, researchers can apply this useful information as a clue to predict protein functions without protein structure information.

1.2 Motivation

Many groups have used the identification of conserved patterns as a method to predict protein function. Some of these groups predict protein motifs using principle component analysis [4-7]. Other groups use structure alignment [8] or sequence alignment [9] as a

method to identify conservation sites. Evolutionary trace analysis was used to predict functional patterns in different phylogenic trees and look for functional important residues [8, 10-13]. However, these methods always use protein structure or protein sequence information to predict protein conservation patterns and may miss these conservation patterns because of the noises from other protein structures which are far apart from ligand-binding site.

1.3 Related Works

1.3.1 Sequence alignment tools

There are several famous sequence alignment tools, such as CLUSTALW [9], T-COFFEE [14], and BLAST [15]. Using protein sequence alignment to search for conservation residues is a popular approach now. Here we take CLUSTALW as an example.

CLUSTALW performs a global multiple sequence alignment through three steps:

(1) Perform all-against all pair-wise alignments

(2) Produce a phylogenic tree by alignment scores

(3) Perform multiple sequence alignments according to phylogenic tree relationships.

However, CLUSTALW only use protein sequence information and generate alignment only depend on “protein side” information.

1.3.2 Structure alignment tools

As in most cases protein functions always have higher relationship with protein three-dimensional structures than protein sequences, for proteins with low sequence identity may form similar three-dimensional structures and have similar function, using structure

alignment as a method to identify conservation residues seems a more convincing approach. There are several famous tools, such as DALI [16], VAST[17], CE[8]. Here we take CE as an example.

CE, combinatorial extension, is a fast and accurate structure alignment tools. This algorithm uses local aligned fragment pairs (AFPs) to extend alignment path and lead to a single optimal alignment.

However, as CE undergoes structure alignments only focus on “protein side” information, when ligand-binding proteins binding with same ligand only have similar structures in ligand-binding sites, structure alignments only focus on “protein side” information may be disturbed by structure information other than ligand-binding sites and led to bad alignments of ligand-binding sites.

1.4 Thesis Overview

In Chapter 2, we proposed a new ligand-based multiple structure alignment approach, MuLiSA, multiple ligand-bound structure alignments. The main difference between MuLiSA and other tools is that we first superimpose the ligands of proteins but not protein itself. In this way, the ligand-binding sites are superimposed naturally. Then we could identify the conservation residues according to these positions in which were superimposed along with ligands. We also introduced datasets and methods for conservation residues identification, pattern identifying and protein function prediction.

In Chapter 3, we applied MuLiSA to ATP-binding proteins, ADP-binding protein and HEM-binding proteins. We verify the structure similarity matrix generated from MuLiSA alignment results using SCOP classification and also compare pattern candidates and conservation residues we identified with PROSITE patterns. We also use protein sequence datasets to verify the profiles of pattern candidates. Finally, we use profiles of pattern

candidates to undergo protein function prediction.

In Chapter 4, we summarized the protein function prediction results of ATP-binding proteins, motor proteins (because of the ambiguous annotations about ADP-binding proteins), kinesin proteins and HEM-binding proteins. Although the coverage rates are only 23.51%, 47.64% and 13.60% of ATp-binding proteins, motor proteins and HEM-binding proteins, the prediction accuracy of kinesin prediction is as high as 86.49%. We also list several predicted proteins and approaches that may improve the alignment performance and possible applications of .MuLiSA for future works.

Chapter 2

Materials and Methods

2.1 Overview

Identification of conservation patterns and residues in proteins by multiple ligand-bound structure alignments encompasses a variety of sequential computational phases, including dataset preparation, dataset clustering, multiple ligand-bound structure alignments, post-alignment analysis and entropy calculation, tool verify and protein function prediction (Figure 1).

In dataset preparation, we first select one kind of ligand-binding protein that we are interested and get ligand-binding protein list from PDBsum [18] database. Because we need precise protein structures to identified conservation residues and motifs, we only select protein structures resolved by X-ray diffraction. Then we select ligand-binding domains using programs from SCOP database [19].

In data clustering, we generate all-against-all multiple ligand-bound structure alignments of these selected ligand-binding domains and generate one structure similarity matrix and one sequence identity matrix for each kind of ligand-binding proteins. Once we have these two matrixes, we select non-redundant protein domains, and undergo protein domain clustering.

In the main step of MuLiSA, first we choose the alignment center domains C of each domain cluster based on structure similarity. Second, we undergo C centered multiple ligand-bound structure alignment. After we generate the alignments, z-score calculation of position entropy can help us to identify conservation residues of each domain cluster. For we believed that the functional important motifs mostly composed of functional important

residues, we identified pattern candidates by conservation residues extension.

Finally, we used SCOP [19] and PROSITE [2] databases to verify our results; and then we generate profiles of pattern candidates and use them to search for protein sequence with these patterns in SWISS-PROT database [20].

Appendix A shows the computational steps in the overall workflow and the programs we used in this study.

2.2 Preparation of training datasets and verifying datasets

2.2.1 Preparation of ligand-binding protein list

We have applied MuLiSA to three kinds of ligand-binding proteins, which are ATP-binding proteins, ADP-binding proteins, and HEM-binding proteins. The ligand-binding protein lists were taken from PDBsum database [18].

2.2.2 Preparation of ligand-binding domains

In order to get ligand-binding domains, first we need to get ligand-binding protein structures. Protein structure three-dimensional information was downloaded from Protein Data Bank (PDB) database[21] according to ligand-binding protein lists getting from PDBsum database [18]. The ligand-binding domains were chosen by our program GetDomain.c (see appendix B) and were downloaded from Structure Classification of Proteins (SCOP) database [19].

Ligand-binding domains were chosen with four criteria, they are as follows:

1. When one of distances between atoms of residues of the domain and atoms of ligands is near than 5Å, we think that this domain is a ligand-binding domain.

2. Because multiple ligand-bound structure alignment first superimposed the ligands of aligned proteins, we only choose protein domains which only bind with one ligand.
3. We only choose ligand-binding domains which the ligand they bind is only bind by one protein domain.
4. We only choose one protein domain in one protein structure.

Because the SCOP domain files do not contain ligand information, after choosing these domains we must add back ligand information from Protein Data Bank (PDB) database [21] into these protein domain files. It must be mentioned that we only choose protein domains solved by x-ray crystallography because we think that these structures are more convincing.

2.2.3 Datasets for verification

To verify whether our alignment results is reasonable and can reflect protein function information, we use the classification of Structural Classification of Proteins (SCOP) database [19] as the benchmark of our structure similarity matrix for non-redundant domain clustering. PROSITE patterns from PROSITE database [2] were also used to quality assessment and refinement of multiple ligand-bound structure alignments. The protein sequences and annotations were downloaded from SWISS-PROT database [20] and were used for profile verification and protein function prediction.

2.3 Methods

2.3.1 Multiple ligand-bound structure alignment

The main idea of this tool is that we try to align together conservation residues of proteins at ligand-binding sites by ligand superimposition; and then identify conservation

residues and patterns by z-score of entropy calculation. Because we have to change the three-dimensional coordinates of proteins along with superimposed ligands, we developed a structure superimpose tool to deal with this problem. We developed this program MuLiSA from ICP algorithm[22] (see appendix C), this program can make proteins and ligands rotation and displacement on three-dimensional space. After we get the superimposed protein structures, we regard two residues are aligned together based on three order rules:

- Rule 1: Cβ or Cα (Gly) atom of amino acid residues in 1Å
- Rule 2: Cβ or Cα (Gly) atom of same amino acid residues in 4Å
- Rule 3: Cβ or Cα (Gly) atom of same group amino acid residues in 4Å or Cβ or Cα (Gly) atom of different group amino acid residues in 2Å.

The amino acid groups are defined as follows:

- ✓ Basic amino acids: lysine, arginine, and histidine.
- ✓ Acidic amino acids: aspratate, glutamate, asparagine, and glutamine.
- ✓ Aromatic amino acids: phenylalanine, and tryptophan.
- ✓ Aliphatic amino acids: glycine, alanine, valine, leucine, isoleucine, and methionine.
- ✓ Hydroxyl containing amino acids: serine, threonine, and tyrosine;
- ✓ Disulfide-bond forming amino acid: cysteine.
- ✓ Cyclic amino acid: proline.

In ATP-binding proteins and ADP-binding proteins, because of the high divergence of phosphate groups, we aligned whole ligand and only “ribose plus base region” first and then choose the better one as the alignment result.

2.3.2 Sequence identity matrix and structure similarity matrix

If two protein domains have similar function and have highly similar structures in ligand-binding sites, these two protein domain structures should fit well in three-dimensional space. We introduced structure similarities in accordance with multiple ligand-bound structure alignments to present this information.

S^T_{ab} is the structure similarity of protein domain *a* and protein domain *b*. L_a is the length (residue numbers) of protein domain *a*, L_b is the length (residue numbers) of protein domain *b*, and *L* is the aligned residue number of protein domains *a* and *b*. S^T_{ab} is given as

$$S^T_{ab} = \frac{L}{\min\{L_a, L_b\}} \qquad (1)$$

We also generate un-gapped sequence identity matrix between protein domains for non-redundant protein domain selection based on only aligned residues of protein domains *a* and *b*S^E_{ab} is the un-gapped sequence identity of protein domain *a* and protein domain *b*. *mt* is the number of identical aligned residues of protein domain *a* and protein domain *b*; *mmt* is the number of non-identical aligned residues of protein domain *a* and protein domain *b*.

$$S^E_{ab} = \frac{mt}{mt + mmt} \qquad (2)$$

2.3.3 Non-redundant protein domains and alignment center *C* selection

Redundant protein domains must be removed because the profiles generated from alignments may be incredible. We regarded two protein domains are redundant protein domains when their structure similarity and sequence identity are both above 0.8; therefore, we first cluster these protein domains and only choose one with no mutation residues and with the smallest X-ray diffraction resolution.

In order to generate a convincing multiple alignments, we must choose an alignment

center domains *C* before we generate this alignments. In structure similarity matrixes, the non-redundant protein domain of one cluster which has the highest structure similarity with other protein domains than others was selected as the alignment center *C* of this cluster. This protein domain was used to be the alignment center of multiple ligand-bound structure alignment. Figure 2 shows one example.

2.3.4 Identification of conservation residues and pattern candidates

To identify these conservation residues, we used entropy (S_p), defined as

$$S_P = -\sum_{i=1}^{20} f_{pi} * \ln\left(f_{pi}\right) \qquad (3)$$

where *i* and f_{pi} denote the i^{th} amino acid type, the probability of finding the amino acid type *i* at position *p*. The entropy is 0 when this position is totally conserved.

In order to estimate the statistical significance of the position entropy, z-score was applied to identify relative conservation positions.

$$Z_p = \frac{X_p - \mu}{\sigma} \qquad (4)$$

where Z_p is the z-score value of position *p*, σ is the standard deviation of all positions entropy, μ is the average value of all positions entropy and X_p is the entropy of position *p*. We identified a conservation position *p* when $Z_p > 2.5$.

To identify pattern candidates, we extend protein segment from conservation residues. First we extend from conservation residues to residues with z-score larger than 1.0 next to these conservation residues. When there is one "gap" (gap: residues with z-score less than 1.0) between two residues, "Gap tolerance" let us to exted the segment. For example, if one is larger than 1.0 and the other is larger than 2.5 (the sum of z-score of these two residues is larger than 3.5), we linked this "gap" and we extend this protein segment. If *n* "gaps" occurs, the sum of z-score of these two gap gapped residues must larger than *n+2* and we can

continue to extend the protein segment. We only choose protein segments as pattern candidates extending from conservation residues and the length is equal or longer than 5 residues (Figure 3).

2.3.5 Profile generation

We generate alignment profiles of pattern candidates (discovered by our MuLiSA) and PROSITE patterns from multiple ligand-bound structure alignments.

$$PF_{pi} = \left\{ f_p^i \right\} \quad \text{where} \, 1 \le i \le 20 \qquad (5)$$

where PF_p is the profile of position p; f_p^i is the probability of the i^{th} amino acid type at position p.

2.3.6 Profile score calculation

We use profiles to search for matched protein segments in protein sequences. The search window size is the length of profiles and shifts one residue each time. Each protein sequence should have *N-(n-1)* (*N* is the length of this sequence and *n* is the length of this pattern) profile scores, and we suppose the segment with the highest profile search score of this protein sequence should be the pattern candidate that we are looking for.

The scoring function is as follows:

$$S = \frac{\sum_{p=1}^{n} \sum_{i=1}^{20} PF_{pi}}{n} \qquad (6)$$

Where S is the profile score, n is the length of a pattern, PF_{pi} is the profile value of amino acid type i at position p. The score is 1 when a segment perfectly matches this profile.

Chapter 3

Protein Function Prediction and Conservation Residues Identification of ATP-, ADP-, and HEM-Binding Proteins

In order to identify the wealth of information present in protein structures, we analyzed conservation residues and patterns in multiple ligand-bound structure alignments.

To infer a major functional role from residue conservation, a function-based clustering is necessary before identifying conservation residues. Statistically, the bias of conservation may be from not having enough and convincing data, this is why we remove structures too much similar, the redundant domains, select alignment center domain C and generate alignments with clusters have more than four protein domains.

Most sequence and structure alignment techniques are protein-based alignment; in other words, these techniques analyze residue conservation only by comparing protein structure or protein sequence similarity. However, local alignment error sometimes happens when the sequence identity is less than 25% in sequence alignment or protein structures are much similar at regions far away from protein functional important region in structure alignment.

At the present, we have applied MuLiSA to ATP-, ADP-, and HEM-binding proteins and identified several conservation residues and pattern candidates. We have generated sequence profiles from multiple alignments and used them to discover protein sequences which may have these profiles. We also proved that MuLiSA is better than other tools in several cases and can discover functional information when comparing with SCOP [19] and PROSITE database[2]. Our major intention was to extract protein structure information from ligand-binding proteins and apply this information to protein function prediction. Table 1 shows some statistics about the dataset we used in this study. We applied MuLiSA to three

kinds of ligand-binding proteins; they are ATP-binding proteins, ADP-binding proteins and HEM-binding proteins. Through getting ligand-binding protein lists, selecting ligand-binding domains, domain clustering, non-redundant domains and alignment center C selection, we use MuLiSA and z-score of entropy calculation to identified conservation residues and pattern candidates of each cluster. These identified conservation residues may be functional important and we survey the literature and it proves that some of these identified conservation residues are critical to ligand-binding or correlate with conformation stability. After pattern candidate identification, we generate profiles of these pattern candidates and use these profiles predict protein functions.

3.1 ATP-binding proteins

3.1.1 Overview

ATP, adenosine triphosphate, is the major energy currency of the cell. It transfers energy from chemical bonds to endergonic reactions of the cell. ATP powers most of the energy-consuming activities of cells, such as muscle contraction, synthesis of polysaccharides, active transport of ions and nerve impulse. Because of ATP is a so important compound and because of the large number of experimental data, like ATP-binding protein structures and literatures, we choose ATP-binding proteins as our first research target. We have generated structure similarity matrix of non-redundant ATP-binding domains for functional-based domain clustering, and we also identified conservation residues and pattern candidates. Finally, we used profiles of pattern candidates to undergo protein function prediction.

3.1.2 Structure similarity matrix and alignment center C selection

Figure 4 shows the structure similarity matrixes and SCOP classifications of 25 non-redundant ATP-binding domains. When comparing with classifications of SCOP database [19], protein domains with higher structure similarities are usually clustered together and they are always belong to same SCOP families. As we all agree that SCOP database [19] is a convincing domain structural and functional classification database, it tells us that the multiple ligand-bound alignment and structure similarity calculation is reasonable and can reflect structural and functional information.

In Figure 4(A), domains belong to the same SCOP families are with same colors. The bold values means the structure similarity is larger than the average value of the row; in other words, the domain in this row is much similar with these compared domains than others. In this matrix, we find that most domains of same SCOP family usually have higher structure similarity with each other (see the regions with red frame), it tell us that the multiple ligand-bound structure alignment and structure similarity calculation is reasonable and can reflect structural and functional information. Figure 4(B) shows the SCOP classification of protein domains in Figure 4(A).

3.1.3 Conservation residues identified from ATP-binding domains

After selecting alignment center *C* of each cluster, we use multiple ligand-bound structure alignment tool, MuLiSA, to generate multiple alignments.

We have identified several conservation residues (with z-score of position entropy > 2.5) of protein domains in “Protein kinases catalytic subunit family” and “Class I aminoacyl-tRNA synthetases (RS), catalytic domain family”. In Table 2, conservation residues were identified and listed; the bold residues are these residues, verified by previous studies, that are

important in ATP-binding or conformation stability [23-32]. For example, in "Human Cyclin-Dependent Kinase 2 protein domain (SCOP code: d1hck__)", we have identified residues A31, K129, N132 and D145 which interact with ATP through forming hydrogen bonds. Except for these four residues, we also identified six conservation residues and we believe that these residues are very likely in playing important role in ATP-binding.

Figure 5 shows the multiple ligand-bound structure alignment results and the identified conservation residues in "Protein kinases, catalytic subunit family" of ATP-binding domains. In Figure 5(A), the identified conservation residues, aligned positions with z-score of entropy calculation > 2.5, are close to ATP in three-dimensional space. It implies that these conservation residues may play important role in ATP-binding. In Figure 5(B), the labeled residue numbers belong to protein domain d1phk__, which is the selected alignment center C of this cluster; and the red framed region means the PROSITE patterns. We observed that most identified conservation residues were on these PROSITE pattern region, it tell us that identifying pattern candidates from conservation residues extension may be a reasonable approach.

3.1.4 Pattern candidates identified from ATP-binding domains

We have identified pattern candidates of "Protein kinases catalytic subunit family" and "Class I aminoacyl-tRNA synthetases (RS), catalytic domain family" of ATP-binding domains. Table 3 lists these pattern candidates. We only choose the pattern candidates which are equal or longer than 5 residues and extending from identified conservation residues with z-score of entropy calculation > 2.5. Table 4 shows the comparison of PROSITE patterns and our defined pattern candidates that overlap with PROSITE patterns of ATP-binding domains. These pattern candidates are partially overlapping with PROSITE patterns. However, the new pattern candidates which do not overlap with PROSITE patterns in Table 3, may be new clues

to search for ATP-binding proteins. For example, although the pattern candidate 1 in “Protein kinases catalytic subunit family” is overlapping with PROSITE pattern: Serine/ Threonine protein kinases active-site signature, there are also pattern candidate 2 and 3 in “Protein kinases catalytic subunit family” that do not overlap with PROSITE pattern.We found that identified pattern canididates are near ATP in 3-D space, therefore we believe that these two pattern candidates may be new clues to search for ATP-binding proteins (Figure 6).

All of these pattern candidates and PROSITE patterns were used to generate profiles and we will use these profiles for protein function prediction.

3.1.5 Profile verification and protein function prediction of ATP-binding proteins

In order to use profiles generated from our alignments to predict protein function, first we need to verify that the profiles we generated from our alignments is reasonable and convincing. Therefore, we use protein sequences which have PROSITE patterns: PS00178, PS00107, PS00108, PS00411, PS00190, PS00435, PS00436, PS00086 and PS00191. These PROSTIE patterns belong to 8 clusters listed in Table 1. Because pattern candidates identified from one cluster should be meaningful for sequences of this cluster, when we use profiles generated from these pattern candidates to search for protein sequences of this cluster, the sequences of this cluster should have higher profile scoring scores. In other words, a good pattern candidate can separate protein sequences of the cluster that have this pattern candidate from protein sequences of other clusters that don’t have this pattern candidate.

In order to compare with the performance of pattern candidates and PROSITE patterns, we also generated profiles of PROSITE patterns from our multiple alignments. If the performance of pattern candidates in one cluster is better than PROSITE patterns, we may find a novel pattern that is more meaningful than PROSITE patterns in this cluster. In Figure 7, we observed that our defined pattern candidate is worse than PROSITE pattern; however,

because the profile of PROSITE pattern is generated from our alignment, and the performance is good, it proved that the profile generated from our alignments is reasonable and convincing.

In order to verify the effectiveness of profiles generated from our alignments in protein function prediction, we compare the performance in profile search between dataset 1, which contains protein sequences with PROSITE pattern; and dataset 2, which contains protein sequences not only with PROSITE pattern but also have “ATP-binding” annotations in SWISS-PROT database. In Figure 8, dataset 1 contains protein sequences contain PROSITE pattern: aminoacyl-transfer RNA synthetases class-I signature and dataset 2 contains protein sequences contain not only PROSITE pattern: aminoacyl-transfer RNA synthetases class-I signature but also have “ATP-binding” annotations in SWISS-PROT database. We observed that the area under curves of dataset 2 is larger than the area under curves of dataset 1. Because the profile of pattern candidates were generated from alignments of ATP-binding domains and the protein sequences in dataset 1 are not all have “ATP-binding” annotations in “KW” of SWISS-PROT database, we suppose that the profile of pattern candidate is more convincing in ATP-binding proteins but not proteins only with PROSITE patterns.

In Table 5, we summarized the average hit rate of true positive rate 50%, 60%, 70%, 80%, 90% and 100% in dataset 1: sequences with PROSITE pattern, and database 2: sequences with PROSITE pattern and SWISS-PROT annotations for profile verification. We observed that whether in dataset 1 or dataset 2, the hit rate of PROSITE patterns are all higher than pattern candidates. Thus, the PROSITE pattern is really meaningful for protein sequences which have these PROSITE patterns.

However, we also observed that the hit rates in dataset 2 are generally higher than hit rates in dataset 1. Because dataset 1 only contains sequences with PROSITE patterns but database 2 contains sequences with PROSITE pattern and SWISS-PROT annotations, it tell us that the profiles we generated from multiple alignments of ATP-binding proteins may be more meaningful for protein sequences with “ATP-binding” annotations in SWISS-PROT

database.

Second, we used profiles of pattern candidates and PROSITE patterns of ATP-binding proteins to search for SWISS-PROT protein sequences that might have these patterns, and we suppose that the protein sequences with these pattern candidates may be ATP-binding proteins. We use all profiles of identified pattern candidates to search all protein sequences in SWISS-PROT database and give each sequence a profile scoring score. The given profile scoring score is the highest score of all profiles search. In this way, we can get a profile scoring ranking list in ATP-binding protein prediction (Figure 9). The sequences with higher profile score have higher possibility to be ATP-binding proteins. When one protein sequence has high profile score but not have "ATP-binding" annotations in SWISS-PROT database, we regard this protein might be an ATP-binding protein because it contains this pattern candidate.

Figure 9 shows the profile scoring list of protein function prediction in ATP-binding proteins. Two points must be mentioned. First, the framed sequences all have "ATP-binding" annotations (except for P27604 and P25169); because these sequences all match novel pattern candidate, pattern candidate 2 in "Protein kinases catalytic subunit family" , we regard this pattern candidate is a new pattern of ATP-binding proteins. Second, the non-labeled sequences, P27604 and P25169, are the sequences that match profiles but don't have "ATP-binding" annotations in SWISS-PROT database, hence these two proteins might be the ATP-binding proteins but not identified yet.

In Table 6, we summarized the true-positive rates, profile scoring scores, and z-score of profile scoring scores of top 100, 500, 1000, 1500, 2000, 2500 and 3000 ranked sequences in profile scoring ranking list. We also compare the hit rates between pattern candidates and PROSITE patterns. We observed when protein sequences with profile scoring score 0.600, the true positive rate is 82.27% and the z-score is 2.87. Thus when protein sequences with profile scoring score higher than 0.600, we can say these protein sequence may be ATP-binding proteins with 82.27% confidence.

When comparing with the hit rate of our defined pattern candidates and PROSITE patterns, we observed that almost all the top 3000 ranked protein sequences with "ATP-binding" annotations were all searched by pattern candidates. Although some of pattern candidates partially overlapped with PROSITE patterns, it tells us that the pattern candidates are useful for protein function prediction in ATP-binding proteins.

3.2 ADP-binding proteins

3.2.1 Overview

ADP, adenosine diphosphate, is a universe energy intermediate of the cell. ADP is the hydrolysis product of ATP. It can also transfers energy from chemical bonds to endergonic reactions of the cell. The main difference between ATP and ADP is that ATP contains two high energy bonds but ADP only have one. Because of ADP is also a universe energy intermediate of the cell, it is also an important compound and we choose ADP-binding proteins as our second research target.

We have also generated structure similarity matrix of non-redundant ADP-binding domains for functional-based domain clustering, and we also identified conservation residues and pattern candidates. Finally, we used profiles of pattern candidates to undergo protein function prediction.

3.2.2 Structure similarity matrix of ADP-binding domains

Figure 10 shows the structure similarity matrixes and SCOP classifications of 30 non-redundant ATP-binding domains. When comparing with SCOP classifications, protein domains with higher structure similarity are usually clustered together and they are always

belong to same SCOP families. It also tells us that the multiple ligand-bound structure alignments and structure similarity calculation in ADP-binding proteins is reasonable and can reflect structural and functional information.

In Figure 10(A), we also observed that most domains of same SCOP family usually have higher structure similarity with each other (see the regions with red frame). Figure 8(B) shows the SCOP classification of protein domains in Figure 10(A). We also chose alignment center C of each cluster in ADP-binding domains.

3.2.3 Conservation residues identified from ADP-binding domains

We have also identified several conservation residues in protein domains of "motor proteins family". In Table 7, conservation residues were identified and listed; the bold residues are residues that were announced on literature that are important in ADP-binding or conformation stability[33-40].

Figure 11 shows the multiple ligand-bound structure alignment result and identified conservation residues in "motor proteins family" of ADP-binding domains. In Figure 11(A), the identified conservation residues are closed to ADP in three-dimensional space. It implies that these conservation residues may play important role in ADP-binding. In Figure 11(B), the labeled residue numbers are belonged to protein domain d1goja_, which is the selected alignment center C of this cluster, and the red framed region means the PROSITE patterns. We observed that most identified conservation residues were on these region, it tell us that identifying pattern candidates from conservation residues extension may be a reasonable approach.

3.2.4 Pattern candidates identified from ADP-binding domains

We have identified pattern candidates of "motor proteins family" of ADP-binding domains. Table 8 lists these pattern candidates. Table 9 shows the comparison of PROSITE patterns and our defined pattern candidates that overlap with PROSITE patterns in ADP-binding domains. These pattern candidates are partially overlapping with PROSITE patterns. However, the new pattern candidates which do not overlap with PROSITE patterns in Table 8, may be new clues to search for ADP-binding proteins. We also found that identified pattern canididates are near ADP in 3-D space, therefore we believe that these three pattern candidates may be new clues to search for ADP-binding proteins (Figure 12). All of these pattern candidates were also used to generate profiles and we will use these profiles for protein function prediction.

3.2.5 Profile verification and protein function prediction of ADP-binding proteins

In order to compare with the performance of pattern candidate and PROSITE patterns, we also generated profiles of PROSITE patterns from our multiple alignments. In Figure 13, we observed that pattern candidate is worse than PROSITE pattern; however, because the profile of PROSITE pattern is generated from our alignments, and the performance is good, it also proved that the profile generated from our alignments is reasonable and convincing.

In order to verify the effectiveness of profiles generated from our alignments in protein function prediction, we also compared the performance in profile search between different datasets. However, because of the ambiguous annotations about ADP-binding proteins and we only chose one domain cluster, "motor proteins family", in ADP-binding proteins, we chose protein sequences contain not only PROSITE pattern: Kinesin motor domain signature but also "motor protein" annotations in SWISS-PROT database.

In Figure 14, dataset 1 contains protein sequences with PROSITE pattern: Kinesin motor domain signature; dataset 2 contains protein sequences contain not only PROSITE pattern but also “motor protein” annotations in SWISS-PROT database. We observed that the area under curves of dataset 2 is also larger than the area under curves of dataset 1. Because the profiles of pattern candidates were generated from motor protein domains alignments and the protein sequences in dataset 1 not all have “motor protein” annotations in SWISS-PROT database, we suppose that the profiles of pattern candidates are more convincing in motor proteins but not proteins only with PROSITE patterns.

In Table 10, we summarized the average hit rate of true positive rate 50%, 60%, 70%, 80%, 90% and 100% in dataset 1: sequences with PROSITE pattern, and database 2: sequences with PROSITE pattern and SWISS-PROT annotations for profile verification. We observed that whether in dataset 1 or dataset 2, the hit rate of PROSITE patterns are all higher than pattern candidates. Thus, the PROSITE pattern is really meaningful for protein sequences which have these PROSITE patterns.

However, we also observed that the hit rates in dataset 2 are generally higher than hit rates in dataset 1. Because dataset 1 only contains sequences with PROSITE patterns but database 2 contains sequences with PROSITE pattern and SWISS-PROT annotations, it tell us that the profiles we generated may be more meaningful for protein sequences with “motor protein” annotations in SWISS-PROT database.

Second, we used profiles of pattern candidates and PROSITE patterns of motor proteins to search for SWISS-PROT protein sequences that might have these patterns; and we suppose the protein sequences with these pattern candidates may be motor proteins. We use all profiles of identified pattern candidates to search all protein sequences in SWISS-PROT database and give each sequence a profile scoring score. The given profile score is the highest score of all profiles search. In this way, we can get a profile scoring list in motor protein prediction (Figure 15). The sequences with higher profile scoring score have higher possibility to be

motor proteins. When one protein sequence has high profile score but not have “motor” annotations in SWISS-PROT database, we regard this protein might be an motor protein because it contains this pattern candidate.

Figure 15 shows the profile scoring list of protein function prediction in motor proteins. Two points must be mentioned. First, the framed sequences all have “motor protein” annotations; because these sequences all match novel pattern candidates, we regard this pattern candidate is a new pattern of motor proteins. Second, the non-labeled sequences are the sequences that match profiles but don’t have “motor protein” annotations in SWISS-PROT database; hence these proteins might be motor proteins that not identified yet.

In Table 11, we summarized the true-positive rates, profile scoring scores, and z-score of profile scoring scores of top 10, 50, 100, 150, 200, 250 and 300 ranked sequences in profile scoring ranking list. We also compared the hit rates between profiles of pattern candidates and PROSITE patterns. We observed when protein sequences with profile scoring score 0.875, the true positive rate is 91.00% and the z-score is 5.76. Thus when protein sequences with profile scoring score higher than 0.875, we can say these protein sequence may be motor proteins with 91.00% confidence.

When comparing the hit rate between profiles of pattern candidates and PROSITE patterns, we observed that all the top 300 ranked protein sequences with “motor protein” annotations were all searched by pattern candidates. Although some of pattern candidates may partially overlap with PROSITE patterns, it tells us that the pattern candidates are useful for protein function prediction in motor proteins.

3.3 HEM-binding proteins

3.3.1 Overview

Heme is a member of a family of compounds called porphyrins, which consist of four pyrrole rings. Heme metabolism is an important metabolic pathway because many important hemoproteins contain heme as a prosthetic group. For example, hemoglobin is a very important hemoprotein and it is an oxygen carrier in the blood. There are also cytochromes, which participate in important electron transfer reactions, and tryptophan oxygenase which is a hemoprotein of intermediary metabolism. Therefore, we choose HEM-binding proteins as the third research target in our research.

We have also generated structure similarity matrix of non-redundant HEM-binding domains for functional-based domain clustering, and we also identified conservation residues and pattern candidates. Finally, we used profiles of pattern candidates to undergo protein function prediction.

3.3.2 Structure similarity matrix of HEM-binding domains

Figure 16 shows the structure similarity matrix and SCOP classifications of non-redundant HEM-binding domains. Because the structure similarity matrix of all the non-redundant HEM-binding domains (131 non-redundant domains) is too large, we only choose structure similarity matrix with 40 HEM-binding domains. The protein domains with higher structure similarity are also clustered together and always belong to same SCOP families..

Figure 16 is meaningful. Because ATP and ADP are similar in three-dimensional structure, structure similarity matrixes of these two kinds of ligand-binding proteins only tell

us that our approach, MuLiSA, can apply to ATP-binding domains and ADP-binding domains. However, because HEM structure is different from ATP and ADP, and the structure similarity matrix is still similar with SCOP classification, we have confidence that our approach, MuLiSA, can apply to different kinds of ligand-binding proteins.

3.3.3 Conservation residues identified from HEM-binding domains

We have also identified several conservation residues of protein domain clusters in HEM-binding proteins. In Table 12, conservation residues were identified and listed; the bold residues are residues that were announced on literature that are important in HEM-binding or conformation stability [41-92].

Figure 17 shows the multiple ligand-bound structure alignment result and identified conservation residues in "Cytochrome b5 family" of HEM-binding domains. In Figure 17(A), the identified conservation residues are closed to heme in three-dimensional space. It implies that these conservation residues may play important role in HEM-binding. In Figure 17(B), the labeled residue numbers were belonged to protein domain d1cyo__, which is the selected alignment center C of this cluster, and the red framed region means the PROSITE patterns. We observed that most identified conservation residues were on these region, it also tell us that identifying pattern candidates from conservation residues extension may be a reasonable approach.

3.3.4 Pattern candidates identified from HEM-binding domains

We have identified pattern candidates of "CCP-like family", "Cytochrome P450 family", "Cytochrome b5 family", "monodomain cytochrome c family" and "monodomain cytochrome c family" of HEM-binding domains. Table 13 lists these pattern candidates. Table 14 shows

the comparison of PROSITE patterns and our defined pattern candidates that overlap with PROSITE patterns of HEM-binding domains. These pattern candidates are partially overlapping with PROSITE patterns. However, the new pattern candidates which do not overlap with PROSITE patterns in Table 14, may be new clues to search for HEM-binding proteins. All of these pattern candidates were also used to generate profiles and we will use these profiles for protein function prediction.

3.3.5 Profile verification and protein function prediction of HEM-binding proteins

In Figure 18, we observed that pattern candidate is better than PROSITE pattern. Although this pattern candidate partially overlaps with this PROSITE pattern, it means that the pattern candidates may be more meaningful than PROSITE pattern for protein sequences with "Heme" annotations in SWISS-PROT database; and because the profile of PROSITE pattern is generated from our alignment, it also proved that the profile generated from our alignments is convincing.

In order to verify the effectiveness of profiles generated from our alignments in protein function prediction, we also compare the performance in profile search between datasets 1, which contains protein sequences with PROSITE pattern; and dataset 2, which contains protein sequences not only with PROSITE pattern but also have "Heme" annotations in SWISS-PROT database. In Figure 19, dataset 1 contains protein sequences contain PROSITE pattern: cytochrome b5 family, heme-binding domain signature and dataset 2 contains protein sequences not only contain PROSITE pattern but also have "Heme" annotations in SWISS-PROT database. We observed that the area under curves of dataset 2 is larger than area under curves of dataset 1. Because the profiles of pattern candidates were generated from HEM-binding domains alignments and the protein sequences in dataset 1 are not all have "Heme" annotations in SWISS-PROT database, we suppose that the profile of pattern

candidate is more meaningful in HEM-binding proteins but not proteins only with PROSITE pattern.

In Table 5, we summarized the average hit rate of true positive rate 50%, 60%, 70%, 80%, 90% and 100% in dataset 1: sequences with PROSITE pattern, and database 2: sequences with PROSITE pattern and SWISS-PROT annotations for profile verification. We observed that although most hit rates in dataset 1and dataset 2 of PROSITE patterns are all higher than our defined pattern candidates, there is an pattern candidate with higher hit rate than PROSITE pattern. Although this pattern candidate partially overlaps with this PROSITE pattern, it means that the pattern candidates may be more meaningful than PROSITE pattern for protein sequences with "Heme" annotations in SWISS-PROT database; and because the profile of PROSITE pattern is generated from our alignment, it also proved that the profile generated from our alignments is convincing.

Second, we use profiles of pattern candidates of HEM-binding proteins to search for SWISS-PROT protein sequences that might have these pattern candidates, and we suppose that the protein sequences with these pattern candidates may be the HEM-binding proteins.

In Figure 20, we also observed there are seven protein sequences which match the pattern candidates but not have "Heme" annotations in SWISS-PROT database, hence these seven proteins might be HEM-binding proteins but not identified yet.

In Table 16, we summarized true-positive rates, profile scoring scores, and z-score of profile scoring scores of top 100, 200, 300, 400, 500, 600 and 700 ranked sequences in profile scoring ranking list. We also compared with the hit rate of pattern candidates and PROSITE patterns. We observed that when protein sequences with profile scoring score 0.744, the true positive rate is 80.50% and the z-score is 4.00. Thus when protein sequences with profile scoring score higher than 0.744, we can say these protein sequences may be HEM-binding proteins with 80.50% confidence.

When comparing the hit rate between pattern candidates and PROSITE patterns, we

observed that almost all the top 700 ranked protein sequences with annotations were searched by pattern candidates. Although some of pattern candidates may partially overlap with PROSITE patterns, it tells us the pattern candidates are useful in protein function prediction of HEM-binding proteins.

3.4 Tool comparison: multiple ligand-bound structure alignments is better than CE and CLUSTALW

Because multiple ligand-bound structure alignments only focus on ligand-binding sites, we neglect noise from protein structure apart from the ligand-binding sites and get the functional-dependent alignments of ligand-binding domains. Figure 21 and Figure 22 shows two examples: the multiple ligand-bound structure alignments are better than famous sequence and structure alignment tools, CLUSTALW and CE. We used PROSITE patterns as the benchmark of alignments.

In Figure 21(A), we find that only the alignments of MuLiSA can align together the PROSITE defined patterns together (PROSITE pattern: P-x(0,2)-[GSTAN]-[DENQGAPK]-x-[LIVMFP]-[HT]-[LIVMYAC]-G-[HNTG]-[LIVMFYSTAGPC]) of two domains, d1maua_ and d1gtra2. In Figure 21(B) and 21(C), we find that the shift of conservation patterns of CE alignment result. In fact, for CE uses only protein structure information to undergo structure alignment, we find that in this case the bad alignment of conservation patterns was because of a huge structure similar region apart from ATP-binding site, and it did disturb the alignment of PROSITE patterns. In other words, through ligand superimposition can only focus on ligand-binding sites and disperse noises from other region, thus the identified conservation residues and patterns will be much more related to ligand-binding.

In Figure 22 shows another remarkable example when protein domains belong to

different SCOP folds classifications. There are 23 domains belongs to “monodomain cytochrome c family” and 5 domains belong to “cytochrome c' family”. We find that CE and CLUSTALW both can’t align the PROSITE patterns together when domains belong to different SCOP fold; however, MuLiSA aligned these PROSITE patterns well. In other words, in spite of protein domains belong to different SCOP fold, alignments focusing on ligand-binding site through ligand superimposition can help us to discover conservation residues and patterns at ligand-binding sites.

Chapter 4

Conclusions

4.1 Summary

We have applied MuLiSA to three kinds of ligand-binding proteins; they are ATP-binding proteins, ADP-binding proteins, and HEM-binding proteins. We have identified several conservation residues and pattern candidates. We have also proved that these identified conservation residues may play important role in ligand-binding or binding site conformation stability. We also find that protein sequences with PROSITE patterns of ligand correlated signatures not necessarily have annotations in SWISS-PROT database, and the hit rate of dataset difference shows that the profiles we generated have higher hit rate in dataset contains sequences with PROSITE pattern and SWISS-PROT annotations; it means that the profiles we generated from ligand-binding proteins with three-dimensional structures is meaningful for protein sequences with SWISS-PROT annotations. When we use these profiles to predict protein functions, we find that protein sequences with profile scoring score 0.744 in HEM-binding proteins prediction have 80.50 % chance to be HEM-binding proteins, protein sequences with profile scoring score 0.875 in motor proteins prediction have 82.27 % chance to be motor proteins, and protein sequences with profile scoring score 0.600 in ATP-binding proteins prediction have 82.27 % chance to be ATP-binding proteins.

Also, we find that in protein function prediction about ATP-binding proteins, motor proteins and HEM-binding proteins, the coverage rates of pattern candidates are 23.51%, 47.64%, and 13.60%; and the prediction accuracy of kinesin proteins is 86.49% (Table 17). Because of the prediction accuracyof kinesin proteins is high, we think the reason of worse

prediction accuracy of ATP-binding proteins, motor proteins and HEM-binding proteins might be the dataset we used to identify pattern candidates is too small; therefore pattern candidates of other proteins were not identified. However, recent developments in X-ray crystallography and NMR have made it faster in solving protein structures. In the near future, when there are more ATP-binding proteins, motor proteins and HEM-binding proteins three-dimensional structures, we can broaden the training dataset and we believe that we can identify more pattern candidates and increase the prediction accuracy.

Table 18 list 10 protein sequences in profile scoring lists of protein function prediction in ATP-binding proteins, motor proteins and HEM-binding proteins, and they may have potentials to be certain ligand-binding proteins.

4.2 Major Contributions

We have developed MuLiSA, a multiple ligand-bound structure alignment technique, based on functional-dependent ligand information to evaluate residue and pattern conservation. The main difference between our tool and others is that we first superimpose the ligands of proteins but not protein itself. In this way, the ligand-binding sites are superimposed naturally. Then we could identify the conservation residues and pattern candidates according to these positions and segments which were superimposed along with ligands. Although the prediction accuracy in ATP-binding proteins and HEM-binding proteins is not good, we proposed a novel tool to identify ligand-binding specificity-determining residues in a different way. This tool may help researchers to looking for functional important residues and do further research.

4.3 Future works

There are still works to do to improve MuLiSA.

First, the alignment algorithm must be improved. In present results, we observed that sometimes there are gaps gapped in well-aligned segments, and the conservation patterns always forms secondary structure segment. To solve this problem, adding secondary structure information and prevent daps in well-aligned segment in the alignment algorithm may be a practicable solution to improve the alignment results.

Second, if we can find proteins binding similar compounds in different biochemical reaction step, such as ATP, ADP and AMP binding by same proteins, through multiple ligand-bound structure alignments, we can observe the residue variation in a reaction, and it may help us to make it clear that the importance and the role of function-dependent residues in a continuous reaction.

Third, multiple ligand-bound structure alignments can be modified to be an “active site-based multiple structure alignments”. When we knows functional important region of proteins, we can superimposed these region and identified more functional important residues for further research.

Table 1. Statistics of proteins, domains and pattern candidates.

Ligand name	No. of proteins [a]	No. of selected domains [b]	No. of non-redundant domains	Clusters [c]	Selected alignment center C	No. of important residues	No. of pattern candidates
ATP	173	60	46	Protein kinases catalytic subunit (7)	d1phk__	10	1
				Class I aminoacyl-tRNA synthetases (RS), catalytic domain (4)	d1maua_	16	3
ADP	302	140	73	motor proteins (8)	d1goja_	8	4
Heme	1145	860	131	CCP-like (13)	d1llp__	11	3
				Cytochrome P450 (13)	d1eupa_	12	3
				Cytochrome b5 (5)	d1cyo__	4	1
				Monodomain cytochrome c (23)	d1i54a_	3	1
				Cytochrome c' (4)	d1i54a_ [d]	3	1

[a] Number of ligand-binding proteins in PDBsum database.
[b] Number of ligand-binding domains selected by our program.
[c] The domain clusters that according to structure similarity and SCOP database classification; the domain names are based on SCOP database nomenclature. We only choose domain clusters with domain number > 3 because the alignments are more statistical meaningful; and we only choose domain clusters with PROSITE patterns because we need benchmarks to verify our results. The numbers in the parentheses are the non-redundant domain numbers of each cluster.
[d] We choose same alignment center C of domain clusters: monodomain cytochrome c and cytochrome c', because same pattern candidates were identified in these clusters.

Table 2. Conservation residues identified from ATP-binding domains.

Family [a]	Domain	Conservation residues [b]															
Protein kinases catalytic subunit	d1phk__	G26	A46	L111	D149	K151	P152	N154	L156	D167	T186						
	d1atpe_	G50	A70	L128	**D166** [c]	K168	P169	**N171**	L173	**D184**	T201						
	d1qmza_	G11	A31	L87	**D127**	**K129**	P130	**N132**	L134	**D145**	**T165**						
	d1csn__	G19	A39	L92	**D131**	**K133**	P134	N136	L138	D154	T181						
	d1hck__	G11	**A31**	L87	D127	**K129**	P130	**N132**	L134	**D145**	T165						
	d1gol__	G30	A50	L110	**D147**	**K149**	P150	N152	L154	**D165**	T188						
	d1h1wa_	G89	A109	L167	D205	K207	P208	N210	L212	D223	T245						
	Z-score [e]	**2.980**	**2.980**	**2.980**	**2.980**	**2.980**	**2.980**	**2.980**	**2.980**	**2.980**	**2.980**						
Class I aminoacyl-tRNA synthetases (RS), catalytic domain	d1maua_	P10	**G17**	L23	D41	S81	**Y125**	**D132**	L135	P172	V179	**K192**	**M193**	S194	**K195**	L206	L272
	d1n77a2	[d]	G17	L23		T186	D194	L197	P228	G274	**K243**			**H15**	L253		
	d1gtra2[+]	P35	G42	I47	D67	S100	Y211	L221	P253	G314		M268	S269	K270	L39	L327	
	d1h3ea1	P46	G54	L59	D78	S129	Y175	D182	V184		G18	K232	M233	S234	K235	L243	L292
	Z-score	**2.879**	**5.218**	**2.879**	**2.879**	**2.879**	**2.879**	**2.879**	**2.879**	**2.879**	**2.879**	**2.879**	**2.879**	**2.879**	**2.879**	**5.218**	**2.879**

[a] The SCOP database families.
[b] Conservation residues identified by MuLiSA with z-score of entropy calculation > 2.5.
[c] Bold residues are residues that were announced on literature which are important in ligand-binding or conformation stability.
[d] The spare spaces are gaps in the alignments.
[e] The position z-score of entropy calculation.
[+] The reference of this protein domain was not found; hence no residues were labeled.

Table 3. Pattern candidates identified from ATP-binding domains

Family	Domain	Pattern candidates [a]		
		1 [b]	2	3
Protein kinases catalytic subunit	d1phk__	150..... RDLKPENILL RDLKPEN-LI RDLKPQNLLI RDIKPDNFLI -DLKPQNLLI RDLKPSNLLL FDLKPENILL + ++ + +	.....17 IKLTDFG IQVTDFG IKLADFG IYVVDFG IKLADL- LKICDFG I-ITDFG +	..190.... TPSYLAPEI TPEYLAPEI TLWYRAPEI TA-YM--R- T--YL-P-L TRWYRAP-I T-A-VLLK- +
	d1atpe_			
	d1qmza_			
	d1csn__			
	d1hck__			
	d1gol__			
	d1h1wa_			
Class I aminoacyl-tRNA synthetases (RS), catalytic domain		1		
	d1maua_	....20... TIGNYIGAL TVGTYL-IL HIGH-A--I HLGH-AVVL + +		
	d1n77a2			
	d1gtra2+			
	d1h3ea1			

Table 4. Comparison of PROSITE patterns and pattern candidates of ATP-binding domains

Family	Domain	PROSITE patterns [a]		Pattern candidates [b]
Protein kinases catalytic subunit		PS00107 Protein kinases ATP-binding region signature. [LIV]-G-{P}-G-{P}-[FYWMGSTNH]-[SGA]-{PW}-[LIVCAT]-{PD}-x-[GSTACLIVMFY]-x(5,18)-[LIVMFYWCSTAR]-[AIVP]-[LIVMFAGCKR]-K. [c]	PS00108 Serine/Threonine protein kinases active-site signature. [LIVMFYC]-x-[HY]-x-D-[LIVMFY]-K-x(2)-N-[LIVMFYCT](3).	1
	d1phk__ d1atpe_ d1qmza_ d1csn__ d1hck__ d1gol__ d1h1wa_	....30.....40..... LGRGVSSVVRRCIHKPTCKEYAVK LGTGS-GRVMLVKHKESGNHYAMK IGEG-T-VVYKARNKLT-EVVALK IGEGSFGVIFEG--K----QVAIK IG-GTY-VVYKARNKLTGEVVALK IG---Y-EV-SC---PN--R-AIR LGEGS-S-V-LARELATSREYAIK + +	...150..... IVHRDLKPENILL LIYRDLKPEN-LI VLHRDLKPQNLLI -V-RDIKPDNFLI --H-DLKPQNLLI I-HRDLKPSNLLL V-RFDLKPENILL + ++ + +	150..... RDLKPENILL RDLKPEN-LI RDLKPQNLLI RDIKPDNFLI -DLKPQNLLI RDLKPSNLLL FDLKPENILL + ++ + +
Class I aminoacyl-tRNA synthetases (RS), catalytic domain		PS00178 Aminoacyl-transfer RNA synthetases class-I signature P-x(0,2)-[GSTAN]-[DENQGAPK]-x-[LIVMFP]-[HT]-[LIVMYAC]-G-[HNTG]-[LIVMFYSTAGPC].		1
	d1maua_ d1n77a2 d1gtra2 d1h3ea1	0.....2 PSGVITIGNY --A--TVGTY PNGY-HIGH- PT--LHLGH- + +		....20... TIGNYIGAL TVGTYL-IL HIGH-A--I HLGH-AVVL + +

Table 5. Hit rate comparison of dataset difference in profile verification of ATP-binding proteins

Family	PROSITE patterns and pattern candidates [a]	Dataset 1 [c]		Dataset 2 [d]	
		No. of sequence	Hit rate [e]	No. of sequence	Hit rate
Protein kinases catalytic subunit	Protein kinases ATP-binding region signature	859	85.15%	773	89.18%
	Serine/ Threonine protein kinases active-site signature.		85.73%		86.67%
	Pattern candidate 1 [b]		84.79%		86.76%
	Pattern candidate 2		64.19%		68.35%
	Pattern candidate 3		71.37%		75.43%
Class I aminoacyl-tRNA synthetases (RS), catalytic domain	Aminoacyl-transfer RNA synthetases class-I signature	1129	26.61%	1056	50.42%
	Pattern candidate 1		20.18%		37.43%

[a] PROSITE patterns and pattern candidates that we identified.
[b] Pattern candidate 1 of "Protein kinases catalytic subunit family"(see Table 3).
[c] Dataset 1: sequences only with PROSITE patterns
[d] Dataset 2: sequences with PROSITE patterns and SWISS-PROT annotations
[e] Average hit rate when true positive rate are 50%, 60%, 70%, 80%, 90% and 100%.

Table 6. Hit rate comparison of pattern candidates and PROSITE patterns in protein function prediction of ATP-binding proteins

No. of top ranked sequence [a]	True-positive rate	Profile scoring score	Z-score of profile scoring score [b]	Hit rate of all pattern candidates	Hit rate of PROSITE pattern
100	100.00% (100)	0.840	6.52	100.00% (100)	0.00% (0)
500	98.40% (492)	0.720	4.70	100.00% (492)	0.00% (0)
1000	95.70% (957)	0.650	3.63	99.79% (955)	0.21% (2)
1500	82.27% (1234)	0.600	2.87	97.65% (1205)	2.35% (29)
2000	76.65% (1533)	0.583	2.61	80.43% (1503)	19.57% (30)
2500	70.28% (1757)	0.567	2.37	94.25% (1656)	5.75% (101)
3000	61.53% (1846)	0.556	2.20	94.53% (1745)	5.47% (101)

[a] The top ranked sequence number. For example, 100 in this column means the 100 ranked sequences with highest profile scoring score in profile scoring ranking list of ATP-binding protein prediction.

[b] Z-score of profile scoring scores. The average of all SWISS-PROT sequence scores is 0.411515; the standard deviation of all SWISS-PROT sequence scores is 0.065701.

Table 7. Conservation residues identified from ADP-binding domains

Family	Domain	Conservation residues							
motor proteins	d1goja_	**P16**	**G88**	G93	K94	S206	**D235**	**G238**	**E240**
	d1bg2__	**P17**	G85	G90	K91	**S202**	D231	**G234**	**E236**
	d1br2a2	P126	G177	G182	K183	**S179**	**D465**	G468	E178
	d2ncda_	P357	G434	G439	G440	S551	D580	G583	E585
	d1f9ta_	P395	G474	G479	K480	S597	**D626**	G629	E631
	d1i5sa_	P14	G97	G102	K103	**S215**	D248	**G251**	E253
	d1lkxa_	P50	G101	G106	K107	S103	D386	G389	E102
	d2kin.1	P17	G86	G91	K92	S203	D232	G235	E237
	Z-score	**3.029**	**3.029**	**3.029**	**3.029**	**3.029**	**3.029**	**3.029**	**3.029**

Table 8. Pattern candidates identified from ADP-binding domains

Family	Domain	Pattern candidates			
		1	2	3	4
motor proteins	d1goja_	0..... VVARFRP VMCRFRP -IC---P VFCRIRP V--R-RP VAVRVRP VVI---P VMCRFRP +	.....90.....100 VFAYGQTGAGKSYTMMG IFAYGQTSSGKTHTMEG I---G--GAGKTET-N- VFAYGQTGSGKTYTM-G IFAYGQTGSGKTFTML- IFAYGQTGAGKSYTMMG V-ISG--GAGKTEAS-- IFAYGQTSSGKTHTMEG + ++	..210.. SRSHSIF SRSHSIF SI--T-I SR-H--- SRSHSIF SRSHAVF SI--KY- SRSHSIF +	.....240 LVDLAGSE LVDLAGSE ILDIAG-E LVDLAGSE LVDLAGSE LVDLAGSE MLDI-G-E LVDLAGSE + + +
	d1bg2__				
	d1br2a2				
	d2ncda_				
	d1f9ta_				
	d1i5sa_				
	d1lkxa_				
	d2kin.1				

Table 9. Comparison of PROSITE patterns and pattern candidates of ADP-binding domains

Family	Domain	PROSITE patterns	Pattern candidates
motor proteins		PS00411 Kinesin motor domain signature. [GSA]-[KRHPSTQVM]-[LIVMF]-x-[LIVMF] -[IVC]-D-L-[AH]-G-[SAN]-E.	4
	d1goja_	30.....240 GQLFLVDLAGSE GKL-LVDLAGSE --IRILDIAG-E LLI-LVDLAGSE H-I-LVDLAGSE --ISLVDLAGSE G-M-MLDI-G-E GNLYLVDLAGSE + + +	.....240 LVDLAGSE LVDLAGSE .ILDIAG-E LVDLAGSE LVDLAGSE LVDLAGSE MLDI-G-E 'LVDLAGSE + + +
	d1bg2__		
	d1br2a2		
	d2ncda_		
	d1f9ta_		
	d1i5sa_		
	d1lkxa_		
	d2kin.1		

Table 10. Hit rate comparison of dataset difference in profile verification of motor proteins

Family	PROSITE patterns and pattern candidates [a]	Dataset 1 [b]		Dataset 2 [c]	
		No. of sequence [d]	Hit rate [e]	No. of sequence	Hit rate
motor proteins	Kinesin motor domain signature	95	99.10%	89	99.50%
	Pattern candidate 1		69.83%		72.47%
	Pattern candidate 2		83.50%		85.24%
	Pattern candidate 3		97.19%		97.56%
	Pattern candidate 4		98.35%		98.75%

[a] PROSITE patterns and pattern candidates that we identified.
[b] Dataset 1: sequences with PROSITE pattern
[c] Dataset 2: sequences with PROSITE pattern and SWISS-PROT “motor protein” annotations.
[d] Number of sequences recorded which have PROSITE patterns in this cluster.
[e] Average hit rate when true positive rate are 50%, 60%, 70%, 80%, 90% and 100%.

Table 11 Hit rate comparison of pattern candidates and PROSITE patterns in protein function prediction of motor proteins

Top number of sequence [a]	True-positive rate	Profile scoring score	Z-score of profile scoring score [b]	Hit rate of all pattern candidates	Hit rate of PROSITE pattern
10	100% (10)	1.000	7.44	100.00% (10)	0.00% (0)
50	100% (50)	1.000	7.44	100.00% (50)	0.00% (0)
100	91.00% (91)	0.875	5.76	100.00% (91)	0.00% (0)
150	66.00% (99)	0.750	4.08	100.00% (99)	0.00% (0)
200	50.50% (101)	0.750	4.08	100.00% (101)	0.00% (0)
250	40.80% (102)	0.750	4.08	100.00% (102)	0.00% (0)
300	34.00% (102)	0.667	2.97	100.00% (102)	0.00% (0)

[a] The top ranked sequence number.
[b] Z-score of profile scoring scores. The average of all SWISS-PROT sequence scores is 0.445968; the standard deviation of all SWISS-PROT sequence scores is 0.074513.

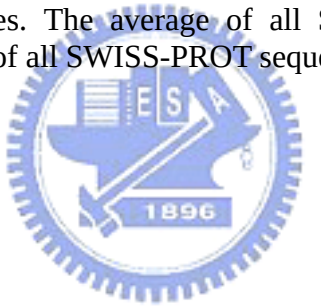


Table 12. Conservation residues identified from HEM-binding domains

Family	Domain	Conservation residues										
CCP-like	d1llp__	**R43**	**H47**	**G66**	D107	G131	R132	P145	V170	**H176**	F200	**D238**
	d1cca__	R48	H52	G65	D106	G129	R130	P145	V169	**H175**	F198	**D235**
	d1oafa_	**R38**	**H42**	G55	D95	G118	R119	P132	V157	**H163**	F186	**D208**
	d1hsr__	**R52**	**H56**	G75	D116	G140	R141	**P154**	V178	H184	F208	D246
	d1h5ma_	**R38**	**H42**	G48	D99	G122	R123	**P139**	V164	**H170**	F229	**D247**
	d1b80a_	R43	**H47**	G66	D107	G131	R132	P145	V170	**H176**	F200	D238
	d1bgp__	**R45**	**H49**	**G55**	D108	G131	R132	P149	V173	**H179**	F232	**D250**
	d1mnp__	R42	H46	G62	D104	G128	R129	P142	V167	H173	F197	D242
	d1fhfa_	**R38**	**H42**	G48	D99	G122	R123	**P139**	V163	**H169**	F228	D246
	d1pa2a_	**R38**	**H42**	G48	D99	G122	R123	**P139**	V163	**H169**	F228	D246
	d1qgja_	R38	H42	G48	D96	G119	R120	P135	V159	H165	F224	D242
	d1qpaa_	**R43**	**H47**	**G66**	D107	G131	R132	P145	V170	**H176**	F200	**D238**
	d1scha_	**R38**	**H42**	**G48**	D99	G122	R123	**P139**	V163	**H169**	F221	**D239**
	Z-score	**2.662**	**2.662**	**2.662**	**2.662**	**2.662**	**2.662**	**2.662**	**2.662**	**2.662**	**2.662**	**2.662**

Table 12. Continued.

Family	Domain	Conservation residues											
Cytochrome P450	d1eupa_	**A241**	G242	E280	R283	D322	R336	F344	G345	G347	H349	C351	G353
	d1dz4a_	V247	G248	E287	R290	D328	R342	F350	G351	G353	H355	**C357**	G359
	d1bu7a_	A264	G265	E320	R323	D363	R378	F393	G394	G396	R398	C400	G402
	d1cpt__	A267	G268	E306	R309	**D348**	R362	F370	G371	G373	**H375**	C377	G379
	d1n6ba_	**A294**	**G295**	E351	R354	D394	H408	F425	S426	G428	R430	C432	G434
	d1e9xa_	A256	G257	313	R316	I355	R369	F387	G388	G390	H392	C394	G396
	d1ehea_	A239	G240	E278	R281	**D321**	R335	F345	G346	G348	H350	C352	A354
	d1gwia_	**A242**	**G243**	E281	R284	D324	**R339**	F348	G349	G351	H353	**C355**	G357
	d1io7a_	A209	G210	E246	R249	D288	R302	F310	G311	G313	H315	C317	G319
	d1izoa_			E282	R285	D324	R338	**Q352**	G353	G355	**H341**	**C363**	**G365**
	d1lfka_	**A236**	G237	E275	R278	D318	R332	F340	G341	G343	H345	**C347**	G349
	d1n4ga_$^{+}$	A233	G234	E272	R275	D315	R329	F338	G339	G341	H343	C345	G347
	d1n97a_	A221	G222	E260	R263		R314	F329	G330	G332	R334	C336	G338
	Z-score	**2.705**	**3.208**	**3.723**	**3.723**	**2.705**	**3.208**	**3.208**	**3.208**	**3.723**	**2.697**	**3.723**	**3.208**
Cytochrome b5	d1cyo__	E11	T33	H39	H63								
	d1b5m__	E11	T33	**H39**	**H63**								
	d1cxya_	E13	T58	**H70**	**H42**								
	d1icca_	E11	T55	**H63**	H39								
	d1mj4a_	E10	T34	H40	H65								
	Z-score	**3.253**	**3.253**	**3.253**	**3.253**								

Table 12. Continued.

Family	Domain	Conservation residues		
monodomain cytochrome c	d1i54a_	C14	**C17**	**H18**
	d1cot__	**C15**	**C18**	**H19**
	d1c75a_	**C32**	**C35**	**C36**
	d1c2ra_	**C13**	**C16**	**H17**
	d1c52__	**C11**	**C14**	**H15**
	d1c6ra_	**C15**	**C18**	**H19**
	d1cc5__$^{+}$	C19	C22	H23
	d1ccr__	C22	**C25**	**H26**
	d1ycc__	C14	C17	H18
	d1co6a_	**C13**	**C16**	**H17**
	d1ctj__	**C15**	**C18**	**H19**
	d1cxc__	C15	C18	H19
	d1cyi__	**C14**	**C17**	H18
	d1dw0a_	**C43**	C46	**H47**
	d1f1fa_	**C14**	**C17**	H18
	d1fj0b_	**C13**	**C16**	**H17**
	d1gdva_	**C14**	**C17**	**H18**
	d1hroa_	**C19**	**C22**	**H23**
	d1jdla_	**C15**	**C18**	**H19**
	d1ls9a_	**C17**	**C20**	**H21**
	d2mtac_$^{+}$	C57	C60	H61
	d3c2c__$^{+}$	C14	C17	H18
	d351c__	**C12**	**C15**	**H16**
Cytochrome c'	d1a7va_	**C113**	**C116**	**H117**
	d1bbha_	**C121**	**C124**	**H125**
	d1cgo__	**C116**	**C119**	**H120**
	d1cpq__	**C118**	**C121**	**H122**
	Z-score	**3.505**	**3.505**	**3.505**

Table 13. Pattern candidates identified from HEM-binding domains

Family	Domain	Pattern candidates		
CCP-like		1	2	3
	d1llp__	...... RLVFHD RLAWH- RLAWHS RIVFHD RLHFHD RLVFHD RLHFHD RLHFHD RLTFHD RLHFHD RLHFHD RMVFHD RL-FHD + +	170..... ELVWMLSAH EVVALMGAH DIVALS-GH EVVDLLAAH DLVALS-GH ELV-MLSAH DLVTIS-GH DLVTLS-GH EVVSLLASH DLVALS-AH DVVALS-AH ETV-LLSAH ELVTLS-AH + +	.....200. PFDSTPGIFD PWN--NVAF- PW-S-PLIFD PLDSTPQVFD DFD-TPTIFD PFDSTPGIFD VLD-TPNVFD NLDSTPDQFD PFDSTPFTFD NLDSTPDAFD PLDST-DTFD PFDSTPGQFD PFDTTP--FD +
	d1cca__			
	d1oafa_			
	d1hsr__			
	d1h5ma_			
	d1b80a_			
	d1bgp__			
	d1mnp__			
	d1fhfa_			
	d1pa2a_			
	d1qgja_			
	d1qpaa_			
	d1scha_			
Cytochrome P450		1	2	3
	d1eupa_	240.....250 LLLAGFEASVSLI LLVGG-DTVVNFL FLIAGHETTSGLL IATAGHDTTSSSS LFGAGTETTSTTL LMFAG-HTSSGTA LLVAGNATMVNMI MVAAGHETTISLI LLIAGNETTTNLI VL---V-AI-YFL VMLAG-DNIS-MI FFGAGVISTGS-L LLVAGHETVASAL ++	280... VEEILR CEELLR LNEALR VDEAVR I-EIQR LKETLR VEELCR VEETLR IEEALR VQEV-R VDELIR VEELLR FQEALR + +	....350.....360 FGQGIHFCMGRPLAKLE FGHGSHLCLGQHLARRE FGNG-RACIGQQFALHE FGWGAHMCLGQ-LAKLE FSAGKRMCVGE-LARME FGAGRHRCVGAAFAIMQ FGFGDHRCIAH-LAKIE FGHGPHVCPGAALSRME FGSGIHLCLGAPLARLE QGGGGHRCPGE-ITIV- FGHGVHHCLGAALARLE FGRGQHFCPGS-LGRRH FGLGQRLCLGR--ALLE ++ + + + +
	d1dz4a_			
	d1bu7a_			
	d1cpt__			
	d1n6ba_			
	d1e9xa_			
	d1ehea_			
	d1gwia_			
	d1io7a_			
	d1izoa_			
	d1lfka_			
	d1n4ga_			
	d1n97a_			
Cytochrome b5		1		
	d1cyo__	......40.. TKFLEEHPGG TRFLSEHPGG T-W--EH-AA T-F-EDH-A- T-F-D-HPGG + +		
	d1b5m__			
	d1cxya_			
	d1icca_			
	d1mj4a_			

Table 13. Continued.

Family	Domain	Pattern candidates
monodomain cytochrome c		1
	d1i54a_	
	d1cot__	.....
	d1c75a_	CAQCH
	d1c2ra_	CKACH
	d1c52__	CISCH
	d1c6ra_	CKTCH
	d1cc5__	CAGCH
	d1ccr__	CAACH
	d1ycc__	CVNCH
	d1co6a_	CAQCH
	d1ctj__	CLQCH
	d1cxc__	CLVCH
	d1cyi__	CAACH
	d1dw0a_	CQTCH
	d1f1fa_	CAACH
	d1fj0b_	CTTCH
	d1gdva_	CAACH
	d1hroa_	CMTCH
	d1jdla_	CAACH
	d1ls9a_	CITCH
	d2mtac_	CMACH
	d3c2c__	CAACH
	d351c__	CSGCH
Cytochrome c'	d1a7va_	CLACH
	d1bbha_	CVACH
	d1cgo__	CKSCH
	d1cpq__	CASCH
		CAACH
		CKACH
		+ ++

Table 14. Comparison of PROSITE patterns and pattern candidates of HEM-binding domains

Family	Domain	PROSITE patterns		Pattern candidates	
CCP-like		PS00436 Peroxidases active site signature. [SGATV]-x(3)-[LIVMA]-R-[LIVMA]-x-[FW]-H-x-[SAC].	PS00435 Peroxidases proximal heme-ligand signature. [DET]-[LIVMTA]-x(2)-[LIVM]-[LIVMSTAG]-[SAG]-[LIVMSTAG]-H-[STA]-[LIVMFY].	1	2
	d1llp__	.40.....5 AHESIRLVFHDS	170.....1 ELVWMLSAHSV	..... RLVFHD	170..... ELVWMLSAH
	d1cca__	GPVLVRLAWH-S	EVVALMGAHAL	RLAWH-	EVVALMGAH
	d1oafa_	APLMLRLAWHSA	DIVALS-GHTI	RLAWHS	DIVALS-GH
	d1hsr__	VRKILRIVFHDA	EVVDLLAAHSL	RIVFHD	EVVDLLAAH
	d1h5ma_	AASILRLHFHDC	DLVALS-GHTF	RLHFHD	DLVALS-GH
	d1b80a_	AHESIRLVFHDS	ELV-MLSAHSV	RLVFHD	ELV-MLSAH
	d1bgp__	AA-LLRLHFHDC	DLVTIS-GHTI	RLHFHD	DLVTIS-GH
	d1mnp__	GASLMRLHFHDC	DLVTLS-GHTF	RLHFHD	DLVTLS-GH
	d1fhfa_	AHEVIRLTFHDA	EVVSLLASHSV	RLTFHD	EVVSLLASH
	d1pa2a_	GASLIRLHFHDC	DLVALS-AHTF	RLHFHD	DLVALS-AH
	d1qgja_	AASLIRLHFHDC	DVVALS-AHTF	RLHFHD	DVVALS-AH
	d1qpaa_	AHEALRMVFHDS	ETV-LLSAHSI	RMVFHD	ETV-LLSAH
	d1scha_	-ASLLRL-FHDC + +	ELVTLS-AHTI + +	RL-FHD + +	ELVTLS-AH + +
Cytochrome P450		PS00086 Cytochrome P450 cysteine heme-iron ligand signature. [FW]-[SGNH]-x-[GD]-x-[RKHPT]-x-C-[LIVMFAP]-[GAD].		3	
	d1eupa_	....350... FGQGIHFCMG		....350.....360 FGQGIHFCMGRPLAKLE	
	d1dz4a_	FGHGSHLCLG		FGHGSHLCLGQHLARRE	
	d1bu7a_	FGNG-RACIG		FGNG-RACIGQQFALHE	
	d1cpt__	FGWGAHMCLG		FGWGAHMCLGQ-LAKLE	
	d1n6ba_	FSAGKRMCVG		FSAGKRMCVGE-LARME	
	d1e9xa_	FGAGRHRCVG		FGAGRHRCVGAAFAIMQ	
	d1ehea_	FGFGDHRCIA		FGFGDHRCIAH-LAKIE	
	d1gwia_	FGHGPHVCPG		FGHGPHVCPGAALSRME	
	d1io7a_	FGSGIHLCLG		FGSGIHLCLGAPLARLE	
	d1izoa_	QGGGGHRCPG		QGGGGHRCPGE-ITIV-	
	d1lfka_	FGHGVHHCLG		FGHGVHHCLGAALARLE	
	d1n4ga_	FGRGQHFCPG		FGRGQHFCPGS-LGRRH	
	d1n97a_	FGLGQRLCLG ++ + + + +		FGLGQRLCLGR--ALLE ++ + + + +	
Cytochrome b5		PS00191 Cytochrome b5 family, heme-binding domain signature. [FY]-[LIVMK]-x(2)-H-P-[GA]-G		1	
	d1cyo__	....40.. FLEEHPGG		......40.. TKFLEEHPGG	
	d1b5m__	FLSEHPGG		TRFLSEHPGG	
	d1cxya_	W--EH-AA		T-W--EH-AA	
	d1icca_	F-EDH-A-		T-F-EDH-A-	
	d1mj4a_	F-D-HPGG +		T-F-D-HPGG + +	

Table 14. Continued.

Family	Domain	PROSITE patterns	Pattern candidates
monodomain cytochrome c		PS00190 Cytochrome c family heme-binding site signature. C-{CPWHF}-{CPWR}-C-H-{CFYW}.	**1**
	d1i54a_	2	.....
	d1cot__	CAQCHT	CAQCH
	d1c75a_	CKACHM	CKACH
	d1c2ra_	CISCH-	CISCH
	d1c52__	CKTCHS	CKTCH
	d1c6ra_	CAGCH-	CAGCH
	d1cc5__	CAACH-	CAACH
	d1ccr__	CVNCHV	CVNCH
	d1ycc__	CAQCHT	CAQCH
	d1co6a_	CLQCHT	CLQCH
	d1ctj__	CLVCHS	CLVCH
	d1cxc__	CAACH-	CAACH
	d1cyi__	CQTCHV	CQTCH
	d1dw0a_	CAACH-	CAACH
	d1f1fa_	CTTCH-	CTTCH
	d1fj0b_	CAACH-	CAACH
	d1gdva_	CMTCHR	CMTCH
	d1hroa_	CAACH-	CAACH
	d1jdla_	CITCHT	CITCH
	d1ls9a_	CMACHR	CMACH
	d2mtac_	CAACH-	CAACH
	d3c2c__	CSGCH-	CSGCH
	d351c__	CLACHT	CLACH
Cytochrome c'	d1a7va_	CVACH-	CVACH
	d1bbha_	CKSCH-	CKSCH
	d1cgo__	CASCH-	CASCH
	d1cpq__	CAACH-	CAACH
		CKACH-	CKACH
		+ ++	+ ++

Table 15. Hit rate comparison of dataset difference in profile verification of HEM-binding proteins

Family	PROSITE patterns and pattern candidates [a]	Dataset 1 [b]		Dataset 2 [c]	
		No. of sequence	Hit rate [d]	No. of sequence	Hit rate
CCP-like	Peroxidases active site signature.	205	8.86%	151	100.00%
	Peroxidases proximal heme-ligand signature.		8.39%		55.72%
	Pattern candidate 1		4.86%		46.67%
	Pattern candidate 2		6.94%		49.81%
	Pattern candidate 3		4.95%		9.54%
Cytochrome P450	Cytochrome P450 cysteine heme-iron ligand signature.	687	86.05%	675	86.45%
	Pattern candidate 1		65.49%		68.34%
	Pattern candidate 2		38.09%		40.55%
	Pattern candidate 3		86.13%		86.26%
Cytochrome b5	Cytochrome b5 family, heme-binding domain signature.	88	79.43%	78	79.30%
	Pattern candidate 1		77.13%		82.53%
monodomain cytochrome c and Cytochrome c'	Cytochrome c family heme-binding site signature.	1130	87.12%	897	84.08%
	Pattern candidate 1		86.93%		84.02%

[a] PROSITE patterns and pattern candidates that we identified.
[b] Dataset 1: sequences with PROSITE pattern
[c] Dataset 2: sequences with PROSITE pattern and SWISS-PROT annotations
[d] Average hit rate when true positive rate are 50%, 60%, 70%, 80%, 90% and 100%.

Table 16. Hit rate comparison of pattern candidates and PROSITE pattern in protein function prediction of HEM-binding proteins

Top number of sequence [a]	True-positive rate	Profile scoring score	Z-score of profile scoring score [b]	Hit rate of all pattern candidates	Hit rate of PROSITE pattern
100	92.00% (92)	0.798	4.72	100.00% (92)	0.00% (0)
200	80.50% (161)	0.744	4.00	96.27% (155)	3.73% (6)
300	69.00% (207)	0.708	3.52	97.10% (201)	2.90% (6)
400	69.75% (279)	0.692	3.30	87.81% (245)	12.19% (34)
500	70.40% (352)	0.685	3.21	90.34% (318)	9.66% (34)
600	60.33% (362)	0.685	3.21	90.61% (328)	9.39% (34)
700	57.86% (405)	0.669	2.99	91.60% (371)	8.40% (34)

[a] The top ranked sequence number.
[b] Z-score of profile scoring scores. The average of all SWISS-PROT sequence scores is 0.436928; the standard deviation of all SWISS-PROT sequence scores is 0.071717.

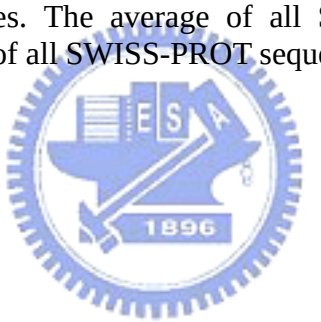


Table 17. Prediction accuracy and coverage rates in protein function prediction

Ligand name	No. of SWISS-PROT annotated protein sequence [a]	No. of true hit of top 100% ranked annotated sequence [b]	PROSITE pattern profile search	All pattern candidates profile search	Novel pattern candidates profile search	Prediction accuracy of all pattern candidates
ATP	13484	3462	8.43% (292)	91.57% (3170)	62.74% (2172)	23.51%
ADP [c]	212	101	0% (0)	100% (101)	7.92% (8)	47.64%
Heme	4111	678	17.55% (119)	82.45% (559)	2.95% (20)	13.60%
Kinesin	111	96	0.00% (0)	100.00% (96)	8.33% (8)	86.49%

[a] Number of annotated SWISS-PROT protein sequences. Annotations: “ATP-binding”, “motor protein”, “Heme” and “kinesin” in “KW” of SWISS-PROT database. There are 151047 protein sequences in SWISS-PROT database.

[b] Number of true hit annotated protein sequences of the top 100% ranked protein sequences. For example, there are 3462 the true hit protein sequence of top 13484 sequences in ATP-binding prediction profile scoring ranking list.

[c] We only choose motor proteins as our protein function prediction target.

Table 18. 10 predicted protein sequences with high scores in profile scoring ranking lists of protein function prediction in ATP-binding proteins, motor proteins and HEM-binding proteins.

	Predicted ATP-binding proteins	Predicted motor proteins	Predicted HEM-binding proteins
1	(295[a]) P27604[b] (Adenosylhomocysteinase[c])	(85) P44531 (Ferric cations import ATP-binding protein fbpC 1)	(62) Q60613 (Adenosine A2a receptor)
2	(304) P25169 (Sodium/potassium-transporting ATPase beta chain)	(105) Q9QYX7 (Piccolo protein)	(82) P29274 (Adenosine A2a receptor)
3	(774) P20357 (Microtubule-associated protein 2)	(108) Q9PU36 (Piccolo protein [Fragment])	(83) Q10024 (Putative diacylglycerol kinase K06A1.6)
4	(855) Q8A407 (Adenosylhomocysteinase)	(111) Q9Y6V0 (Piccolo protein)	(87) P92127 (Variant-specific surface protein VSP4A1 [Precursor])
5	(886) Q92TC1 (Adenosylhomocysteinase)	(118) Q96RT7 (Gamma-tubulin complex component 6)	(96) P55493 (Hypothetical 65.5 kDa protein y4IJ)
6	(882) Q96RU7 (Neuronal cell death inducible putative kinase)	(125) Q9JVP2 (Aminomethyltransferase)	(98) P46616 (Adenosine A2a receptor)
7	(925) P34611 (B-box type zinc-finger protein ncl-1)	(126) P35100 (ATP-dependent Clp protease ATP-binding subunit clpA homolog, chloroplast [Precursor])	(108) Q81MN9 (Polyphosphate kinase)
8	(938) Q9WTQ6 (Neuronal cell death inducible putative kinase)	(127) Q8CFL8 (Zinc finger SWIM domain containing protein 3)	(112) P55019 (Solute carrier family 12 member 3)
9	(947) Q8K4K2 (Neuronal cell death inducible putative kinase)	(131) Q7MDL6 (Adenosine deaminase)	(115) P11413 (Glucose-6-phosphate 1-dehydrogenase)
10	(952) Q89HP6 (Adenosylhomocysteinase)	(146) P00815 (Histidine biosynthesis trifunctional protein)	(121) Q9QZY5 (T-cell surface glycoprotein CD1e [Precursor])

[a] Ranking serial number in profile scoring ranking lists.
[b] SWISS-PROT accession numbers.
[c] Protein name in SWISS-PROT database.

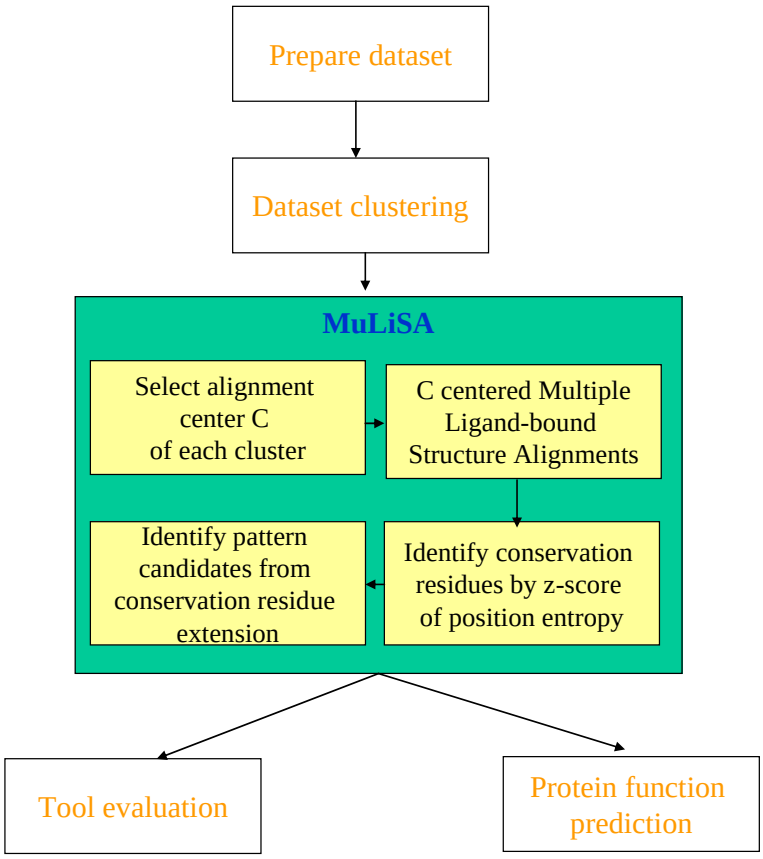


Figure 1. The workflow of analysis and identification of conservation patterns and residues in proteins by MuLiSA. This flow starts from dataset preparation and clustering, followed by multiple ligand-bound structure alignments (MuLiSA), tool evaluation and protein function prediction.

	d1gtra2	d1h3ea1	d1maua2	d1n77a2
d1gtra2	**1**	**0.3**	**0.39**	**0.34**
d1h3ea1	**0.3**	**1**	**0.44**	**0.36**
d1maua2	**0.39**	**0.44**	**1**	**0.36**
d1n77a2	**0.34**	**0.36**	**0.36**	**1**

d1gtra2 ： 0.3 + 0.39 + 0.34 = 1.03

d1h3ea1 ： 1.10

d1maua2 ： 1.19

d1n77a2 ： 1.06

Figure 2. The alignment center C selection. The alignment center C was selected when domain of one cluster which has the highest structure similarity with other protein domains than others. In this case, we select domain d1maua2 as the alignment center C of this cluster.

A

```
                                340.......350.....
d1eupa_.ent                     TRGHLSFGQGIHFCMGRP
d1dz4a_.ent_HEM                 V--HTTFGHGSHLCLGQH
d1bu7a_.ent_HEM                 ----KPFGNG-RACIGQQ
d1cpt__.ent_HEM                 NR-HLGFGWGAHMCLGQ-
d1n6ba_.ent_HEM                 -K--MPFSAGKRMCVGE-
d1e9xa_.ent_HEM    ...........Y---IPFGAGRHRCVGAA....
d1ehea_.ent_HEM                 ---PLGFGFGDHRCIAH-
d1gwia_.ent_HEM                 -R-HISFGHGPHVCPGAA
d1io7a_.ent_HEM                 N-PHLSFGSGIHLCLGAP
d1izoa_.ent_HEM                 --M-IPQGGGGHRCPGE-
d1lfka_.ent_HEM                 ---HVAFGHGVHHCLGAA
d1n4ga_.ent_HEM                 ---HLAFGRGQHFCPGS-
d1n97a_.ent_HEM                 ---YFPFGLGQRLCLGR-
z-score > 2.500                       ++ + + + +
```

B

```
Entropy
ave:-1.959799   sd:0.526345
Position:  336  ResNum:  338   ET:-2.458       z-Score:-0.947
Position:  337  ResNum:  339   ET:-2.311       z-Score:-0.668
Position:  338  ResNum:  340   ET:-2.565       z-Score:-1.150
Position:  339  ResNum:  341   ET:-1.517       z-Score:0.841
Position:  340  ResNum:  342   ET:-1.692       z-Score:0.508
Position:  341  ResNum:  343   ET:-1.479       z-Score:0.913
Position:  342  ResNum:  344   ET:-0.271       z-Score:3.208
Position:  343  ResNum:  345   ET:-0.271       z-Score:3.208
Position:  344  ResNum:  346   ET:-2.205       z-Score:-0.465
Position:  345  ResNum:  347   ET:0.000        z-Score:3.723
Position:  346  ResNum:  348   ET:-2.352       z-Score:-0.745
Position:  347  ResNum:  349   ET:-0.540       z-Score:2.697
Position:  348  ResNum:  350   ET:-1.845       z-Score:0.219
Position:  349  ResNum:  351   ET:0.000        z-Score:3.723
Position:  350  ResNum:  352   ET:-1.479       z-Score:0.913
Position:  351  ResNum:  353   ET:-0.271       z-Score:3.208
Position:  352  ResNum:  354   ET:-1.672       z-Score:0.548
Position:  353  ResNum:  355   ET:-2.205       z-Score:-0.465
```

Figure 3. Identification of conservation residues at positions with z-score > 2.5. (A) The multiple alignments of four protein sequences. (B) The entropy and z-score values of each position. Figure 2(A) shows the alignment results of 13 protein sequences belongs to “Cytochrome P450 family”. The numbers on the top of Figure 2(A) are the residue numbers of d1eupa_. The “+” symbols denotes the positions with z-score > 2.5, and we can observe in Figure 2(B) that these positions are with z-scores 3.208 and 3.723. The framed region is the possible pattern candidate.

A

	d1gtra_	d1h3ea_	d1maua_	d1n77a2	d1gn8a_	d1f9aa_	d1jaga_	d1jjva_	d1n5ia_	d1e2qa_	d1b0ua_	d1ji0a_	d1do0a_	d1j7ka_	d1nsf__	d1asza_	d1b76a1	d1e24a2	d1gol__	d1csn__	d1qmza_	d1hck__	d1atpe_	d1phk__	d1h1wa
d1gtra_	0	0.3	0.39	0.34	0.46	0.44	0.24	0.22	0.09	0.19	0.21	0.24	0.2	0.25	0.22	0.28	0.22	0.27	0.23	0.26	0.26	0.23	0.23	0.25	0.21
d1h3ea_	0.3	0	0.44	0.36	0.48	0.48	0.23	0.16	0.13	0.16	0.15	0.2	0.18	0.24	0.23	0.28	0.27	0.32	0.17	0.23	0.2	0.19	0.19	0.21	0.21
d1maua_	0.39	0.44	0	0.36	0.45	0.49	0.25	0.21	0.16	0.16	0.17	0.25	0.16	0.29	0.21	0.29	0.3	0.34	0.18	0.23	0.21	0.21	0.23	0.25	0.21
d1n77a2	0.34	0.36	0.36	0	0.4	0.37	0.2	0.16	0.08	0.16	0.17	0.22	0.22	0.26	0.21	0.34	0.26	0.28	0.24	0.26	0.26	0.26	0.27	0.28	0.26
d1gn8a_	0.46	0.48	0.45	0.4	0	0.49	0.25	0.2	0.1	0.28	0.28	0.21	0.3	0.35	0.28	0.37	0.42	0.36	0.31	0.25	0.24	0.23	0.29	0.23	0.25
d1f9aa_	0.44	0.48	0.49	0.37	0.49	0	0.24	0.2	0.05	0.23	0.29	0.21	0.26	0.35	0.2	0.43	0.41	0.35	0.26	0.21	0.2	0.22	0.27	0.23	0.24
d1jaga_	0.24	0.23	0.25	0.2	0.25	0.24	0	0.23	0.18	0.29	0.18	0.12	0.29	0.16	0.22	0.26	0.26	0.3	0.31	0.28	0.29	0.27	0.32	0.28	0.23
d1jjva_	0.22	0.16	0.21	0.16	0.2	0.2	0.23	0	0.06	0.5	0.25	0.19	0.36	0.12	0.38	0.26	0.2	0.26	0.28	0.24	0.22	0.29	0.27	0.29	0.23
d1n5ia_	0.09	0.13	0.16	0.08	0.1	0.05	0.18	0.06	0	0.1	0	0	0.13	0.19	0.11	0.22	0.18	0.12	0.18	0.19	0.19	0.18	0.19	0.15	0.2
d1e2qa_	0.19	0.16	0.16	0.16	0.28	0.23	0.29	0.5	0.1	0	0.34	0.16	0.37	0.19	0.39	0.22	0.26	0.26	0.27	0.22	0.21	0.26	0.25	0.25	0.23
d1b0ua_	0.21	0.15	0.17	0.17	0.28	0.29	0.18	0.25	0	0.34	0	0.28	0.23	0.23	0.18	0.21	0.22	0.22	0.21	0.15	0.12	0.12	0.19	0.14	0.13
d1ji0a_	0.24	0.2	0.25	0.22	0.21	0.21	0.12	0.19	0	0.16	0.28	0	0.17	0.2	0.12	0.31	0.25	0.23	0.11	0.11	0.1	0.11	0.13	0.13	0.06
d1do0a_	0.2	0.18	0.16	0.22	0.3	0.26	0.29	0.36	0.13	0.37	0.23	0.17	0	0.33	0.41	0.22	0.22	0.23	0.24	0.26	0.26	0.27	0.26	0.27	0.29
d1j7ka_	0.25	0.24	0.29	0.26	0.35	0.35	0.16	0.12	0.19	0.19	0.23	0.2	0.33	0	0.24	0.29	0.32	0.26	0.19	0.23	0.23	0.22	0.26	0.25	0.21
d1nsf__	0.22	0.23	0.21	0.21	0.28	0.2	0.22	0.38	0.11	0.39	0.18	0.12	0.41	0.24	0	0.21	0.18	0.25	0.39	0.34	0.37	0.32	0.33	0.34	0.34
d1asza_	0.28	0.28	0.29	0.34	0.37	0.43	0.26	0.26	0.22	0.22	0.21	0.31	0.22	0.29	0.21	0	0.43	0.55	0.15	0.2	0.17	0.19	0.18	0.19	0.19
d1b76a1	0.22	0.27	0.3	0.26	0.42	0.41	0.26	0.2	0.18	0.26	0.22	0.25	0.22	0.32	0.18	0.43	0	0.4	0.15	0.19	0.16	0.16	0.19	0.18	0.14
d1e24a2	0.27	0.32	0.34	0.28	0.36	0.35	0.3	0.26	0.12	0.26	0.22	0.23	0.23	0.26	0.25	0.55	0.4	0	0.21	0.23	0.24	0.21	0.22	0.25	0.21
d1gol__	0.23	0.17	0.18	0.24	0.31	0.26	0.31	0.28	0.18	0.27	0.21	0.11	0.24	0.19	0.39	0.15	0.15	0.21	0	0.46	0.51	0.54	0.46	0.49	0.51
d1csn__	0.26	0.23	0.23	0.26	0.25	0.21	0.28	0.24	0.19	0.22	0.15	0.11	0.26	0.23	0.34	0.2	0.19	0.23	0.46	0	0.55	0.46	0.54	0.51	0.53
d1qmza_	0.26	0.2	0.21	0.26	0.24	0.2	0.29	0.22	0.19	0.21	0.12	0.1	0.26	0.23	0.37	0.17	0.16	0.24	0.51	0.55	0	0.64	0.65	0.63	0.57
d1hck__	0.23	0.19	0.21	0.26	0.23	0.22	0.27	0.29	0.18	0.26	0.12	0.11	0.27	0.22	0.32	0.19	0.16	0.21	0.54	0.46	0.64	0	0.52	0.51	0.54
d1atpe_	0.23	0.19	0.23	0.27	0.29	0.27	0.32	0.27	0.19	0.25	0.19	0.13	0.26	0.26	0.33	0.18	0.19	0.22	0.46	0.54	0.65	0.52	0	0.78	0.63
d1phk__	0.25	0.21	0.25	0.28	0.23	0.23	0.28	0.29	0.15	0.25	0.14	0.13	0.27	0.25	0.34	0.19	0.18	0.25	0.49	0.51	0.63	0.51	0.78	0	0.54
d1h1wa_	0.21	0.21	0.21	0.26	0.25	0.24	0.23	0.23	0.2	0.23	0.13	0.06	0.29	0.21	0.34	0.19	0.14	0.21	0.51	0.53	0.57	0.54	0.63	0.54	0

B

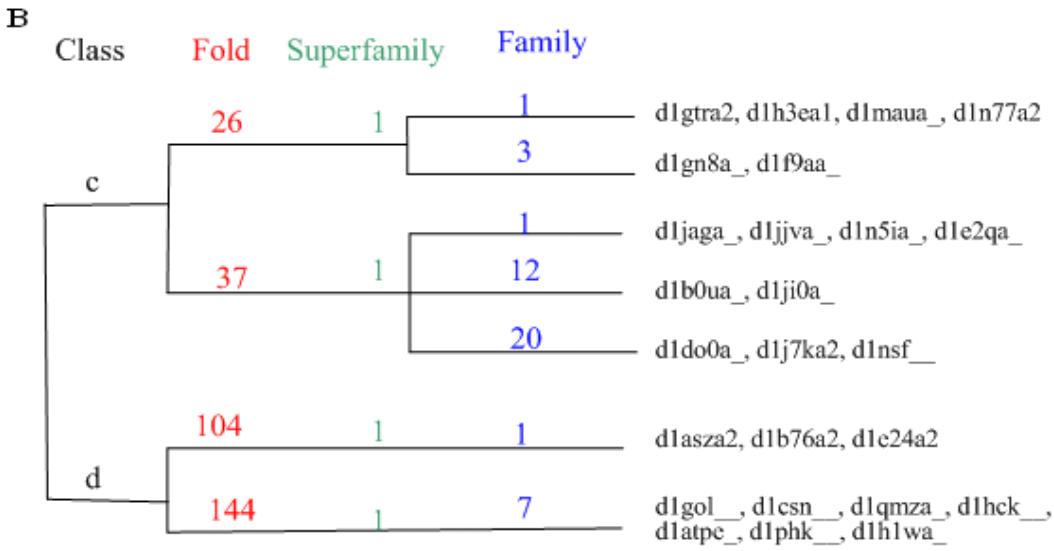


Figure 4. (A) Structure similarity matrix of 25 non-redundant ATP-binding domains; (B) SCOP classification of 25 non-redundant ATP-binding domains. In Figure 4(A), domains belong to same SCOP families are with same colors. The bold values means the structure similarity is larger than the average value of the row; in other words, the domain in this row is much similar with these compared domains than others. In this matrix, we find that most domains of same SCOP family usually have higher structure similarity with each other (see the regions with red frame), it tells us that the multiple ligand-bound structure alignment and structure similarity calculation is reasonable and can reflect structural and functional information. In Figure 4(B), protein domains were classified according to SCOP classification hierarchy: class, fold, superfamily, and family. The protein domains were named by SCOP database nomenclature.

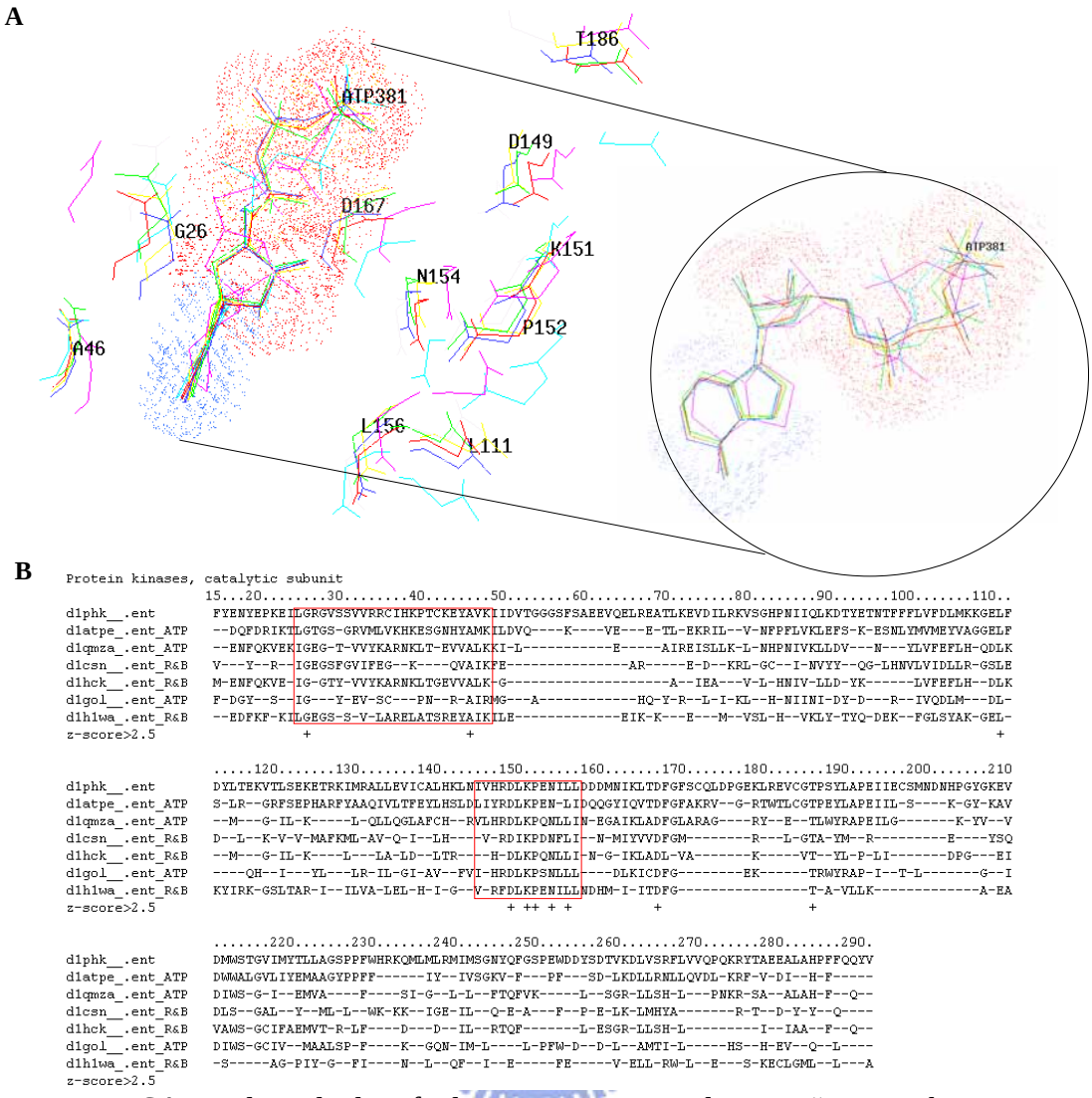


Figure 5. MuLiSA result and identified conservation residues in “Protein kinases, catalytic subunit family” of ATP-binding domains. (A) Three-dimensional distributions of identified conservation residues and the ligand superimposition. Yellow: d1phk__; blue: d1atpe_; green: d1qmza_; red: d1csn__; grey: d1hck__; pink: d1gol__; light blue: d1h1wa_; (B) Multiple ligand-bound structure alignment result of “Protein kinases, catalytic subunit family” domains. In Figure 5(A), the identified conservation residues, aligned positions with z-score of entropy calculation > 2.5, are close to ATP in three-dimensional space. It implies that these conservation residues may play important role in ATP-binding. In Figure 5(B), the labeled residue numbers belong to protein domain d1phk__, which is the selected alignment center C of this cluster; and the red framed region means the PROSITE patterns. We observed that most identified conservation residues were on these PROSITE pattern region, it tell us that identifying pattern candidates from conservation residues extension may be a reasonable approach.

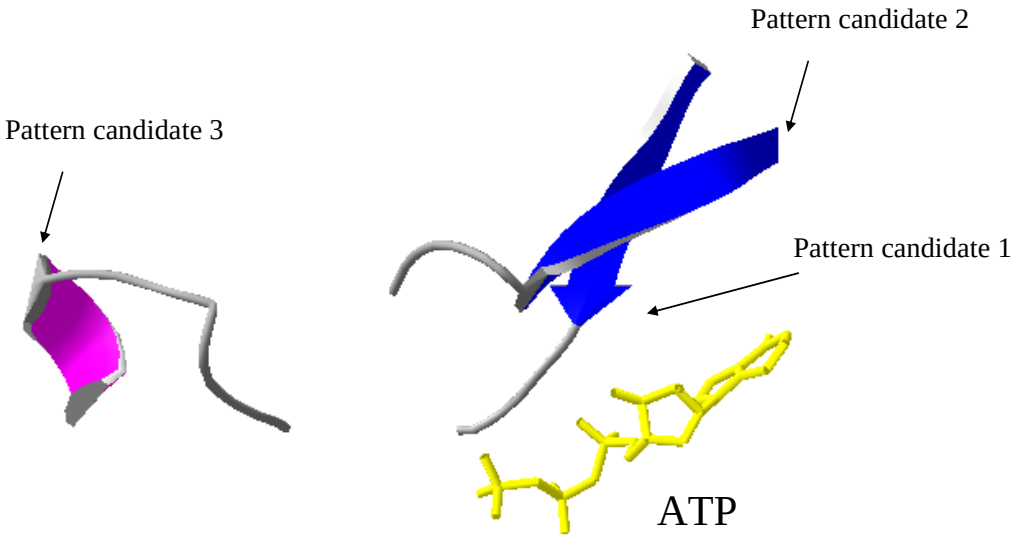


Figure 6. Three pattern candidates of “Class I aminoacyl-tRNA synthetases (RS), catalytic domain family” on three-dimensional space. Pattern candidate 1 is overlapping with PROSITE pattern PS00108; pattern candidate 2 and 3 are novel pattern that we identified. All three pattern candidates are closed to ATP; hence they may be important in ATP-binding.

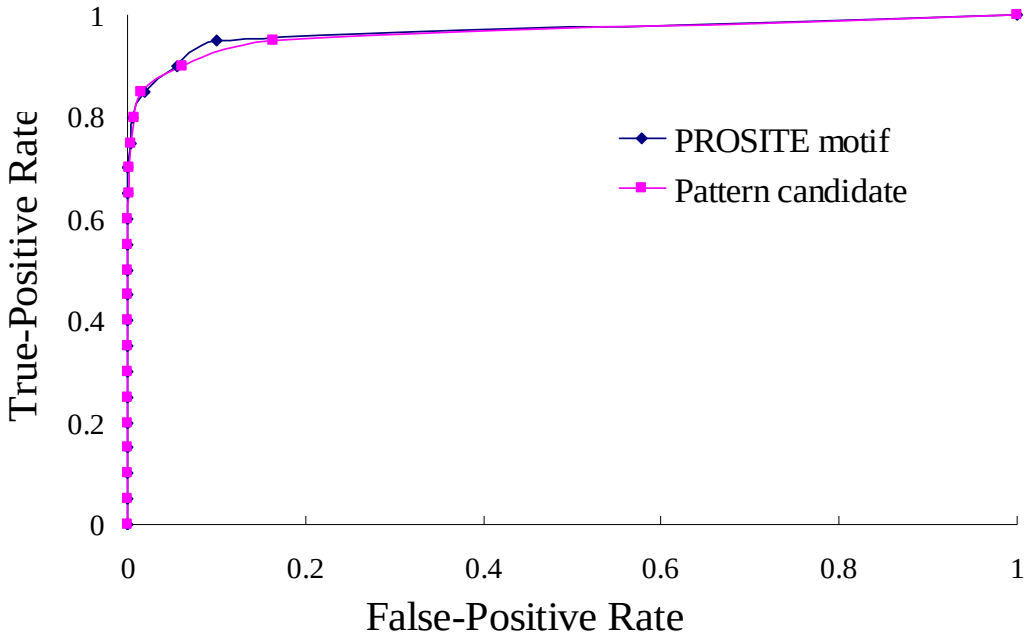


Figure 7. Comparison of pattern candidate 1 and PROSITE pattern: Serine/ Threonine protein kinases active-site signature in “Protein kinases catalytic subunit family” for profile verification of ATP-binding proteins. In ROC curve, the area under curves represents the goodness of the test. We observed that our defined pattern candidate is worse than PROSITE pattern; however, because of that the profile of PROSITE pattern is generated from our alignments, it proved that the profile generated from our alignments is reasonable.

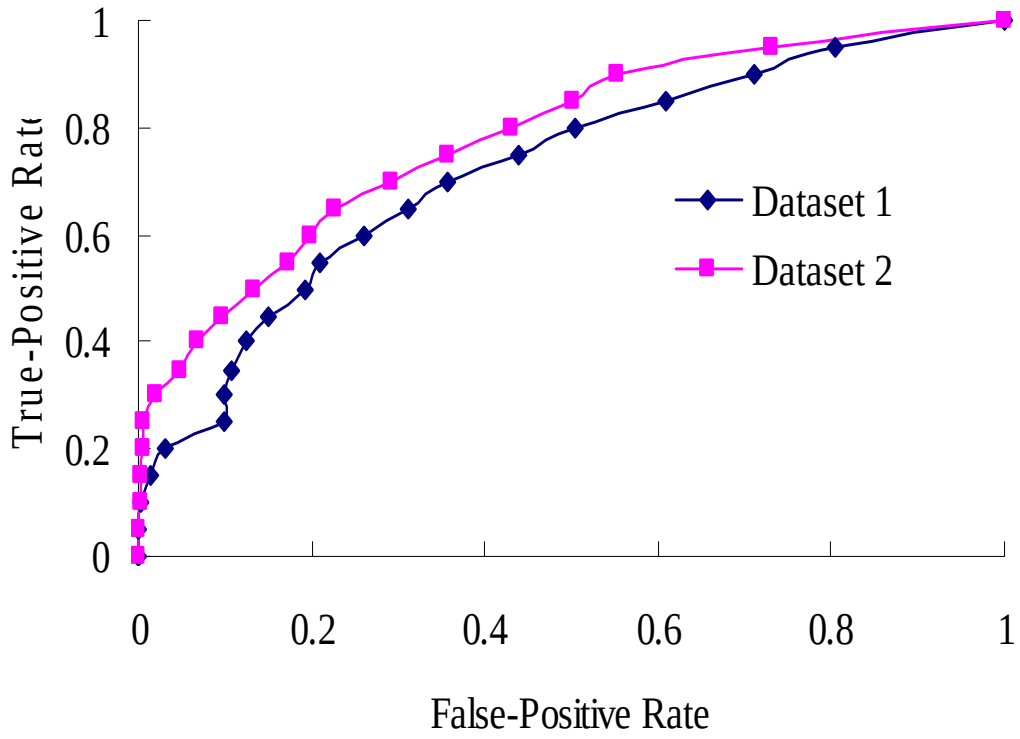


Figure 8. Comparison of datasets used in profile search by pattern candidate 1 of “Class I aminoacyl-tRNA synthetases (RS), catalytic domain family” of ATP-binding domains. Dataset 1: protein sequences contain PROSITE pattern: aminoacyl-transfer RNA synthetases class-I signature. Dataset 2: protein sequences contain PROSITE pattern: aminoacyl-transfer RNA synthetases class-I signature and also have “ATP-binding” annotations in SWISS-PROT database. We observed that the area under curves of dataset 2 is larger than area under curves of dataset 1. Because the profile of pattern candidates were generated from ATP-binding domains alignments and the protein sequences in dataset 1 are not all have “ATP-binding” annotations, we think that the profile of pattern candidate is more meaningful in ATP-binding proteins but not proteins only with PROSITE pattern.

ATP	Seq.	Score	Pattern
+	P48963	0.867	126-RDLKPQNLLI-135
+	P97377	0.867	126-RDLKPQNLLI-135
+	O62846	0.867	164-RDLKPENLLI-173
+	P43063	0.867	132-RDLKPQNLLI-141
	.		
	.		
	.		
+	P23647	0.767	124-RDLKPQNVLL-133
+	Q38775	0.767	144-RDLKPHNLLM-153
+	Q92398	0.767	148-RDLKPGNLLV-157
+	P41892	0.762	145-IKLADFG-151
+	Q14004	0.762	237-IKLADFG-243
+	Q9FZ36	0.762	210-IKLADFG-216
+	Q9R1U5	0.762	163-IKLADFG-169
+	Q09898	0.762	345-IKLTDFG-351
+	P50527	0.762	519-IKLTDFG-525
+	O22040	0.762	211-IKLADFG-217
+	Q03497	0.762	753-IKLTDFG-759
+	Q00534	0.762	159-IKLADFG-165
+	Q60670	0.762	163-IKLADFG-169
	P27604	0.762	13-IKLADFG-19
+	P34722	0.762	513-IKLADFG-519
+	O75962	0.762	2873-IKLADFG-2879
+	Q9NYV4	0.762	873-IKLADFG-879
+	P34102	0.762	10-IKLTDFG-16
+	P34103	0.762	10-IKLTDFG-16
+	Q9Z1M4	0.762	208-IKLTDFG-214
+	Q12236	0.762	317-IKLTDFG-323
+	P38991	0.762	241-IKLTDFG-247
	P25169	0.762	185-IKLTDFG-191
+	Q9UBS0	0.762	208-IKLTDFG-214
+	Q64261	0.762	159-IKLADFG-165
+	P53681	0.760	275-RDLKPENFLF-284
+	Q9HBH9	0.760	157-RDLKPENILC-166
	.		

Figure 9. Profile scoring list of protein function prediction in ATP-binding proteins. The protein sequences with SWISS-PROT “ATP-binding” annotations were labeled by “+” symbol on ATP column. The protein accession numbers in SWISS-PROT database are list on Seq. column. Values on “Score” column are the profile scoring scores. The “Pattern column” shows the matched protein sequence segment, the residue numbers of the first and the last residues are shown. Two points must be mentioned. First, the framed sequences all have “ATP-binding” annotations (except for P27604 and P25169); because these sequences all match our new finding pattern candidate, we regard this pattern candidate is a new pattern of ATP-binding proteins. Second, the non-labeled sequences, P27604 and P25169, are the sequences that match profiles but don’t have “ATP-binding” annotations in SWISS-PROT database, hence these two proteins might be ATP-binding protein that not identified yet.

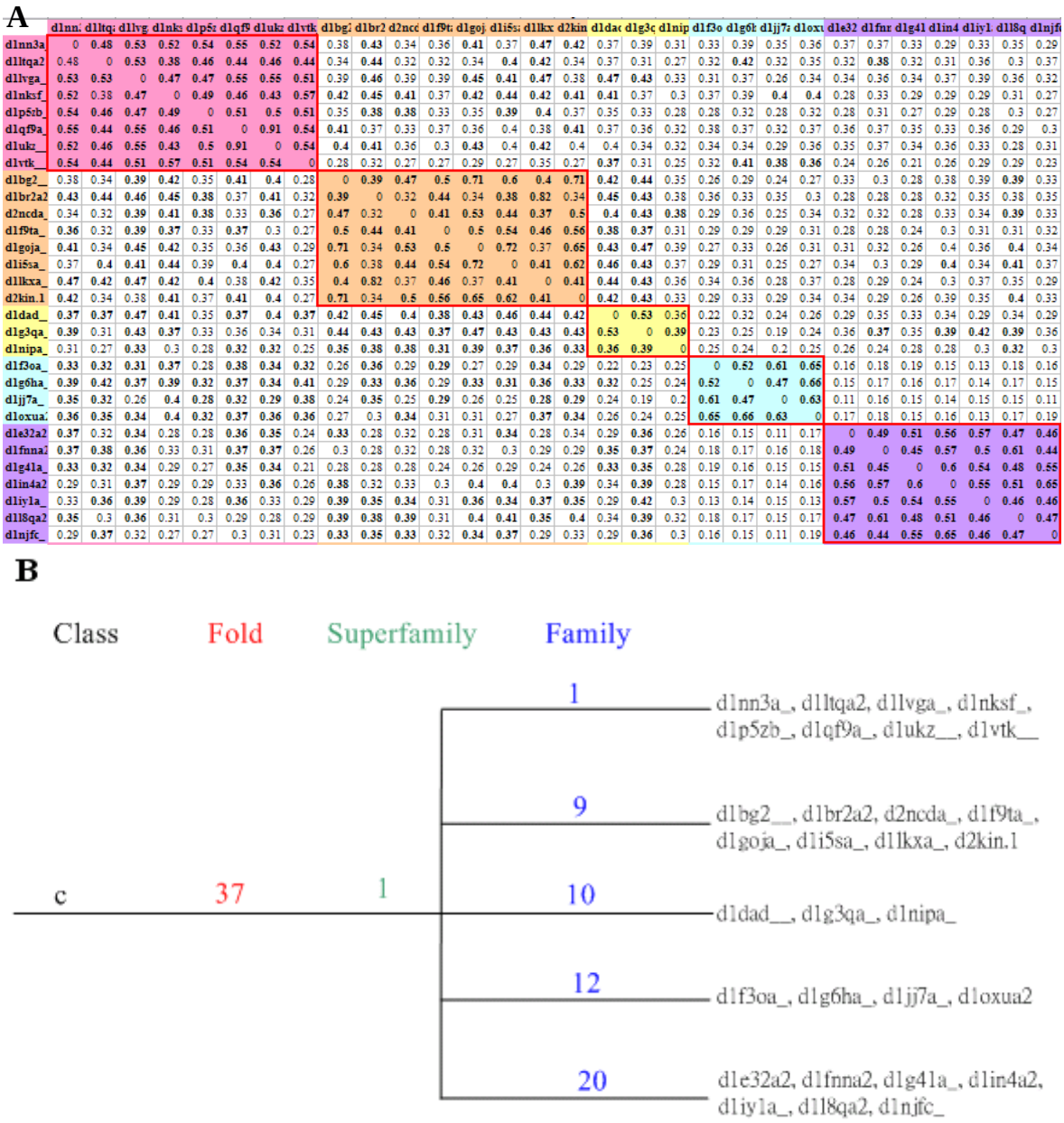
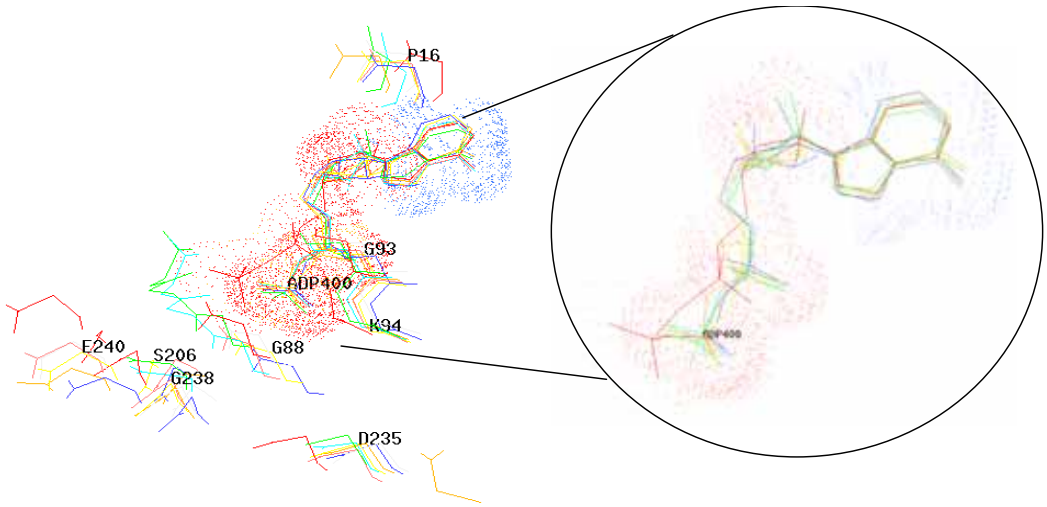


Figure 10. (A) Structure similarity matrix of 30 non-redundant ADP-binding domains; (B) SCOP classification of 30 non-redundant ADP-binding domains. In Figure 10(A), domains belong to same SCOP families are with same colors. The bold values means the structure similarity is larger than the average value of the row. In this matrix, we find that most domains of same SCOP family usually have higher structure similarity with each other (see the regions with red frame). In Figure 10(B), protein domains were classified according to SCOP classification hierarchy.

A



B

```
motor proteins
                .......10........20........30........40........50........60........70........80........90.......100
d1goja_.ent     SSSANSIKVVARFRPQNRVEIESGGQPIVTFQGPDTCTVDSKEAQGSFTFDRVFDMSCKQSDIFDFSIKPTVDDILNGYNGTVFAYGQTGAGKSYTMMG
d1bg2__.ent_R&B ----ETIKVMCRFRPLN-SEVNRG-KYIAKFQGEDTVVI----A---YAFDRVFQSSTSQEQVYN--AK-IV-DVLEGYNGTIFAYGQTSSGKTHTMEG
d1br2a2.ent_ADP ----E----IC---P-NR-NK---L--L---E--RD-MV-L---NI---------I--Y---I----LV--I-FV-DAS-FGI---G--GAGKTET-N-
d2ncda_.ent_R&B T--LDGIRVFCRIRP-----R---MQ---TYHDESTVELQS-QVQ-IFSFDQVF-PLS-QSDIFE-M--PS--QI---S-I-VFAYGQTGSGKTYTM-G
d1f9ta_.ent_R&B ----D-T-V--R-RPA-L-S--SD---IV--QF-D-I-VETK--E-SF--R--FD---TD-DVS--V-H--I-SIL--G--NIFAYGQTGSGKTFTML-
d1i5sa_.ent_ADP -----SVKVAVRVRP-N-RE--S----II---G--T-T-PVKNI--SFSFD-SYW-YASQKQVYR--GEEMLQ-A-EGYNVCIFAYGQTGAGKSYTMMG
d1lkxa_.ent_ADP -------HVVI---P--K-NQN--V---L-----D-TN-E--RIE-LF--D---NI-----DMF---A--Y--M-G---N-CV-ISG--GAGKTEAS--
d2kin.1.ent_ADP -----SIKVMCRFRPLN-AEIR---K-IP-F-G-ET-------IG-P--FDRVLPPNTTQEQVYN--AK--V-DII---NGTIFAYGQTSSGKTHTMEG
z-score > 2.500                 +                                                                       +    ++

                .......110.......120.......130.......140.......150.......160.......170.......180.......190.......200
d1goja_.ent     TSIDDPDGRGVIPRIVEQIFTSILSSAANIEYTVRVSYMEIYMERIRDLLAPQNDNLPVHEEKNRGVYVKGLLEIYVSSVQEVYEVMRRGGNARAVAATN
d1bg2__.ent_R&B --LHDPEGMGIIPRIVQDIF-YI-S--NN-E-KVKVSYFEIYLDKIRDLLD-SK-NLSVH-E--R--YVKGCTE--V-S-DEV-DTIDEGKS-R-V-M--
d1br2a2.ent_ADP ---E------IA--ALE-LE--------------KVN---Y--Y---RLIW---E---------------------A----GM-M-I---A--GF-LAT-
d2ncda_.ent_R&B VPV-E--S-GVIPRLD------G------L-Y---VA---LYV--I--AL----DN--V-AN----------E-L------DP-NL-HHM--TRM--A--
d1f9ta_.ent_R&B ---------GIIP-IV--L-----------------T-VE-Y-N----L----------E----R-ITHK-V-----M-S-E-I---A---L---A-----
d1i5sa_.ent_ADP KQ--E-DQQGIIPQLCEDLFSRIN-TN-NM-YSVEVSYMEIYCERVRDLL---N-NLRVRE--LLGPYVE-LSKLAVTSYNDI-DLM--G--AR-V----
d1lkxa_.ent_ADP ----------IML-LL--P-SS--S-------TG-I-I-NYL------V----------------------N--------MDLISL--L-L-G-L-SV
d2kin.1.ent_ADP --L-DPQLKGIIPRIA-DI------Y---E----VRVSYFEIYLDKIRDLL-----NL-VHE----D-YVKGC-E--E----DV-D-I--G---R--VM--
z-score > 2.500

                .......210.......220.......230.......240.......250.......260.......270.......280.......290.......300.
d1goja_.ent     MNQESSRSHSIFVITITQKNVETGSAKSGQLFLVDLAGSEKVGKTGASGQTLEEAKKINKSLSALGMVINALTDGKSSHVPYRDSKLTRILQESLGGNSRT
d1bg2__.ent_R&B V--ESSRSHSIFLI---LKQ-------SGKL-LVDLAGSEV-------G-----------SN--L--NG--S-----------SKMARILYQ-VLGG-RT
d1br2a2.ent_ADP --E-SSI--T-I-ADG-------------IRILDIAG-EH------I---A---L--IE--T---L-M-H--L--------T--LQL-K----TL-----T
d2ncda_.ent_R&B ---RTSR-H---LL-IY-G------H-SLLI-LVDLAGSE-I----E-----------E-LLS-L--VINAL-Q----HIPMRN--LT--L-NSLGGNSKT
d1f9ta_.ent_R&B -----SRSHSIFECSV-K----------H-I-LVDLAGSER--------------RT-QNSI--L---GDPI--------NY----LSKIL--L-L-YS-C
d1i5sa_.ent_ADP M-NESSRSHAVFNIIFTQK--D-ITEKV---ISLVDLAGSE--------G-----LKG---SLT-LG-V---AS------IPYRDSVLTWLL-E-LLGNSRT
d1lkxa_.ent_ADP KSEE-SI--KY--G-I-N---------KG-M-MLDI-G-EHV--TV-N--F---L---E--LE-L----E--SE-------T----LI-AL-A-L---C--
d2kin.1.ent_ADP ----SSRSHSIFLI-IHFEN---V---KGNLYLVDLAGSEKIV----------------KSLSALG-VI-VL----T---PYRDSKMTRILQDSL-GN-RT
z-score > 2.500       +                          +  + +
```

Figure 11. MuLiSA result and identified conservation residues in “motor proteins family” of ADP-binding domains. (A) Three-dimensional distributions of identified conservation residues and the ligand superimposition. Yellow: d1goja_; blue: d1bg2__; green: d1br2a2; red: d2ncda_; grey: d1f9ta_; orange: d1i5sa_; brown: d2kin.1; light blue: d1lkxa_; (B) Multiple ligand-bound structure alignment result of “motor proteins family” domains. In Figure 11(A), the identified conservation residues are closed to ADP in three-dimensional space. It implies that these conservation residues may play important role in ADP-binding. In Figure 11(B), the labeled residue numbers were belonged to protein domain d1goja_, which is the selected alignment center C of this cluster, and the red framed region means the PROSITE patterns.

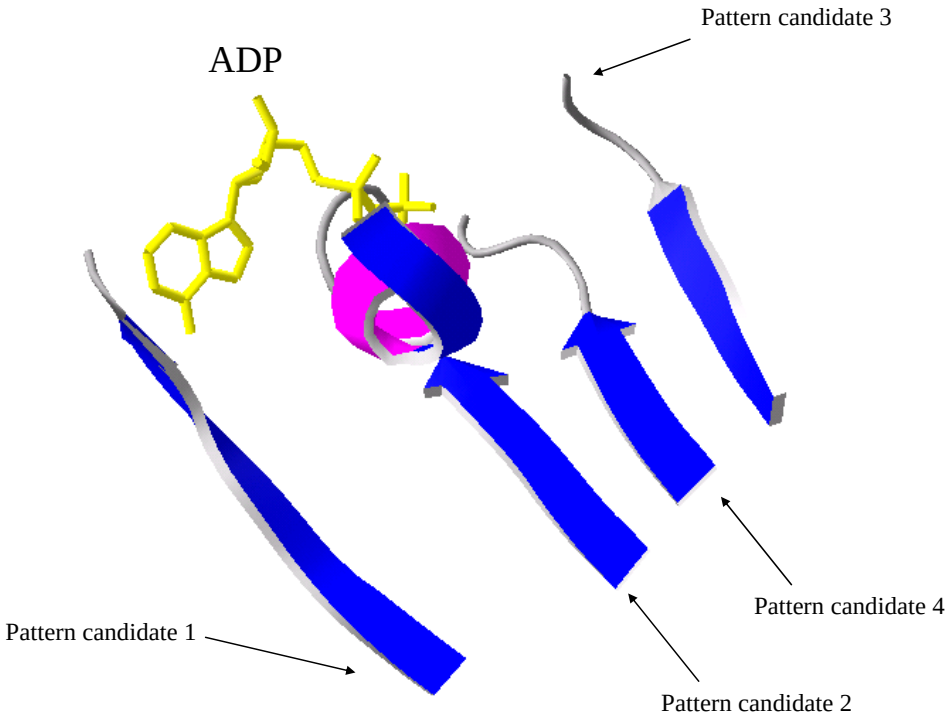


Figure 12. Three pattern candidates of “motor proteins family” on three-dimensional space. Pattern candidate 4 is overlapping with PROSITE pattern PS00411; pattern candidate 1, 2 and 3 are novel pattern that we identified. All three pattern candidates are closed to ADP; hence they may be important in ADP-binding.

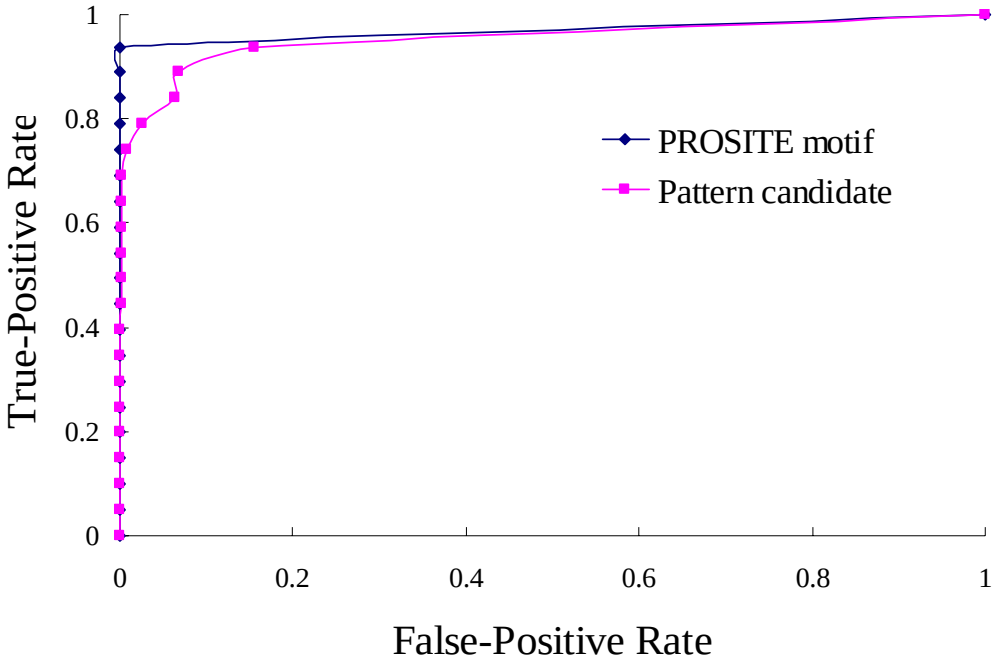


Figure 13. Comparison of pattern candidate 4 and PROSITE pattern: Kinesin motor domain signature in “motor proteins family” for profile verification of ADP-binding domains. We observed that our defined pattern candidate is worse than PROSITE pattern; however, because of that the profile of PROSITE pattern is generated from our alignment, it proved that the profile generated from our alignments is reasonable.

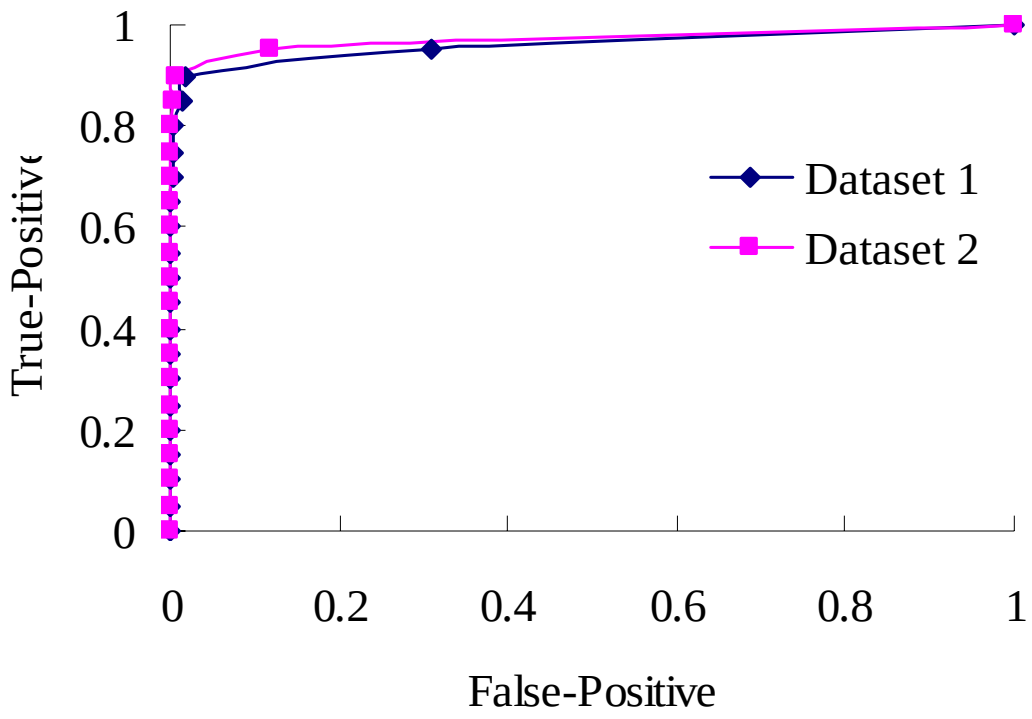


Figure 14. Comparison of datasets used in profile search by pattern candidate 1 of “motor proteins family” of ADP-binding domains. Dataset 1: protein sequences contain PROSITE pattern: Kinesin motor domain signature. Dataset 2: protein sequences contain PROSITE pattern: Kinesin motor domain signature and also have “motor protein” annotations in SWISS-PROT database. We observed that the area under curves of dataset 2 is larger than area under curves of dataset 1. Because the profile of pattern candidates were generated from motor proteins domains alignments and the protein sequences in dataset 1 are not all have “motor protein” annotations, we think that the profile of pattern candidate is more meaningful in motor proteins but not proteins only with PROSITE pattern.

motor	Seq.	Score	Pattern
+	O43093	1.000	232-LVDLAGSE-239
+	Q9EQW7	1.000	250-LVDLAGSE-257
+	P46870	1.000	239-LVDLAGSE-246
+	Q92376	1.000	719-LVDLAGSE-726
+	O95239	1.000	238-LVDLAGSE-245
+	P17119	1.000	624-LVDLAGSE-631
+	P17120	1.000	320-LVDLAGSE-327
+	O88658	1.000	245-LVDLAGSE-252
+	P45962	1.000	469-LVDLAGSE-476
+	P17210	1.000	236-LVDLAGSE-243
+	Q60575	1.000	246-LVDLAGSE-253
	.		
	.		
	.		
+	O14782	1.000	244-LVDLAGSE-251
+	P82266	1.000	263-LVDLAGSE-270
+	P24339	0.881	283-SRSHSIF-289
+	P97329	0.881	377-SRSHSIF-383
+	O95235	0.881	378-SRSHSIF-384
	O88338	0.875	421-AVDLAGSE-428
+	P70096	0.875	484-LVDLAGNE-491
+	P18105	0.875	224-IVDLAGSE-231
+	Q922S8	0.875	486-LVDLAGNE-493
	Q8X9C8	0.875	136-LVGLAGSE-143
+	Q62909	0.875	436-LVDLAGNE-443
	Q8FJU6	0.875	136-LVGLAGSE-143
+	Q14807	0.875	272-LIDLAGSE-279
+	Q15058	0.875	601-LIDLAGSE-608
+	P46872	0.875	245-MVDLAGSE-252
	P75746	0.875	136-LVGLAGSE-143
+	O35787	0.875	245-LVNLAGSE-252
	P44531	0.875	123-LVDLAGFE-130
	Q8ZD89	0.875	136-LVGLAGSE-143
+	P79955	0.875	537-LIDLAGSE-544
+	Q91636	0.875	494-LVDLAGNE-501
	.		

Figure 15. Profile scoring list of protein function prediction in motor proteins. The protein sequences with SWISS-PROT “motor protein” annotations were labeled by “+” symbol on motor column. The protein accession numbers in SWISS-PROT database are list on Seq. column. Values on “Score” column are the profile scoring scores. The “Pattern column” shows the matched protein sequence segment, the residue numbers of the first and the last residues are shown. Two points must be mentioned. First, the framed sequences all have “motor” annotations; because these sequences all match our new finding pattern candidate, we regard this pattern candidate is a new pattern of motor proteins. Second, the non-labeled sequences are the sequences that match profiles but don’t have “motor protein” annotations in SWISS-PROT database; hence these proteins might be motor proteins that not identified yet.

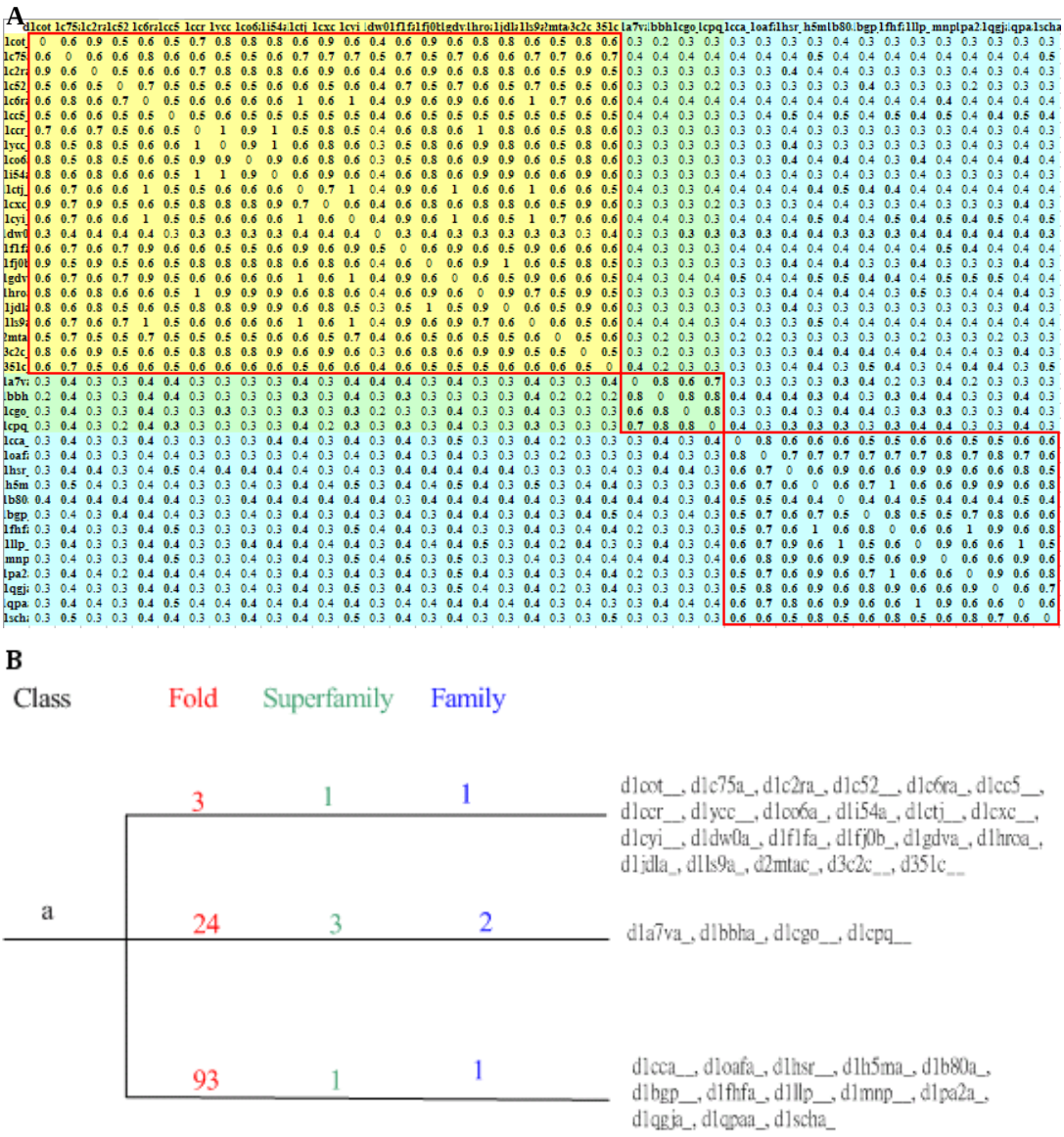
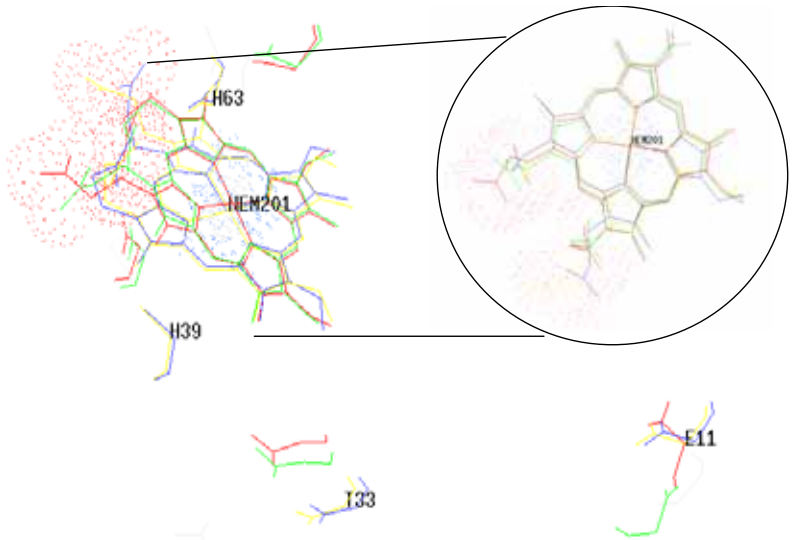


Figure 16. (A) Structure similarity matrix of 40 non-redundant HEM-binding domains; (B) SCOP classification of 40 non-redundant HEM-binding domains. In Figure 16(A), protein domains belong to same SCOP families are with same colors. In this matrix, we find that most domains of same SCOP family usually have higher structure similarity with each other (see the regions with red frame); it tell us that the multiple ligand-bound structure alignment and structure similarity calculation is reasonable and can reflect structural and functional information. In Figure 16(B), protein domains were classified according to SCOP classification hierarchy: class, fold, superfamily, and family.

A



B

```
Cytochrome b5

                  ........10........20........30........40........50
d1cyo__.ent       SKAVKYYTLEEIQKHNNSKSTWLILHYKVYDLTKFLEEHPGGEEVLREQA
d1b5m__.ent_HEM   --AVTYY-LEEVAKRN-AEETWMVIHGRVYDITRFLSEHPGGEEVLLEQA
d1cxya_.ent_HEM   ----------EV----------M-LPS-GWEST-W--EH-AA--LI-Q--
d1icca_.ent_HEM   ----------EV-R-D------M-LRT-G-DAT-F-EDH-A---LI--Y-
d1mj4a_.ent_HEM   ---I------E--------S-G-PL---VVWIT-F-D-HPGG--KL--LA
z-score > 2.500             +                       +     +

                  ........60........70........80........
d1cyo__.ent       GGDATENFEDVGHSTDARELSKTFIIGELHPDDRSKIT
d1b5m__.ent_HEM   GADATESFEDVGHSPDAREMLKQYYIGDVHPND-----
d1cxya_.ent_HEM   V-TLV-Y--N--H---M--S------------------
d1icca_.ent_HEM   V--L--F--E--H---G--------A-N-K--------
d1mj4a_.ent_HEM   ---A---W---VH-SHVRELLKYF--VL----------
z-score > 2.500               +
```

Figure 17. MuLiSA result and identified conservation residues in “Cytochrome b5 family” of HEM-binding domains. (A) Three-dimensional distributions of identified conservation residues and the ligand superimposition. Yellow: d1cyo__; blue: d1b5m__; green: d1cxya_; red: d1icca_; grey: d1mj4a_; (B) Multiple ligand-bound structure alignment result of “Cytochrome b5 family” domains. In Figure 17(A), the identified conservation residues are closed to heme in three-dimensional space. It implies that these conservation residues may play important role in HEM-binding. In Figure 17(B), the labeled residue numbers were belonged to protein domain d1cyo__, which is the selected alignment center C of this cluster, and the red framed region means the PROSITE patterns.

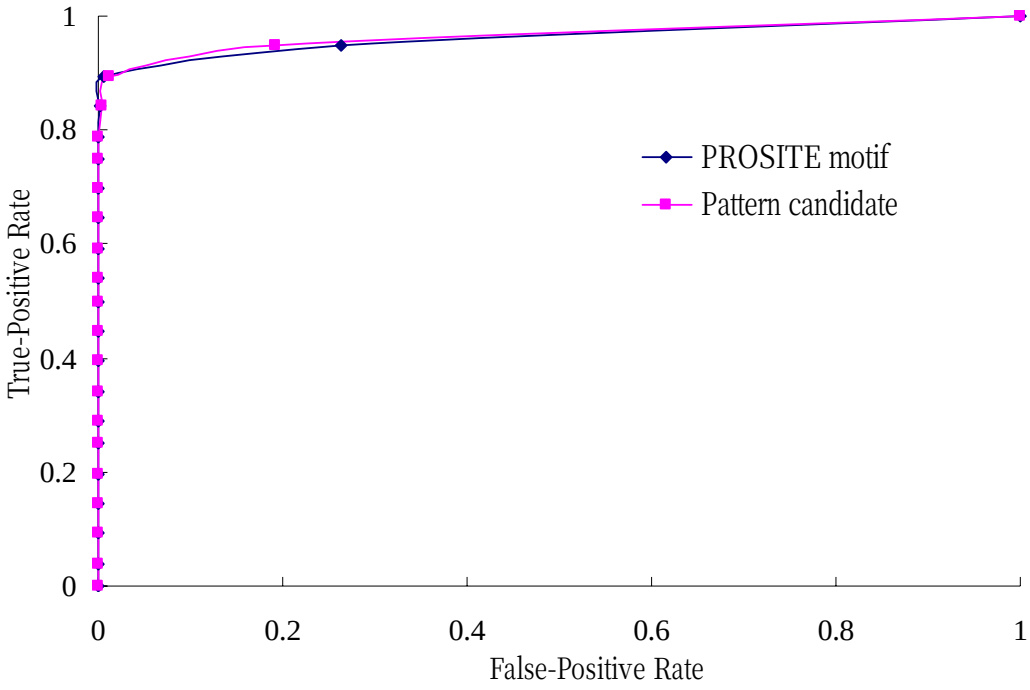


Figure 18. Comparison of pattern candidate 1 and PROSITE pattern: cytochrome b5 family, heme-binding domain signature in “cytochrome b5 family” for profile verification of HEM-binding domains in dataset 2. We observed that our defined pattern candidate is a little better than PROSITE pattern. Although this pattern candidate partially overlaps with this PROSITE pattern, it means that the pattern candidates identified by our approach may be more meaningful than PROSITE pattern for protein sequences with “Heme” annotations in SWISS-PROT database; and because of that the profile of PROSITE pattern is generated from our alignment, it also proved that the profile generated from our alignments is reasonable.

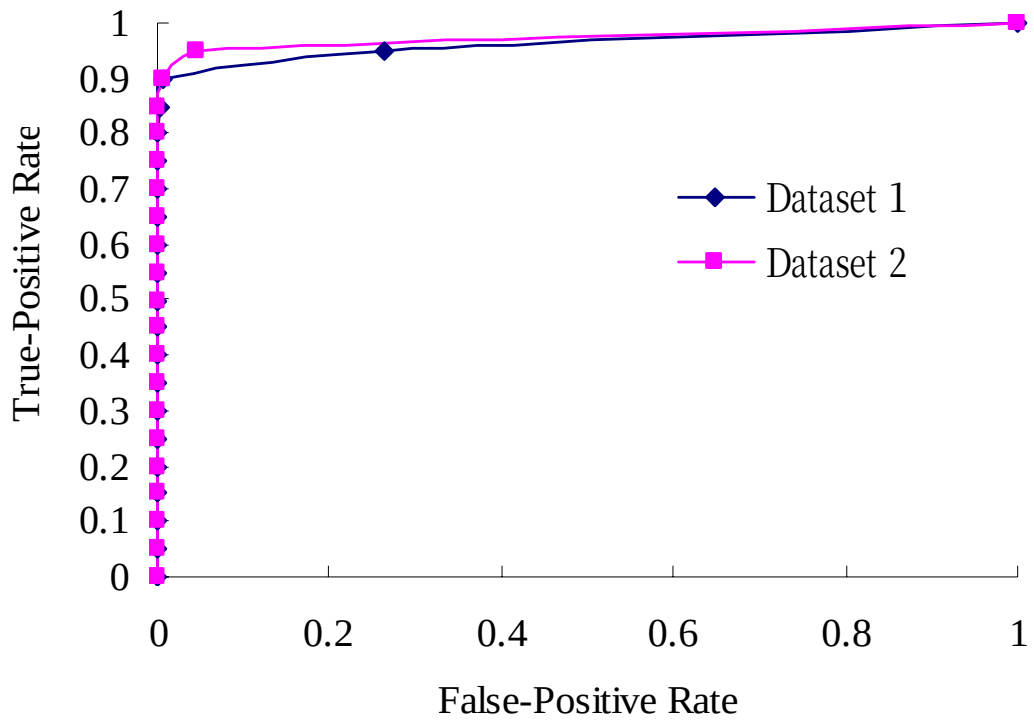


Figure 19. Comparison of datasets used in profile search by pattern candidate 1 of “cytochrome b5 family” of HEM-binding domains. Dataset 1: protein sequences contain PROSITE pattern: cytochrome b5 family, heme-binding domain signature. Dataset 2: protein sequences contain PROSITE pattern and have “Heme” annotations in SWISS-PROT database. We observed that the area under curves of dataset 2 is larger than area under curves of dataset 1. Because the profile of pattern candidates were generated from HEM-binding domains alignments and the protein sequences in dataset 1 are not all have “Heme” annotations, we think that the profile of pattern candidate is more meaningful in HEM-binding proteins but not proteins only with PROSITE pattern.

Heme	Seq.	Score	Pattern
+	Q9FJZ9	0.852	69-RLHFHD-74
+	P12437	0.852	93-RLHFHD-98
+	P15233	0.852	47-RLHFHD-52
+	P15232	0.852	66-RLHFHD-71
+	Q05855	0.852	61-RLHFHD-66
	.		
	.		
	.		
+	Q50925	0.798	334-CAACH-338
+	P19136	0.796	66-RLTFHD-71
+	P20010	0.796	66-RLTFHD-71
+	Q02567	0.796	63-RLTFHD-68
+	P20013	0.796	70-RLTFHD-75
	Q81MN9	0.796	123-RLTFHD-128
+	Q9LVL1	0.778	66-RLFFHD-71
+	Q9SK52	0.778	67-RLIFHD-72
+	O81755	0.778	50-RLLFHD-55
	P55019	0.778	887-RLGFHD-892
+	P00434	0.778	38-RLFFHD-43
	Q89A58	0.778	118-RLRFHD-123
	P11413	0.778	369-RLQFHD-374
+	Q9SY33	0.778	87-RLIFHD-92
+	Q96510	0.778	63-RLFFHD-68
+	Q9SLH7	0.778	66-RLQFHD-71
+	Q9FJR1	0.778	69-RLFFHD-74
	P06308	0.778	146-RLRFHD-151
	Q9QZY5	0.778	100-RLYFHD-105
	Q15345	0.778	626-RLSFHD-631
+	O48677	0.778	58-RLFFHD-63
+	Q9LVL2	0.778	57-RLFFHD-62
+	Q9FMR0	0.778	64-RLYFHD-69
+	Q96518	0.778	61-RLFFHD-66
+	Q9SZE7	0.778	63-RLYFHD-68
	.		
	.		

Figure 20. Profile scoring list of protein function prediction in HEM-binding proteins. The protein sequences with SWISS-PROT “Heme” annotations were labeled by “+” symbol on “Heme” column. We observed there are seven protein sequences which match the pattern candidate we identified but not have “Heme” annotations in SWISS-PROT database, hence these seven proteins might be HEM-binding protein but not identified yet.

A

```
MuLiSA:
d1maua_.ent      MKTIFSGIQ PSGVITIGNY IGALRQFVELQHEYNCYFCIVDQHAITVWQDPHELRQNIRR
d1gtra2.ent_ATP  --G---PPE PNGY-HIGH- A--I-NFFQA--G------FDD---V----------EYI--

C.E.:
d1maua_.ent      MKTIFSGIQ PSGVITIGNY IGALRQFVELQHEYNCYFCIVDQHAITVWQDPHELRQNIRR
d1gtra2.ent      --------- ---------- ----------------------------------------

CLUSTALW:
d1maua_.ent      MKTIFSGIQ PSGVITIGNY IGALRQFVELQHEYNCYFCIVDQHAITVWQDPHELRQNIRR
d1gtra2.ent      --TNFIRQI IDEDLASGKH T---TVHTRFPPEPNGYLHIGHAKSICLNFG---IAQDYKG
```

B **C**

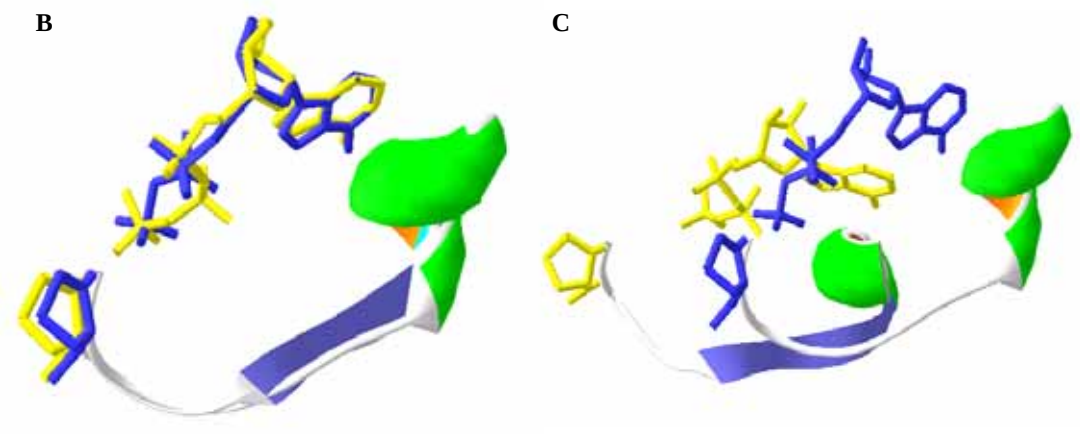


Figure 21. The comparison of MuLiSA, CE, and CLUSTALW results of two Class I aminoacyl-tRNA synthetases (RS), catalytic domains: d1maua_ and d1gtra2. (A) Alignment comparison between three methods. The shadowed region is the PROSITE defined patterns; (B) 3D structure alignment result of MuLiSA; (C) 3D structure alignment result of CE. In Figure 21(A), only the alignments of MuLiSA can align the PROSITE defined patterns together (PROSITE pattern: P-x(0,2)-[GSTAN]-[DENQGAPK]-x-[LIVMFP]-[HT]-[LIVMYAC]-G-[HNTG]-[LIVMFYSTAGPC]) of two domains, d1maua_ and d1gtra2. In Figure 21(B), two ATPs were nearly superimposed and the PROSITE patterns also aligned well. However, in Figure 21(C), we can see that the PROSITE patterns were shifted. In fact, for CE uses only protein structure information to undergo structure alignment, we find that in this case the bad alignment of conservation patterns was because of a huge structure similar region apart from ATP-binding site, and it did disturb the alignment of PROSITE patterns.

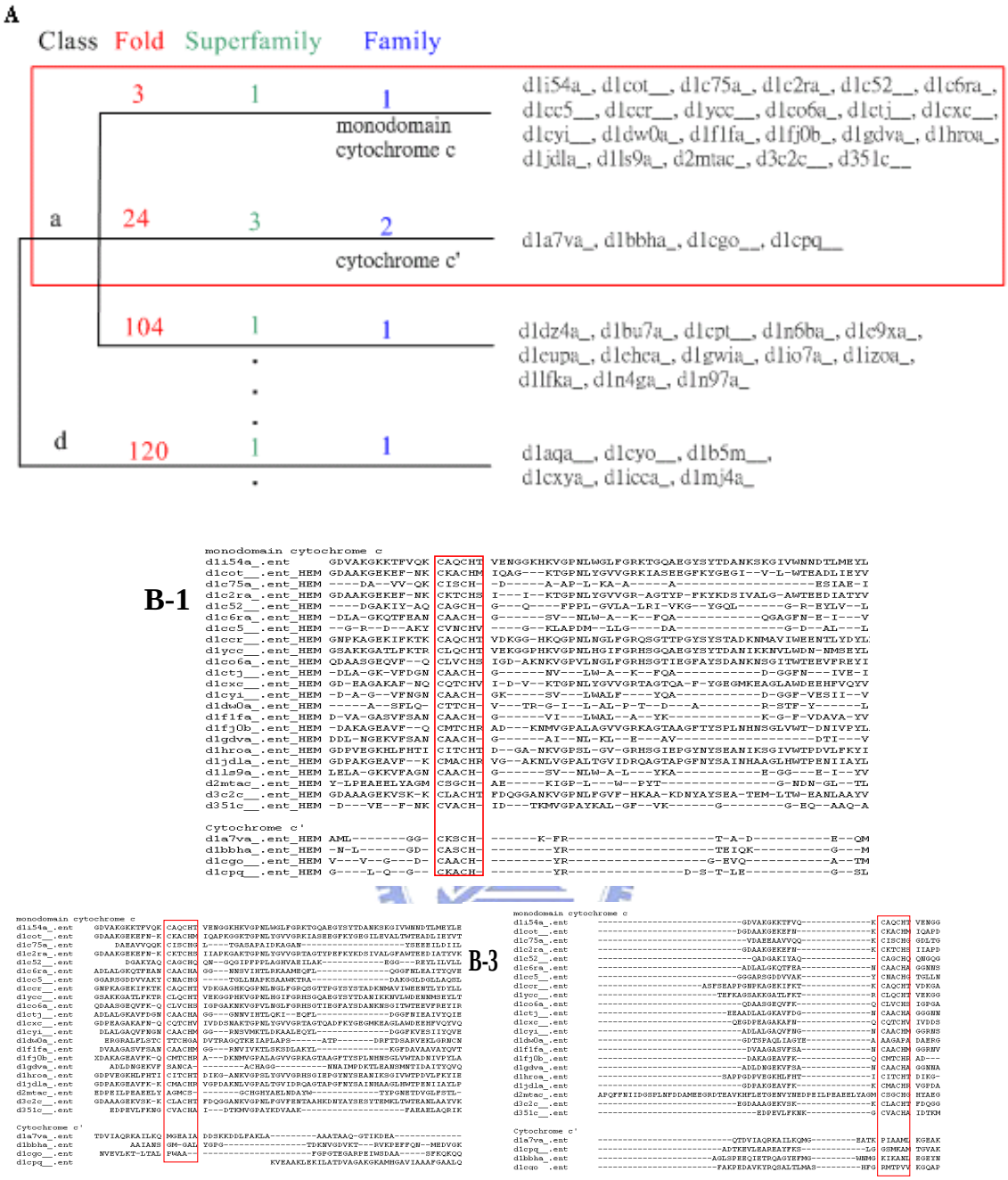


Figure 22. The comparison of MuLiSA, CE, and CLUSTALW results of two domain families, monodomain cytochrome c and cytochrome c', which have same conservation patterns (PROSITE pattern: C-{CPWHF}-{CPWR}-C-H-{CFYW}) but belong to different SCOP fold. (A) SCOP classification of HEM-binding domains; (B-1) Alignment result of MuLiSA; (B-2) Alignment result of CE; (B-3) Alignment result of CLUSTALW. In Figure 22(A), there are 23 domains belongs to “monodomain cytochrome c family” and 5 domains belong to “cytochrome c' family”. What’s most important is that these two families belong to different folds, it means domain structures of these two protein families should be different. In Figure 18(B-2) and (B-3), CE and CLUSTALW both can’t align the PROSITE patterns together when domains

belong to different SCOP fold; however, in Figure 22(B-1), MuLiSA aligned these PROSITE patterns well. For MuLiSA aligned the conservation patterns by ligand superimposition first, we think that when proteins have similar function in ligand-binding but with different protein structures, MuLiSA can exclude protein structure noise and only focus on ligand-binding sites, so MuLiSA aligned well; on the other hand, CE and CLUSTALW consider the whole protein structure or sequence information, so information in ligand-binding site may be disturbed by whole protein information.

References

1. Page, R.D.M. and E.C. Holmes, *Molecular evolution: a phylogenetic approach*. 1998, Blackwell, Oxford. p. 228-279.
2. Falquet, L., et al., *The PROSITE database, its status in 2002.* Nucleic Acids Research, 2002. **30**: p. 235-238.
3. Bateman, A., et al., *The Pfam protein families database.* Nucleic Acids Research, 2004. **32**: p. D138-D141.
4. Casari, G., C. Sander, and A. Valencia, *A method to predict functional residues in proteins.* Nature Structural Biology, 1995. **2**: p. 171-178.
5. Pietrokovski, S., J.G. Henikoff, and S. Henikoff, *The Blocks database--a system for protein classification.* Nucleic Acids Research, 1996. **24**: p. 197-200.
6. Jones, S. and J.M. Thornton, *Prediction of protein-protein interaction sites using patch analysis.* Journal of Molecular Biology, 1997. **272**: p. 133-143.
7. Shatsky, M., R. Nussinov, and H.J. Wolfson, *Flexible protein alignment and hinge detection.* Proteins: Structure Function and Genetics, 2002. **48**: p. 242-256.
8. Shindyalov, I.N. and P.E. Bourne, *Protein structure alignment by incremental combinatorial extension (CE) of the optimal path.* Protein Engineering, 1998. **11**(9): p. 739-747.
9. Higgins, D., et al., *CLUSTALW: improving the sensitivity of progressive multiple sequence alignment through sequence weighting,position-specific gap penalties and weight matrix choice.* Nucleic Acids Research, 1994. **22**: p. 4673-4680.
10. Sali, A., et al., *From comparisons of protein sequences and structures to protein modelling and design.* Trends in Biochemical Sciences, 1990. **15**: p. 235-240.
11. Lichtarge, O., H.R. Bourne, and F.E. Cohen, *An evolutionary trace method defines binding surfaces common to protein families.* Journal of Molecular Biology, 1996. **257**: p. 342-358.
12. Innis, C.A., J. Shi, and T.L. Blundell, *Evolutionary trace analysis of TGF-beta and related growth factors: implications for site-directed mutagenesis.* Protein Engineering, 2000. **13**: p. 839-847.
13. Landgraf, R., I. Xenarios, and D. Eisenberg, *Three-dimensional cluster analysis identifies interfaces and functional residue clusters in proteins.* Journal of Molecular Biology, 2001. **307**: p. 1487-1502.
14. Notredame, C., D.G. Higgins, and J. Heringa, *T-Coffee: a novel method for fast and accurate multiple sequence alignment.* Journal of Molecular Biology, 2000. **302**: p. 205-217.
15. Altschul, S.F., et al., *Basic local alignment search tool.* Journal of Molecular Biology,

1990. **215**: p. 403-410.
16. Holm, L. and C. Sander, *Protein structure comparison by alignment of distance matrices.* Journal of Molecular Biology, 1993. **233**: p. 123-138.
17. Gibrat, J.F., T. Madej, and S.H. Bryant, *Surprising similarities in structure comparison.* Current Opinion in Structural Biology, 1996. **6**: p. 377-385.
18. Laskowski, R.A., et al., *PDBsum: a Web-based database of summaries and analyses of all PDB structures.* Trends in Biochemical Sciences, 1997. **22**: p. 488-490.
19. Murzin, A.G., et al., *SCOP: a structural classification of proteins database for the investigation of sequences and structures.* Journal of Molecular Biology, 1995. **247**: p. 536-540.
20. Boeckmann, B., et al., *The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003.* Nucleic Acids Research, 2003. **31**: p. 365-370.
21. Berman, H.M., et al., *The Protein Data Bank.* Nucleic Acids Research, 2000. **28**: p. 235-242.
22. Besl, P.J. and N.D. McKay, *A method for registration of 3-D shapes.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1992. **14**: p. 239-256.
23. Owen, D.J., et al., *Two structures of the catalytic domain of phosphorylase kinase: an active protein kinase complexed with substrate analogue and product.* Structure, 1995. **3**: p. 467-482.
24. Zheng, J.H., et al., *2.2 A refined crystal structure of the catalytic subunit of cAMP-dependent protein kinase complexed with MnATP and a peptide inhibitor.* Acta Crystallographica Section D-Biological Crystallography, 1993. **49**: p. 362-365.
25. Brown, N.R., et al., *The structural basis for specificity of substrate and recruitment peptides for cyclin-dependent kinases.* Nature Cell Biology, 1999. **1**: p. 438-443.
26. Xu, R.M., et al., *Crystal structure of casein kinase-1, a phosphate-directed protein kinase.* EMBO Journal, 1995. **14**: p. 1015-1023.
27. Schulze-Gahmen, U., H.L. De Bondt, and S.H. Kim, *High-resolution crystal structures of human cyclin-dependent kinase 2 with and without ATP: bound waters and natural ligand as guides for inhibitor design.* Journal of Medicinal Chemistry, 1996. **39**: p. 4540-4546.
28. Robinson, M.J., et al., *Mutation of position 52 in ERK2 creates a nonproductive binding mode for adenosine 5'-triphosphate.* Biochemistry, 1996. **35**: p. 5641-5646.
29. Biondi, R.M., et al., *High resolution crystal structure of the human Pdk1 catalytic domain defines the regulatory phosphopeptide docking site.* EMBO Journal, 2003. **21**: p. 4219-4228.
30. Retailleau, P., et al., *Interconversion of ATP binding and conformational free energies by tryptophanyl-tRNA synthetase: structures of ATP bound to open and closed, pre-transition-atate conformations.* Journal of Molecular Biology, 2003. **325**: p. 39-63.

31. Sekine, S., et al., *ATP binding by glutamyl-tRNA synthetase is switched to the productive mode by tRNA binding.* EMBO Journal, 2003. **22**: p. 676-688.
32. Yaremchuk, A., et al., *Class I tyrosyl-tRNA synthetase has a class II mode or tRNA recognition.* EMBO Journal, 2002. **21**: p. 3829-3840.
33. Song, Y.H., et al., *Structure of a fast kinesin: implications for ATPase mechanism and interactions with microtubules.* EMBO Journal, 2001. **20**: p. 6213-6225.
34. Kull, F.J., et al., *Crystal structure of the kinesin motor domain reveals a structural similarity to myosin.* Nature, 1996. **380**: p. 550-555.
35. Dominguez, R., et al., *Crystal structure of a vertebrate smooth muscle myosin motor domain and its complex with the essential light chain: visualization of the pre-power stroke state.* Cell, 1998. **94**: p. 559-571.
36. Sablin, E.P., et al., *Direction determination in the minus-end-directed kinesin motor ncd.* Nature, 1998. **395**: p. 813-816.
37. Yun, M., et al., *A structural pathway for activation of the kinesin motor ATPase.* EMBO Journal, 2001. **20**: p. 2611-2618.
38. Kikkawa, M., et al., *Switch-based mechanism of kinesin motors.* Nature Cell Biology, 2001. **411**: p. 439-445.
39. Kollmar, M., et al., *Crystal structure of the motor domain of a class-I myosin.* EMBO Journal, 2002. **21**: p. 2517-2525.
40. Sack, S., et al., *X-ray structure of motor and neck domains from rat brain kinesin.* Biochemistry, 1997. **36**: p. 16155-16165.
41. Choinowski, T., et al., *The crystal structure of lignin peroxidase at 1.70 A resolution reveals a hydroxy group on the cbeta of tryptophan 171: a novel radical site formed during the redox cycle.* Journal of Molecular Biology, 1999. **286**: p. 809-827.
42. Goodin, D.B. and D.E. McRee, *The Asp-His-Fe triad of cytochrome c peroxidase controls the reduction potential, electronic structure, and coupling of the tryptophan free radical to the heme.* Biochemistry, 1993. **32**: p. 3313-3324.
43. Sharp, K.H., et al., *Crystal structure of the ascorbate peroxidase-ascorbate complex.* Nat.Struct.Biol, 2003. **10**: p. 303-307.
44. Itakura, H., Y. Oda, and K. Fukuyama, *Binding mode of benzhydroxamic acid to Arthromyces ramosus peroxidase shown by X-ray crystallographic analysis of the complex at 1.6 A resolution.* FEBS Letters, 1997. **412**: p. 107-110.
45. Berglund, G.I., et al., *The catalytic pathway of Horseradish peroxidase at high resolution.* Nature, 2002. **417**: p. 463-468.
46. Blodig, W., et al., *Autocatalytic formation of a hydroxy group at C beta of trp171 in lignin peroxidase.* Biochemistry, 1998. **37**: p. 8832-8838.
47. Henriksen, A., K.G. Welinder, and M. Gajhede, *Structure of barley grain peroxidase refined at 1.9-A resolution. A plant peroxidase reversibly inactivated at neutral pH.* Journal of Biological Chemistry, 1998. **273**: p. 2241-2248.

48. Sundaramoorthy, M., et al., *Preliminary crystallographic analysis of manganese peroxidase from Phanerochaete chrysosporium.* Journal of Molecular Biology, 1994. **238**: p. 845-848.
49. Henriksen, A., et al., *Structure of soybean seed coat peroxidase: a plant peroxidase with unusual stability and haem-apoprotein interactions.* Protein Science, 2001. **10**: p. 108-115.
50. Ostergaard, L., et al., *Arabidopsis ATP A2 peroxidase. Expression and high-resolution structure of a plant peroxidase with implications for lignification.* Plant Molecular Biology, 2000. **44**: p. 231-243.
51. Mirza, O., et al., *Arabidopsis thaliana peroxidase N: structure of a novel neutral peroxidase.* Acta Crystallographica Section D-Biological Crystallography, 2000. **56**: p. 372-375.
52. Schuller, D.J., et al., *The crystal structure of peanut peroxidase.* Structure, 1996. **4**: p. 311-321.
53. Cupp-Vickery, J.R., R. Anderson, and Z. Hatziris, *Crystal structures of ligand complexes of P450Eryf exhibiting homotropic cooperativity.* Proceedings of the National Academy of Sciences of the United States of America, 2000. **97**: p. 3050-3055.
54. Schlichting, I., et al., *The catalytic pathway of cytochrome P450Cam at atomic resolution.* Science, 2000. **287**: p. 1615-1622.
55. Sevrioukova, I.F., et al., *Structure of a cytochrome P450-redox partner electron-transfer complex.* Proceedings of the National Academy of Sciences of the United States of America, 1999. **96**: p. 1863-1868.
56. Hasemann, C.A., et al., *Crystal structure and refinement of cytochrome P450terp at 2.3 A resolution.* Journal of Molecular Biology, 1994. **236**: p. 1169-1185.
57. Wester, M.R., et al., *Structure of a substrate complex of mammalian cytochrome P450 2C5 at 2.3 A resolution: evidence for multiple substrate binding modes.* Biochemistry, 2003. **42**: p. 6370-6379.
58. Podust, L.M., T.L. Poulos, and M.R. Waterman, *Crystal structure of cytochrome P450 14alpha-sterol demethylase (Cyp51) from Mycobacterium Tuberculosis in complex with azole inhibitors.* Proceedings of the National Academy of Sciences of the United States of America, 2001. **98**: p. 3068-3073.
59. Shimizu, H., et al., *Crystal structures of cytochrome P450Nor and its mutants (Ser286 Val, Thr) in the ferric resting state at cryogenic temperature: a comparative analysis with monooxygenase cytochrome P450S.* Journal of Inorganic Biochemistry, 2000. **81**: p. 191-205.
60. Podust, L., et al., *The 1.92 A structure of Streptomyces coelicolor A3(2) Cyp154C1:a new monooxygenase that functionalizes macrolide ring systems.* Journal of Biological Chemistry, 2003. **278**: p. 12214-12221.

61. Park, S.Y., et al., *Crystallization and preliminary X-ray diffraction analysis of a cytochrome P450(Cyp119) from Sulfolobus solfataricus.* Acta Crystallographica Section D-Biological Crystallography, 2000. **56**: p. 1173-1175.
62. Lee, D.S., et al., *Substrate recognition and molecular mechanism of fatty acid hydroxylation by cytochrome P450 from Bacillus subtilis.Crystallographic, Spectroscopic, and Mutational Studies.* Journal of Biological Chemistry, 2003. **278**: p. 9761-9767.
63. Zerbe, K., et al., *Crystal structure of Oxyb, a cytochrome P450 implicated in an oxidative phenol coupling reaction during vancomycin biosynthesis.* Journal of Biological Chemistry, 2002. **277**: p. 47476-47485.
64. Yano, J.K., et al., *Preliminary characterization and crystal structure of a thermostable cytochrome P450 from Thermus thermophilus.* Journal of Biological Chemistry, 2003. **278**: p. 608-606.
65. Durley, R.C.E. and F.S. Mathews, *Refinement and structural analysis of bovine cytochrome b(5) at 1.5 angstrom resolution.* Acta Crystallographica Section D-Biological Crystallography, 1996. **52**: p. 65-76.
66. Rodriguez-Maranon, M.J., et al., *13C NMR spectroscopic and X-ray crystallographic study of the role played by mitochondrial cytochrome b5 heme propionates in the electrostatic binding to cytochrome c.* Biochemistry, 1996. **35**: p. 16378-16390.
67. Kostanjevecki, V., et al., *Structure and characterization of Ectothiorhodospira Vacuolata cytochrome b558, a prokaryotic homologue of cytochrome b5.* Journal of Biological Chemistry, 1999. **274**: p. 25614-25620.
68. Altuve, A., et al., *Probing the differences between rat liver outer mitochondrial membrane cytochrome b5 and microsomal cytochrome b5.* Biochemistry, 2001. **40**: p. 9469-9483.
69. Tezcan, F.A., et al., *Electron tunneling in protein crystals.* Proceedings of the National Academy of Sciences of the United States of America, 2001. **98**: p. 5002-5006.
70. Benning, M.M., T.E. Meyer, and H.M. Holden, *X-ray structure of the cytochrome c2 isolated from Paracoccus denitrificans refined to 1.7-A resolution.* Archives of Biochemistry and Biophysics, 1994. **310**: p. 460-466.
71. Benini, S., et al., *Crystal structure of oxidized Bacillus Pasteuriicytochrome C(553) at 0.97-A resolutionx.* Biochemistry, 2000. **39**: p. 13115-13126.
72. Benning, M.M., et al., *Molecular structure of cytochrome c2 isolated from Rhodobacter capsulatus determined at 2.5 A resolution.* Journal of Molecular Biology, 1991. **220**: p. 673-685.
73. Than, M.E., et al., *Thermus thermophilus cytochrome-c552: a new highly thermostable cytochrome-c structure obtained by MAD phasing.* Journal of Molecular Biology, 1997. **271**: p. 629-644.
74. Schnackenberg, J., et al., *Amino acid sequence, crystallization and structure*

determination of reduced and oxidized cytochrome c6 from the green alga Scenedesmus obliquus. Journal of Molecular Biology, 1999. **290**: p. 1019-1030.

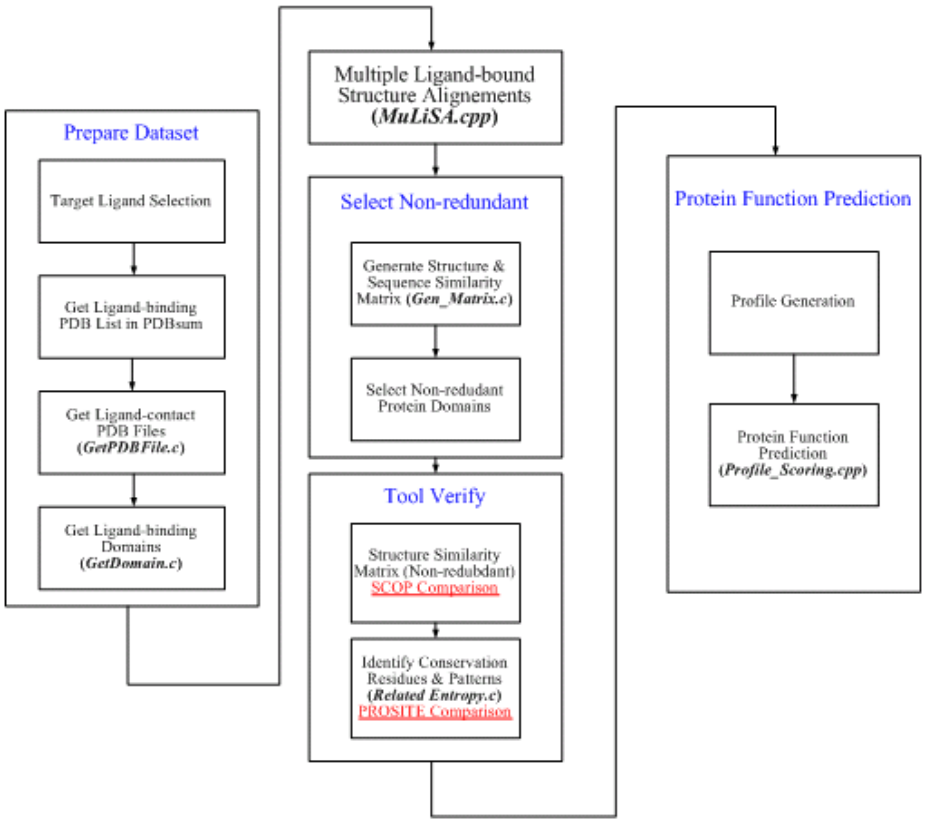
75. Ochi, H., et al., *Structure of rice ferricytochrome c at 2.0 A resolution.* Journal of Molecular Biology, 1983. **166**: p. 407-418.
76. Louie, G.V. and G.D. Brayer, *High-resolution refinement of yeast iso-1-cytochrome c and comparisons with other eukaryotic cytochromes c.* Journal of Molecular Biology, 1990. **214**: p. 527-555.
77. Sogabe, S. and K. Miki, *Refined crystal structure of ferrocytochrome c2 from Rhodopseudomonas Viridis at 1.6 A resolution.* Journal of Molecular Biology, 1995. **252**: p. 235-247.
78. Frazao, C., et al., *Ab initio determination of the crystal structure of cytochrome c6 and comparison with plastocyanin.* Structure, 1995. **3**: p. 1159-1169.
79. Axelrod, H.L., et al., *Crystallization and -ray structure determination of cytochrome-c(2) from Rodobacter-sphaeroides in 3 crystal froms.* Acta Crystallographica Section D-Biological Crystallography, 1994. **50**: p. 596-602.
80. Kerfeld, C.A., et al., *The structure of chloroplast cytochrome c6 at 1.9 A resolution: evidence for functional oligomerization.* Journal of Molecular Biology, 1995. **250**: p. 627-647.
81. Leys, D., et al., *Crystal structures of an oxygen-binding cytochrome c from Rhodobacter sphaeroides.* Journal of Biological Chemistry, 2000. **275**: p. 16050-16056.
82. Sawaya, M.R., et al., *Structures of cytochrome c-549 and cytochrome c6 from Arthrospira maxima.* Biochemistry, 2001. **40**: p. 9215-9225.
83. Geremia, S., et al., *Cleavage of the iron-methionine bond in c-Type cytochromes: crystal structure of oxidized and reduced cytochrome c(2) from Rhodopseudomonas palustris and its ammonia complex.* Protein Science, 2002. **11**: p. 6-17.
84. Yamada, S., et al., *Structure of cytochrome c6 from the red alga Porphyra yezoensis at 1.57 A resolution.* Acta Crystallographica Section D-Biological Crystallography, 2000. **56**: p. 1577-1582.
85. Benning, M.M., T.E. Meyer, and H.M. Holden, *Molecular structure of a high potential cytochrome c2 isolated from Rhodopila globiformis.* Archives of Biochemistry and Biophysics, 1996. **333**: p. 338-348.
86. Camara-Artigas, A., J.C. Williams, and J.P. Allen, *Structure of cytochrome c2 from Rhodospirillum centenum.* Acta Crystallographica Section D-Biological Crystallography, 2001. **57**: p. 1498-1505.
87. Dikiy, A., et al., *Structural basis for the molecular properties of cytochrome c(6).* Biochemistry, 2002. **41**: p. 14689-14699.
88. Matsuura, Y., T. Takano, and R.E. Dickerson, *Structure of cytochrome c551 from Pseudomonas aeruginosa refined at 1.6 A resolution and comparison of the two redox*

forms. Journal of Molecular Biology, 1982. **156**: p. 389-409.

89. Shibata, N., et al., *Basis for monomer stabilization in Rhodopseudomonas palustris cytochrome c' derived from the crystal structure.* Journal of Molecular Biology, 1998. **284**: p. 751-760.
90. Ren, Z., T. Meyer, and D.E. McRee, *Atomic structure of a cytochrome c' with an unusual ligand-controlled dimer dissociation at 1.8 A resolution.* Journal of Molecular Biology, 1993. **234**: p. 433-445.
91. Dobbs, A.J., et al., *Three-dimensional structure of cytochrome c' from two Alcaligenes species and the implications for four-helix bundle structures.* Acta Crystallographica Section D-Biological Crystallography, 1996. **52**: p. 356-368.
92. Tahirov, T.H., et al., *High-resolution crystal structures of two polymorphs of cytochrome c' from the purple phototrophic bacterium rhodobacter capsulatus.* Journal of Molecular Biology, 1996. **259**: p. 467-479.

Appendix

A. Flow chart with Programs



In this research, we used six C or C++ language written programs, they are as follows:

1.GetPDBFile.c: get PDB files which are on PDBsum lists.

2.GetDomain.c: get ligand binding domains from above selected PDB files.

3.MuLiSA.cpp: ligand superimposition and generate alignments. Thanks for Mr. K.P. Liu's help.

4.Gen_Matrix.c: generate structure similarity matrixes and sequence identity matrixes.

5.Related Entropy.c: calculate position entropy and z-socre of multiple alignments.

Profile_Scoring.cpp: search for SWISS-PROT sequences and generate profile scoring lists. Thanks for Mr. D.K. Yang's help.

B. Source code of program GetDomain.c

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <string.h>
#include <stdlib.h>
//*****************************************Parameter//
#define NearLigand      5                                   // Lignad-contact cutoff
#define LIGAND          "HEM"                               // Ligand Name
#define LIGAND_ATOM 43                                      // Ligand atom number
//Parameter*****************************************//


#define PDBpath "H:\\PDB\\"                 // PDB File Path
#define SCOPpath "H:\\SCOP\\"               // SCOP File Path
#define targetpath ".\\target\\"            // target path
#define pdbhead "pdb"                       // PDB File Name Head
#define pdbtail ".ent"                      // PDB File Name Tail

#define MAXFILENUM  2000                    // MAX File Number
#define MAXATOM     100000                  // MAX Atom Number
#define MAXRES      1000                    // MAX residue number
#define MAXLEN      150                     // MAX length of each line
#define MAXLIG      50                      // MAX number of ligand of one PDB file
#define MAXLIGANT   100                     // MAX atom number of ligand
#define MAXDOMAIN   50                      // MAX number of domain of one PDB file
#define NAMELEN     5                       // PDB ID length+1

void Initial(void);                 // Initialize Variables
void ReadFileList(void);            //* Read File List(outlist.txt) Function
void GetDomain(void);               //** Get All Domain names
void GetLigRes(void);               //***// Get Ligand-contact Residues
void Res_To_Domain(void);           /****/// Find Domains From Residues
void SCOP_List(void);               //***** Write SCOP List File
void SCOP_File(void);               //****** Write SCOP File with Ligand

void ReadPDB(int, char*);           //***// Read PDB File to PDBTEMP structure
void SelectLigRes(int,int);         //***// Select Lignad-Residues distance & Store the Residues inside cutoff
double Distance(int,int,int);       //***// Count distance
int Belong_To_Domain(int,int,int,int);  /****/// Check Domains & Residues

struct Protein_Domain
{
        char    Name[20];                       // Store PDB IDs
        int     Useful;                         // Useful protein?? (0:NO; 1;YES)
        int     Usefuldomain[MAXDOMAIN]; // Useful domain?? (0:NO; 1;YES)
        int     domainNUM;                      // Domain number of protein
        int     ligandNUM[MAXDOMAIN];   // Ligand number of each domain
       int     domain_lig_resnum[MAXDOMAIN][MAXLIG];   // Store residue number of ligand of each
       ligand-contact domain
        char    domain_lig_chain[MAXDOMAIN][MAXLIG][2]; // Store chain ID of ligand of each
ligand-contact domain
        char    domain_name[MAXDOMAIN][20];             // Store domain names (ex:d1a0a__)
        char    domain_class[MAXDOMAIN][20];            // Store domain class (ex:c.26.1.1)
        char    domain_region[MAXDOMAIN][MAXLEN];       // Store domain region (ex:- or A: or
A:78-156,A:249-463)
}Protein[MAXFILENUM];

struct PDBFile
{
        char    HEADER[MAXATOM][7];             // HEADER
        int     ATOM_NUM[MAXATOM];              // Atom Number
        char    ATOM_NAME[MAXATOM][5];          // Atom Name
```

```
        char    RES_NAME[MAXATOM][4];           // Residue Name
        char    CHAIN_ID[MAXATOM][2];           // Chain ID
        int     RES_NUM[MAXATOM];                       // Residue Number
        double      X[MAXATOM];                         // X-coordinates
        double      Y[MAXATOM];                         // Y-coordinates
        double      Z[MAXATOM];                         // Z-coordinates

}PDBTEMP;

struct LigInfo
{
        char    HEADER[MAXLIGANT][7];                   // HEADER
        int     ATOM_NUM[MAXLIGANT];                    // Atom Number
        char    ATOM_NAME[MAXLIGANT][5];                // Atom Name
        char    RES_NAME[MAXLIGANT][4];                 // Residue Name
        char    CHAIN_ID[MAXLIGANT][2];                 // Chain ID
        int     RES_NUM[MAXLIGANT];                     // Residue Number
        double      X[MAXLIGANT];                       // X-coordinates
        double      Y[MAXLIGANT];                       // Y-coordinates
        double      Z[MAXLIGANT];                       // Z-coordinates

//}LIGTEMP[MAXLIG];
}*LIGTEMP;

struct Ligand_Res
{
        int ligand_num;                         // Ligand Number of a File
        int res_num[MAXLIG];                    // Number of Contact Residues of Each Ligand
        int ligand_resnum[MAXLIG];              // Residue Numbers of Each Ligand
        char ligand_chain[MAXLIG][2];           // Chain ID of Each Ligand
        char chain[MAXLIG][MAXRES][2];          // Chain ID of Each Ligand-Contact Residue
        int res[MAXLIG][MAXRES];                // Residue Number of Each Ligand-Contact Residue
//}Contact_Res[MAXFILENUM];
}*Contact_Res;

static int PDBNUM;              // Store total PDB file number
static int ATOMNUM;             // Store Atom Numbers of Ligands

void main(void)
{
/*
        printf("%d\n",sizeof( struct Protein_Domain)*MAXFILENUM);
        printf("%d\n",sizeof(struct PDBFile));
        printf("%d\n",sizeof(struct LigInfo)*MAXLIG);
        printf("%d\n",sizeof(struct Ligand_Res)*MAXFILENUM);
        printf("%d\n",(sizeof( struct Protein_Domain)*MAXFILENUM + sizeof(struct PDBFile) + sizeof(struct
LigInfo)*MAXLIG + sizeof(struct Ligand_Res)*MAXFILENUM)/1024);
        getch();
*/
        ATOMNUM=LIGAND_ATOM;                    // Get Atom Numbers of Ligands

        Initial();                              // Initialize Variables
        printf("1!!\n");
        //getch();

        ReadFileList();                         // Function of read file list (outlist.txt)
        printf("2!!\n");
        //getch();

        GetDomain();                            // Fnction of getting all domain names (SCOP_dir1.65.txt)
        printf("3!!\n");
        //getch();

        Contact_Res=(struct Ligand_Res*)malloc(MAXFILENUM*sizeof(struct Ligand_Res));
        LIGTEMP=(struct LigInfo*)malloc(MAXLIG*sizeof(struct LigInfo));

        GetLigRes();                            // Function of getting ligand-contacting residues
```

```
	printf("4!!\n");
	//getch();

	free(LIGTEMP);

	Res_To_Domain();			// Function of Find Domains From Residues
	printf("5!!\n");
	//getch();

	SCOP_List();				// Function of Write SCOP List File
	printf("6!!\n");
	//getch();

	SCOP_File();				// Function of Write SCOP File with Ligand
	printf("7!!\n");

	free(Contact_Res);

}

void Initial(void)			// Function of initialize
{
	int i,j,k;

	for(i=0;i<MAXFILENUM;i++){

		Protein[i].Name[0]='\0';
		Protein[i].Useful=0;
		Protein[i].domainNUM=0;

		for(j=0;j<MAXDOMAIN;j++){

			Protein[i].domain_name[j][0]='\0';
			Protein[i].domain_class[j][0]='\0';
			Protein[i].domain_region[j][0]='\0';
			Protein[i].ligandNUM[j]=0;
			Protein[i].Usefuldomain[j]=0;

			for(k=0;k<MAXLIG;k++){

				Protein[i].domain_lig_resnum[j][k]=0;
				Protein[i].domain_lig_chain[j][k][0]='\0';
			}
		}
	}
}

void ReadFileList(void)		// Function of read file list (outlist.txt)
{

	FILE *list;			// File pointer to read outlist.txt
	int line;			// Count FILE(line) number

	char ltemp[MAXLEN];	// Temp record

	//////////////////////////////////////////
	///
	///Open & Read outlist.txt
	///
	//////////////////////////////////////////

	if((list=fopen("outlist.txt","r"))==NULL)
	{
		printf("Open outlist.txt Error!\n");
	}
	else
```

```
	{
		line=0;

		while(fgets(ltemp,MAXLEN,list)!=NULL){

			if(line==0)			// Read total PDB file number
			{
				strtok(ltemp," \n");
				PDBNUM=atoi(ltemp);
				//printf("%d\n",PDBNUM);
			}
			else				// Store PDB IDs
			{
				strtok(ltemp," \n");
				strncpy(Protein[line-1].Name,ltemp,strlen(ltemp));
				Protein[line-1].Name[strlen(ltemp)]='\0';
				printf("!!%s!!\n",Protein[line-1].Name);
				//getch();
			}

			line++;
		}
	}

	fclose(list);
}

void GetDomain(void)		// Get All Domain names of proteins
{
	FILE *scop;			// File pointer to read SCOP_dir1.65.txt

	char ltemp[MAXLEN],strtemp[MAXLEN];	// Temp record

	int i,temp;			// Variables

	if((scop=fopen("SCOP_dir1.65.txt","r"))==NULL)
	{
		printf("Open SCOP_dir1.65.txt Error!\n");
	}
	else
	{
		for(i=0;i<PDBNUM;i++){		//for all PDB

			rewind(scop);
			Protein[i].domainNUM=0;

			while(fgets(ltemp,MAXLEN,scop)!=NULL){

				if( strncmp(Protein[i].Name,ltemp+8,4)==0 )
				{
					//Get domain name
					strncpy(Protein[i].domain_name[Protein[i].domainNUM],ltemp,7);
					Protein[i].domain_name[Protein[i].domainNUM][7]='\0';
				   printf("3:domain_name:!!%s!!\n",Protein[i].domain_name[Protein[i].domainNUM]);

					// Get domain region
					strtemp[0]='\0';
					strcpy(strtemp,ltemp+13);
					strtok(strtemp," \t\n");

				  strncpy(Protein[i].domain_region[Protein[i].domainNUM],strtemp,strlen(strtemp));
					Protein[i].domain_region[Protein[i].domainNUM][strlen(strtemp)]='\0';

				 printf("domain_region:!!%s!!\n",Protein[i].domain_region[Protein[i].domainNUM]);

					// Get domain class
					temp=strlen(strtemp);
```

```
                                strtemp[0]='\0';
                                strcpy(strtemp,ltemp+14+temp);
                                strtok(strtemp," \t\n");
                                strcpy(Protein[i].domain_class[Protein[i].domainNUM],strtemp);
                                Protein[i].domain_class[Protein[i].domainNUM][strlen(strtemp)]='\0';

                               printf("domain_class:!!%s!!\n",Protein[i].domain_class[Protein[i].domainNUM]);
                                printf("\n");

                                Protein[i].domainNUM++;
                        }
                    }
                }
                fclose(scop);
        }
}

void GetLigRes(void)                    // Function of getting ligand-contacting residues
{
        FILE *pdb;                              // File pointers of pdb file
        FILE *nopdb,*NMRpdb;                    // File pointer of note files (no PDB file & NMR PDB file)

        char filetemp[MAXLEN];                  // Temp record file name
        char ltemp[MAXLEN];                     // Temp record
        int line;                               // Count ATOM & HETATM
        int i,j;

        nopdb=fopen("nopdb.txt","w");
        NMRpdb=fopen("NMRpdb.txt","w");

        printf("PDBNUM:%d\n",PDBNUM);
        for(i=0;i<PDBNUM;i++){//for all PDB

                printf("%s\n",Protein[i].Name);
                sprintf(filetemp,"%s%s%s%s",PDBpath,pdbhead,Protein[i].Name,pdbtail);
                printf("4:%s\t%d\n",filetemp,i);

                //initial
                for(j=0;j<MAXATOM;j++){

                        PDBTEMP.HEADER[j][0]='\0';
                        PDBTEMP.ATOM_NUM[j]=0;
                        PDBTEMP.ATOM_NAME[j][0]='\0';
                        PDBTEMP.RES_NAME[j][0]='\0';
                        PDBTEMP.CHAIN_ID[j][0]='\0';
                        PDBTEMP.RES_NUM[j]=0;
                        PDBTEMP.X[j]=0;
                        PDBTEMP.Y[j]=0;
                        PDBTEMP.Z[j]=0;
                }

                if((pdb=fopen(filetemp,"r"))==NULL)//If no PDB File
                {
                        printf("Open %s Error!\n",filetemp);
                        fprintf(nopdb,"%s\n",filetemp);//record PDB ID in nopdb.txt
                }
                else            // Read PDB File
                {
                        line=0;
                        while(fgets(ltemp,MAXLEN,pdb)!=NULL){

                                if( strncmp(ltemp,"MODEL",5)==0 )// Neglect NMR structure
                                {
                                        printf("NMR:%s\n",Protein[i].Name);
                                        fprintf(NMRpdb,"%s\n",Protein[i].Name);//record PDB ID in NMRpdb.txt
                                        //getch();
                                        break;
```

```
                        }
                        if(strncmp(ltemp,"HETATM",6)==0 || strncmp(ltemp,"ATOM    ",6)==0) //Get
        HETATM & ATOM
                        {
                                ReadPDB(line,ltemp);            // Read PDB File to PDBTEMP structure
                                line++;                         // Count ATOM & HETATM
                        }
                }
                fclose(pdb);

                /////// Select ligand-contact residues    (inside cutoff distance)

                SelectLigRes(i,line);
        }
    }
    fclose(nopdb);
    fclose(NMRpdb);
}

void ReadPDB(int line, char *ltemp)                     // Function of read PDB File to PDBTEMP structure
{
    char strtemp[MAXLEN];                               // Temp record

    //printf("%d\n",line);
    strncpy(PDBTEMP.HEADER[line],ltemp,6);              // Get HEADER
    PDBTEMP.HEADER[line][6]='\0';
    //printf("%s ",PDBTEMP.HEADER[line]);

    strncpy(strtemp,ltemp+7,4);                         // Get atom number
    strtemp[4]='\0';
    PDBTEMP.ATOM_NUM[line]=atoi(strtemp);
    //printf("%5d ",PDBTEMP.ATOM_NUM[line]);

    strncpy(PDBTEMP.ATOM_NAME[line],ltemp+12,4);        // Get atom name
    PDBTEMP.ATOM_NAME[line][4]='\0';
    //printf("%s ",PDBTEMP.ATOM_NAME[line]);

    strncpy(PDBTEMP.RES_NAME[line],ltemp+17,3);         // Get residue name
    PDBTEMP.RES_NAME[line][3]='\0';
    //printf("%s ",PDBTEMP.RES_NAME[line]);

    strncpy(PDBTEMP.CHAIN_ID[line],ltemp+21,1);         // Get chain ID
    if( strncmp(PDBTEMP.CHAIN_ID[line]," ",1)==0 )
    {
        PDBTEMP.CHAIN_ID[line][0]='-';
    }
    PDBTEMP.CHAIN_ID[line][1]='\0';

    //printf("%s ",PDBTEMP.CHAIN_ID[line]);

    strncpy(strtemp,ltemp+22,4);                        // Get residue number
    strtemp[4]='\0';
    PDBTEMP.RES_NUM[line]=atoi(strtemp);
    //printf("%4d ",PDBTEMP.RES_NUM[line]);

    strncpy(strtemp,ltemp+30,8);                        // Get x-coordinate
    strtemp[8]='\0';
    PDBTEMP.X[line]=atof(strtemp);
    //printf("%5.3lf ",PDBTEMP.X[line]);

    strncpy(strtemp,ltemp+38,8);                        // Get y-coordinate
    strtemp[8]='\0';
    PDBTEMP.Y[line]=atof(strtemp);
    //printf("%5.3lf ",PDBTEMP.Y[line]);

    strncpy(strtemp,ltemp+46,8);                        // Get z-coordinate
    strtemp[8]='\0';
```

```
        PDBTEMP.Z[line]=atof(strtemp);
        //printf("%5.3lf\n",PDBTEMP.Z[line]);
}

void SelectLigRes(int pdb,int line)                    // Function of select Lignad-Residues distance
& Store the Residues inside cutoff
{

        int lig;                    // Ligand number
        int ligatm;                 // Ligand atom number
        int resnumtemp;             // Count ligand change residue number
        char chainIDtemp[2];        // Change Ligand of chain ID
        int flag;
        int i,j,k,l;

        lig=0;
        ligatm=0;
        resnumtemp=0;
        chainIDtemp[0]='\0';

        //initial
        Contact_Res[pdb].ligand_num=0;

        for(j=0;j<MAXLIG;j++){

                Contact_Res[pdb].res_num[j]=0;
                Contact_Res[pdb].ligand_resnum[j]=0;
                Contact_Res[pdb].ligand_chain[j][0]='\0';

                for(k=0;k<MAXRES;k++){

                        Contact_Res[pdb].chain[j][k][0]=0;
                        Contact_Res[pdb].res[j][k]=0;
                }
        }

/*
        struct Ligand_Res
        {
                int ligand_num;                                 // Ligand Number of a File
                int res_num[MAXLIG];                            // Number of Contact Residues of Each Ligand
                int ligand_resnum[MAXLIG];                      // Residue Numbers of Each Ligand
                char ligand_chain[MAXLIG][2];                   // Chain ID of Each Ligand
                char chain[MAXLIG][MAXRES][2];                  // Chain ID of Each Ligand-Contact Residue
                int res[MAXLIG][MAXRES];                        // Residue Number of Each Ligand-Contact
Residue

        }*Contact_Res;
*/
        //initial
        for(i=0;i<MAXLIG;i++){

                for(j=0;j<MAXLIGANT;j++){


                        LIGTEMP[i].HEADER[j][0]='\0';
                        LIGTEMP[i].ATOM_NUM[j]=0;
                        LIGTEMP[i].ATOM_NAME[j][0]='\0';
                        LIGTEMP[i].RES_NAME[j][0]='\0';
                        LIGTEMP[i].CHAIN_ID[j][0]='\0';
                        LIGTEMP[i].RES_NUM[j]=0;
                        LIGTEMP[i].X[j]=0;
                        LIGTEMP[i].Y[j]=0;
                        LIGTEMP[i].Z[j]=0;
                }
        }
/*
```

```
	struct LigInfo
	{
		char	HEADER[MAXLIGANT][7];			// HEADER
		int	ATOM_NUM[MAXLIGANT];			// Atom Number
		char	ATOM_NAME[MAXLIGANT][5];		// Atom Name
		char	RES_NAME[MAXLIGANT][4];			// Residue Name
		char	CHAIN_ID[MAXLIGANT][2];			// Chain ID
		int	RES_NUM[MAXLIGANT];			// Residue Number
		double	X[MAXLIGANT];				// X-coordinates
		double	Y[MAXLIGANT];				// Y-coordinates
		double	Z[MAXLIGANT];				// Z-coordinates

	}LIGTEMP[MAXLIG];
*/


	for(i=0;i<line;i++){		// Store ligand information

		if(		strcmp(PDBTEMP.HEADER[i],"HETATM")==0		&&
strcmp(PDBTEMP.RES_NAME[i],LIGAND)==0 )
		{
			if( resnumtemp==0 )
			{
				resnumtemp=PDBTEMP.RES_NUM[i];
				strcpy(chainIDtemp,PDBTEMP.CHAIN_ID[i]);
				Contact_Res[pdb].ligand_resnum[lig]=PDBTEMP.RES_NUM[i];
				Contact_Res[pdb].ligand_chain[lig][0]=PDBTEMP.CHAIN_ID[i][0];
				Contact_Res[pdb].ligand_chain[lig][1]='\0';
			}
			if(	resnumtemp!=0	&&	(resnumtemp!=PDBTEMP.RES_NUM[i]	||
strcmp(chainIDtemp,PDBTEMP.CHAIN_ID[i])!=0) )// If more than one Ligand
			{
				lig++;
				ligatm=0;
				resnumtemp=PDBTEMP.RES_NUM[i];
				strcpy(chainIDtemp,PDBTEMP.CHAIN_ID[i]);
				Contact_Res[pdb].ligand_resnum[lig]=PDBTEMP.RES_NUM[i];
				Contact_Res[pdb].ligand_chain[lig][0]=PDBTEMP.CHAIN_ID[i][0];
				Contact_Res[pdb].ligand_chain[lig][1]='\0';
			}

			strcpy(LIGTEMP[lig].HEADER[ligatm],PDBTEMP.HEADER[i]);
			//printf("%s ",LIGTEMP[lig].HEADER[ligatm]);

			LIGTEMP[lig].ATOM_NUM[ligatm]=PDBTEMP.ATOM_NUM[i];
			//printf("%5d ",LIGTEMP[lig].ATOM_NUM[ligatm]);

			strcpy(LIGTEMP[lig].ATOM_NAME[ligatm],PDBTEMP.ATOM_NAME[i]);
			//printf("%s ",LIGTEMP[lig].ATOM_NAME[ligatm]);

			strcpy(LIGTEMP[lig].RES_NAME[ligatm],PDBTEMP.RES_NAME[i]);
			//printf("%s ",LIGTEMP[lig].RES_NAME[ligatm]);

			strcpy(LIGTEMP[lig].CHAIN_ID[ligatm],PDBTEMP.CHAIN_ID[i]);
			//printf("%s ",LIGTEMP[lig].CHAIN_ID[ligatm]);

			LIGTEMP[lig].RES_NUM[ligatm]=PDBTEMP.RES_NUM[i];
			//printf("%5d ",LIGTEMP[lig].RES_NUM[ligatm]);

			LIGTEMP[lig].X[ligatm]=PDBTEMP.X[i];
			//printf("%8.3lf ",LIGTEMP[lig].X[ligatm]);

			LIGTEMP[lig].Y[ligatm]=PDBTEMP.Y[i];
			//printf("%8.3lf ",LIGTEMP[lig].Y[ligatm]);

			LIGTEMP[lig].Z[ligatm]=PDBTEMP.Z[i];
			//printf("%8.3lf\n",LIGTEMP[lig].Z[ligatm]);
```

```
                //printf("%d\t%d\n",lig,ligatm);

                ligatm++;
            }
        }

        Contact_Res[pdb].ligand_num=lig+1;          // Record ligand number of the PDB
        //printf("%d*****%d\t%d\n",pdb,Contact_Res[pdb].ligand_num,lig);
        //getch();

        ////// Count distance of ligand & residue atoms & store it

        //printf("%d\t%d\t%d\n",lig,ATOMNUM,line);
        for(i=0;i<=lig;i++){                                    // For each ligand

            for(j=0;j<ATOMNUM;j++){                             // For all ligand atoms

                resnumtemp=0;

                for(k=0;k<line;k++){                            // For all PDB file lines

                    if( strcmp(PDBTEMP.HEADER[k],"ATOM  ")==0)
                    {
                        //printf("%lf\n",Distance(i,j,k));
                        //getch();

                        if (Distance(i,j,k)<=NearLigand) // If close
                        {
                            //neglect same residue number (more than one atom close to ligand of same residue)
flag=1;


                            for(l=0;l<Contact_Res[pdb].res_num[i];l++){ // Check for repeat

                            if(Contact_Res[pdb].res[i][l]==PDBTEMP.RES_NUM[k]           && strcmp(Contact_Res[pdb].chain[i][l],PDBTEMP.CHAIN_ID[k])==0 )
                                {
                                    flag=0;
                                }
                            }

                            if(flag==1)// Store information
                            {

        Contact_Res[pdb].chain[i][Contact_Res[pdb].res_num[i]][0]=PDBTEMP.CHAIN_ID[k][0];
                                    Contact_Res[pdb].chain[i][Contact_Res[pdb].res_num[i]][1]='\0';

        Contact_Res[pdb].res[i][Contact_Res[pdb].res_num[i]]=PDBTEMP.RES_NUM[k];

        //printf("%5d!!\t%s!!\n",Contact_Res[pdb].res_num[i],Contact_Res[pdb].chain[i][Contact_Res[pdb].res_num[i]]);
                                Contact_Res[pdb].res_num[i]++;
                            }

                            /*
                                struct Ligand_Res
                                {
                                    int ligand_num;             // Ligand Number of a File
                                    int res_num[MAXLIG];    // Number of Contact Residues of Each Ligand

                                    char chain[MAXLIG][MAXRES][2];  // Chain ID of Each Ligand-Contact Residue

                                    int res[MAXLIG][MAXRES];    // Residue Number of Each Ligand-Contact Residue

                                }Contact_Res[MAXFILENUM];
                            */
```

```
                                }
                        }
                }
        }
}
/*
        printf("%3d!!\n",Contact_Res[pdb].ligand_num);
        printf("%s\n",Protein[pdb].Name);

        for(i=0;i<Contact_Res[pdb].ligand_num;i++){


        printf("lig_resnum:%3d\tchain:%s\n",Contact_Res[pdb].ligand_resnum[i],Contact_Res[pdb].ligand_chain[i
]);

                printf("**%d!!\n",Contact_Res[pdb].res_num[i]);

                for(j=0;j<Contact_Res[pdb].res_num[i];j++){

                        printf("%3d\t%s\n",Contact_Res[pdb].res[i][j],Contact_Res[pdb].chain[i][j]);
                }

        }
        getch();
*/

}

void Res_To_Domain(void)                        // Function of Find Domains From Residues
{
        int domainflag,proteinflag;
        int i,j,k,l;

        ///////*************************Record contact ligand-number of each domain
        //printf("yes!\n");
        for(i=0;i<PDBNUM;i++){                                  //Each PDB
                //printf("%d**********%d\n",i,PDBNUM);

                for(j=0;j<Contact_Res[i].ligand_num;j++){               //Each ligand of this PDB

                        for(k=0;k<Contact_Res[i].res_num[j];k++){       //Each ligand-contact residue of this PDB

                                for(l=0;l<Protein[i].domainNUM;l++){    //Each domain of this PDB

                                        //printf("%d\t%d\t%d\t%d!!!out\n",i,j,k,l);
                                        if( Belong_To_Domain(i,j,k,l)!=0 )              // Check Domains & Residues
                                        {
                                                if( Protein[i].ligandNUM[l]==0 )
                                                {

        Protein[i].domain_lig_resnum[l][Protein[i].ligandNUM[l]]=Contact_Res[i].ligand_resnum[j];

        Protein[i].domain_lig_chain[l][Protein[i].ligandNUM[l]][0]=Contact_Res[i].ligand_chain[j][0];
                                                        Protein[i].domain_lig_chain[l][Protein[i].ligandNUM[l]][1]='\0';

                                                        Protein[i].ligandNUM[l]++;
                                                }
                                                else
                                                {

        if( Protein[i].domain_lig_resnum[l][Protein[i].ligandNUM[l]-1]!=Contact_Res[i].ligand_resnum[j] ||

        Protein[i].domain_lig_chain[l][Protein[i].ligandNUM[l]-1][0] != Contact_Res[i].ligand_chain[j][0] )
                                                        {

        Protein[i].domain_lig_resnum[l][Protein[i].ligandNUM[l]]=Contact_Res[i].ligand_resnum[j];
```

```
Protein[i].domain_lig_chain[l][Protein[i].ligandNUM[l]][0]=Contact_Res[i].ligand_chain[j][0];

Protein[i].domain_lig_chain[l][Protein[i].ligandNUM[l]][1]='\0';

                                        Protein[i].ligandNUM[l]++;
                                }
                        }

                        /*
                        struct Protein_Domain
                        {
                                /////char        Name[20];
                                // Store PDB IDs
                                /////int         Useful;
                                // Useful protein?? (0:NO; 1;YES)
                                /////int         Usefuldomain[MAXDOMAIN];
                                // Useful domain?? (0:NO; 1;YES)
                                ////////////int             domainNUM;
                                // Domain number of protein
                                int             ligandNUM[MAXDOMAIN];
                                // Ligand number of each domain
                                int             domain_lig_resnum[MAXDOMAIN][MAXLIG];
                                // Store residue number of ligand of each ligand-contact domain
                                char    domain_lig_chain[MAXDOMAIN][MAXLIG][2];
                                // Store chain ID of ligand of each ligand-contact domain
                                ////////////char domain_name[MAXDOMAIN][20];
                                // Store domain names (ex:d1a0a__)
                                ////////////char domain_class[MAXDOMAIN][20];
                                // Store domain class (ex:c.26.1.1)
                                ////////////char domain_region[MAXDOMAIN][20];
                                // Store domain region (ex:- or A: or A:78-156,A:249-463)

                        }Protein[MAXFILENUM];
                        */
                                }
                        }
                }
        }
        //printf("yes1!\n");

        for(i=0;i<PDBNUM;i++){

                printf("5:%s!\n",Protein[i].Name);
//              printf("Protein[i].domainNUM:%d\n",Protein[i].domainNUM);

//              for(j=0;j<Protein[i].domainNUM;j++){

//                      printf("%s!!%s!!\n",Protein[i].domain_name[j],Protein[i].domain_region[j]);
//                      printf("ligandNUM:%2d!!\n",Protein[i].ligandNUM[j]);

//                      for(k=0;k<Protein[i].ligandNUM[j];k++){

//
        printf("%d!!!%s!!!\n",Protein[i].domain_lig_resnum[j][k],Protein[i].domain_lig_chain[j][k]);

//                      }
//              }
//              printf("**********************************************\n");
//              getch();
        }

        //********** Protein.Usefuldomain tag (Only get domains with only one ligand)

        //printf("Protein.Usefuldomain tag!!\n");
        for(i=0;i<PDBNUM;i++){                  // Protein.Usefuldomain tag
```

```
		domainflag=999;

		//printf("Protein[i].domainNUM:%d\n",Protein[i].domainNUM);

		for(j=0;j<Protein[i].domainNUM;j++){

			//printf("ligandNUM:%2d!!\n",Protein[i].ligandNUM[j]);

			if( Protein[i].ligandNUM[j]==1 && domainflag!=999 )
			{

	if( strcmp(Protein[i].domain_region[j]+1,Protein[i].domain_region[domainflag]+1)==0 )
				{
					Protein[i].Usefuldomain[j]=0;
				}
				else
				{
					Protein[i].Usefuldomain[j]=1;
				}
			}
			if( Protein[i].ligandNUM[j]==1 && domainflag==999 )
			{
				Protein[i].Usefuldomain[j]=1;
				domainflag=j;;
			}

			//printf("*******%d******\n",Protein[i].Usefuldomain[j]);
		}
	}

	//*** Protein.Useful tag (Only get proteins with only one ligand-contact domain)

	//printf("Protein.Useful tag!!\n");
	for(i=0;i<PDBNUM;i++){				// Protein.Useful tag

		proteinflag=0;

		for(j=0;j<Protein[i].domainNUM;j++){

			if( Protein[i].Usefuldomain[j]==1 )
			{
				Protein[i].Usefuldomain[j]=1;
				proteinflag++;
			}
		}
		if( proteinflag==1)
		{
			Protein[i].Useful=1;
		}
	}
/*
	for(i=0;i<PDBNUM;i++)
	{
		printf("%d!!\t%d!!\n",i,Protein[i].Useful);
	}
*/

}

int Belong_To_Domain(int pdb,int lig,int res,int domain)		// Function of Check Domains & Residues
{

	int flag,series_flag;		// flag:1 for belong to domain; series_flag:concern domain_region without chain(ex:d1lfi_1	1lfi	1-334 c.94.1.2)
	int region_num;				// Count region number (ex:2 for A:15-114,A:308-346)seperated by ","
```

```
        char *token;
        char domains[5][MAXLEN];      // Temp store FULL domain region
        char chains[5][2];            // Temp store chain IDs of domain region ('A' for A:15-114)
        int series[5][2];              // Temp store residue series of domain region(ex:15 & 114 for A:15-114)
        int adjust;                   // Count series number (0 & 1)
        int comma;                    // Check domain region number
        int i;

        //initial
        for(i=0;i<5;i++){

                domains[i][0]='\0';
                chains[i][0]='\0';
                series[i][0]=0;
                series[i][1]=0;
        }

/*
        printf("%s!!\t%s!!\t%s!!\n",Protein[pdb].domain_name[domain],Protein[pdb].domain_region[domain],Prote
in[pdb].domain_class[domain]);
        printf("\n");
        printf("%5d!!\t%s!!\n",Contact_Res[pdb].res[lig][res],Contact_Res[pdb].chain[lig][res]);
        getch();
*/

        //printf("%d\t%d\t%d\t%d!\n",pdb,lig,res,domain);

        region_num=0;
        comma=0;
        for(i=0;i< strlen(Protein[pdb].domain_region[domain]);i++){

                if( Protein[pdb].domain_region[domain][i]==',' )
                {
                        comma=1;
                }
        }
        //printf("comma:%d\n",comma);

        if(comma==1)
        {
                //printf("comma=1\n");

                token=strtok(Protein[pdb].domain_region[domain],",");
                while(token != NULL){

                        //printf("%d:\t%s\n",region_num,token);

                        strcpy(domains[region_num],token);
                        token=strtok(NULL,",");
                        //printf("%s!\n",domains[region_num]);
                        //getch();
                        region_num++;
                }
        }
        else
        {
                //printf("comma!=1\n");
                //printf("domains[region_num]:%s!\n",domains[region_num]);
                //printf("Protein[pdb].domain_region[domain]:%s\n",Protein[pdb].domain_region[domain]);

                strcpy(domains[region_num],Protein[pdb].domain_region[domain]);
                //printf("domains[region_num]:%s!!\n",domains[region_num]);
                domains[region_num][strlen(Protein[pdb].domain_region[domain])]='\0';
                //printf("domains[region_num]:%s!!!\n",domains[region_num]);
                region_num++;
        }
```

```
	//printf("%d\t%d\t%d\t%d!!\n",pdb,lig,res,domain);

	for(i=0;i<region_num;i++){

		series_flag=0;
		if( (65<=domains[i][0] && domains[i][0]<=90) || domains[i][0]==45 || ( (49<=domains[i][0] &&
domains[i][0]<=57) && domains[i][1]==58) )
		{
			//printf("%c!\n",domains[i][0]);
			chains[i][0]=domains[i][0];
			chains[i][1]='\0';
			//printf("%s!!\n",chains[i]);
			//getch();
			series_flag=1;
		}
		else
		{
			chains[i][0]=' ';
			chains[i][1]='\0';
			//printf("%s!!\n",chains[i]);
		}

		if(series_flag==1)
		{
			token=strtok(domains[i],":\t\n ,-");

			adjust=0;
			while(token != NULL){

				//printf("token:%s\n",token);
				//getch();
				token=strtok(NULL,":\t\n ,-");
				if(token==NULL)
				{
					break;
				}
				series[i][adjust]=atoi(token);
				//printf("%5d!!!\n",series[i][adjust]);
				adjust++;
			}
		}
		else
		{
			token=strtok(domains[i],":\t\n ,-");
			adjust=0;
			while(token != NULL){

				series[i][adjust]=atoi(token);
				//printf("%5d!!!\n",series[i][adjust]);
				token=strtok(NULL,":\t\n ,-");
				adjust++;
			}
		}
		//printf("%d!!\n",series[i][0]);
		//printf("%d!!\n",series[i][1]);
		//getch();
	}

	flag=0;

////////////////Compare domain region & residue number
	//printf("%d\t%d\t%d\t%d!!!\n",pdb,lig,res,domain);
	//getch();

	for(i=0;i<region_num;i++){

		//printf("%c!!%c\n",chains[i][0],Contact_Res[pdb].chain[lig][res][0]);
```

```
		//getch();

		if( chains[i][0]==Contact_Res[pdb].chain[lig][res][0] )
		{
			if( series[i][0]==0 && series[i][0]==0 )
			{
				flag=1;
			}
			else
			{
				if( series[i][0]<=Contact_Res[pdb].res[lig][res] &&
Contact_Res[pdb].res[lig][res]<=series[i][1] )
				{
					flag=1;
				}
			}
		}
	}
	if(flag==1)
	{
		return 1;
	}
	else
	{
		return 0;
	}
}

void SCOP_List(void)			// Function of Write SCOP List File
{
	FILE *scoplist;//File pointer to record SCOP list (with & without SCOP file)

	int i,j;

	scoplist=fopen("SCOP_List.txt","w");

	for(i=0;i<PDBNUM;i++){

		//printf("!!!!%d!!!!\n",Protein[i].Useful);//
		if( Protein[i].Useful==1 )//
		{
			for(j=0;j<Protein[i].domainNUM;j++){

				//printf("****%d****\n",Protein[i].Usefuldomain[j]);//
				if(Protein[i].Usefuldomain[j]==1) // Write domain name, domain region & domain class
				{
					fprintf(scoplist,"%s\t%s \t%s\n",Protein[i].domain_name[j],
Protein[i].domain_class[j], Protein[i].domain_region[j]);
				}
			}
		}
	}

	fclose(scoplist);
}

void SCOP_File(void)			// Function of Write SCOP File with Ligand
{
	FILE *fptr,*fptw,*fptr2; // fptr: read SCOP domain file; fptw: Write SCOP file to target folder; fptr2: read
PDB file for write ligand information to target SCOP file
	FILE *nofile; // write noSCOP.txt
	char path1[MAXLEN],path2[MAXLEN],path3[MAXLEN];
	char temp[4];
	char ltemp[MAXLEN];
	char restemp[MAXLEN],chaintemp[2];
	int lig_atm;
	int i,j;
```

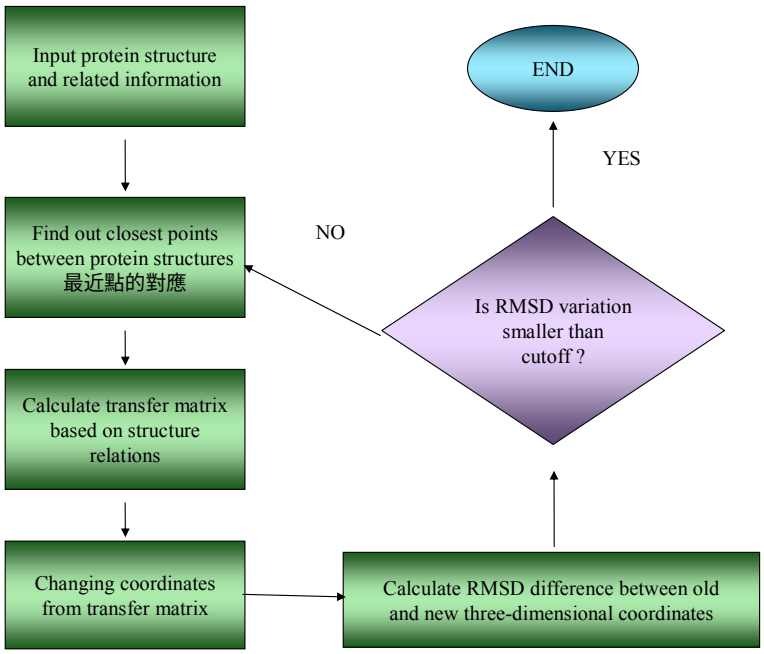
```
if((nofile=fopen("noSCOP.txt","w"))==NULL)
{
        printf("god\n");
        exit(1);
}

else
{
        for(i=0;i<PDBNUM;i++){

                if( Protein[i].Useful==1 )
                {
                        for(j=0;j<Protein[i].domainNUM;j++){

                                if( Protein[i].Usefuldomain[j]==1 )
                                {
                                        temp[0]='\0';
                                        temp[0]=Protein[i].domain_name[j][2];
                                        temp[1]=Protein[i].domain_name[j][3];
                                        temp[2]='\\';
                                        temp[3]='\0';
                                        path1[0]='\0';

sprintf(path1,"%s%s%s%s",SCOPpath,temp,Protein[i].domain_name[j],pdbtail);

                                        if((fptr=fopen(path1,"r"))==NULL)
                                        {
                                                printf("no!!\n");
                                                printf("path1:%s\n",path1);
                                                fprintf(nofile,"%s\n",path1);
                                                //fprintf(nofile,"aaaaa\n");
                                                printf("no1!!\n");
                                        }
                                        else
                                        {
                                                fclose(fptr);
                                        }
                                }
                        }

                }
        }
        fclose(nofile);
}

for(i=0;i<PDBNUM;i++){                          // Get domain File without Ligand
        //printf("yes!!\n");

        if( Protein[i].Useful==1 )
        {
                //printf("yes1!!\n");
                for(j=0;j<Protein[i].domainNUM;j++){

                        if( Protein[i].Usefuldomain[j]==1 )
                        {
                                //printf("yes2!!\n");
                                temp[0]='\0';
                                temp[0]=Protein[i].domain_name[j][2];
                                temp[1]=Protein[i].domain_name[j][3];
                                temp[2]='\\';
                                temp[3]='\0';

                                path1[0]='\0';
                                sprintf(path1,"%s%s%s%s",SCOPpath,temp,Protein[i].domain_name[j],pdbtail);

                                if((fptr=fopen(path1,"r"))==NULL)
```

```
{
        printf("no!!\n");
        printf("path1:%s\n",path1);
        fprintf(nofile,"%s\n",path1);
        printf("no1!!\n");
}
else
{
        //printf("yes3!!\n");
        printf("7:path1:%s\n",path1);
        path2[0]='\0';//SCOP File (get domain)
        sprintf(path2,"%s%s%s",targetpath,Protein[i].domain_name[j],pdbtail);
        printf("7:path2:%s\n",path2);

        fptw=fopen(path2,"w");

        while(fgets(ltemp,MAXLEN,fptr)!=NULL){//Write SCOP File

                if( strncmp(ltemp,"END",3)!=0 )
                {
                        //printf("%s!!",ltemp);
                        fprintf(fptw,"%s",ltemp);
                        //printf("write\n");
                }
        }

        path3[0]='\0';//PDB File (get ligand)
        sprintf(path3,"%s%s%s%s",PDBpath,pdbhead,Protein[i].Name,pdbtail);
        printf("path3:%s\n",path3);

        fptr2=fopen(path3,"r");

        lig_atm=0;
        while(fgets(ltemp,MAXLEN,fptr2)!=NULL){

                //printf("%s!\n",ltemp);//check1
                //getch();


                if( strncmp(ltemp,"HETATM",6)==0 )  // Get HETATM
                {
                        //printf("%s!!\n",ltemp);//check2
                        //getch();

                        restemp[0]='\0';                        // Get residue number
                        strncpy(restemp,ltemp+22,4);
                        restemp[4]='\0';

                        chaintemp[0]='\0';
                        strncpy(chaintemp,ltemp+21,1);          // Get chain ID
                        if( strncmp(chaintemp," ",1)==0 )
                        {
                                chaintemp[0]='-';
                        }
                        chaintemp[1]='\0';


        //printf("%d\t%s\n",Protein[i].domain_lig_resnum[j][0],restemp);

        //printf("%c\t%c\n",Protein[i].domain_lig_chain[j][0][0],chaintemp[0]);

                        if(  Protein[i].domain_lig_resnum[j][0]==atoi(restemp)  &&
Protein[i].domain_lig_chain[j][0][0]==chaintemp[0] && lig_atm<LIGAND_ATOM)
                        {
                                //printf("%s!!!\n",ltemp);//check3
                                //getch();
```

```
                                                fprintf(fptw,"%s",ltemp);
                                                lig_atm++;
                                            }
                                        }
                                    }
                                    fprintf(fptw,"END\n");

                                    fclose(fptr);
                                    fclose(fptw);
                                    fclose(fptr2);
                                }
                            }
                        }
                    }
                }
        fclose(nofile);
}

double Distance(int lig,int ligatm,int atm)                    // Function of count distance
{
        double dx,dy,dz;
        double dis,sum;

        dx=(LIGTEMP[lig].X[ligatm]-PDBTEMP.X[atm])*(LIGTEMP[lig].X[ligatm]-PDBTEMP.X[atm]);
        dy=(LIGTEMP[lig].Y[ligatm]-PDBTEMP.Y[atm])*(LIGTEMP[lig].Y[ligatm]-PDBTEMP.Y[atm]);
        dz=(LIGTEMP[lig].Z[ligatm]-PDBTEMP.Z[atm])*(LIGTEMP[lig].Z[ligatm]-PDBTEMP.Z[atm]);

        sum=dx+dy+dz;
        dis=sqrt(sum);
        return(dis);
}
```

C. ICP algorithm

The flow chart of the algorithm is as follows:



Procedures

Step 1: Input protein structure and related information.

Record three-dimensional coordinates of proteins and ligands.

Step 2: Find out closest points between protein structures.

Step 3: Calculate transfer matrix based on structure relations.

$$\vec{\mu}_p = \frac{1}{\mu_p}\sum_{i=1}^{N_p}\vec{p}_i \quad \text{and} \quad \vec{\mu}_x = \frac{1}{\mu_x}\sum_{i=1}^{N_x}\vec{x}_i$$

where *p* is data, *x* is model, *Np* is paired points of data, *Nx* is paired points of model. This formula is to calculate the geometry center of data and model.

$$\sum_{px} = \frac{1}{N_p}\sum_{i=1}^{N_p}\left[\left(\vec{p}_i - \vec{\mu}_p\right)\left(\vec{x}_i - \vec{\mu}_x\right)^t\right] = \frac{1}{N_p}\sum_{i=1}^{N_p}\left[\vec{p}_i \cdot \vec{x}_i^t\right] - \vec{\mu}_p\vec{\mu}_x^i$$

This formula is used to calculate Covariance Matrix.

$$Q\left(\sum_{px}\right)=\begin{bmatrix} tr\left(\sum_{px}\right) & \Delta^T \\ \Delta & \sum_{px}+\sum_{px}^{T}-tr\left(\sum_{px}\right)I_3 \end{bmatrix}$$

This formula is used to calculate Symmetric Matrix, Δ is $[A_{23}\ A_{31}\ A_{12}]^T$ and Aij is $(\sum px-\sum px^T)_{ij}$, Symmetric Matrix can also be transferred into formula like follows.

$$\mathbf{Q}=\begin{bmatrix} \mathbf{tr(C)} & \mathbf{C_{12}-C_{21}} & \mathbf{C_{20}-C_{02}} & \mathbf{C_{01}-C_{10}} \\ \mathbf{C_{12}-C_{21}} & \mathbf{2C_{00}-tr(C)} & \mathbf{C_{01}+C_{10}} & \mathbf{C_{02}+C_{20}} \\ \mathbf{C_{20}-C_{02}} & \mathbf{C_{01}+C_{10}} & \mathbf{2C_{11}-tr(C)} & \mathbf{C_{12}+C_{21}} \\ \mathbf{C_{01}-C_{10}} & \mathbf{C_{02}+C_{20}} & \mathbf{C_{12}+C_{21}} & \mathbf{2C_{22}-tr(C)} \end{bmatrix}$$

When we get Symmetric Matrix, we should calculate eigenvalue of Symmetric Matrix: λ and Eigenvector: V [q0,q1,q2,q3], and select the eigenvector with the largest eigenvalue as the rotation vector.

$$R=\begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0+q_2) \\ 2(q_1q_2+q_0q_3) & q_0^2+q_2^2-q_1^2-q_3^2 & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & q_0^2+q_3^2-q_1^2-q_2^2 \end{bmatrix}$$

The above formulas the rotation matrix generated based on paired points, and the optimized translation vector is as follows:

$$\overrightarrow{q_\Gamma}=\overrightarrow{\mu_x}-R\left(\vec{q}R\right)\overrightarrow{\mu_p}$$

Through the last formula, we can transfer points of different coordinate systems into one coordinate system, and get the optimal solution.

Step 4: Transfer model's coordinates base on matrix.

When we get the geometry transfer matrix, we can transfer protein residues' coordinates based on superimposed ligand coordinates. The new coordinates are the result of ligand-superimpose.

Step 5: Calculate RMSD changes of new coordinates and old coordinates.

We major the similarity of old coordinates and new coordinates by calculating RMSD. The formula are as follows:

$$f(\vec{q}) = \frac{1}{N_p} \sum_{i=1}^{N_p} \left\| \overrightarrow{x_i} - R(\vec{q}R) \overrightarrow{p_i} - \vec{q}\Gamma \right\|^2$$