A COMPUTATIONAL FRAMEWORK FOR INFERRING STRUCTURE, FUNCTION, AND EVOLUTION OF PROTEINS

A Dissertation in
Computer Science and Engineering

by

Yoojin Hong

© 2010 Yoojin Hong

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

December 2010
The dissertation of Yoojin Hong was reviewed and approved* by the following:

Webb Miller
Professor of Computer Science and Engineering and Biology
Chair of Committee
Dissertation Co-Advisor

Randen L. Patterson
Assistant Professor of Biology
Dissertation Co-Advisor

Padma Raghavan
Professor of Computer Science and Engineering

Jayanth Banavar
Professor of Physics

Damian B. van Rossum
Research Associate Professor of Biology
Dissertation Co-Advisor

Raj Acharya
Professor of Computer Science and Engineering
Head of the Department of Computer Science and Engineering

*Signatures are on file in the Graduate School
ABSTRACT

There are an overwhelming number of protein sequences without structure or function annotation in the post-genomic era. To facilitate protein annotation, we created a new computational framework, called Adaptive BLASTing, to infer the structure, function, and evolution of proteins from their amino acid sequences. The aims of this framework are to (i) detect particular characteristics of proteins that one is interested from their amino acid sequences, (ii) identify distant relationships among proteins in the twilight zone of sequence similarity (i.e., <25% pairwise sequence identity), and (iii) present the measured information in various formats to facilitate the analysis of proteins from different perspectives. In Adaptive BLASTing, position-specific scoring matrix (PSSM) libraries are generated for specific protein characteristics. By using low-identity alignments to compare query sequences with the PSSM libraries, the sequences are represented by vectors of the scores for each PSSM library. Then, distant relationships among proteins are identified based on the correlation (distance) between their vectors. In addition, we developed methods to detect conserved signals that represent a particular characteristic, which can be used to identify the characteristic in proteins. We tested Adaptive BLASTing on three different protein analyses, such as evolutionary analysis of highly divergent proteins, protein fold recognition, and characterization of protein structures. In each of these studies, our method outperforms other benchmarking methods. Taken together, we suggest that Adaptive BLASTing is a promising method for protein annotation. For example, by using multiple PSSMs to represent a single protein family, our method can detect relationships among proteins in highly divergent families more sensitively than traditional profile-based sequence analysis methods. Adaptive BLASTing also enables evolutionary studies to be performed without multiple sequence alignment (MSA) algorithms. The performance of MSA algorithms drops with low sequence similarities and large numbers of protein sequences. As a result, Adaptive
BLASTing allows evolutionary relationships to be identified in large sets of highly divergent protein sequences. Finally, this method is applicable to study proteins for any characteristics of proteins, because a PSSM library can be built for any characteristics in theory. Although no existing sequence analysis method is perfect, Adaptive BLASTing can be a promising method to quickly and accurately annotate protein sequences.
# TABLE OF CONTENTS

**LIST OF FIGURES AND TABLE**................................................................. vii  
**GLOSSARY**............................................................................................... ix  
**ACKNOWLEDGEMENTS** .......................................................................... xi  

**Chapter 1** Introduction ............................................................................. 1  
1.1 Inference of Protein Information ................................................................ 1  
1.2 Protein Sequence Analysis Methods .......................................................... 2  
1.3 Objective and Approach ........................................................................... 6  
1.4 Dissertation Overview ............................................................................. 8  

**Chapter 2** The *Adaptive BLASTing* Framework .......................................... 9  
2.1 Background .............................................................................................. 9  
2.1.1 BLAST ............................................................................................. 9  
2.1.2 Position-Specific Scoring Matrix ......................................................... 10  
2.1.3 Seeded Alignment ............................................................................ 11  
2.2 Motivation and Idea ................................................................................ 12  
2.2 The Framework Overview ....................................................................... 13  
2.2.1 Template PSSM Library Construction .................................................. 14  
2.2.2 Low Identity Alignments ..................................................................... 15  
2.2.3 Vector Representation of Protein Sequences ....................................... 17  
2.2.4 Similarity/Distance Measure .............................................................. 19  
2.3 Discussion ............................................................................................... 20  

**Chapter 3** Fast and Efficient Seeded Alignment Algorithm............................. 22  
3.1 Motivation ............................................................................................... 22  
3.2 Definitions ............................................................................................... 22  
3.3 Observations and Solution Overview ....................................................... 24  
3.4 Algorithm ............................................................................................... 27  
3.5 Performance Evaluation ......................................................................... 31  
3.5.1 Experiment Setup and Datasets ......................................................... 31  
3.5.2 Execution Time and Accuracy ........................................................... 32  
3.6 Summary ............................................................................................... 33  

**Chapter 4** Fold Recognition of Proteins with Twilight Zone of Sequence Similarity ......... 35  
4.1 Background ............................................................................................ 35  
4.1.1 Protein Structure ............................................................................. 35  
4.1.2 SCOP Database ............................................................................... 37  
4.1.3 Fold Recognition ............................................................................. 37  
4.2 Motivation ............................................................................................... 38  
4.3 Method ................................................................................................. 39
4.4 Results.........................................................................................................................41
  4.4.1 Alignment Comparison and Information Content.............................................42
  4.4.2 Performance Evaluation....................................................................................44
  4.4.3 Application for Fold Classification.....................................................................47
  4.5 Discussion..............................................................................................................49

Chapter 5. MSA-Free Evolutionary Study with Highly Divergent Retroelements ..........51
  5.1 Background ............................................................................................................51
    5.1.1 Multiple Sequence Alignment........................................................................51
    5.1.2 Inference of Evolutionary History of Proteins.............................................53
    5.2 Motivation ..........................................................................................................54
    5.3 Method and Results .............................................................................................55
      5.3.1 Dataset ..........................................................................................................55
      5.3.2 Statistics on Sequence Similarity...................................................................56
      5.3.3 Method ...........................................................................................................57
      5.3.2 Result .............................................................................................................60
    5.4 Discussion ..........................................................................................................61

Chapter 6. Structural and Functional Characterization of Proteins..................................63
  6.1 Background ..........................................................................................................63
    6.1.1 Domain Detection Algorithms .......................................................................63
    6.1.2 Protein Secondary Structure Prediction.......................................................64
    6.1.3 Prediction of Critical Residues of Proteins....................................................65
    6.2 Motivation ..........................................................................................................66
    6.3 Method .................................................................................................................67
    6.4 Study of Membrane-spanning Proteins..............................................................71
      6.4.1 Functional Classification..............................................................................71
      6.4.2 Structure Characterization............................................................................73
    6.5 Study of Ankyrin-repeat Proteins.......................................................................75
    6.6 Study of Transient Receptor Potential Channels...............................................77
    6.7 Discussion ..........................................................................................................80

Chapter 7. Conclusion and Future Works ........................................................................82

References ....................................................................................................................86
LIST OF FIGURES AND TABLE

Figure 1.1: Pairwise alignment details ................................................................. 3
Figure 1.2: The examples of sequence alignments ................................................ 5
Figure 2.1: The sketch of seed alignment algorithm ............................................. 12
Figure 2.2: The illustration of the Adaptive BLASTing framework ..................... 14
Figure 3.1: Adaptive GDDA-BLAST alignment details regarding seed embedding ... 24
Figure 3.2: Four basic steps of Adaptive GDDA-BLAST ..................................... 27
Figure 3.2: Performance Comparison of GDDA-BLAST and Adaptive GDDA-BLAST .... 33
Figure 4.1: Different level of protein structure .................................................... 36
Figure 4.2: Computational pipeline for fold recognition ....................................... 40
Figure 4.3: Characterization of structural similarity scores ................................. 43
Figure 4.4: Fold recognition performance of structural sequence profiles ............. 46
Figure 4.5: Comparative statistics to show independence between predictions of different methods ................................................................. 47
Figure 4.6: 3-D structure prediction of CASP9 targets ....................................... 48
Figure 5.1: Phylogeny of 88 RT sequences with statistical supports ..................... 59
Figure 6.1: Prediction of Protein Structure .......................................................... 65
Figure 6.2: The example of further analysis on positional conservation graph ....... 69
Figure 6.3: Graphical representation outputs from Adaptive BLASTing ............... 70
Figure 6.4: Functional classification of 74 Transmembrane Proteins .................... 72
Figure 6.5: The result of searching 1F88 against NCBI CDD by rps-BLAST .......... 73
Figure 6.6: Adaptive BLASTing analysis of 1F88 using ILB PSSM library ............ 74
Figure 6.7: Adaptive BLASTing analysis of 1N11 using Ankyrin-repeat PSSM library .... 76
Figure 6.8: Domain prediction by rps-BLAST and Adaptive BLASTing. ....................... 78
Figure 6.9: Structure modeling of TRPC3 Ankyrin-repeat. ........................................ 80
Table 3.1: Residue of a chimera sequence. ................................................................. 23
GLOSSARY

Protein: a molecule composed of a linear sequence of amino acids which fold into a unique 3-D structure that allow it to perform various biological functions in living cells.

Amino acid: a chemical building block of proteins. There are 20 different amino acids. Amino acids are linked with each other to form amino acid sequences of proteins.

Protein domain: a basic protein unit, which is a part of protein sequence, that can independently function, evolve, or maintains its structure (i.e., functional, evolutionary, or structural domain)

Homology: any similar characteristic of organisms that suggests they share a common origin.

Secondary structures: standard local structures of parts of protein sequence. Alpha-helix and beta-sheet are the two main secondary structure elements of proteins.

Fold: a unique combination of secondary structure elements. By the definition of SCOP database, proteins have a common fold if they have same major secondary structures in the same arrangement with the same topological connections.

N-terminus: the start of a protein sequence. C-terminus refers to the end of a protein sequence.

Twilight zone of sequence similarity: less than 25-30% pairwise sequence identity

Pairwise alignment: an alignment of two biological sequences. Pairwise sequence identity is the percentage of identically aligned residues in a pairwise alignment.

Multiple Sequence Alignment (MSA): an alignment of three or more biological sequences. Conserved regions in MSA are associated with structurally, functionally, or evolutionarily related pairs of aligned sequences.
Position Specific Scoring Matrix (PSSM): a statistical representation of patterns of homologous protein sequences. In general, the substitution scores of each 20 amino acids at each position are encoded in the \((20 \times \text{sequence length})\) dimension of a PSSM.

Phylogenetic tree: a tree-like representation of the inferred evolutionary relationship among species or other entities (e.g., protein sequences) based on similarities or distances on their characteristics (e.g., the number of different amino acids of two protein sequences to be compared).

Monophyly: the making of a monophyletic group having all descendents from a common ancestor.

Transmembrane proteins: proteins that have, for the most part, highly hydrophobic helices which transverse lipid-bilayers of cell membrane. They play important roles in the functioning of cells, such as detecting/propagating the signals from outside the cell or transport ions or small molecules through membrane.

Ankyrin-repeat: a structural motif in proteins that comprise two alpha-helices separated by loops. This has \(~30-34\) amino acid residues and mediate protein-protein interactions, some of which associate with a number of human diseases.
ACKNOWLEDGEMENTS

Dr. Randen L. Patterson, Dr. Damian B. van Rossum, Dr. Webb Miller, Dr. Padma Raghavan, Dr. Raj Acharya, Dr. Jayanth Banavar, Dr. Gaurav Bhardwaj, Dr. Kyung Dae Ko, Dr. Sree V Chintapalli, Gue Su Chang, Morgen Patterson, Foram Dave.

Seungman Hong, Heekyung Kim, Myungene Hong, Kyungene Hong, Jeongdu Moon, Gwangsook Bae, Yongwoong Moon, Yonghan Moon, Dr. Yongma Moon, Max Terang Moon.
Chapter 1

INTRODUCTION

Proteins are linear chains of amino acids which adopt unique structures that allow them to perform various biological functions [Baker 2000]. The similar function and/or structure of proteins may indicate that they have a common evolutionary ancestor. Thus, the structure, function, and evolution of proteins are interrelated. These characteristics (i.e. protein information) are specified in their amino acid sequences. Many computational methods have been developed to interpret such information, and consequently characterize proteins based on their amino acid sequences. Yet, in the post-genomic era, there are an overwhelming number of protein sequences whose function or structure has not been annotated.

1.1 INFERENCE OF PROTEIN INFORMATION

As a result of numerous genome sequencing projects, enormous number of protein sequences are deposited in sequence databases without any function or structure annotation. Consequently, one has started to adopt computational sequence analysis methods for quick annotation of proteins.

In biology, homology refers to the similar characteristics of organisms inherited from their common ancestors. In the protein world, homologous proteins are proteins that have descended and evolved from a common ancestor. With respect to their amino acid sequences, their sequences diverged from the same sequence over the course of evolution. The function and structure of homologous proteins often remains the same or differs only slightly. Therefore, in many computational methods, sequence similarity of an unknown protein to a known protein has been used as a key indicator for proteins sharing the same or similar function and structure (i.e.,
homology-based inference). However, challenge arises when related proteins do not have a significant sequence similarity. A small number of conserved residues (as low as 8% identity) can coordinate the function or structure of proteins. When sequence identity is below 25-30% (i.e., the twilight zone) [Rost 1999], computational methods often fail to infer the relationship among proteins.

1.2 PROTEIN SEQUENCE ANALYSIS METHODS

Protein sequences are a sequence of 20 amino acids (a string of characters), which vary in length, but are usually 100-1,000 amino acids long [Lipman 2002]. Different compositions of amino acids, with varying lengths, encode different protein structure and/or function. It is much harder to determine the structure and function of proteins than obtaining their amino acid sequence. In fact, the gap between available protein sequences and sequences whose function or structure is annotated is increasing in public sequence databases [Edwards 2009]. Computational sequence analysis methods help us to predict the structure and/or function of proteins by analyzing their amino acid sequences, which will help decrease the aforementioned gap.

The simplest way of obtaining protein information from protein sequences is to directly compare the sequences of two proteins (unknown protein as a query sequence vs. known protein as a target sequence). If the two proteins are homologous sequences, they diverged because mutations were introduced. Some amino acids are replaced by different amino acids (substitutions), while some amino acids are deleted or newly inserted (indels), while the rest remain the same. Amino acid substitutions and indels are represented as mismatches and gaps in an alignment. Using the scores of amino acid matches/mismatches (substitution scores) and gap penalty, alignment algorithms aim at aligning protein sequences in an optimal way.
Needleman-Wunsch, Smith-Waterman, and BLAST (Basic Local Alignment Search Tool) [Smith 1981, Needleman 1970, Altschul 1990] are the most popular alignment algorithms. Needleman-Wunsch and Smith-Waterman are dynamic programming algorithms used to find the optimal global and local alignments. These algorithms are guaranteed to find the optimal alignments, but their computational cost is expensive. BLAST finds a near optimal local alignment using a heuristic approach which makes it practical for quickly searching huge sequence databases.

Various substitution scoring matrices, such as PAM and BLOSUM families [Dayhoff 1978, Henikoff 1992], have been introduced. For each pair of amino acids, a substitution score is assigned based on the observed frequency of the occurrence of the pairs in the related proteins. Figure 1 shows an example of alignment from BLAST and BLOSUM62 scoring matrix.

![Alignment Details](image)

**Figure 1.1**: Pairwise alignment details. (a) BLAST Alignment example (b) BLOSUM62 scoring matrix
While BLAST has remained one of the most used sequence database search tools due to its speed, approaches such as Position Specific Scoring Matrix (PSSM) and Hidden Markov Model (HMM) have been introduced for better sensitivity to weak sequence similarity (e.g., PSI-BLAST, FFAS, SAM-T2K, HHsearch [Altshul 1997, Jaroszewski 2005, Karplus 1998, Soding 2005]). Protein divergence arises differently in different protein families. In other words, amino acid substitutions and indels are introduced differently in the sequences of proteins in different protein families. Thus, the sequences of a highly divergent protein family often fail to be found related by current computational methods. The aforementioned methods, which are often called profile-based methods, involve the use of statistical representation of protein sequences in a protein family (i.e., a profile). Profile-based methods use information from multiple sequences rather than a single sequence. Park et al. reported that sequence analysis algorithms that use information from multiple sequences show a three-fold higher performance in detecting remote homology among the proteins with less than 25-30% sequence identity [Park 1998].

PSSM-based algorithms (e.g., PSI-BLAST, FFAS) generate a position-specific substitution matrix using a set of protein sequences from a particular protein family such that the scoring matrix is specific to that protein family. A set of homologous sequences are aligned to each other by multiple sequence alignment algorithms (More details are described in chapter 5). These algorithms can align divergent protein sequences using scoring matrices such as PSSM despite a relatively low sequence similarity. An example of multiple sequence alignment is shown in Figure 1.2. The conserved regions in a multiple sequence alignment are associated with functionally and/or structurally important region of the homologous proteins. The position-specific substitution score of each amino acid is calculated from the frequency of an amino acid in a specific position of a multiple sequence alignment (chapter 2 for more details). Using the position-specific scoring matrix, alignments are generated for remotely related proteins by weighting conserved positions in their protein family greater than non-conserved positions.
HMM-based algorithms (e.g., SAM-T2K, HHsearch) generate an HMM model whose parameters are trained using sequences from a protein family. Using the model, the probability of the occurrence of the query sequence is calculated, with a high probability indicating that the query sequence is related to the HMM model represents. In general, HMM-based algorithms outperform PSSM-based algorithms since HMM-based algorithms incorporate position-specific gap penalties [Krogh 1994, Soding 2005]. For this reason, PSSMs work well for shorter sequences. However, PSSM-based algorithms have advantages. It is simpler to build PSSMs than HMMs. In addition, traditional alignment algorithms based on a dynamic programming approach can be used to align sequences using PSSMs. Both PSSM- and HMM-based algorithms require multiple sequence alignments of homologous sequences whose quality is easily degraded when in the twilight zone of sequence similarity and when there are large numbers of protein sequences to deal with. This indicates that the performance of PSSM- and HMM-based algorithms can be limited by the performance of multiple sequence alignment algorithms.

Figure 1.2: The examples of sequence alignments. (a) pairwise alignment (b) multiple sequence alignment. Part of a multiple sequence alignments of nine helix-loop-helix proteins as constructed by DIALIGN. Conserved regions (blue boxes) are associated with two parts of functional domain of the proteins ((i)DNA-binding basic region and first alpha-helix and (ii)second alpha-helix) [Morgenstern 1997]
1.4 OBJECTIVE AND APPROACH

Despite many efforts, there is no established foolproof sequence analysis method that can completely annotate unknown proteins. In this dissertation, we are proposing a new computational framework using related algorithms to infer the function, structure, and evolution of proteins based on their amino acid sequences. This framework is suggested in order to (i) measure information within amino acid sequences for particular protein characteristics that a user defines, (ii) reveal the relationship among proteins in the twilight zone of sequence similarity, and (iii) provide the measured information in various output formats to study proteins on various levels.

In the case of proteins in the twilight zone of sequence similarity, the direct comparison of their sequences provides very limited information on the relationship among the proteins. In our framework, remote relationship among proteins is inferred on the basis of their relationship to other proteins, under the assumption that if two proteins are related (aligned) to the same or a similar set of proteins, they are likely to be related to each other. We also take into account low-identity alignments, which were traditionally considered an insignificant source of information, in order to utilize all possible sources of information. In fact, we propose a new sequence alignment algorithm to generate low-identity alignments between highly divergent protein sequences. Using this algorithm, we calculate the score of a query protein relative to each of the different targeted proteins on the basis of the alignment. These scores are encoded into a vector to represent the query protein. The relationship of the proteins is measured by the correlation between their vectors.

We further develop this framework to reveal the remote relationship among proteins on the basis of their relationship with different protein families, not individual proteins. For this purpose, we propose an algorithm to measure a protein against each protein family. In our method,
instead of using a single representative PSSM for a whole protein family, we maintain the PSSMs of all sequences in the protein family (PSSM library). This method is advantageous when we define an extremely divergent protein family, because, in such cases, we can obtain multiple alignments for each protein. If a query protein is related to a certain protein family, the multiple alignments to that protein family should contain a common signal which is specific to that protein family. This common signal is found when the alignments are overlapped onto the query sequence; with the overlapped information indicating the protein family–specific signals found in the query protein. We also developed the scoring schemes to measure such information. Once the protein is scored against each protein family, the protein is once again encoded in a vector for further analysis.

Once a protein family is defined on the basis of protein characteristic of interest and its library is built, the protein can be studied at a different level of resolution in our framework - for example, if a protein has a certain structure, which region of the protein contributes to that structure, or which residues of the protein are critical to maintaining that structure can be isolated. To summarize, the proposed framework, termed as Adaptive BLASTing, has the following features:

- It allows users to define any protein function or structure of interest in such a way that it can isolate and measure the signals specific to that function or structure from protein sequences.
- It measures, analyzes, and visualizes the function- or structure-specific signals, which provides answers to various protein related questions.
- It reveals the remote relationship among proteins based on their relationship with different proteins or protein families.
- It makes use of low-identity alignments, which are traditionally considered insignificant.
It encodes protein sequences in feature vectors, which allows for the application of traditional data mining and machine learning algorithms to protein sequence analysis.

The objective of this dissertation is to propose a framework for protein sequence analysis, to develop appropriate algorithms that work within the framework, to describe its usefulness, and to provide promising results in the study of various biological problems.

1.4 DISSERTATION OVERVIEW

The remainder of this dissertation is organized as follows. Chapter 2 discusses the overview of the Adaptive BLASTing framework and the ideas behind it. Chapter 3 introduces a new alignment algorithm that is incorporated into the framework. The algorithm adaptively modifies protein sequences such that highly divergent sequences can be aligned. In Chapter 4, Adaptive BLASTing is applied for fold recognition of proteins in the twilight zone of sequence similarity. Chapter 5 describes the evolutionary study of highly divergent proteins using our framework without using multiple sequence alignment (MSA) algorithms. Chapter 6 introduces the promising results from Adaptive BLASTing for predicting functional and structural elements in proteins. Finally, the conclusion and future directions are discussed in Chapter 7.
The question of how to obtain protein information from protein amino acid sequences still isn’t fully solved. It is especially challenging when proteins with functional, structural, and/or evolutionary relationship are in the twilight zone of sequence similarity [Schaffer 1999, Jaroszewski 2002, Altschul 1997, Yona 2002]. We are proposing a new computational framework for protein sequence analysis, termed as Adaptive BLASTing, to address this. In this chapter, we describe the ideas and the overview of the framework with related algorithms and scoring schemes. Its utilities and performance are shown separately in chapters 4, 5, and 6, respectively, with being applied to various biological problems.

2.1 BACKGROUND

2.1.1 BLAST

BLAST (Basic Local Alignment Search Tool) is a heuristic local alignment algorithm which was proposed for quickly searching large sequence database [Altschul 1990]. Given a query sequence, BLAST constructs a list of words (e.g., 3 amino acids) that align to query words with a score of at least $T$ and search database sequences which have a word on the list. Given such a word pair in the query and the database sequences (i.e., hit), BLAST extends the hit in both directions to generate a high-scoring alignment. To save computation, the extension is stopped when the alignment score falls below a certain threshold, as compared to the best alignment score that is obtained during the extension. Since the extension step is the most time-consuming, the new
‘two-hit’ approach starts the extension only when there are two non-overlapping word pairs within a distance $A$ to each other. Due to its heuristic approach, BLAST is fast enough to search a large sequence database. For each alignment whose score is $S$, BLAST reports an e-value which is the expected number of alignments with score at least $S$ that are expected to be obtained in a database search by chance. The e-value is calculated as follows: $E = K m n e^{-\lambda S}$, where $m$, $n$, and $S$ are the size of the sequence database, a query sequence length, and an alignment score, respectively, and $K$ and $\lambda$ are scaling factors for the scoring matrix used. BLAST returns alignments whose e-value is less than a threshold (e.g., default e-value threshold 0.01). While BLAST still remains one of the most frequently used algorithm to search sequence databases, its sensitivity is not satisfactory in the twilight zone of sequence similarity.

### 2.1.2 POSITION SPECIFIC SCORING MATRIX

Positional Specific Scoring Matrix (PSSM) is a statistical representation of patterns of protein sequences – also called a profile, Position Weight Matrix (PWM), or Position-specific Weight Matrix (PSWM). It is believed that sequence database searches which use the pattern shown in multiple homologous proteins (using the PSSM which represent the pattern) is a much more effective for detecting weak relationship among proteins than using a single sequence as the query [Altshul 1997, Park 1998]. To build a PSSM, homologous proteins are first aligned to each other and the observed frequency of each of 20 amino acids at each position is measured. Given the frequencies, the substitution scores of each 20 amino acids at each position are calculated and encoded in the $(20 \times \text{query length})$ dimension of a PSSM. As position-specific substitution scores, the following log-odds score is popularly used: $s_{i,j} = \log \left( \frac{p_{i,j}}{b_j} \right)$ where $s_{i,j}$, $p_{i,j}$, and $b_j$ are the substitution score of the amino acid $j$ at the position $i$, the probability of
observing amino acid $j$ at the position $i$ and the background probability of observing an amino acid $j$, respectively.

PSI-BLAST (Position-Specific Iterated BLAST) improves the sensitivity of BLAST for distant relationship by using PSSMs [Altschul 1997]. PSI-BLAST first searches a query sequence against a sequence database. A PSSM is generated from a multiple sequence alignment of the database sequences which are returned with e-values that are smaller than a threshold. Then, the PSSM is used in the place of the query sequence for the sequence database search. By repeating these steps, PSI-BLAST detects weak relationship among proteins. While PSI-BLAST searches a PSSM against a sequence database, rps-BLAST (Reverse PSI-BLAST) searches a sequence against a database of pre-calculated PSSMs [Schaffer 1999]. In addition to these sequence–PSSM comparison approaches, various PSSM-PSSM comparison approaches, such as prof_sim, or FFAS [Yona 2002, Jaroszewski 2005], have been introduced and shown that they outperform sequence-PSSM approaches.

### 2.1.3 SEENED ALIGNMENT

A protein domain is a basic unit of a protein that can independently function, evolve, or maintains its structure. Such domains are conserved in protein sequences. To detect highly divergent domains in proteins, a simple sequence alignment algorithm, called GDDA-BLAST (Gestalt Domain Detection Algorithm-BLAST), was introduced by Patterson et al. [Patterson 2005]. Since such domains are only identified as low-identity alignments, conventional domain detection algorithms (e.g., CDD, Pfam, SMART) often fail to detect them. To overcome this problem, GDDA-BLAST generates seeded alignments using modified query sequences (Figure 2.1(a)). GDDA-BLAST embeds the N- or C-terminal of a target domain sequence, termed as *seed*, at every position of a query sequence. Since the modified query sequences are guaranteed to have a
hit (exact matches up to the length of the embedded seed) to the target domain sequence, a traditional sequence alignment algorithm, such as BLAST, align the rest of the query sequence to the domain sequence. Despite the artificial hit in the modified query sequences, both query and domain sequences must share similar regions to generate a significant alignment initiated by the hit. If significant alignments exist with consecutive modified query sequences, it is believed that those alignments with low-identity are not random and the query region covered in the alignments may correspond to highly divergent domain in the query protein (Figure 2.1(b)).

2.2 MOTIVATION AND IDEA

Proteins may not be completely independent when comparing their characteristics. For example, protein A can be structurally more similar to protein B than to another protein C even though three of them have no structural homology. Therefore, our assumption is that remotely related proteins are likely to share the similarities with the characteristics of different protein families than completely independent proteins are. In other words, the relationship among proteins is

![Figure 2.1: The sketch of seed alignment algorithm. (a) Seeded alignment step (b) Domain detection step](image-url)
indicated on the basis of their relationship to different proteins or different protein families which share function, structural, or evolutionary characteristics. To implement the idea in our framework, we are proposing the methods to measure the information of different protein characteristics using protein sequences. Based on this information, we measure the relationships among proteins.

### 2.3 THE FRAMEWORK OVERVIEW

The basic idea that is described above is implemented as follows. We first define protein characteristic to measure. A set of PSSMs (i.e., template library) is then prepared using the reference sequences from the protein family that has such characteristics. To measure the relationship of a query protein to the protein family, we align the query protein sequence with each of the PSSMs in the library. We then calculate scores for each PSSM or to each template library using the alignments (i.e., template-specific score). The relationships of a query protein to different proteins or different protein families are represented as a vector of template-specific scores. Functional, structural, or evolutionary relationships among proteins are determined by the correlation of their vectors. Further, the quantitative information that measures a specific protein characteristic of a query sequence is given in various output formats. Analyzing the outputs helps us to study proteins especially for a specific protein function or structure of interest. We term our framework as *Adaptive BLASTing* due to its adaptability to different purposes of studying proteins (e.g., user-defined template libraries, user-defined way of running BLAST or BLAST-like seeded alignment algorithm, and various outputs) and the use of BLAST algorithm as the foundation of the framework. The overall framework is shown in Figure 2.2. More details are described in the following sections.
2.3.1 TEMPLATE PSSM LIBRARY CONSTRUCTION

A template PSSM library is essential to measure information from protein sequences. It determines information content to be measured – what kind of functional, structural, or evolutionary characteristics are measured from amino acid sequences. Here we describe how to build a template library as an information source.

To build a template library for a specific functional or structural characteristic of interest, we first collect reference sequences that are already annotated to have that characteristic. For reference sequence collection, we use publicly available knowledge base sources, such as NCBI CDD (Conserved Domain Database), Pfam, SMART, PDB (Protein Data Bank), SCOP (Structural Classification of Proteins), CATH, and many other protein databases. [Marchler-
Bauer 2005, Bateman 2004, Letunic 2006, Berman 2000, Murzin 1995, Orengo 1997]. The references sequences are then expanded by simple sequence database search (e.g. BLAST, PSI-BLAST). Redundant or highly similar sequences are removed to avoid measuring redundant information. Algorithm 2.1 describes how reference sequences are expanded and PSSMs are generated to construct a template PSSM library.

**Algorithm 2.1:** PSSM library construction

**Input:** a set of protein sequences $X$

**Output:** a set of PSSMs $L$

1. // Step 1; expand sequences
   for each sequence $x$ ($\in X$) do
   $G \leftarrow$ sequences returned after PSI-BLAST searching of NR database
   for each sequence $\gamma$ ($\in G$) do
   $sim(x, \gamma)$ $\leftarrow$ sequence identity of $x$ and $\gamma$
   if $sim(x, \gamma) < \phi$, $X' \leftarrow \gamma$

2. // Step 2; remove redundancy
   for each sequence $x_c$ ($\in X'$) do
   for each sequence $x_v$ ($\in X'$) do
   $sim(x_c, x_v)$ $\leftarrow$ sequence identity of $x_c$ and $x_v$
   if $sim(x_c, x_v) \geq \sigma$, then remove $x_v$ from $X'$

3. // Step 3; PSSM generation
   for each sequence $x$ ($\in X'$) do
   $L \leftarrow$ PSSM generated by PSI-BLAST using NR database
   return $L$

**2.3.2 LOW IDENTITY ALIGNMENTS**

Remote relationship among proteins is implied from sequence alignments with the twilight zone of sequence similarity [Rost 1999, Yona 2002, Blake 2001]. By taking account of low-identity alignment as well, we use as much information as possible from protein sequences for inferring relationship among proteins. For collecting informative signals, we use either of seeded alignments or BLAST alignments.
As previously described, the seeded alignment algorithm was proposed to generate low-identity alignments between highly divergent sequences. In the next chapter, we propose a new seeded alignment algorithm which is faster and more efficient. Seeded alignments are generated for each pair of a query and a template PSSM. The seeded alignments are filtered out by predetermined thresholds to remove noisy/random alignments. As the thresholds, coverage (i.e. alignment length as a function of the length of template PSSM) and pairwise identity are used. For a pair of a query and a template PSSM, multiple seeded alignments (up to query sequence length $\times$ 2) can be generated since modified query sequences are made by embedding N- and C-terminal of a master sequence of a template PSSM to every position of the query sequence. Seeded alignments that satisfy the thresholds to determine alignment significance are called as a hit. If a template PSSM has at least one significant seeded alignment to a query, the template PSSM is positive.

BLAST reports e-value with alignments. E-value indicates how high the alignment score is likely to rise by chance. Alignments with lower e-value are thus considered statistically more significant. Low-identity alignments typically have low alignment scores so that their e-value is often large. Low-identity alignments are often considered insignificant because of their high e-value. To distinguish random and informative low-identity alignments from BLAST, we again use coverage threshold. Commonly accepted e-value threshold of BLAST is $\sim 10^{-2}$. We, however, utilize alignment with e-value up to $10^{10}$ using 60-80% coverage threshold. Our results, which are shown in the following chapters, indicate that low-identity alignments are a rich source of information that can be used to unmask the fundamental properties of proteins including protein structure, function, and evolution. In addition, coverage threshold is validated as an effective filter to eliminate noisy low-identity alignments.
2.3.3 VECTOR REPRESENTATION OF PROTEIN SEQUENCES

Protein sequence is represented in a vector of scores for each template PSSM or each template library in our framework. A template-specific score reflects how much a query sequence has the information related to the protein characteristic that a template represents. We are proposing scoring schemes to calculate the template-specific score using the alignments between a query protein sequence and the template.

**Template PSSM-specific score based on alignment characteristics:** First, we calculate a template-specific score simply on the basis of the characteristics of the alignment(s) between a query and a template PSSM, such as pairwise identity ($I$), coverage ($C$), or hit ratio in case of seeded alignments ($H$). Let $A_{q,k}$ be an alignment (or a set of alignments when using seeded alignments) between a query protein $q$ and a template PSSM $k$. The characteristic of an alignment $A_{q,k}$ are scored as follows:

\[
I = \frac{s}{|A_{q,k}|}, \quad C = \frac{a_s - a_e + 1}{l}, \quad H = \frac{h_{q,k}}{2n}
\]

where $s$, $|A_{q,k}|$, $a_e$, $a_s$, $n$, $l$, $h_{q,k}$ are the number of exact matches of $A_{q,k}$, the length of $A_{q,k}$, start position of $A_{q,k}$ in $q$, end position of $A_{q,k}$ in $q$, the length of $q$, the length of $k$, and the number of hits of $q$ to $k$, respectively. For a template PSSM-specific score, we define the two simple scores which are *composite score* (i.e., $avg.I \times avg.C \times H$) when using seeded alignments and *product score* (i.e., $I \times C$) when using BLAST alignments.

**Template library-specific score based on positional scoring of a query sequence:** Not all amino acids in protein sequence are critical for a particular protein function or structure [Capra 2007]. Our goal is to measure function- or structure-specific information only. Template PSSMs in a library are all generated from related proteins that have the functional/structural characteristic of interest. Therefore, we expect that the alignments between a query sequence and the PSSMs to
contain the information in common to the characteristic. Given the alignments, we score each
query residue and then calculate a template library-specific score for the query. In this calculation,
we only use one alignment to each PSSM. Thus, when using seeded alignments, we re-align the
region of the query sequence, which the seeded alignments cover, with the PSSM by Smith-
Waterman algorithm [Smith 1981].

Let $A(Q, T)$ be the set of alignments between a query protein $Q$ and a template library $T$,
$n$ be the length of $Q$, and $m$ be the number of PSSMs in $T$. Each alignment $a(t_k) \in A(Q, T)$
between $Q$ and the PSSM $t_k$ $(1 \leq i \leq m)$ in $T$ is associated with a continuous interval $\tau(a(t_k))$ of residues
on $Q$. We define a positional score $ps_i$ for each amino acid residue $q_i$ $(1 \leq i \leq n)$ on $Q$ and a
template library $T$-specific score $ts$ as follows:

$$
ps_i = \sum_{a \in A} s(q_i, \gamma(A)) - \mu, \quad 1 \leq i \leq n
$$

$$
\sum_{i=1}^{n} ts_i
$$

where

- $\mu$ is the average of $ps$

- $ps_i' = \begin{cases} ps_i & \text{if } ps_i > 0 \\ 0 & \text{otherwise} \end{cases}$

- $\gamma(A)$ is the template residue aligned with $q_i$ in $A$

- $A = \{a \in A(Q, T) : q_i \in \tau(a(t_k)), a(t_k) \in A(Q, T)\}$ is the alignment to the PSSM $t_k$ in $T$,

- $a(t_k)$, which contains $q_i$.

$$
s(i, j) = \begin{cases} BLOSUM62(i, j) & \text{if } BLOSUM62(i, j) > 0 \\ 0 & \text{otherwise} \end{cases}
$$
The idea is that after accumulating the informative signal (positive BLOSUM scores) on each query residue, the residues with strong signal (with high positional scores) are considered critical to the functional, structural, or evolutionary characteristic that a measuring template library represents. Noisy signals could be collected while using multiple low identity alignments. To take into account only a pure population of informative signals, we use normalized positional scores that are positive when calculating a template library-specific score of a query protein; normalization is simply done by subtracting the average of positional scores of query residues.

Positional analysis, which was described above for positional scoring, is useful for predicting functionally or structurally critical residues of proteins. Typically, for such predictions, related protein sequences are aligned in a multiple sequence alignment (MSA) and residues that show high conservation among the sequences are predicted as critical residues [Capra 2007, Fischer 2008]. While MSA algorithms often fail in twilight zone of sequence similarity, our framework can be used to study critical residues in highly divergent proteins, MSA-free. The examples that our framework is applied to for such problems are shown in the chapter 6.

2.3.4 SIMILARITY/DISTANCE MEASURE

Once protein sequences are encoded in a vector of template-specific scores, any standard similarity or distance metrics can be used to infer their relationship. Among many, we use Euclidean distance and Pearson’s correlation coefficient which are the two most popular metrics. Performance of other metrics in our framework would be open to a future study.
**Evolutionary distance measure:** *Euclidean distance* is used to measure the evolutionary distance between two proteins with their vectors $X$ and $Y$.

$$ED(X, Y) = \sqrt{\sum_{i=1}^{n} (X_i - Y_i)^2}$$

**Functional/Structural similarity measure:** *Pearson’s correlation coefficient* is used to measure functional or structural similarity between two proteins with their vectors $X$ and $Y$.

$$PC(X, Y) = \frac{1}{n} \sum_{i=1}^{n} \left( \frac{X_i - \mu_X}{\delta_X} \right) \left( \frac{Y_i - \mu_Y}{\delta_Y} \right)$$

where $\mu_X$ and $\delta_X$ are the average and standard deviation of $X$, respectively.

### 2.4 DISCUSSION

We proposed the idea of *Adaptive BLASTting* framework with the related algorithms for (i) template PSSM library construction (to control information/signal source), (ii) alignment generation (to collect informative signals), (iii) scoring schemes for template-specific scores (to measure the information in collected signals), and (iv) similarity/distance measures among proteins (to reveal the relationship among proteins using the measured information). The framework combined with the algorithms allows us to measure the relationship among proteins on the basis of their relationship to different proteins or protein families. Depending on how template libraries are built (how protein families are defined), we can relate proteins differently. For example, if we build template libraries for different protein functions, two proteins with the different structures but the same function would be related with large similarity score, while the proteins with the same structures but different functions get low similarity score. If we build template libraries for different structures, the result would be the other way around. This framework can be adaptively used for different purposes of studying proteins.
We measure information that is specific to a certain protein characteristic from amino acid sequences. Proper visualization and analysis of the quantitative information should be helpful to scrutinize individual proteins and related methods are described in chapter 6 with examples of studying various types of proteins for different function and structure.
3.1 MOTIVATION

Despite of the potential of GDDA-BLAST (chapter 2), its computational cost is high. This is because the cost of GDDA-BLAST is approximately proportional to the length of query sequence used to generate the seeded alignments – i.e., one has to run BLAST against modified query sequences generated by embedding a seed at every position of the query sequence. Furthermore, this step should be repeated for all template sequences used to measure the query sequence. In practice, this is prohibitively expensive. To address this challenge, we are introducing a new seeded alignment algorithm, called Ada-BLAST (Adaptive GDDA-BLAST), which is as sensitive as the original GDDA-BLAST but orders of magnitudes faster than GDDA-BLAST. In a nutshell, Ada-BLAST exploits the similarity among modified query sequences to adaptively avoid expensive computations.

3.2 DEFINITIONS

Note that a template sequence to measure a query protein sequence is denoted as a target sequence in this chapter. Let a target sequence be $X$ and a query sequence be $Y$. The length of $X$ is denoted as $|X|$. Assume that $|X|$ and $|Y|$ are $n$ and $m$, respectively. A subsequence of $X$ from the $i^{th}$ residue to the $j^{th}$ residue is denoted by $x_{i,j}$ such that $0 \leq i \leq j \leq n-1$. A subsequence whose length is 1, such as $x_{i,i}$, is simply represented as $x_i$. Concatenation of two sequences, $X$ and $Y$, is represented as
Two subsequences that are aligned in an alignment are represented with (\(\)). For example, \((x_{a,b}, y_{c,d})\) represents that \(x_{a,b}\) and \(y_{c,d}\) are aligned.

**Table 3.1:** Residue of a chimera sequence.

<table>
<thead>
<tr>
<th>Chimera index</th>
<th>(y_{0,q-1})</th>
<th>N-terminal seed (S)</th>
<th>(y_{q,m-1})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(c_0)</td>
<td>(y_0)</td>
<td>(x_0)</td>
<td>(y_q)</td>
</tr>
<tr>
<td>(c_1)</td>
<td>(y_1)</td>
<td>(x_1)</td>
<td>(y_{q+1})</td>
</tr>
<tr>
<td>(\cdots)</td>
<td>(\cdots)</td>
<td>(\cdots)</td>
<td>(\cdots)</td>
</tr>
<tr>
<td>(c_{q-1})</td>
<td>(y_{q-1})</td>
<td>(x_{k-1})</td>
<td>(y_{m-1})</td>
</tr>
</tbody>
</table>

A modified (chimera) query sequence is generated by embedding either the N- or C-terminal of \(X\) as a “seed”, denoted by \(S\), to every position of \(Y\). Usually \(p\%\) of \(X\) (i.e., \(k\) residues of \(X\) where \(k = \lceil X \times p \times 0.01 \rceil\)) is used as a seed. Thus, N- and C-terminal seeds are \(x_{0,k-1}\) and \(x_{n-k,n-1}\), respectively. A chimera with a seed at position \(q\) of \(Y\) is \(y_{0,q-1} \& S \& y_{q,m-1}\) and represented with \(C(q)\). An example of a chimera is shown in Table 3.1. This table shows an example of chimera sequence with an N-terminal seed of length \(k\) inserted into the position \(q\) of the original query sequence \(Y\). The length of the resulting chimera sequence is \((m+k)\), where \(m\) is the length of the original query sequence. To align the target sequence \(X\) and the query sequence \(Y\), GDDABLAST generates \((m \times 2)\) number of chimera sequences embedding N- and C-terminal seeds from \(X\) at each position of \(Y\). Each chimera is then aligned to \(X\) using rps-BLAST. For each query, rps-BLAST is run independently, yielding a total of \((m \times 2 \times \#\) of target sequences) BLAST executions. Moreover, the same procedure needs to be repeated for the total number of queries.

Note that chimera sequences differ only by the position of the seed. This implies that for two subsequent chimeras, much of the computation can be reused. In the Ada-BLAST approach, we re-use the outputs from each step of rps-BLAST for efficient computation. For clarity, we define the outputs of each step as follow. In the first step of rps-BLAST, we find hits between \(X\) and \(Y\). A hit of \((x_{a,a+w-1}, y_{b,b+w-1})\), where \(w\) is a word size, is denoted as \(h(a,b)\). After ungapped
extensions on two neighboring hits in the second step, we obtain HSPs (High Scoring Sequence Pairs) extending two hits without gaps. An HSP to align \( x_{a,a+r} \) and \( y_{b,b+r} \), is denoted \( hsp(a,b,r) \). If an HSP has a score high enough to trigger gapped extension, in the third step, then an alignment is generated extending the HSP with gaps to both directions from a residue pair in the highest scored region of the HSP. The pair from which gapped extension is started is often referred to as a seed as well in BLAST [Altschul 1990]. In order to avoid confusion, it is denoted as a GE starting pair to distinguish it from the embedded seed of GDDA-BLAST.

3.3 OBSERVATIONS AND SOLUTION OVERVIEW

![Diagram showing adaptive GDDA-BLAST alignment details regarding seed embedding.](image)

Figure 3.1: Adaptive GDDA-BLAST alignment details regarding seed embedding. (a) Limited region of interest with the seed embedding position. The diagonal line represents the alignment with the seed in different locations. The examples illustrate the region of interest of the N-terminal seeds. Similarly for the C-terminal seed, it is the upper-left corner of the seed. (b) The corresponding hits of a query and a chimera sequence. This example illustrates that the hits between the target sequence (X) and the query sequence (Y) can be reused for aligning a chimera sequence (C) against the target sequence (X). (c) The seed positions selected given a partial alignment. Ranges on the top and bottom represent the seed embedding positions of N-terminal and C-terminal seeds, respectively.
**Observation 1.** *Seeding limits the search space.* Since a seed provides an exact match, it is very likely that the GE starting pair is in an HSP that includes the seed. Moreover, we are only interested in the alignments that include the seed because the other alignments can be found through the conventional methods using the original query sequence. This limits the search space of rps-BLAST. For example, when a seed is inserted at the position 0 of a query sequence, our search space will be the region in gray, as shown in Figure 3.1(a, left), and every time the seed insertion position is moved to the right, our search space is reduced, as is shown in Figure 3.1(a, right). Note that, in case of the chimeras with C-terminal seeds, the search space is limited to the upper-left corner from the start position of the seed.

**Observation 2.** *Chimeras share hits.* Because chimeras are the same sequence, except for the position of a seed, most of their hits are conserved. Therefore, we can reuse the hits between $X$ and $Y$ to compute the alignments of any chimera sequences. Consider a chimera, $C(q)$. Let $h_{c(q)}$ be a hit obtained after the first rps-BLAST step between $X$ and $C(q)$. The relation between $X$-$Y$ hits (i.e., $h$) and $X$-$C(q)$ hits (i.e., $h_{c(q)}$) can be defined as follows:

\[
h(a, b) = \begin{cases} 
  h_{c(q)}(a, b) & \text{if } 0 \leq b \leq q - w \\
  h_{c(q)}(a, b + k) & \text{if } q \leq b \leq m \\
  \text{no hit} & \text{otherwise}
\end{cases}
\]

where $k$ is the seed length, $q$ is the seed embedding position, and $w$ is the size of a hit.

Proof is omitted because it is clearly shown in Figure 3.1(b). Note that we are not interested in the hits of lemma 1(a) (Figure 3.1(b) (c,d)) because they are outside of the region of interest by observation 1. Therefore, we only use the hits of lemma 1(b) (Figure 3.1(b) (a,b)) for alignment.
Observation 3. Chimeras share HSPs after ungapped extension. Rps-BLAST performs ungapped extension on neighboring hits resulting in HSPs. Similar to observation 2, we can define the relationship between an HSP of X-Y (i.e., hsp) and that of X-C(q) (i.e., hsp_{C(q)}) as follows:

**Lemma 2.**

\[
\begin{aligned}
\text{hsp}(a, b, r) &= \begin{cases} 
\text{hsp}_{C(q)}(a, b, r) & \text{if } 0 \leq b \leq q - r \\
\text{hsp}_{C(q)}(a, b + k, r) & \text{if } q \leq b \leq m \\
\text{no HSP} & \text{otherwise}
\end{cases}
\end{aligned}
\]

where \( r \) is the length of the HSP. The proof is straightforward from lemma 1.

Observation 4. Chimeras share alignment paths in gapped extension. Gapped extension is rps-BLAST starts at a GE starting pair that is a central residue pair in the highest scoring segment of any HSP whose score is sufficiently high. Different alignments can be generated if the gapped extension is performed on different GE starting pairs and there is no guarantee that the same GE starting pair will be selected for different chimera sequences. However, as shown in Figure 3.1(c) (boxed residues represent the seed), if a portion of a target sequence is conserved in a query sequence, then it is very likely that the conserved region is aligned for multiple neighboring chimera sequences. We exploit this property to speed up the alignment process.

Observation 5. Not every chimera produces a useful alignment. Even though a seed provides artificial matches, it cannot be extended if there are insufficient neighboring HSPs to connect to. Therefore, we can significantly reduce the computational complexity of the alignment process by embedding seeds only into a limited number of query positions that are likely to be extended.
Hence, in Ada-BLAST, we align the query and the target sequence first, and we then compute the seed embedding position from the alignment result before aligning the chimera sequences.

### 3.4 ALGORITHM

Ada-BLAST works according to four basic steps, as shown in Figure 3.2. First, we find the conserved regions by generating non-overlapping local alignments between the query and the target sequence [Haung 1990]. We call these partial alignments. Second, for each partial alignment from step 1, seed embedding positions are determined. Third, we produce final alignments including the seeds. Finally, we filter out the non-significant alignments using quality measures such as the %coverage and %identity of the alignments to the corresponding PSSM.

![Figure 3.2](image)

Figure 3.2: Four basic steps of Adaptive GDDA-BLAST. (i) Step 1: Find multiple non-overlapping local alignments. (ii) Step 2: Select seed embedding positions in query sequence. (iii) Step 3: Generate final alignments with seed. (iv) Step 4: Filter out non-significant alignments using coverage and pairwise identity of the alignment.

**Step 1.** Find multiple non-overlapping local alignments. Huang et al [Haung 1990] proposed an algorithm to find the multiple non-overlapping local alignments between two sequences. We adopted this algorithm to generate partial alignments between the query and the target sequence. As a scoring matrix, any given substitution matrix (e.g., BLOSUM62, BLOSUM45, PAM30 [Dayhoff 1978, Henikoff 1992]) or PSSM of a target sequence can be used.
The local alignments are found as follows. First, the hits between the query and the target sequence are found. A hit is three consecutive residues with a score larger than a threshold (i.e., \textit{minimum word score}). For each hit, to generate an HSP, an ungapped extension is performed until the score drops below a threshold (\textit{HSP drop-off score}). When a new HSP is constructed, the hits involved in the HSP are removed to prevent subsequent HSPs from extending over them. This ensures that all local alignments produced later will be non-overlapping. We keep only the HSPs with scores larger than the threshold (\textit{minimum HSP score}). Gapped extension is then performed for each HSP to generate partial alignments.

We keep only the partial alignments whose lengths are greater than the threshold (i.e., \textit{minimum partial alignment length}). Minimum partial alignment length is proportionally determined to the length of the target sequence. Since a partial alignment is a locally best alignment, it is not likely that a seed will extend further than the end position of the partial alignment. In the last step, a final alignment will be filtered by the coverage of the alignment over a target sequence. This pre-filtering on partial alignments can remove the seed embedding positions where a seed cannot be extended enough to have sufficient coverage to become a final alignment.

\textbf{Step 2. Select seed embedding positions.} In this step, seed insertion positions in the query sequence are selected given the partial alignments obtained from step 1. As discussed in observation 1, a final alignment is generated by extending a seed from its end positions. Since a partial alignment is a locally optimal alignment, the extension of a seed can be converged with the partial alignment if a seed is inserted nearby and the score of the path is high enough. Because of the relatively high penalty of gaps used in sequence alignment methods, an alignment is usually generated with HSPs connected with small numbers of gaps in between. In addition, the score of partial alignment on either side of the gaps must be higher than the gap penalty [Rognes
The gapped extension usually starts from the seed and extends to the partial alignment because the score of the alignment with the seed is typically much higher than that of the partial alignment. For this reason, we can compute the seed insertion positions simply with the score of a seed and the distance from the seed to the partial alignment.

Given a seed $S$ of score $\text{Score}(S)$ (i.e., the distance from the seed to the partial alignment) is computed as follows: $G(S) = \left\lfloor \frac{\text{Score}(S) - \text{GOP}}{\text{GEP}} \right\rfloor - 1$, where $\text{GOP}$ and $\text{GEP}$ are gap opening penalty and gap extension penalty, respectively. Given a query sequence $Y$ and a partial alignment $(x_{a,r}, y_{b,s})$, the query position $q$ is subject to embedding a seed of length $k$ as follows:

For N-terminal seed:

$$\max[-k, \epsilon - G(S)] \leq q \leq \min[\epsilon + G(S), |Y| - k - 1], \quad \text{where} \quad \epsilon = b - a \quad (1)$$

For C-terminal seed:

$$\max[k, |Y| + (\epsilon - G(S)) - 1] \leq q \leq \min[|Y| + k - 1, |Y| + (\epsilon + G(S)) - 1] \quad (2)$$

where $\epsilon = s - r$.

Note that the query embedding position $q$ is computed relative to the original query sequence positions. For example, if the N-terminal seed is inserted at the beginning of the query, $q$ is then $-k$ in order to preserve the original query sequence positions in the alignment. Recall that the region of interest starts immediately after the seed and in this way we can preserve the original positions for the subsequent computations. The insertion positions for the C-terminal seeds are also represented similarly. For C-terminal calculations, if $\max[k, |Y| + (\epsilon - G(S)) - 1]$ is larger than $\min[|Y| + k - 1, |Y| + (\epsilon + G(S)) - 1]$, no C-terminal seed is inserted. The idea of maximum gap has been described previously for connecting HSPs with gaps [Rognes 1998].
Step 3. Generate final alignments with a seed. For each query position \( q \) identified in step 2, we perform an alignment with the seed \( S \) inserted in the respective position to generate the final alignments. The final alignments are generated by running dynamic programming starting at the end position of the seed, \((|S| - 1, q)\), and proceeding to \((|X| - 1, |Y| - 1)\).

Since we are working with highly divergent sequences producing low-identity alignments, it is reasonable to consider the scenario that a longer alignment with lower score can be biologically more meaningful than a shorter alignment with higher score [Huang 2006]. Motivated by this observation, during the alignment, we adjust the alignment score with respect to the length of the alignment as follows:

\[
s_a(a, b) = \begin{cases} 
  s_c(a, b) \log a & \text{if } a > 1 \\
  s_c(a, b) & \text{otherwise}
\end{cases}
\]

where \((a, b)\) is a cell in the dynamic programming matrix, and \(s_c(a, b)\) and \(s_a(a, b)\) are the score before and after adjustment, respectively. Note that \(a\) represents the alignment length at position \((a, b)\) in the dynamic programming matrix. If we have the best score at \((a, b)\), we have the final alignment \((x_{S|S| - 1, a}, y_{q, b})\).

Step 4. Filter out non-significant alignments. Not all alignments produced from the previous step are informative. In this step, we prune out insignificant alignments using the metrics, % coverage and pairwise identity. Given an alignment \((x_{a, r}, y_{b, s})\), the coverage of the alignment to a target sequence is calculated as follows:

\[
Coverage = \begin{cases} 
  \frac{r + 1}{|X|} & \text{if } N \text{-terminal seed} \\
  \frac{|X| - a}{|X|} & \text{otherwise}
\end{cases}
\]

where \(X\) is a target sequence. The pairwise identity considered here is the identity of the
alignment excluding the matches in a seed: i.e., $pairwise\ identity = \frac{\# of\ matches}{\alpha} \times 100$, where
\[
\alpha = (\# \ of \ matches + \# \ of \ mismatches + \# \ of \ gaps) \ in \ an \ alignment \ excluding \ the \ seed.
\]
If the pairwise identity and coverage of a final alignment are greater than the thresholds, minimum identity and minimum coverage, the alignment is returned to the user.

3.5 PERFORMANCE EVALUATION

3.5.1 EXPERIMENT SETUP AND DATASETS

Both GDDA-BLAST and Ada-BLAST were implemented in C, and compiled for both Linux and Windows environments. GDDA-BLAST utilizes rps-BLAST in NCBI BLAST 2.2.15 package to compute the alignments. In order to validate our approach, we tested both for execution time and accuracy. The execution time experiment was conducted on a dedicated machine with 1.8GHz Intel Core\textsuperscript{TM} 2 duo processor and 2GB memory running Windows Vista. The experiment for accuracy was performed on a server with eight Dual-core 2.4 GHz AMD Opteron processors and total of 32G memory running Linux. Note that for the execution time experiment we used a less-equipped dedicated machine instead of the server shared by others in order to measure the execution time more accurately.

For test data set, SABmark Twilight zone benchmark set was used [van Walle 2005]. This dataset contains protein sequences that are classified by SCOP classification [Murzin 1995]. The sequences in the same group of SABmark Twilight zone set have sequence similarity below 25% identity, making this one of the most challenging test sets for sequence analysis algorithms. As a template library, PSSMs from NCBI CDD were used [Marchler-Bauer 2005]. The pre-
classified set of protein sequences serves as gold-standard based on which we can measure our method.

3.5.2 EXECUTION TIME AND ACCURACY

To compare the execution time of GDDA-BLAST and Ada-BLAST, we ran both methods with 602 query sequences randomly chosen from the SABmark Twilight zone set and 51 target sequences randomly selected from the CDD database. Figure 3.3(a) shows the per-query alignment execution time when a given query is run against the 51 PSSMs in the library. The lengths of the 602 query sequences range from 34 to 759 amino acids. Note that the running time of GDDA-BLAST increases linearly as a function of query sequence length. Conversely, Ada-BLAST shows much better scalability with respect to the size of query because it inserts a seed only at the positions where the seed is likely to be extended. Moreover, the performance gain is maximized when the two sequences compared are of low-identity because the number of likely seed embedding positions is limited. This makes Ada-BLAST an attractive alternative for the alignment of highly divergent sequences. Overall, Ada-BLAST is 19.3 times (± 15.29 S.D.) faster than GDDA-BLAST on average while it achieves more than 100 times speed-up in many occasions.

To see if Ada-BLAST alignments can be used to encode informative phylogenetic profiles for proteins as GDDA-BLAST alignments do, we challenged structural homology detection in highly divergent protein sequences. For the test, we used 534 sequences from 61 fold groups in SABmark Twilight zone set. We employed 23,511 NBCI CDD PSSMs as a measuring PSSM set, and each 534 queries were encoded in a vector of composite scores. We used Pearson’s correlation coefficient to measure the structural similarity between two proteins using their vectors. We performed Receiver Operating Characteristic (ROC) curve analysis [Bradley
1995] to measure the performance of GDDA-BLAST and Ada-BLAST. ROC curve shows the sensitivity of each method at different false positive rates, where a left-shifted curve is considered more accurate. Sensitivity and false positive rate are calculated as follows: \(\text{Sensitivity} = \frac{TP}{TP + TN}\), \(\text{Specificity} = \frac{TN}{TN + FP}\), \(\text{False Positive Rate} = 1 - \text{Specificity}\), where \(TP\) is # of true positives, \(TN\) is # of true negatives and \(FP\) is the number of false positives. To calculate sensitivity at different false positive rates, given each query, we considered the sequences with \(k\) (highest Pearson’s correlation coefficient to the query) as related, if increased \(k\) from 1 to ~40. As shown in Figure 3.3(b), the performance difference of GDDA-BLAST when compared with Ada-BLAST is negligible.

Figure 3.3: Performance Comparison of GDDA-BLAST and Adaptive GDDABBLAST. (a) Per-query running time of GDDA-BLAST and Adaptive GDDABBLAST. (b) Fold recognition performance of GDDA-BLAST, Adaptive GDDA-BLAST, PSI-BLAST and SAM-T2K on SABmark Twilight zone set is shown with ROC curves.

3.6 SUMMARY

We proposed here an upgraded seeded alignment algorithm for obtaining low-identity sequence alignments, termed as Ada-BLAST. Despite the great potential of GDDA-BLAST, it suffers
from its high computational cost. Ada-BLAST addressed this problem by exploiting the similarity among modified query sequences to adaptively avoid expensive computations. The experimental results show that Ada-BLAST is more than 19 times faster than GDDA-BLAST while maintaining sensitivity. As Ada-BLAST performs in the twilight zone of sequence similarity, this approach can be harnessed to decode the most challenging protein datasets, and scaled up to screen proteomes and the vast quantities of sequence being obtained from metagenomic studies.
Chapter 4

FOLD RECOGNITION OF PROTEINS WITH TWILIGHT ZONE OF SEQUENCE SIMILARITY

4.1 BACKGROUND

4.1.1 PROTEIN STRUCTURE

Protein structure is broken into four levels (Figure 4.1). Primary structure refers to the linear sequence of amino acids. Secondary structure is regular local structures. Alpha-helix and beta-sheet are two common secondary structure elements. Unique combinations of secondary structure elements are often called as protein folds. Tertiary structure refers to the three dimensional (3-D) structure of a single protein molecule. Quaternary structure is the associated structure of multiple protein molecules to make an active unit. Not all proteins have a quaternary structure. Each molecule folds into a stable tertiary structure and then they associate with each other to form a quaternary structure.

Regarding protein structure, the ultimate goal of computational algorithms is to predict 3-D structures of proteins from their primary structures. The 3-D structures of proteins are determined by experimental techniques such as X-ray crystallography or Nuclear Magnetic Resonance (NMR). These techniques are, however, time-consuming so that the gap between available protein sequences and available protein structures is increasing in public protein databases [Edwards 2009]. At present, the most reliable computational algorithms for 3-D protein structure prediction (e.g. I-TASSER, MODELLER, ROBETTA [Zhang 2007, Sali 1993, Chivian 2003]), are based on comparative modeling (homology modeling). Comparative modeling predicts the 3-D structure of a target protein using the structure of a protein that may be
homologous to the target protein and the sequence alignment between the target protein sequence and the template protein sequence. Using a suitable template structure thus determines the quality of a predicted structure. Fold recognition algorithms can assist to find good template structures given target protein sequences.

Figure 4.1: Different level of protein structure\(^1\)

\(^1\) http://en.wikipedia.org/wiki/Protein_structure
4.1.2 SCOP DATABASE

The Structural Classification of Proteins (SCOP) database provides the classification of protein structures on hierarchical levels based on their structural, functional, and evolutionary relationship [Murzin 1995]. SCOP database consists of 4 classification levels: class, fold, superfamily, and family. In the family level, proteins are classified on the basis of (i) sequence identities that are >30% or (ii) functions or structures that are very similar regardless of their sequence similarity. Superfamilies consist of families with similar functional or structural protein structures, thus suggesting that a common evolutionary origin is probable. At the fold level, the superfamilies are classified in the same fold if their proteins have the same secondary structure topology. At the class level, different folds are classified into the same class on the basis of their secondary structures comprising of (i) all alpha-helices, (ii) all beta-sheets, or (iii) both of alpha-helices and beta-sheets. Based on the definition of folds in the SCOP classification, the proteins in the same fold do not necessarily have homology in evolutionary sense. Since they have structural similarities, however, some often said the proteins in the same fold or with structural similarities have “structural” homology [Rinderknecht 1978, Rossmann 1976].

4.1.4 FOLD RECOGNITION

As discussed, fold recognition is the first step to build high quality 3-D structures of proteins by comparative modeling. This is done either by (i) aligning the sequence of a target protein with the sequences of the proteins whose structures are already resolved (e.g., FFAS, HHpred, and SAMP 
T2K [Jaroszewski 2005, Soding 2005, Karplus 2005]) or by (ii) aligning the secondary structures that are predicted from the target protein sequence directly to the structurally resolved structures [Soding 2005, Francesco 1999, Russell 1996]. Note that HHpred is categorized in both
approaches since this works with or without secondary structure prediction. The first approach relates with the sequence alignment problem for remotely related proteins, while the second approach relates with the problem of secondary structure prediction. Since structures are more conserved than sequences [Baker 2000, Grinshin 2001], profile-based alignment algorithms, which perform well in twilight zone of sequence similarity, are frequently used for fold recognition. As an example for the second approach, Rost et al. proposed a fold recognition algorithm based on the secondary structure prediction algorithm, PHD, which the same authors previously proposed (see chapter 6 for the details on secondary structure prediction). Given a target sequence, they predict its secondary structures and generate 1-D structure profile at which each residue is assigned one of the three secondary structure states (alpha-helix, beta-sheet, and others). For structurally resolved proteins, observed 1-D structure profiles are also generated. The 1-D structural profile of a target sequence is aligned with those of structurally resolved proteins to find the proteins that have structural homology with the target protein. According to [Soding 2005], the second approach outperforms the first one only slightly.

4.2 MOTIVATION

According to [Chothia 1992], the number of distinct native state protein folds is extremely limited. In addition, structure is more conserved than sequence similarity [Baker 2000, Grinshin 2001]. Taken together, these attributes underscore the inverse protein folding problem; whereby the vast and varied numbers of primary amino acid sequences that exist in biology occupy a relatively limited number of structural folds. Due to the extreme divergence (≤25% pairwise identity) that can exist between structurally resolved (template) sequences and structurally unknown (target) sequences, fold recognition is often compromised. Thus, the crucial information specifying protein structure must be contained in a very small fraction of the amino acid sequence, making
the informative points hard to measure. Therefore, a solution to the inverse protein folding problem must be the ability to identify these information points and use them to relate targets to appropriate template sequences.

We apply our method, Adaptive BLASTing framework for structural homology detection. Herein, we report that structural sequence profiles (vector representations encoding the structural information of proteins) are a fast and robust method for fold recognition which works in the twilight zone of sequence similarity. Our findings demonstrate that structural sequence profiles are an interoperable quantitatively method for analyzing primary amino acid sequences and selecting appropriate templates for structural modeling.

4.3 METHOD

Structural sequence profiles are derived from user-defined libraries of Position-Specific Scoring Matrices (PSSMs) of structurally similar proteins, which contain a frequency table for substitutions that occur in related sequences; PSSMs are a powerful measure of homology. Indeed, it is well-established that PSSMs contain more information than individual sequences [Henikoff 1997, Altschul 1997, Schaffer 1999]. We take advantage of the increased information content of PSSMs and the speed of BLAST to quantify alignments within a structural sequence profile. Three features make our method distinct from traditional sequence analysis methods. First, we measure targets with multiple structure-specific PSSM libraries. Second, we quantify low-identity alignments, which are traditionally considered statistically insignificant. Third, we consider all relationships (to the same fold and different folds) to extract meaningful signals, which appear to be important for measurements in the twilight zone [Ko 2008, Hong 2009, Bhardwaj 2009, Ko 2009].
Our method involves four steps to infer remote structural homology between proteins (Fig. 4.2).

First, a PSSM library is generated for each SCOP fold (1086 folds in SCOP 1.65) [Murzin 1995] using SCOP domain sequences in each fold whose sequence identity is less than 40% to each other. Except in cases where large numbers of structures already exist (e.g. SCOP fold b.1; Immunoglobulin-like beta-sandwich fold which already has >1000 sequences), all SCOP fold groups are expanded by PSI-BLAST [Altschul 1997] search against NCBI NR database using the SCOP domains in the fold groups as queries. The settings for PSI-BLAST is 3 maximum number of iterations, 30 maximum number of database sequences returned at each iteration (-b option), and other options remained as default. The sequences similar to the SCOP sequences (≥90% identity) are removed. For each fold group, redundant or highly similar sequences (≥40% identity by Needleman-wunsch algorithm [Needleman 1970]) are also eliminated. Fold-specific libraries for 1086 fold groups are then built by generating PSSMs from
the sequences obtained from PSI-BLAST. Fold-specific PSSMs are then compiled as a BLAST compatible database [Marchler-Bauer 2005] (Fig. 4.2-b).

Second, each query sequence is then searched against the SCOP fold PSSM libraries using rps-BLAST [Altschul 1997, Schaffer 1999]. The alignments returned from the search are filtered out if they do not satisfy our e-value and coverage thresholds (i.e., alignment length as a function of library PSSM length). In the studies, alignments are collected using e-values of either 0.01, no coverage or e-value $10^{-10}$, 80% coverage thresholds.

Third, a fold-specific score is calculated. The given alignments returned from the rps-BLAST search of each query against a fold-specific PSSM library, each amino acid of a query which is identically or positively (identical, but conserved) aligned is scored with BLOSUM62 score of the aligned pairs. These scores are summed for each amino acid of the query (i.e., positional score). The fold-specific score is calculated using:

$$\sum_{i=1}^{n} r_i$$

if $r_i > 0$ where $n$ is the length of a protein sequence and $r_i$ is a positional score of $i^{th}$ amino acid of the protein. Then each query is encoded in a structural sequence profile a vector of fold-specific scores (Fig. 4.2c).

Fourth, as a quantitative measure of how two targets are similar (i.e. the structural similarity score), we calculated the Pearson’s correlation coefficient between their vectors.

### 4.4 RESULTS

We tested the efficacy of this approach using the TZ-SABmark, a carefully curated set of fold-specific sequences of remote homology [van Walle 2005]. Each fold-specific sequence group represents a SCOP fold classification of related sequences with ≤25% sequence identity. From the original TZ-SABmark, 534 sequences from the first 61 fold groups (avg. length of $135.27 \pm 89.39$
s.d.) were used as a test set. For the test, fold-specific libraries were built with SCOP domains after excluding TZ-SABmark sequences to avoid debate.

### 4.4.1 Alignment Comparison and Information Content

We first evaluate sequence similarity between TZ-SABmark test sequences and the sequences used for building fold-specific libraries. Figure 4.3 (a) plots cumulative frequency distributions of pairwise %identity between pairs of TZ-SABmark test sequences and PSSMs from their true- and false-fold groups. These statistics demonstrate that ~95% of all same-fold pairs have <20% pairwise identity. Indeed, this distribution is negligibly distinct from comparisons of different-fold pairs. Additionally, we compare the sequence similarity between the SCOP reference sequences and those PSSMs which were obtained through their PSI-BLAST expansion. The sequences used to define fold-specific libraries are also in the twilight zone. Taken together, this indicates that our information source is: (a) derived from low-identity alignments, (b) improved by including intermediate sequences in the library, and (c) not due to redundancy.

It is reasonable to consider that a protein would have a larger fold-specific score for its true-fold than for its false-folds; this is confirmed in Figure 4.3 (b) that demonstrates that our fold libraries are specific. We observe that 99.8% of the query sequences have fold-specific scores ≤0.1 for different-folds, while only 24.3% of them have scores ≤0.1 for same-folds. Given these data, if we annotate each protein by the highest fold-specific score, the folds of 70.8% of TZ-SABmark test sequences can be predicted correctly (e-value 0.01, no coverage). Figure 4.3 (c) shows cumulative frequencies of structural similarity score between pairs of same-fold (blue, 3,428 pairs) and different-fold (red, 65,536 pairs) query sequences. ~24.2% of same-fold pairs have structural similarity scores >0.1, while only ~0.2% of different-fold pairs have scores >0.1. Figure 4.3 (d) plots structural similarity scores between same/different-fold pairs versus their
pairwise % identity. We observe an independent trend between structural similarity score and pairwise identity whereby true positives distribute to higher structural similarity scores (see Fig. 4.3 (e), (f), (g) for the statistics of e-value $10^{10}$, 80% coverage threshold setting).

Figure 4.3: Characterization of structural similarity scores. (a) The distribution of %pairwise identity between pairs of TZ-SABmark sequences and the library sequences of the same-fold (blue) and different-fold (red), and %pairwise identity between the PDB reference sequences and PSI-BLAST expanded sequences (green). (b),(e) The distribution of query sequence scores for each fold-specific library (c),(f) Cumulative frequencies of the structural similarity scores between pairs of same-fold (blue) and different-fold (red) query sequences. 3,428 same-fold pairs and 65,536 different-fold pairs were measured from 534 sequences. (d),(g) Structural similarity
4.4.2 PERFORMANCE EVALUATION

SAM-T2K, prof_sim, HHsearch 1.5.0 and FFAS03 are used as benchmark methods [Jaroszewski 2005, Yona 2002, Soding 2005, Karplus 2005]. For SAM-T2K, blastall in NCBI BLAST 2.2.15 is used for target2k script in SAM3.5 package searching a sequence database to collect sequences for HMM generations for 534 test sequences. When a query sequence is scored given a HMM model by hmmscore, Smith-Waterman algorithm was used by default. For prof_sim, sequence profiles are generated by PSI-BLAST and profile-profile alignment is done with local alignment setting. For HHsearch 1.5.0, PSI-BLAST is used to build HMMs of TZ-SABmark test sequences with the setting of –j 5 –h 0.001. Database of TZ-SABmark HMMs is generated and query HMM is searched with default settings. For all four benchmark methods, NCBI NR database with 6,419,591 protein sequences is used as a sequence database. FFAS03 was run by a member of Gozik lab² to a false-positive rate ~0.01. In the result of each method, all-against-all comparison of TZ-SABmark test sequences is done, and for each sequence, all other sequences are sorted by structural similarity score (in case of our method) e-value/p-value (in case of benchmarking methods) for ROC curve analysis.

In Figure 4.4 (a), we compare ROC curves of our method with two different settings (e-value 0.01, no coverage and e-value $10^{10}$, 80% coverage thresholds, see Fig. 4.4 (b), (c) for the results of different thresholds) versus four traditional fold recognition methods (FFAS03, ²http://bioinformatics.burnham.org)
HHsearch, prof_sim and SAM-T2K). The results demonstrate that *structural sequence profiles* of both settings outperform these benchmarking methods. Additionally, we tested how our method performs with fold-specific libraries built with SCOP domains which have 30-40% and less than 25% sequence identity to TZ-SABmark test sequences (Fig. 4.4 (d)). Given the fold-specific libraries from SCOP domains with 30-40% sequence identity, we achieve ~60% of total sensitivity. While it is true that the SCOP domains which have relatively high sequence similarity to TZ-SABmark sequences help construct very informative fold-specific libraries, fold-specific libraries from SCOP domains with less than 30% identity still constitute ~50% of total sensitivity, indicating that *structural sequence profiles* can perform as good as FFAS which shows the best performance among all benchmarking methods with this very limited set of reference sequences.

The sensitivity of *structural sequence profiles* using only statistically significant alignments from rps-BLAST (e-value 0.01, no coverage) are ~0.6 at false positive of 0.01. *Structural sequence profiles* of e-value $10^{-10}$, 80% coverage obtain similar sensitivity at a false positive rate ~0.04, but its sensitivity increases up to ~0.7 at a false positive rate 0.1 due to the obtained additional alignments. Intriguingly, the alignments obtained from both filtering strategies reside in the twilight zone.

Figure 4.5 (a) quantifies the independence between predictions of *structural sequence profiles* with two different settings (e-value 0.01, no coverage vs. e-value $10^{-10}$, 80% coverage) for true-positives, false-positives, and false-negatives. Interestingly, we observe a significant number of unique true-positive pairs at both e-value settings. This suggests that comparative measurements are likely to be useful for the identification of true-positive pairs. We made the same comparison between our method (e-value 0.01, no coverage), FFAS03, and HHsearch (Fig. 4.5 (b), see Fig. 4.5 (c) for comparisons using e-value $10^{-10}$, 80% coverage threshold). The diagrams indicate that *structural sequence profiles* obtain more unique true-positive pairs and false-negative pairs while predicting fewer false-positive pairs. The most dramatic increase
occurs between unique true positives whereby structural sequence profiles obtain 5.6 fold increase over FFAS03 and a 10.3 fold increase over HHsearch.

Figure 4.4: Fold recognition performance of structural sequence profiles. (a) Comparison of ROC curves of structural sequence profiles with two different settings, FFAS03, HHsearch, prof_sim, and SAM-T2K. (b) Comparison of ROC curves of structural sequence profiles with different coverage threshold when the e-value threshold is fixed at 10^{-10}. (c) Performance of structural sequence profiles in 4 different combination of e-value and coverage threshold. (d) Performance of structural sequence profiles with the three sets of fold-specific libraries generated with SCOP domains whose redundancy is filtered with three different identity thresholds.
4.4.3 APPLICATIONS FOR FOLD CLASSIFICATION

Based on the promising results described above, we sought to perform a forward-engineering and blind experiment. We obtain two targets from the ongoing 9th Critical Assessment of Techniques for Protein Structure Prediction (CASP) competition. Fortunately, the results of these two targets have been already released and we present our results for these newly acquired structures here (T0520 and T0523, Fig 4.6 (a), (b) respectively). These proteins are termed as “human/server targets” because they are considered challenging targets containing low homology domains.

In our pipeline, we first determine which of the SCOP fold libraries has the highest fold-specific score with the target. To select the best template structures from the fold, all SCOP

<table>
<thead>
<tr>
<th></th>
<th>True positive</th>
<th>False positive</th>
<th>False negative</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>e-val 0.01, no cov.</td>
<td>e-val 10^10, 80% cov.</td>
<td>e-val 0.01, no cov.</td>
</tr>
<tr>
<td>a</td>
<td>28.2% 43.2% 28.6%</td>
<td>22.3% 0.9% 76.9%</td>
<td>20.4% 59.2% 20.4%</td>
</tr>
<tr>
<td>b</td>
<td>49.4%</td>
<td>8.5% 3.5% 25.6%</td>
<td>27.1%</td>
</tr>
<tr>
<td></td>
<td>4.4% 5.8% 2.8%</td>
<td>31.1% 4.4% 34.7%</td>
<td>3.0% 48.3% 7.1%</td>
</tr>
<tr>
<td>FFAS03 prof_sim</td>
<td>FFAS03 prof_sim</td>
<td>FFAS03 prof_sim</td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>35.3%</td>
<td>8.1% 2.6% 27.7%</td>
<td>27.2%</td>
</tr>
<tr>
<td></td>
<td>8.4% 12.4% 5.5%</td>
<td>31.3% 4.9% 35.1%</td>
<td>1.6% 21.6% 4.9%</td>
</tr>
<tr>
<td>FFAS03 prof_sim</td>
<td>FFAS03 prof_sim</td>
<td>FFAS03 prof_sim</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 4.5:** Comparative statistics to show independence between predictions of different methods (structural sequence profiles, FFAS03, prof_sim) or structural sequence profiles with different settings (e-value 0.01, no coverage vs. e-value 10^10, 80% coverage) for true-positives, false-positives, and false-negatives
domains in the fold (SCOP 1.75) and the target were represented in a vector of (%identity × %coverage) score of an alignment to each PSSM in the fold-specific library and then performed hierarchical clustering using Pearson’s correlation coefficient as a similarity metric. The SCOP domains with the highest Pearson’s correlation coefficient to the target are then used as templates for template-based structure modeling (purple boxes). Sequence identity between the SCOP domains and these targets are highly divergent (grey text). Following our pipeline, the sequences of the selected SCOP domains and the target are aligned using MUSCLE [Edgar 2004] and threaded models were generated using Modeller [Eswar 2006]. We observe that both targets accord well to the crystalline structures, with the carbon-backbones of our models deviating less than 1.5 angstroms from observed. These results suggest that the Sequence profiling framework holds promise for structural modeling.

Figure 4.6: 3-D structure prediction of CASP9 targets.
4.5 DISCUSSION

In this chapter, we reveal the power of *structural sequence profiles* for fold recognition in the “twilight-zone” of sequence similarity. Our results support the hypothesis that *structural sequence profiles* provide a robust user-defined structural modeling application. This is supported by several key findings from our measurements: (i) “twilight-zone” pairwise alignments are informative (Fig. 4.2), (ii) better performance than multiple benchmarking methods in TZ-SABmark by providing more unique true-positive pairs (Fig. 4.3), and (iii) the capability of reconstituting structural fold classifications, including sub-fold groupings that are not encoded in the PSSM library (Fig. 4.5, and Fig. 4.6). A number of broad implications can hence be derived from this study.

Again, we report here that low-identity alignments are a rich source of information, which can be used to unmask the fundamental properties of proteins, including protein structure, function, and evolution using simple arithmetic [Ko 2008, Hong 2009, Ko 2009]. As described above we build diverse scalable PSSM libraries in the *Sequence profiling* framework. We use the information content provided by PSSMs to increase the signal-to-noise ratio inherent to low-identity alignments. In addition, we demonstrate that a coverage threshold is an effective filter of noisy alignments (Fig S2c). When fold-specific scores are encoded into a vector (i.e., *structural sequence profiles*), multiple data mining algorithms can be used reliably to measure fold attributes.

We also evaluate the performance of *structural sequence profiles* correlated using Pearson’s correlation coefficients to relate divergent structural folds. When compared to popular profile-based algorithms such as FFAS03, HHsearch, SAM-T2K and prof_sim, *structural sequence profiles* obtain a significant portion of unique true-positive pairs and reduced false-positives. Taken together, this enhances our increased performance.
Considering the current genomic explosion of sequences, fold-recognition methods are needed as they are a true watershed in Biology. Based on the results presented here, conversion and PSI-BLAST expansion of the PDB into fold-, superfamily-, and family-specific PSSM libraries would, in theory, synergize and improve structural modeling in general. The current weakness of our application is the incomplete PSSM fold-specific libraries. Future work is aimed at expansion and improvement of these libraries using all available information in structural databases.
Chapter 5

MSA-FREE EVOLUTIONARY STUDY WITH HIGHLY DIVERGENT PROTEINS

In this chapter, we describe our study to reconstruct the evolutionary history of highly divergent retroelement proteins using the Adaptive BLASTing framework. Our results show that Adaptive BLASTing is a promising and innovative method to study evolution.

5.1 BACKGROUND

5.1.1 MULTIPLE SEQUENCE ALIGNMENT

Multiple sequence alignment (MSA) is an alignment of three or more biological sequences that are assumed to have a structural, functional, and/or evolutionary relationship. Under the assumption that related proteins are descended from a common evolutionary ancestor, one aligns their sequences to each other by MSA methods and tries to find conserved regions of the sequences as such regions are associated with functionally, structurally, and/or evolutionary critical parts of related proteins. In the MSA shown in Figure 1.2(b), each of the conserved regions is associated with the structural elements that the aligned proteins commonly have in their structures. Thus, MSA methods have been required to study many biological problems, such as reconstruction of phylogenetic trees, generation of profiles (e.g., PSSMs), and prediction of function or structure of proteins.

MSA methods are usually categorized into two approaches: progressive and iterative. Progressive MSA methods first compute pairwise distance for every pair of sequence using their pairwise alignments. A clustering method (e.g., Neighbor-joining [Saitou 1987]) is applied to
generate a guide tree, which shows the relationship among sequences in a tree format. A MSA is built by first aligning the most similar sequences and then successfully adding less related sequences or groups of sequences until the most distant sequence is added. ClustalW [Thompson 1997] is one of the most popular MSA algorithms used in progressive approaches. Progressive MSA methods are relatively fast but their performance is highly dependent on an initial guide tree. Errors in initial alignments cannot be corrected and tend to propagate to the entire alignment. The basic steps of a progressive approach are described in Figure 5.1. In contrast to a progressive approach, iterative MSA methods try to fix the errors in initial alignment by re-aligning previously calculated alignments iteratively. Given a previously calculated MSA, iterative MSA methods continue to improve both the guide tree and MSA, stopping the iterations when they are converged. Dialign [Morgenstern 2004] is one of the iterative MSA methods.

While MSA methods are routinely used for a wide variety of biological problems today, they have limitations [Kemena 2009, Rosenberg 2005]. In the post-genomic era, people try to align greater numbers of sequences. However, as the number of sequences to align is increasing, computational cost of MSA method is also increasing quickly, making it difficult to study large datasets. Low sequence similarity between the sequences in question introduces another challenge. Rosenberg showed that accuracy of MSA quickly drop as starting sequence identity approaches ~50% [Rosenberg 2005]. It is interesting that MSA method yields a MSA with 44% of aligned identity (i.e., the proportion of identically aligned sites in a MSA) while true identity of test sequences is 25-26%. This difference arises due to MSA methods that increase aligned identity to optimize a MSA. Overall, large highly divergent datasets would be the greatest challenge for today’s MSA methods.
5.1.2 INFERENCE OF EVOLUTIONARY HISTORY OF PROTEINS

Biologists study evolutionary history of proteins and that of organisms that contain them by analyzing accumulated mutations in their amino acid sequences. For the analysis, a set of homologous protein sequences are aligned by MSA. The divergence information shown in the MSA is used to infer evolutionary distances among proteins. Automatically generated MSAs are often subjected to manual adjustment to improve their quality. Given the MSA, pairwise distances between all sequences are calculated. For example, p-distance, which is the simplest distance measure, is as follows: 

\[ p - \text{dist} = \frac{n}{L} \]

where \( n \) is the number of amino acid sites which are different in MSA and \( L \) is the total number of sites compared.

Given the pairwise distances, an evolutionary tree, which is also called a phylogenetic tree, is constructed to present the relationships among proteins. In the tree, leaves and interior nodes represent proteins in present day and hypothetical ancestors, respectively. If the tree is rooted, a path from the root (i.e., the common ancestor of all) to a leaf defines an evolutionary path, while an unrooted tree only specifies relationships among proteins, rather than an evolutionary path. Many methods (e.g., Neighbor-joining, Minimum evolution methods [Saitou 1987, Rzhetsky 1992]) have been proposed to build phylogenetic trees. Neighbor-joining method constructs a phylogenetic tree by minimizing the sum of all branch-lengths on the tree. Starting with a star-like tree where each leaf corresponds to individual protein, two nodes are selected to create a new node between the root and the selected nodes iteratively. Two nodes \( i \) and \( j \) are selected to minimize 

\[ q_{ij} = (r - 2)d_{ij} - (R_i + R_j) \]

where \( d_{ij} \) is the distance between nodes \( i \) and \( j \), \( R_i \) is \( \sum_{k=1}^{n} d_{ik} \), and \( r \) is the remaining number of nodes close to the root. When a new node \( A \) is
generated by joining the two nodes, the distance between $A$ and a remaining node, $k$, is replaced by $d_{ak} = (d_{ik} + d_{jk} - d_{ij}) / 2$ [Mailund 2006, Studier 1988].

Minimum evolution method tries to make a tree with a topology which has the smallest sum of all branches by any tree-generating methods. Neighbor joining method is the simplest minimum evolution method.

5.2 MOTIVATION

Inferring evolutionary relationships among highly divergent protein sequences is a daunting task. The problems that arise in the twilight zone of sequence similarity are even more challenging for multiple sequence alignment algorithms, which have been the only computational approach for the evolutionary study of proteins, to the best of our knowledge, other than pairwise alignment algorithms. We showed that structurally related proteins within the twilight zone of sequence similarity are related given carefully generated fold-specific PSSM libraries (chapter 4). Amino acid sequences specify structure, function, and evolution of proteins. Thus, in this chapter, we test our framework to measure evolutionary information of protein sequences, in addition to structural information. The results show that our framework has the capacity to derive evolutionary relationships among highly divergent proteins, entirely independent of multiple sequence alignment.
5.3 METHOD AND RESULTS

5.3.1 DATASET

Self-replicating genetic elements such as retrotransposons use reverse transcriptase (RT, an RNA-dependent DNA polymerase) to multiply via an RNA intermediate copied into DNA [Eickbush 2008]. These highly diverse and likely ancient proteins are extremely effective at replicating and, along with other transposable elements, make up to ~50% of eukaryotic genomes by weight [Eickbush 2008, Boeke 2003]. The first RT was discovered as a retroviral encoded enzyme [Baltimore 1970]. Subsequently, multiple genetic elements from diverse organisms have been shown to encode proteins that share sequence similarity to the retroviral RT, including cellular telomerase [Eickbush 2008]. Given (i) the >20 years of research/literature on this protein family, (ii) the extreme nature of divergence within the known family members, and (iii) its implications for the early evolution of life on earth and major infectious diseases of humans [Darnell 1986], Retroelements are an excellent and rigorous benchmark set to test whether Adaptive BLASTing can measure evolutionary distances.

The RT domain is the only known region common to all classes of Retroelements and therefore is often used for comparative analysis [Eickbush 2008, Xiong 1990, Arkhipova 2003]. Within the highly divergent RT domain, seven conserved motifs in the catalytic region of the enzyme have been identified that enable evolutionary inferences of retrotransposons [Eickbush 2008, Xiong 1990, Poch 1989, Kohlstaedt 1992]. However, limiting the alignment space to these motifs requires potentially subjective manual editing, generating few evolutionary informative sites. Thus, deep evolutionary relationships are often ambiguous at best. Indeed, even these seven conserved motif are as divergent as their functional constraints will allow [Eickbush 2008, Malik
2001]. As a consequence, the precise evolutionary relationship of the Retroelements is still a subject of debate.

We curated 88 RT-containing protein sequences representing 11 groups of Retroelements. The individual groups are from a broad range of taxa and comprise (i) long-terminal-repeat RTs (LTR: containing Ty1/Copia, Ty3/Gypsy, and BEL/Pao subgroups), (ii) retroviruses (e.g., HIV), (iii) pararetroviruses (RT-containing DNA viruses), including hepadnaviruses of animals (e.g., hepatitis B) and caulimoviruses of plants (e.g., cauliflower mosaic virus), (iv) tyrosine recombinase RTs (e.g., DIRS-1 elements), (v) non-LTR retrotransposons, (vi) Penelope-like elements (PLE), (vii) telomerases (TERT), (viii) group II introns (i.e., retrointrons), (ix) Mt Plasmids (i.e., retrolasmsids), (x) Ms DNAs (i.e., retron), and (xi) diversity-generating retroelements (DGR) [Eickbush 2008, Boeke 2003, Goodwin 2000, Doulatov 2004, Medhekar 2007].

5.3.2 STATISTICS ON SEQUENCE SIMILARITY

Quantitative analysis of within-group and between-group variations in sequence similarity was performed by using global pairwise alignments on RT domain of test sequences. Overall, the average percentage identity between 88 RT-sequences was 17.7% (± 6.0% s.d.). Specifically, 3,644 pairs (95.2%) among 3,828 possible pairs of these 88 sequences having <25% sequence identity. Within our dataset, the group with the highest sequence identity is the Mt plasmid group (average 61.2% identity ± 41.8% s.d.), and the group with the lowest sequence identity is the telomerase group (average 17.9% ± 4.4% s.d.). As a whole, RT sequences reside in the twilight zone of sequence similarity, underscoring the reason why deducing evolutionary relationships within the RT family is challenging.
5.3.3 METHOD

**Building Phylogenetic Tree using Adaptive BLASTing.** In this study, we used pre-calculated NCBI CDD domain PSSMs (24,280 PSSMs including 16 RT-specific PSSMs) as template PSSMs. Since we do not use RT-specific PSSMs only, we defined RT domain regions of each of the 88 test sequences to limit our measure only to RT domain regions. The definition of RT domain region was performed as similar to the definition of domain boundary given seeded alignment in GDDA-BLAST (Figure 2.1(b)). Each of 88 sequences was aligned with 16 RT-specific PSSMs by rps-BLAST, and the overlapped region of the resulting alignments over the test sequence is defined as a RT domain boundary. We ran seeded alignment algorithm for every pair of test sequence and template PSSM (10% seed size, 60% coverage, 10% identity thresholds used) and calculated a composite score. Each of the 88 test sequences were represented in a vector of composite scores. Pairwise evolutionary distance between each pair of 88 sequences was calculated using Euclidean distance metrics. We constructed phylogenetic tree of the 88 RT-containing proteins using minimum-evolution method, which is implemented in MEGA4 software [Tamura 2007]. Minimum-evolution method build a phylogenetic tree based on the assumption that the tree with the smallest sum of branch length estimates is most likely to reflect the true evolutionary history [Edgar 2004, Lassmann 2006].

**Building Phylogenetic Tree using MSA as Benchmark Methods.** For comparison, we performed multiple sequence alignment of the 88 RT sequences with various established algorithms that measure evolutionary distance, such as ClastalW, MUSCLE, K-align, and Dialign [Thompson 1997, Edgar 2004, Lassmann 2006, Morgenstern 2004]. Given each of the resulting alignments without manual editing, pairwise distances of 88 RT sequences were calculated using
Poisson correction model with $\gamma$ parameter 1.0 after pairwise deletion of gaps. Then, we built a minimum evolution tree of 88 RT-containing proteins.

**Calculating Statistical Significance of Phylogenetic Tree.** For statistical significance of phylogenetic trees that MSA-based method generated, we performed bootstrap re-sampling with 1,000 replicates and generated consensus tree of the samples by using MEGA4. For the phylogenetic tree that is generated using our framework, we performed both bootstrap and jackknife re-sampling. For re-sampling, the measuring templates PSSMs are randomly selected to generate a random sample, which is a vector of composite scores only to the selected PSSMs. Fitch and Consensus (Majority extended rule) in PHYLIP 3.67$^3$ were used to generate minimum evolution trees of random samples and their consensus tree.

---

5.3.4 RESULT

Figure 5.1 shows the phylogenetic trees that were generated by our method and Dialign. Among four benchmark methods, Dialign generated a phylogenetic tree with the highest statistical support values on the branches, especially on the outer branches. In our phylogenetic tree (Figure 5.1(a)), nearly all of the 88 RT sequence display monophyly. Importantly, we observe multiple within-group and between-group clades in our tree that are corroborated by multiple independent studies. These relationships include (i) a clear segregation of the LTR and prokaryotic clades, (ii) telomerases and PLE occurring sister groups, and (iii) the overall topology of the LTR clade. We tested the robustness of our phylogeny by two types of re-sampling methods, such as bootstrap and jackknife. These statistical support values are displayed in order of jackknife and bootstrap in our tree. Despite no measurable statistical support at some nodes, several noticeable features are found in the statistical values to support reliable branching patterns.

There is no exact answer for evolutionary history of retroelement proteins. To evaluate the performance of reconstructing evolutionary history, it is important to compare the independent results obtained by our method and Dialign with the results in the literature obtained by manual editing of retroelement sequences. Based on random considerations, obtaining similar results for the 14 clades of Retroelements is grossly improbable. For example, non-LTR and LTR elements are clearly distinct, as has been previously suggested [Xiong 1999, Doulatov 2004].
Telomerases and PLEs also form a sister clade which has also been seen in other studies [Arkhipova 2003, Doulatov 2004]. In large part, both phylogenies also recapitulate the results for the topology of the LTR group found in [Goodwin 2000], including retroviruses and pararetroviruses. With the exception of Mt plasmids, the topology for the prokaryotic group is the same in both analyses, which accord with previous studies [Eickbush 2008, Arkhipova 2003, Doulatov 2004]. All MSA methods tested here and manually edited trees in the literature place the Mt plasmids in the prokaryotic group. Conversely, our result places Mt plasmids with the telomerases and PLEs, although this position has no statistical support. Nevertheless, it has been demonstrated that Mt plasmids have 3’ terminal repeats similar to those of chromosomal telomerases, making them the potential precursor of telomerases [51]. Another key difference between the results of Dialign and our method is the placement of the hepadnaviruses. Dialign places the hepadnaviruses between the Bel/Ty1 clades, in contrast to the result obtained with our method and previous reports [Goodwin 2000, Plant 2000].

5.4 DISCUSSION

Within the twilight zone of sequence similarity, statistical support can help eliminate evolutionary ambiguities. Although none of the methods that we tested obtained robust deep-branch statistical support, having an independent approach, such as Adaptive BLASTing, to estimate evolutionary relationships undoubtedly represent an important advance. Even in the nascent stage of applying Adaptive BLASTing for evolutionary studies, it provided measurements that gave independent support for phylogenetic studies and key insight into evolutionary relationships among distantly related and/or rapidly evolving proteins. In fact, Adaptive BLASTing has been further applied to various evolutionary studies since we first proposed it. A phylogenetic tree of 716 retroelement proteins that was generated using 846 RT-specific PSSMs was reported in [Bhardwaj 2010].
addition, Bhardwaj et al. reconstructed a phylogeny of simulated data in the Adaptive BLASTing framework. While the evolutionary history is unknown for any protein family, Rose (Random Model of Sequence Evolution) provides simulated sequences that are generated based on a probabilistic model of protein sequence evolution [Stoye 1998]. Since Rose knows the accurate evolutionary history of the simulated protein sequences, it provides the perfect benchmark data. According to their result, Adaptive BLASTing-based method could recapture ~93% of the true evolutionary history of 67 simulated sequences, which have twilight zone of sequence similarity, while MUSCLE performs poorly. Overall, we proposed Adaptive BLASTing as a promising MSA-free method for evolutionary studies on large divergent datasets. Our method, of course, has a downside due to its innovative way of studying evolution. For example, theories, which have been established in a long history of evolutionary studies, are hard to use in our framework to estimate evolutionary distance (e.g., how evolutionary distance calculated by Euclidean distance can be interpreted in the timescale of evolution?). Further studies on the application of evolution theories in our framework would be open to future work.
Chapter 6

STRUCTURAL AND FUNCTIONAL CHARACTERIZATION OF PROTEINS

In this chapter, we test whether our method can be used to identify specific structural or functional characteristics of proteins from their amino acids. To achieve the goal, we propose methods to visualize conserved signals in low-identity alignments between a query protein sequence and a particular structure- or function-specific PSSM library. Our results show that our method is very promising as a tool for complete annotation of proteins that is the ultimate goal of our studies.

6.1 BACKGROUND

6.1.1 PROTEIN DOMAIN DETECTION

A protein domain is a basic unit of a protein that can independently function, evolve, or maintains its structure. Proteins often contain multiple domains which create new biological functions as combined. According to the divide and conquer strategy, the first step to understanding proteins is the prediction of each individual domain present in the protein. Protein domain prediction can be made by using Profile-based methods like CDD, Pfam, SMART, COG [Marchler-Bauer 2005, Bateman 2004, Letunic 2006, Tatusov 2000] build PSSMs or HMMs from MSAs of conserved domain sequence automatically or manually. To obtain these sequences automatically, domain databases like Pfam, DOMAINER [Bateman 2004, Sonnhammer 1994], cluster protein sequences into groups based on their pairwise alignments. Given a set of sequences in each group, conserved regions are defined by performing MSA on the sequences. Either PSSM or HMM are
generated only by using the sequences for the conserved regions. The generated profiles are deposited in a domain database, facilitating the search of query sequences containing domains with profiles already on the database.

6.1.2 PROTEIN SECONDARY STRUCTURE PREDICTION

Secondary structure prediction algorithms characterize structure of proteins in terms of secondary structure elements, such as alpha-helices, beta-sheets, or loops (See chapter 5 for more details on secondary structure of proteins). These algorithms provide 2-D information of protein structures like which parts of protein sequence corresponds to alpha-helices, beta-strands etc. Figure 6.1 show the output examples from PSIPRED and TMHMM [Jones 1999, McGuffin 1999, Sonnhammer 1998]. PSIPRED predicts the regions of secondary structure elements (alpha-helices, beta-strands, or loops) on protein sequences, while TMHMM predicts especially membrane-spanning regions that comprise mostly highly hydrophobic alpha-helices. 2-D structural information obtained from secondary structure prediction algorithms can further be used by 3-D structure prediction algorithms by providing constraints on 3-D structure modeled by the algorithms, or by fold recognition algorithms (e.g., HHpred) [Rost 1997, Soding 2005].

Secondary structure prediction algorithms can be categorized on the basis of the content of information they use for prediction and by how they use the information. First of all, there are algorithms using statistics [Nagano 1973, Chou 1974]. Chou et al., for example, predicted secondary structure of proteins as calculating probability of each query residue to be alpha-helix, beta-sheet or loop based on the observed frequencies of 20 amino acids in each of the secondary structure elements in the sequence of structurally resolved proteins [Chou 1974]. Lim et al. defined rules for prediction using physiochemical properties learned from structurally resolved proteins [Lim 1974]. Algorithms like PHD or PSIPRED use neural networks (e.g., PHD,
PSIPRED) and are probably most popular today [Rost 1993, Jones 1999, Pollastri 2004].
PSIPRED uses two-stage feed-forward neural networks trained by back-propagation using PSI-
BLAST PSSMs as inputs [Jones 1999, Rumelhart 1994]. While these algorithms using neural
networks remain a powerful tool, they have inherited the drawbacks that neural network
techniques have, such as the requirement of a large training dataset and a training stage that are
very time-consuming. The aforementioned algorithms assign each of secondary structure
elements to each query residue. TMHMM is categorized as a specialized secondary structure
prediction algorithm since it only predicts membrane-spanning helices.

6.1.3 PREDICTION OF CRITICAL RESIDUES OF PROTEINS

Not all amino acid residues in a protein are equally important to perform a function or maintain a
structure of the protein. Mutations on critical residues can result in a loss of protein function
and/or improper protein folding. These critical residues can be found experimentally, but requires

Figure 6.1: Prediction of Protein Structure (a) by PSIPRED (b) by TMHMM
a lot of lab time with repeated steps of mutating candidate residues and consequently observing any change in protein function/structure as a consequence of the mutation. Conveniently, various computational methods have been introduced. While the details of these methods are different, most of these methods heavily depend on MSA which predicts highly conserved residues as critical residues [Capra 2007, Berezin 2003, Wang 2006, Fischer 2007]. Thus twilight zone of sequence similarity is again an obstacle for predicting functionally or structurally important residues from amino acid sequences.

6.2 MOTIVATION

In chapter 2, we introduced the overall framework for Adaptive BLASTing. On access to a function- or structure-specific PSSM library, we measure the information for the specific function or structure from the alignments between a query sequence and the PSSMs of the library. The measured information is used to encode informative vectors for proteins further allowing protein comparison in the format of the vectors to reveal their relationship. We propose the scoring schemes to calculate function- or structure-specific scores given the alignments to PSSM libraries. We expect that proper analysis and visualization of measured information from these alignments can help us study specific protein characteristics of interest. Results obtained after studying proteins using the suggested methods show that Adaptive BLASTing framework is potentially a useful tool for computational biologists or even bench-scientists to study proteins and achieve their common goal of complete annotation of proteins in terms of their function and structure.
6.3 METHOD

We introduce methods to visualize the information measured in the alignments between a query sequence and PSSMs in a function- or structure-specific library in three different graphical representations. These can be used to study proteins for questions like the location of functional or structural domain, and the identification of critical amino acids. While traditional methods provide their results only in either of alignment or sequence logo formats, our method presents results in multiple output formats. Advantageously multiple output formats provide a medium to retest hypothesis of a query protein in comparison to having a single output where no comparison can be made and there is no scope for retest. Again, all method that we propose are under the assumption that the alignments to a PSSM library of protein characteristic $X$ contain the information in common to $X$.

**Alignment Overlap Graph.** Given specific alignments, we overlap the alignments onto a query sequence to define the boundaries which can be associated with the protein characteristic of a measuring PSSM library. The alignment overlap graph is generated simply as described in Algorithm 6.1.

---

**Algorithm 6.1:** Generating alignment overlap graph

**Input:** a query sequence $q$ and a set of alignments $A$

**Output:** an array $Q$ which represents an alignment overlap graph ($|Q| = |q|$)

for each alignment $a \in A$

$s \leftarrow$ start position of $a$ in $q$

$e \leftarrow$ end position of $a$ in $q$

for $i = s$ to $e$

$Q_i \leftarrow 1$

return $Q$

---

**Positional Conservation Graph.** Alignment overlap graph is generated as all query residues equally weighted whether the residue is aligned identically, positively, or deleted in an alignment. In positional conservation graph, residues are weighted differently so that we expect to see the
information related to the protein characteristic of interest in higher resolution. A positional
conservation graph is generated as follows. Let \( A(Q, T) \) be the set of alignments between a query
protein \( Q \) and a template library \( T \), \( n \) be the length of \( Q \), and \( m \) be the number of PSSMs in \( T \).
Each alignment \( a(t_i) \in A(Q, T) \) between \( Q \) and the PSSM \( t_i \) \((1 \leq i \leq m)\) in \( T \) is associated with a
continuous interval \( \tau(a(t_i)) \) of residues on \( Q \). We define a positional conservation score \( ps_i \) for
each amino acid residue \( q_i \) \((1 \leq i \leq n)\) on \( Q \) as follows:
\[
ps_i = \sum_{k=1}^{m} s(q_i, \gamma(A)), \quad 1 \leq i \leq n
\]
where \( \gamma(A) \) is the template residue aligned with \( q_i \) in \( A \),
\[
A_i = \{ A \in A(Q, T) : q_i \in \tau(a(t_i)), a(t_i) \in A(Q, T) \}
\]
is the alignment to the PSSM \( t_i \) in \( T \),
a(\( t_i \)) , which contains \( q_i \), and \( s(i, j) = \begin{cases} BLOSUM62(i, j) & \text{if } BLOSUM62(i, j) > 0 \\ 0 & \text{otherwise} \end{cases} \).

A positional conservation graph displays the conservation of signals that we measure for
a PSSM library in a form of low-identity alignments. This provides insight on which amino acids
are critical for the function or structure that the PSSM library represents. When the graph is
further analyzed via smoothing, baseline correction, or peak selection, peak regions with higher
signals than other regions on average can correspond to functionally or structurally important
regions. The examples of such analysis are described in Figure 6.2. Our results in the following
sections show that positional conservation can characterize secondary structure elements of
proteins.
Amino Acid Conservation Graph. While positional conservation graph represents signal conservation on each query residue in number values, amino acid conservation graph represents most likely conserved amino acids on each query residue. Amino acid conservation graph is almost the same as sequence logos [Crooks 2004, Schneider 1990] except for its requirement of MSAs. We simply overlap the alignments to each PSSM of a PSSM library onto a query sequence (merging the alignments by taking a query sequence as a frame of reference). For each query position, sequence conservation $R_{seq}$ is the difference between the maximum possible entropy ($S_{max}$) and the entropy of the observed amino acid distribution ($S_{obs}$), which is calculated
as follows: \[ R_{seq} = S_{max} - S_{abs} = \log_2 N - \left( - \sum_{n=1}^{N} p_n \log_2 p_n \right) \]
where \( p_n \) is the observed frequency of amino acid \( n \) at a query position and \( N \) is the number of amino acids (\( N=20 \)) [Crooks 2004].

Figure 6.3 show the examples of each of alignment overlap (blue, Figure 6.3 (a)), positional conservation (red, Figure 6.3 (a)), and amino acid conservation (Figure 6.3 (b)) graphs.

**Figure 6.3:** Graphical representation outputs from Adaptive BLASTing (a) blue: alignment overlap graph, red: positional conservation graph (c) amino acid conservation graph
6.4 STUDY OF MEMBRANE-SPANNING PROTEINS

6.4.1 FUNCTIONAL CLASSIFICATION

We first tested whether we can classify transmembrane proteins by their functions. All transmembrane proteins are similar in that they have, for the most part, highly hydrophobic helices that transverse lipid-bilayers of cell membrane yet are all functionally unique. We curated a set of transmembrane containing proteins from a range of different protein families (e.g., voltage-gated Ca\(^{2+}\), K\(^+\), and Cl\(^-\) channels, calcium-activated K\(^+\) channels, cyclic nucleotide gated channels, transient receptor potential channels (TRPs), receptor tyrosine kinases, G-protein coupled receptors (GPCRs), transporters and exchangers). Each of these 74 sequences was analyzed in *Adaptive BLAST* framework using our integral lipid-binding PSSM library (ILB) which contains 38,155 PSSMs we generated. Given the rps-BLAST alignments to those PSSMs, test sequences were represented in vectors of product scores (Chapter 2). Hierarchical clustering (complete linkage) was performed on the vectors using Pearson’s correlation coefficient as a similarity metric. In Figure 6.4, we report the condition (e-value threshold 100) that achieves the highest degree of classification. While the classification is not perfect, the clusters in the dendrogram show robust measures for pairing functionally related sequences. Intriguingly, several clades of potassium and cyclic-nucleotide gated channels are close to TRP channels. All three of these groups have 6 transmembrane-spanning regions (i.e., TMs) and are thought to be structurally related [Yang 2010]. Importantly, these fold/function-specific subgroups are not classification hierarchies in our PSSM library, but emerge as a function in the result. This data demonstrates that we can cluster protein families by their functions while they are all structurally related by using the PSSM libraries generated from proteins having a similar structural characteristics yet diverse function.
Figure 6.4: Functional classification of 74 Transmembrane Proteins
6.4.2 STRUCTURE CHARACTERIZATION

Next, we conducted analyses on a structurally resolved (X-ray Crystallography) transmembrane protein, Bovine Rhodopsin (PDB: 1F88) [Palczewski 2000] in order to see if the information that we measured given ILB PSSM library characterized the structure of the protein. Figure 6.5 depicts the output of rps-BLAST (e-value threshold 0.01) for the domain architecture of 1F88. Notably, rps-BLAST returns alignments for serpentine type 7 TM domain PSSMs. Given ILB PSSM library, we define domain boundaries and secondary structural elements with higher resolution by using additional information below the accepted statistical thresholds (seeded alignments in this study). We compared our performance against TMHMM, which is a HMM-based method designed specifically to predict TM regions on transmembrane proteins (Figure 6.6). While neither TMHMM nor our method accurately model the entire crystal structure, we observe several interesting features. For example, several of the membrane-spanning helices are

Figure 6.5: The result of searching 1F88 against NCBI CDD by rps-BLAST (e-value threshold 0.01)
interrupted by loop regions that are not identified by TMHMM. Indeed, the C-terminus of 1F88 contains 3 small helices, the last of which is a bent-helix that is believed to be parallel to the membrane (aa 288-348, Figure 6.6 (c)). Another region of interest is contained between aa 91 and 111, which is a loop in the crystal structure, but is predicted to be a short helix by us. We expect that this loop may be, under native condition, a bent-helix similar to other regions in the protein. Our amino acid conservation graph also demonstrates that 6 out of the top 8 residues with highest scores (higher scores representing increased conservation) surround the chromophore binding site, which must be conserved for vision to occur (Figure 6.6 (b)). Taken together, the results provide support for our framework as a useful tool to investigate any protein sequence in multi-views.
To further show the utility of Adaptive BLASTing framework for studying proteins, we performed analyses on a structurally resolved ankyrin-repeat protein called Human Ankyrin-R (PDB: 1N11) [Michaely 2002]. While TMHMM or similar methods exist to characterize the
structures of transmembrane proteins, yet there is no such method for ankyrin-repeat proteins. This protein is, thus, the perfect example to add even more value to our method. 1N11 has 12 ankyrin-repeats where each ankyrin-repeat is composed of two alpha helices separated by loops (Figure 6.7 (b)). To see if we can characterize the structure of this ankyrin-repeat protein, we first prepared 449 ankyrin-repeat PSSMS that were generated with the domain sequences returned after a simple keyword search of “ankyrin repeat” against NCBI CDD database. We then generated seeded alignments between 1N11 and each of the PSSMs. Figure 6.7 (a) depicts the

Figure 6.7: Adaptive BLASTing analysis of 1N11 using Ankyrin-repeat PSSM library (a) rpsBLAST result of searching 1N11 against NCBI CDD (b) the structure of 1F88. Each of 12 ankyrin-repeats are labeled. (c) Green: alpha-helices of 1F88 structure annotated, Pink: alignment overlap graph, Blue and orange: positional conservation graph after smoothed and baseline corrected with two different FFT settings.
output of rps-BLAST (e-value threshold 0.01) that predicts the presence of 4 ankyrin-repeat domains in 1N11. In contrast, the alignment overlap graph of Adaptive BLASTing depicts the domain architecture of 1N11 with all 12 ankyrin-repeat domains predicted (pink line in Figure 6.7 (b)). On performing positional analysis as previously done for 1F88, we could see alpha helices in each Ankyrin-repeat although the results are not yet perfect. It is interesting that we observed a signal at the C-terminus of the protein which is a long loop in the structure (red arrow in Figure 6.7 (b)). This long loop is associated with a small fragment of Spectrin-binding domain [Michaely 2002]. The appearance of the small signal thus seems likely due to the function of ankyrin-repeat proteins mediating binding activities.

6.6 STUDY OF TRANSIENT RECEPTOR POTENTIAL CHANNELS

We here studied TRP channel physiology, in particular vertebrate TRPC3, whose physiological role has yet to be clearly elucidated [van Rossum 2008, Caraveo 2006, van Rossum 2005, Patterson 2002], in our framework. The TRP channel super-family is well-known for being involved in all areas of sensory perception, as well as variety of other biological functions [Venkatachalam 2007, Clapham 2003]. The TRPC channels are most closely related to the TRP channel found in drosophila eye which are required for vision [Montell 1989]. Although these channels have been extensively studied, little structural data exists for these channels.
When the human TRPC3 sequence is searched against NCBI CDD database using rpsB
BLAST (e-value threshold 0.01), ankyrin-repeats, the ion-channel, and the TRP_2 domains are
identified (Figure 6.8 (a)). Individual alignments to the positive domains are displayed as
providing links to multiple sources of information for the domains. While informative, these data
cannot be easily extracted in a format that can be used for quantitative comparisons. Conversely,
we provide a positional graph to show conservation for our TM- and ankyrin-repeat-specific
PSSM libraries for TRPC3 (Figure 6.8 (b)). When compared to the rps-BLAST result, we suggest
that both the channel domain boundaries and the ankyrin-repeat boundaries are underestimated.

Similar to our channel boundary prediction, our result given Ankyrin-repeat-specific
library suggests that the number of ankyrin-repeats in the N-terminus of TRPC3 is

Figure 6.8: Domain prediction by rps-BLAST and Adaptive BLASTing (a) rps-BLAST result of
searching TRPC3 against NCBI CDD (b) positional conservation graphs generated by using ILB
(white) and Ankyrin-repeat (green) PSSM libraries
underestimated by rps-BLAST (Green, Figure 6.8 (a)). To investigate where this result could be accurate, we compared the results we obtained for TRPC3 with the results we obtained for the structurally resolved ankyrin-repeats in TRPV6 (Figure 6.9 (a)) given the Ankyrin-repeat library. In the alignment overlap graph, six clear peaks can be observed, which accord to the 6 ankyrin-repeats in the structure (Figure 6.9 (b)). The positional conservation graph (Figure 6.9 (a)) suggests that repeats 3 and 4 are highly conserved, while repeats 1, 2, 5, and 6 are less conserved. A very similar pattern for both the alignment overlap and positional conservation graph is observed for TRPC3 (Figure 6.9 (c)).

As these alignments extend from amino acid 1-280 in human TRPC3, this suggests that the TRP_2 domain (aa 194-260) in TRPC3, which has been implicated in peripheral lipid-binding and vesicle fusion [23, 44], is comprised of ankyrin-repeats. By homology, this also suggests that the TRP2_domain is contained in TRPV channels, which is generally not observed in these channels. To further investigate this hypothesis, we aligned the positive region of TRPC3 for Ankyrin-repeat library by our method with the chicken TRPV4 ankyrin-repeat sequence, which as been structurally resolved, using MUSCLE [Edgar 2004]. Although the overall alignment has low identity (~13%), the resulting structural model obtained from Modeller [Sali 1993] using the alignment was high-quality (Figure 6.9 (d)). Our TRPC3 model mirrors the carbon backbone of TRPV4 structure in the helical portions of the ankyrin-repeats, with the loop regions having a lower correlation; this may be expected if the function of this domain is not conserved between TRPC3 and TRPVs. Interestingly, the known TRP_2 domain in TRPC3 corresponds to ankyrin-repeats 5 and 6 in the TRPV4 structure (Figure 6.9 (d)), suggesting that these ankyrin-repeats may be involved in the lipid-regulation of TRPVs.
In chapter 2, we proposed the Adaptive BLASTing framework where we can measure information for a specific protein characteristic of interest from protein sequences. In the following two chapters, we showed our results to reveal the relationship among remotely related proteins after encoding their sequences in vectors of scores for different protein characteristics. Here, we explored the potential for using the information measured for a particular protein characteristic to study individual proteins. We proposed the methods to display...
the information in three different output formats. All the outputs visualize signal conservation in
protein sequences for a protein characteristic of interest, but in different formats. The multi-view
from these output formats provides confidence for the signal observed or helps to obtain
additional information not displayed in one format but in another.

We showed our results of characterizing the structures of transmembrane and ankyrin-
repeat proteins. In addition to prediction of just domain boundaries of TM and ankyrin-repeats,
secondary structure elements of the proteins could also be predicted. Our results also indicated
that Adaptive BLASTing can be of assistance to identify amino acid residues that are critical for
protein function and/or structure from positional and amino acid conservation graphs. Various
methods are proposed to perform different jobs, such as domain prediction, secondary structure
prediction, and prediction of functionally or structurally important residues, but Adaptive
BLASTing does all the jobs in a single framework.

Although efficient, in order to be a more robust method, Adaptive BLASTing needs to
overcome some limitations. First, we should be able to provide statistical confidence for the
signals measured. For example, rps-BLAST provides e-values for predicted domains, while
TMHMM provides posterior probabilities for predicted TM regions. In contrast, Adaptive
BLASTing only provides alignment overlap or positional conservation scores which are not
comparable when using different set of PSSM libraries. Given these scores, it can only be
expected that regions with “relatively” high scores are likely to be associated with the protein
characteristic of a measuring PSSM library. While the peaks in alignment overlap and positional
conservation graphs correspond to functional sites or structural elements, our procedure to define
their boundaries still requires manual effort (e.g., baseline correction) (Figure 6.2). An automatic
baseline correction for these graphs seems to be a necessary feature.
Chapter 7

CONCLUSION AND FUTURE WORKS

In this study, we proposed a novel computational method called Adaptive BLASTing to infer the structure, function, and evolution of proteins. This method is based on individual PSSM libraries that are constructed for specific protein characteristics of protein structure, function, or evolution. Given an unknown protein sequence, we used low-identity alignments between this query and PSSM libraries to identify the signals that were specific for a particular characteristic. Although low-identity alignments are considered insignificant or unreliable sources of information, we were able to include them in our analysis and our results show that these alignments can be informative when they are properly filtered and analyzed. We also developed scoring schemes to calculate structure- or function-specific (PSSM library-specific) scores for query sequences or residues. In our method, protein sequences are represented by vectors of the scores for different PSSM libraries. The correlation (distances) between these vectors indicated the structural or functional similarity (evolutionary distances) between proteins. In addition, we also represented protein sequences by three different graphical representations of signal conservation. (Alignment overlap, positional conservation, and amino acid conservation). In theory, this method could be used to analyze protein sequences for any characteristics by using suitable PSSM libraries and output formats.

We tested our method on three kinds of protein analyses, namely, evolutionary analysis of highly divergent proteins, protein fold recognition, and characterization of protein structures. In the first test (chapter 4), to analyze the evolutionary history of highly divergent retroelement proteins, we represented 74 retroelement protein sequences, that are in twilight zone of sequence similarity, by score vectors for different protein domains. Then, we reconstructed the
phylogenetic tree of these proteins by using the Euclidean distance between their vectors as the evolutionary distance. Although our result cannot be validated because the true evolutionary history of retroelement proteins is unknown many of the evolutionary relationships in our phylogenetic tree were consistent with those in previous studies. In the second test (chapter 5), to identify relationships among structurally related proteins with low sequence similarity, we constructed PSSM libraries for 1076 unique folds and analyzed 534 test sequences from 61 different groups of protein folds. The structural relationships among these sequences were identified based on the Pearson’s correlation coefficient between their score vectors for different fold-specific PSSM libraries. Our results showed that our method outperformed other sophisticated fold recognition algorithms and revealed distant structural similarities among proteins based on their relationship to different folds. In the third test (chapter 6), to determine whether our method could be used to identify specific characteristics in individual proteins, we used PSSM libraries that were specific for transmembrane domains and ankyrin-repeats to analyze transmembrane and ankyrin-repeat proteins with known structures. Our results showed that we could characterize the structures of these proteins (e.g., prediction of membrane-spanning regions, prediction of ankyrin-repeat helices) by visualizing conserved signals in low-identity alignments between a query protein sequence and a specific PSSM library.

Collectively, these results demonstrated that Adaptive BLASTing is a potentially powerful method to study protein sequences. This method has four advantages that make it superior to other benchmarking methods for protein annotation. First, traditionally, a single profile (e.g., PSSM, HMM) that is generated from a MSA of protein sequences is used to represent a whole protein family. Since these sequences must be aligned in the MSA, the performance of the profile is limited by the performance of the MSA algorithms. In contrast, our method uses multiple profiles (PSSMs) to represent a protein family. As a result, our PSSM libraries could represent more highly divergent protein families. Second, our method uses a novel
representation of protein sequences as vectors of scores to infer structural, functional, or evolutionary relationships among proteins. MSA algorithms, which are the only existing methods to infer the evolutionary distance between protein sequences, are limited to the analysis of relatively small numbers of protein sequences with moderate or high sequence similarities. In contrast, Adaptive BLASTing allows evolutionary relationships to be identified in large sets of highly divergent protein sequences. Third, our method can identify specific characteristics of proteins from their protein sequences. For example, if it is not known whether a protein sequence $A$, which does not have a significant match with any other protein in public databases, has a characteristic $X$, then a PSSM library for $X$ could be constructed from a set of sequences that are known to have $X$ and used to determine if $A$ has $X$ based on the alignments between $A$ and the PSSM library. Fourth, by using a coverage threshold, our method can discriminate between noisy and informative low-identity alignments. In addition, we proposed a method to merge multiple low-identity alignments to reveal their common informative signals. Finally, it is interesting to note that all of these advantages could be implemented in a single computational framework.

As mentioned previously, our method is based on PSSM libraries, which determined the information content of the signals that we calculated from protein sequences. For protein fold recognition, we constructed 1,086 fold-specific PSSM libraries that were based on the fold classification of the SCOP database. By using these PSSM libraries, we could predict protein folds by identifying structural relationships among proteins. However, the SCOP database is much smaller than the Protein Data Bank (PDB) which is the largest protein structure database [Berman 2002]. Since the goal of fold recognition algorithms is to identify the best available template structure for any query protein sequence to predict its 3-D structure, the next step should be to construct PSSM libraries for all unique clusters of similar protein structures in the PDB.

The Gene Ontology (GO) project aims to use controlled vocabularies in three domains: molecular function, biological processes, and cellular components [Ashburner 2000].
Furthermore, the terms in these vocabularies are structured by hierarchical relationships. Since each Go term is associated with gene and proteins, we could use the protein sequences that are associated with each of the GO terms for molecular functions to construct PSSM libraries for individual protein functions. In the future, we envision that any protein function could be analyzed by using a set of function-specific PSSM libraries.

Finally, adaptive BLASTing could be improved by providing a measure of statistical confidence for the PSSM library-specific scores or signal conservation score on sequence residues, similar to the e-value that BLAST returns with alignment score.
REFERENCES


VITA

Yoojin Hong

Ph.D., Department of Computer Science and Engineering  2005 – 2010
The Pennsylvania State University, University Park, U.S.A.

M.S., Department of Computer Science and Engineering  2002 – 2004
The Pennsylvania State University, University Park, U.S.A.

B.S., Department of Computer Science  1997 – 2001
Sookmyung Women’s University, Seoul, South Korea