SIMILARITY SEARCH AND ANALYSIS OF PROTEIN SEQUENCES AND STRUCTURES: A RESIDUE CONTACTS BASED APPROACH

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

AHMET SAÇAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
COMPUTER ENGINEERING

AUGUST 2008

Approval of the thesis:

SIMILARITY SEARCH AND ANALYSIS OF PROTEIN SEQUENCES AND STRUCTURES: A RESIDUE CONTACTS BASED APPROACH

submitted by **AHMET SAÇAN** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Volkan Atalay _____
Head of Department, **Computer Engineering**

Prof. Dr. İ. Hakkı Toroslu _____
Supervisor, **Computer Engineering Department, METU**

Assoc. Prof. Dr. Hakan Ferhatosmanoğlu _____
Co-supervisor, **Computer Sci. & Eng., The Ohio State Univ.**

Examining Committee Members:

Prof. Dr. Volkan Atalay
Computer Engineering, METU _____

Prof. Dr. İ. Hakkı Toroslu
Computer Engineering, METU _____

Assoc. Prof. Dr. Göktürk Üçoluk
Computer Engineering, METU _____

Assist. Prof. Dr. Tolga Can
Computer Engineering, METU _____

Assist. Prof. Dr. Osman Abul
Computer Engineering, TOBB-ETU _____

Date: 07.08.2008

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: Ahmet Saçan

Signature :

ABSTRACT

SIMILARITY SEARCH AND ANALYSIS OF PROTEIN SEQUENCES AND STRUCTURES: A RESIDUE CONTACTS BASED APPROACH

Saçan, Ahmet

Ph.D., Department of Computer Engineering

Supervisor : Prof. Dr. İ. Hakkı Toroslu

Co-Supervisor : Assoc. Prof. Dr. Hakan Ferhatosmanoğlu

August 2008, 117 pages

The advent of high-throughput sequencing and structure determination techniques has had a tremendous impact on our quest in cracking the language of life. The genomic and protein data is now being accumulated at a phenomenal rate, with the motivation of deriving insights into the function, mechanism, and evolution of the biomolecules, through analysis of their similarities, differences, and interactions. The rapid increase in the size of the biomolecular databases, however, calls for development of new computational methods for sensitive and efficient management and analysis of this information.

In this thesis, we propose and implement several approaches for accurate and highly efficient comparison and retrieval of protein sequences and structures. The observation that corresponding residues in related proteins share similar inter-residue contacts is exploited in derivation of a new set of biologically sensitive metric amino acid substitution matrices, yielding accurate alignment and comparison of proteins. The metricity of these matrices has allowed efficient indexing and retrieval of both protein sequences and structures. A landmark-guided embedding of protein sequences is developed to represent subsequences in a vector space for

approximate, but extremely fast spatial indexing and similarity search.

Whereas protein structure comparison and search tasks were hitherto handled separately, we propose an integrated approach that serves both of these tasks and performs comparable to or better than other available methods. Our approach hinges on identification of similar residue contacts using distance-based indexing and provides the best of the both worlds: the accuracy of detailed structure alignment algorithms, at a speed comparable to that of the structure retrieval algorithms. We expect that the methods and tools developed in this study will find use in a wide range of application areas including annotation of new proteins, discovery of functional motifs, discerning evolutionary relationships among genes and species, and drug design and targeting.

Keywords: Protein Sequence and Structure, Similarity Search, Distance Based Indexing, Amino Acid Substitution Matrix

ÖZ

PROTEİN DİZİLERİNİN VE YAPILARININ BENZERLİK ARAMASI VE ANALİZİ: AMİNO ASİT TEMASLARINA DAYALI BİR YAKLAŞIM

Saçan, Ahmet
Doktora, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Hakkı Toroslu
Ortak Tez Yöneticisi : Doç. Dr. Hakan Ferhatosmanoğlu

Ağustos 2008, 117 sayfa

Protein dizisi ve yapısının yüksek-verimli tespitine olanak sağlayan yöntemlerin hasıl olmasının, hayatın dilini çözümleme arzu ve çabamız üzerinde muazzam tesiri olmuştur. Biyolojik moleküllerin benzerliklerinin, farklılıklarının ve birbirleriyle etkileşimlerinin incelenmesinden yola çıkarak bunların vazifesi, işleyişi ve evrimi ile ilgili yeni keşifler yapılabilmesi gayesiyle, genetik ve protein verileri artık fevkalade bir hızla biriktirilmektedir. Ancak biyomoleküler veritabanlarının gün geçtikçe büyümesi, bu bilginin duyarlı ve etkili yönetimi ve analizini sağlayabilecek yeni bilişsel yöntemlerin geliştirilmesini gerekli kılmaktadır.

Bu çalışmada, protein dizilerinin ve yapılarının erişimi ve karşılaştırmasını sahih ve oldukça etkili yapmaya olanak sağlayan yaklaşımlarımızı önermekte ve gerçekleştirmekteyiz. Benzer proteinlerin birbirine karşılık gelen amino asitlerinin diğer amino asitlerle benzer temaslarda bulunduğu gözleminden yola çıkarak, biyolojik olarak anlamlı sonuçlar veren ve proteinlerin doğru hizalanmasını ve karşılaştırılmasını sağlayan yeni bir takım amino asit değişim dizeyleri elde ettik. Bu dizeyleri, hem protein dizilerinin hem de yapılarının etkin bir şekilde indekslenmesinde kullandık. Protein alt-dizilerini vektörel bir temsile indirgeyip yaklaşık, ama

oldukça hızlı indeksleme ve benzerlik araması gerçekleştiren bir metod geliştirdik.

Protein yapılarının erişimi ve karşılaştırılması şu ana kadar ayrı ayrı ele alınnıyordu. Biz ise, aynı anda her iki işlevi gören ve diğer yöntemlerden daha iyi sonuç veren bütünleşik bir yaklaşım önerdik. Yaklaşımımız, benzer amino asit temaslarının uzaklık-tabanlı indeksleme kullanılarak tayinine dayalı olup her iki alanda da en iyi sonuçlara ulaşmıştır: ayrıntılı yapısal hizalama algoritmalarının doğruluğunda sonuç verirken, bunu yapısal erişim algoritmalarının hızıyla karşılaştırılabilir bir hızda gerçekleştirebilmektedir. Bu çalışmamızda geliştirdiğimiz yöntem ve araçların, yeni proteinlerin vazifelerinin tayini, işlevsel desenlerin keşfi, genler ve türler arasındaki evrimsel ilişkilerin açığa çıkarılması, ve yeni ilaçların bulunması ve hedeflenmesi gibi alanlarda uygulama bulacağını beklemekteyiz.

Anahtar Kelimeler: Protein Dizisi ve 3 Boyutlu Yapısı, Benzerlik Araması, Uzaklık Tabanlı Indeksleme, Amino Asit Değişim Tablosu

To
Annem, Babam, Dulcy, Kayra, Yabo, and Gary

ACKNOWLEDGMENTS

I would, first and foremost, like to thank my family (including my mother-in-law Heidi Nemeth) for their love, patience, and encouragement. I would also like to thank my mentors Prof. Dr. Hakkı Toroslu and Assoc. Prof. Dr. Hakan Ferhatosmanoğlu for their support and guidance during the development of this work, at both personal and professional capacities. And many thanks to the faculty and staff at the Computer Engineering Department at METU and to the members of the Database Research Group at OSU, who, with their friendly and jolly spirits, have made my Ph.D. study a bearable, memorable, and pleasent experience.

The work described in this manuscript was partially supported by Turkish Scientific and Research Council (TÜBİTAK) Grant 107E173 and US National Science Foundation (NSF) Grant IIS-0546713.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Over the past few decades, the philosophy and methodology of research in biological sciences have shifted tremendously to make use of *in-silico* modeling and analysis, besides the traditional *in-vitro* and *in-vivo* experimentation. This shift was primarily due to genomic sequences becoming available at an ever increasing rate with the advent of high-throughput sequencing techniques. GenBank [11], a central database of publicly available DNA sequences, has been doubling in size every 15 months; the genomes of more than 800 organisms have been completely sequenced since 1995, and there are close to 3,000 more ongoing genome projects [87, 14]. Following the structural annotation of these genomes, attention is now focused on determining the function of the identified genes. Determining the biological role of these genes using traditional genetics research methods is difficult, costly, and time consuming. Thus, most functional annotation methods compare and contrast the protein of interest with the database of available proteins whose functions are already known.

In biology, two or more structures are said to be *homologous* if they have evolved from a common ancestor. Detecting homologous proteins in the databases is of paramount importance for at least three reasons [158]. Firstly, it enlarges the number of proteins for which functional inference can be made. Secondly, detection of functionally important regions is made easier, since they retain less number of mutations. Thirdly, the detection of very distant relationships might reveal unexpected evolutionary links between organisms. As a consequence, similarity search in genomic databases constitutes an important part of the bioinformatics research.

Most of the current similarity search and protein comparison approaches are purely sequence-based. However, for sequences that have diverged too much over the course of evolution, sequence similarity may not be at detectable levels. On the other hand, 3D structural resem-

blance between ancient homologs is often still identifiable, because the structure is in closer connection with the function, and thus tends to be more conserved [61]. 3D structure can also provide deeper insight into the function of the protein, because it is possible to determine the active sites, and discern substrate level interactions and biochemical functions from the spatial conformation of the amino acid residues.

The phenomenal rate of increase in the protein sequence and structure data have surely opened new doors leading to important biological discoveries. In the meantime, the growing size of these databases, the diversity of the types of information being collected, and the complexity of the queries being sought present new computational challenges and demand new ways of maintaining and searching the deposited data.

In this thesis, we develop efficient and sensitive methods to handle and analyze protein sequence and structure data and tackle the challenges brought by the data size. Specifically, we derive an amino-acid substitution matrix from the interactions formed in protein three-dimensional structures (Chapter 3) that allows a fast but approximate approach to sequence similarity search problem (Chapter 4). The local interactions in the proteins is further used in an effort to detect significant structural motifs (Chapter 5), and in an integrated approach to search and comparison of protein structures (Chapter 6).

1.1 Motivation

Availability of efficient and sensitive methods for the analysis protein sequences and structures is of great value in gaining insight into the structure, function, and biological importance of the proteins. The indexing and similarity search systems significantly increase the ability to manage and process biological data and to discover new knowledge, helping to advance the field of biological science. The application areas that a protein similarity search system takes part include discovery of the genes responsible for certain functions in the metabolic pathway, determination of the function of a newly identified gene, identifying the biological mechanism of a biological function, protein modeling, personalized medicine, and drug design and targeting.

A typical scenario for the study of a disease is to first identify the gene-X of interest that is responsible for the disorder. The online repositories such as OMIM [100] or text-mining

in previous scholarly publications [31] can be used to retrieve already identified genes. If the responsible genes are not already known, quantitative trait loci (QTL) [89] or chemical mutagenesis [9, 74] can be used to locate the gene on the genome, or microarray expression analysis [136] can be performed to identify differentially expressed genes between normal and disease phenotypes.

Sequence: Once the DNA sequence of the gene-X is obtained [125], it is searched against the available genomic sequence databases using BLAST or PSI-BLAST [3]. If there is a high level of similarity between the protein-X[1] and the database hits, the information available for the database proteins can directly be used to annotate the new protein. If, on the other hand, the similarity is not trivial, a multiple sequence alignment [111] can be used to determine the residues conserved across different organisms or across related genes in the same organism. The highly conserved residues are generally critical for the function of the protein; otherwise these residues would not have had any selective pressure to resist mutations during the evolution. The pattern of conserved residues can further be searched against sequence pattern databases [42] to see if the protein-X contains any putative functional motifs.

Structure: When there is too much divergence between related sequences, they may not be found by sequence search tools; and even when they are found, the residue correspondences resulting from sequence alignment may be inaccurate. In such cases, one resorts to a structural analysis of the protein. The "structure" of the protein-X is the locations of its atoms in 3D space, and can be determined using X-ray crystallography or NMR spectroscopy [106]. The structure of protein-X is then searched against the database of protein structures to identify similar protein structures. Fold-level similarities can give clues for the function and biological mechanism of the protein-X, such as presence of a Zinc-finger or OB-fold domain may indicate a regulatory role through DNA-binding. More detailed analysis of spatially local motifs may unveil the active sites on the protein structure.

[1] We assume that gene X is a coding gene and can be translated to the corresponding protein-X.

1.2 Problem Definition

1.2.1 Sequence Alignment and Similarity Search

The sequence of a protein q is represented by a string of length m whose symbols are from the alphabet $\Sigma = \{\alpha_1, \alpha_2, ...\alpha_\sigma\}$, where each α_i corresponds to one of the 20 amino-acids. The *edit distance* from a protein q to a protein s is defined as the total cost of substitution, insertion, and deletion operations required to transform q to s. Alternatively, the *alignment* of the sequences q and s is defined by the set of residue correspondences as illustrated in Figure 1.1, and the alignment score is the sum of the substitution scores for each aligned pair of residues and the gap penalties. The level of sequence homology between two proteins is usually given as the *percent sequence identity* (ratio of identical amino-acid residues) of the aligned residues.

```
q:  ...CALCULATOR...
        |   |  | |
s:  ...COMPU--TER...
```

Figure 1.1: Example of a sequence alignment. The insertions and deletions are represented by an additional gap symbol "-" Equivalent aligned residues are marked with a "|".

The substitution score of the individual residues is looked up from an *amino-acid substitution matrix M* (also known as *similarity matrix*). A substitution matrix, like PAM250 [121] or BLOSUM62 [33], is a 20x20 listing of scores for aligning each amino acid with another amino acid. A *difference matrix* (also known as *distance matrix*) can be readily obtained from a similarity matrix by subtracting each entry from the maximum similarity score in the matrix.

The number of possible sequence alignments is exponential in the length of alignments; the optimal alignment is the one that has the maximum alignment score (or, the minimum edit distance) and can be found using dynamic programming. Optimal alignment of the entire query sequence is called a *global alignment* and can be found using the Needleman-Wunsh algorithm [113], whereas optimal alignment of any two sub-sequences of q and s is called a *local alignment* and can be found using the Smith-Waterman algorithm [144].

Note that the optimal alignment is dependent on the choice of substitution matrix M and may not reflect the biologically most accurate set of residue correspondences. The accuracy of a substitution matrix M can be evaluated as the fraction of the correct correspondences identified with respect to expert-curated sequence alignments. This forms the topic of Chapter 3, where we show that substitution matrices generated from residue contact profiles yield biologically accurate alignments.

Given a database $S = \{s_1, s_2, ...\}$ of protein sequences, the similarity search problem for a given query q is defined as finding the subsequences of the database proteins that give the maximum alignment score. The alignment score is often converted to a statistical significance measure and the database sequences that satisfy a given significance threshold are returned [2]. The success of a similarity search method can be measured as how well it retrieves the proteins that are classified as homologs by experts.

The common heuristic employed in sequence similarity search assumes that homologous proteins share short exact subsequences (*words* of length k) and tries to first identify the matching short words from the database proteins and extend around these seeds [3]. Shorter words would generate too many false seeds from unrelated proteins (e.g., for $k = 1$, almost all of the proteins would need to be checked for extension), while longer words would miss true seeds from the related proteins. Furthermore, the size of the hash table used to access database words is exponential in k, and long words are not feasible. For proteins, $k = 3, 4$ is typically used. One of the active research directions is to extend the basic "short exact match" assumption to obtain a more sensitive and efficient heuristic to identifying true candidate seeds. In Chapter 4, we are concerned with identifying longer, inexact word matches from the database proteins using an approximate sequence embedding and spatial indexing based approach.

1.2.2 Structure Alignment and Similarity Search

The structure of a protein q is defined by the 3D coordinates of its atoms. In the context of structure alignment, often only the alpha carbon atom (CA) of each amino acid residue is considered. The pairwise structure alignment problem is then finding the solution to two inter-related sub-problems: finding the residue correspondences between two proteins, and finding the optimal transformation matrix to superimpose the two structures. An example

structure alignment is given in Figure 1.2.

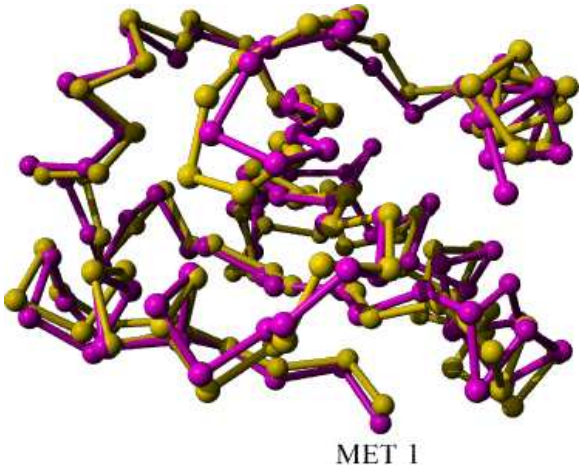


Figure 1.2: Structural alignment of Thioredoxins from humans (3trx, orange) and from the fruit fly (1xwc, pink) Drosophila melanogaster. The sequence identity of the corresponding residues is 43.8%. The 1st residue, which is a Methionine for both proteins and forms the N-terminus, is labeled. The protein backbone chain (rods) is shown as connecting the CA atoms (balls) of the amino acid residues.

The optimality of the superposition is generally measured using two measures: the length of the alignment (N, *coverage*, number of corresponding residues), and the root mean square deviation ($RMSD$, *accuracy*) of the superposition defined as:

$$RMSD = \sqrt{\frac{\sum_i d_i}{N}} \tag{1.1}$$

where d_i is the Euclidean distance between the i^{th} corresponding residues from proteins q and p. Note that a trade-off exists between the length of the alignment and the RMSD error. It is generally possible to produce short structural alignments with very low RMSD error (e.g., aligning only a single residue from each protein would trivially achieve 0.0Åerror). And naturally, a higher RMSD error is incurred for longer alignments. There has been several attempts to summarize the quality of the structural alignment in a single scoring function. The TM-score [178] has recently been proposed as a normalized score for quantifying the quality of a structural alignment. The TM-score is defined as:

$$TM\text{-}score = \frac{1}{L_{target}} \sum_{i}^{N} \frac{1}{1 + (\frac{d_i}{d_0(L_{target})})^2} \tag{1.2}$$

where L_{target} is the length protein of interest (such as the query protein in a database search)[2], d_i is same as above, and $d_0(L_{target})$ is a normalizing factor so that the average TM-score is not dependent on the protein size. d_0 is calculated as the average distance between an aligned pair of residues in a randomly related structural alignment to a protein of length L_{target}, and is approximated by[3]:

$$d_0(L_{target}) = 1.24\sqrt[3]{L_{target} - 15} - 1.8 \tag{1.3}$$

In line with the sequence similarity search, the structural similarity search involves retrieval and alignment of structurally similar database proteins for a given query protein structure. Please note that unlike sequence similarity search where the database retrieval is already associated with an alignment, the structure retrieval and alignment have so far been considered separately. In Chapter 6, we propose and implement an integrated approach where the structure retrieval process inherently entails a high quality structural alignment.

A problem related to structural alignment is to identify the spatial sites common to a set of proteins that are known to have a similar function. In Chapter 5 we present a method to discover such spatial neighborhoods in proteins using the features extracted for a residue-interaction based neighborhood.

1.3 Contributions

In this thesis, we have developed methods and tools for search and analysis of the protein sequence and structure data. Specifically, in Chapter 3:

1. A novel method for deriving amino acid substitution matrices from 4-body contact propensities of amino-acids in 3D protein structures is developed.

2. The resulting substitution matrices are shown to provide comparable alignment accuracy to the matrices that were specifically designed for sequence alignment. This demonstrates the importance and ability of the residue interactions in capturing the evolutionary selective pressures.

[2] This is useful for comparing the alignment quality of different proteins to a single query protein. If a single pairwise structural alignment is being performed, then L_{target} is taken to be the length of the shorter protein.

[3] When L_{target} is smaller than 15, d_0 is fixed to be 0.5Å

3. The new matrices are based on different principles than previous matrices, and are useful in the applications where multiple scoring multiple matrices are needed, or in applications where the main feature of interest for the amino acids is their contact potentials (e.g., contact-based empirical potentials used in protein folding).

4. A subset of the matrices satisfy the *metric* properties, and are especially useful in sequence and structure indexing applications (which are targeted in Chapters 4 and 6). These metric matrices yield better accuracy than previous metric matrices which are based on evolutionary arguments or on conversion from non-metric matrices.

In Chapter 4:

1. A novel method of mapping the protein sequences to a vector space based on a metric-preserving, landmark-guided embedding approach is introduced.

2. A detailed analysis of the dependence of the accuracy of the sequence embedding on the various parameters involved is presented.

3. The approximate representation of the sequences in the vector domain achieves several orders of magnitude speed-up in similarity search when compared to the exact representation, while maintaining comparable accuracy.

In Chapter 5:

1. A new framework for automated discovery of family-specific local sites and the features associated with these sites is proposed.

2. The success of the proposed approach is demonstrated on a case study, and on a challenging classification experiment.

3. The developed method is provided as an extensible software freely available for academic research.

And finally, in Chapter 6:

1. A novel, integrated approach to both search and alignment of protein structures is proposed. Whereas previous research separates the retrieval and alignment problems, this is the first time that these problems are effectively solved together.

2. The proposed approach is shown to achieve comparable or better performance than the popular structure search and alignment tools in pairwise structure alignment, similarity search, and protein classification tasks.

3. On case studies and on large-scale experiments, the effectiveness of the method in retrieving related protein structures, producing high quality structure alignments and identifying cross-fold similarities are demonstrated.

4. The implementation of the approach is made available for use as a publicly accessible web-service.

1.4 Structure of the Thesis

In Chapter 2, we give a brief overview of the molecular biology pertinent to protein sequence and structure data. The Central Dogma of the molecular biology is described. The structure and properties of the DNA, RNA, and proteins and the transfer of information between these biopolymers are discussed. Following the biological background, the current state of the art on comparison and search of the protein sequence and structure data is surveyed.

Chapters 3 and 4 focus on sequence alignment and search, whereas Chapters 5 and 6 focus on structural alignment, search, and motif discovery. Chapter 3 derives a set of amino-acid substitution matrices from residue contact profiles and evaluates the accuracy of these matrices in sequence alignment tasks. A subset of these matrices are utilized later in Chapters 4 and 6.

Chapter 4 seeks to reduce the sequence information to a vector representation using landmark-guided embedding. The embedded sequences are then indexed using a spatial access method for fast retrieval. The developed approach is proposed mainly for the search of similar *k-mers* in the sequence database, which forms the first step in similarity search of the whole protein.

Chapter 5 is concerned about extracting common recurrent local structural sites in a family of proteins. The local structural sites are characterized by means of geometrical and biochemical features. The set of sites common to a family of proteins are shown to be able to successfully discriminate family proteins from other proteins that do not have these sites.

In Chapter 6, we present an integrated approach to protein structure comparison and database

search. The approach is based on representing each residue by its contact environment, and indexing these environments using distance-based indexing for fast retrieval. The metric substitution matrices developed in Chapter 3 are used to ensure correctness of the distance function. The environment hits for a query protein structure are retrieved from the database, and extended to high scoring segment pairs (HSPs), which are then used directly for structural superposition. The accuracy and efficiency of this approach is demonstrated on large protein datasets, and on several case studies.

Finally, In Chapter 7, we discuss the future research directions for each of the studies presented in Chapters 3–6. We remark that each of these studies form the subject matter of individual peer-reviewed publications. We have left each chapter self-contained with its own introduction, background work, and methodology; such that the chapters can also be read out of order, if desired. Consequently, there is some overlap and redundancy in the introduction and presentation of the methods. We further note that the sequence to structure ordering of the chapters was not followed in the actual timeline of this thesis study. Particularly, the local structural motif mining study presented in Chapter 5 was investigated before the others.

CHAPTER 2

BACKGROUND AND RELATED WORK

2.1 Molecular Biology

In this section, we provide a brief overview of the molecular biology pertinent to the protein sequences and structures as they are studied in this thesis. We remark that the biological information reviewed here contains some generalizations for which there are known exceptions. Please refer to [1] for more detailed and biologically oriented information, and to [70] for an overview of molecular biology geared toward computer scientists.

The *Central Dogma* of molecular biology [32] describes the transfer of sequential information between biopolymers, and explains how a strand of DNA corresponds to the amino acid sequence of a protein (Figure 2.1). Below, we first describe each of the biopolymers involved in the Central Dogma, and then describe each of its information transfer steps.

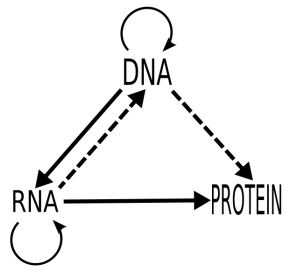


Figure 2.1: The Central Dogma of molecular biology. DNA codes for the production of RNA while the RNA codes for the production of RNA. RNA replication, RNA to DNA reverse transcription, and DNA to protein direct translation are special transfers known to occur, but only under specific conditions, such as in case of some viruses or in-vitro.

2.1.1 DNA

DNA (deoxyribonucleic acid) is composed of a sequence of four types of nucleotide bases: Adenine (A), Thymine (T), Cytosine (C), and Guanine (G). DNA is usually found in double stranded form in the cells, where two DNA strands form a ladder-like *double helix* (Figure 2.2). The two strands are complements of each other in that an Adenine base on one strand is matched with a Thymine on the other strand; similarly, a Cytosine is matched with a Guanine. These base pairs are held together with hydrogen bonds. The covalent bonding of the individual bases on the DNA strands induce a directionality, going from the 5' end (*beginning*) to the 3' end (*end*). The two strands of the double helix are in opposite directions (*anti-parallel*).

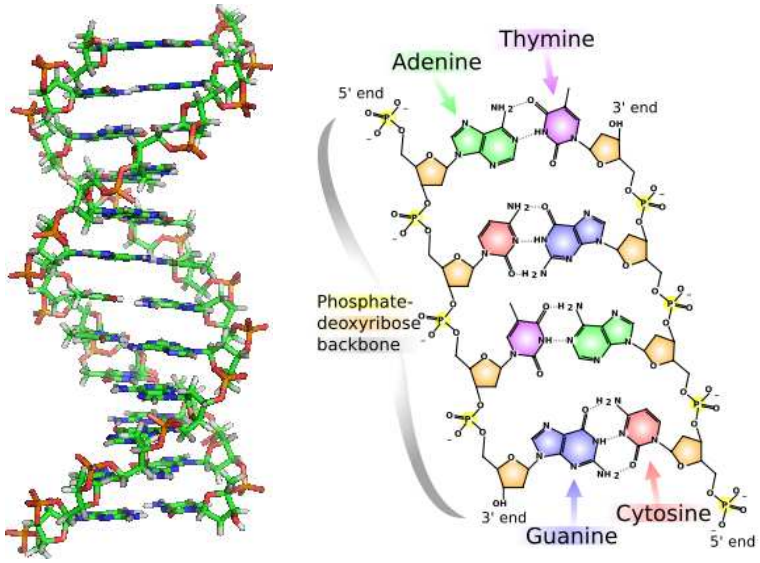


Figure 2.2: The DNA double helix (left) and the chemical structure of DNA (right). Hydrogen bonds between A-T and G-C are shown as dotted lines. (source: [170])

2.1.2 RNA

RNA (ribonucleic acid) is also composed of the four types of nucleotide bases like DNA, except for that the Uracil (U) nucleotide is used in place of the Thymine nucleotide. Unlike DNA, RNA exists as a single stranded molecule. However, sections of RNA can form com-

plex structures (including double-helix) guided by base complementation (between A and U, or G and C).

2.1.3 Proteins

Proteins are formed by polymerization of amino acid residues (Figure 2.3). Each of the 20 standard amino acids (Table 2.1) contain an amino group (NH_2), a carboxy group (CO_2H), and a variable side group (R); all of which are covalently attached to the central alpha carbon atom (C_α). The dihedral angles around the alpha carbon atom in the peptide bond determine the structure of the protein backbone (Figure 2.4).

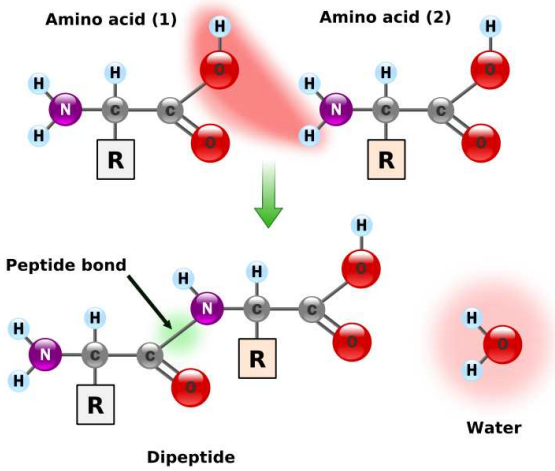


Figure 2.3: Formation of a peptide bond through condensation of two amino acids. The "R" groups on each amino acid represent the variable side-chains. (source: [169])

The side chains (R) determine the differences in the structural and biochemical properties of amino acids. The amino acids can be classified based on these properties; Table 2.2 shows a sample classification. It must be noted that some of the properties governing the classification display a continuum; and there are multitude of amino acid property scales [78] that quantify these properties, such as hydrophobicity, size, charge, and secondary structure preference of the amino acids.

The primary sequence (also denoted as simply *the sequence*) of a protein is simply the linear

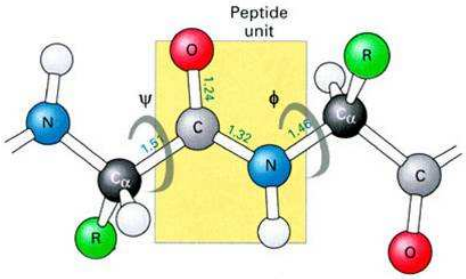


Figure 2.4: The ϕ and ψ dihedral angles of the backbone determine the tertiary structure of the protein. (source: [163])

sequence of its amino acid residues, without any regard to the three dimensional configuration of the protein. The protein sequence is conventionally written in N-terminal to C-terminal order.

The *secondary structure* is the general three dimensional configuration of the local segments, and is for the most part determined by hydrogen bonding between the residues. The common types of secondary structures are *alpha helices*, *beta sheets*, *turns* and *loops*. Unlike alpha helices and beta sheets, the turns and loops are more irregular structures that serve as connector regions between the helical and sheet regions.

Alpha helices are formed by a pattern of hydrogen bonding between the backbone carbonyl oxygen (O_i) of a residue and the hydrogen of the amino group of the fourth following residue (Figure 2.5). This bonding pattern causes a helical formation with 3.6 residues per turn.

The beta sheets are formed where the backbone adopts an "extended" conformation and hydrogen bonds are formed between the carbonyl oxygen and amino groups of two or more adjacent beta strands (Figure 2.6). Based on the directionality of the adjacent strands, the beta sheet is said to be *parallel* or *anti-parallel*.

The *tertiary structure* (also referred to as *the structure*) of the protein is its the three dimensional structure, as defined by its atomic coordinates. The tertiary structures of proteins can be categorized into *folds*, which are usually composed of a well-defined set of secondary structure elements. Figure 2.7 shows two examples of folds: the flavodoxin fold which is composed of helices and sheets, and the globin fold which is composed of only helices.

Table 2.1: The 20 standard amino acids.

Letter code	Abbreviation	Full name
A	Ala	Alanine
C	Cys	Cysteine
D	Asp	Aspartic acid (Aspartate)
E	Glu	Glutamic acid (Glutamate)
F	Phe	Phenylalanine
G	Gly	Glycine
H	His	Histidine
I	Ile	Isoleucine
K	Lys	Lysine
L	Leu	Leucine
M	Met	Methionine
N	Asn	Asparagine
P	Pro	Proline
Q	Gln	Glutamine
R	Arg	Arginine
S	Ser	Serine
T	Thr	Threonine
V	Val	Valine
W	Trp	Tryptophan
Y	Tyr	Tyrosine

2.1.3.1 Protein Sequencing

Sequencing is the process of extracting sequence information from biopolymers. The two direct methods of protein sequencing are the Edman degradation [39] and mass spectrometry [145]. In Edman degradation, the protein is adsorbed onto a solid surface, and a single amino acid is cleaved from the N-terminal by a chemical reagent. The single amino acid is then washed off and identified by chromatography. The cleave-identify cycle is repeated for the rest of the protein, effectively discovering an ordered amino acid composition of the protein.

In mass spectroscopy, a protein is digested into short peptides, which are passed through a high pressure liquid chromatography column. At the end of this column, the solution is sprayed through a narrow, high positive charged nozzle. The charge on the resulting droplets cause them to fragment until only single ions remain. The mass spectrum of these fragments are analyzed and the original peptides are reassembled through a computationally intensive process. Indirect sequencing of the proteins from respective genetic sequences (which can be

Table 2.2: Classification of the 20 standard amino acids. Note that the classification is not clear-cut, and some amino acids belong in more than one category.

category	amino acids
Aliphatic/hydrophobic	Ala, Leu, Ile, Val
Polar	Asn, Gln
Alcoholic	Ser, Thr, (Tyr)
Sulfur-containing	Met, Cys
Aromatic	Phe, Tyr, Trp, (His)
Charged	Arg, Lys, Asp, Glu, (His)
Special	Gly (no R), Pro (cyclic, imino-acid)

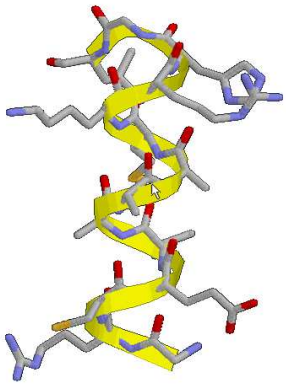


Figure 2.5: Alpha helix is formed by a series of hydrogen bonding between every i^{th} and $i + 4^{th}$ residues. The yellow ribbon is drawn along the backbone to illustrate the helical structure. Side chains are shown in the ball and stick scheme. (source: [163])

DNA or RNA molecules) is also possible by inferring the amino acid composition as coded by the *genetic code*.

2.1.3.2 Protein Structure Determination

The tertiary, three dimensional structure of the proteins is identified mainly by X-ray crystallography [80] or NMR (nuclear magnetic resonance) spectroscopy [106]. In X-ray crystallography, the arrangement of atoms within a crystal is determined from the diffraction pattern of an X-ray beam through the crystal.

The NMR spectroscopy is based on the fact that active nuclei (such as ^1H or ^{13}C) absorb at

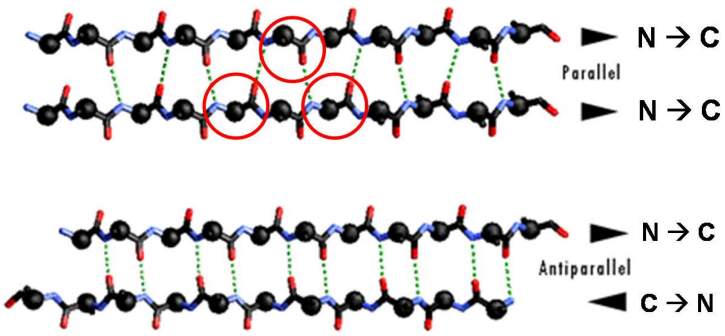


Figure 2.6: Beta sheets are formed by hydrogen bonding between beta strands. (source: [163])

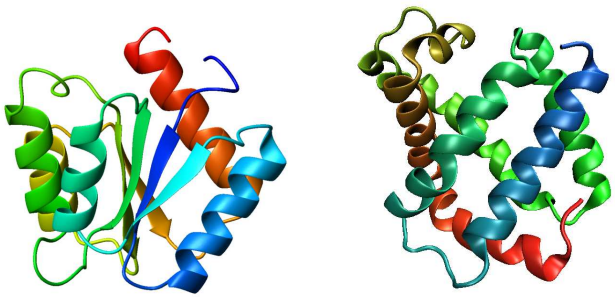


Figure 2.7: The flavodoxin fold (left) and the globin fold (right). (source: [171, 172])

a specific frequency when placed in a magnetic field. Depending on their local chemical environment, different protons in a molecule resonate at slightly different frequencies (*chemical shift*). The 3D structure information is derived from an analysis of the resonant frequency, chemical shift, energy of the absorption, and the intensity of the signal.

2.1.3.3 Protein Structure Databases

Protein Data Bank (PDB) [13] is a publicly available repository of the protein structures. The structures in the PDB have 4-letter identifiers, such as "1ne3", which contains the structure data for the ribosomal protein S28. When the deposited protein structure is composed of more than one polypeptide chain, a chain identifier following the PDB code (separated by a colon ":") is used to refer to each chain, such as "1ne3:A". The PDB database entries are generally made available in a plain text file format.

There are a number of protein structure classification schemes that classify the proteins based on structure, function, and sequence similarity. In this thesis, we use the SCOP [108] classification, which is maintained in a semi-automated fashion by human experts. The SCOP hierarchy categorizes the proteins in four levels: class, fold, superfamily, and family. Note that, a single PDB file may contain multiple *domains*, which are categorized separately. Although dividing a protein into multiple domains is not always straightforward or accurate, a domain is defined as a segment of the protein that can fold or function independently. SCOP domain identifiers are 7-letter codes, such as "d1ne3a_", where the first letter "d" specifies that this is a protein, the 2nd to 5th letters are the 4-letter PDB identifier, the 6th letter is the chain identifier (an underscore "_" may be used if the PDB file contains only a single chain), and the 7th letter is the domain identifier (an underscore "_" may be used if the chain contains only a single domain). The SCOP families are denoted in the "class.fold.superfamily.family" notation, such as the "b.40.4.5" family, which belongs to the all beta proteins class ("b"), the OB-fold ("b.40"), the Nucleic acid-binding superfamily ("b.40.4"), and the Cold-shock DNA-binding domain-like family ("b.40.4.5").

This completes our discussion of the protein sequence and structure and other biopolymers involved in the Central Dogma. We now turn to the process of the information transfer between these biopolymers.

2.1.4 Information transfer in the Central Dogma

DNA replication. The genetic information is transmitted from parents to progeny through a faithful replication of DNA. The DNA replication is carried out by a complex set of proteins that unwind the double strand, and synthesize the complementary strands using each of the original strands as templates. Proofreading and error checking mechanisms exist to ensure that the resulting double-stranded DNA molecules are near-identical replicas of the original DNA.

Transcription (or gene expression) is the synthesis of messenger RNA (mRNA) from a section of the DNA (Figure 2.8). This coding section of the DNA is defined to be a *gene*. The mRNA is generated using one of the strands of the double-stranded DNA as the *template*. The template strand is also called the *anti-sense* strand, and the other strand of the DNA that is not

serving as a template is called the *sense* strand. The resulting mRNA is the complement of the template strand, and is thus identical to the sense strand (except for the T to U nucleotide replacement).

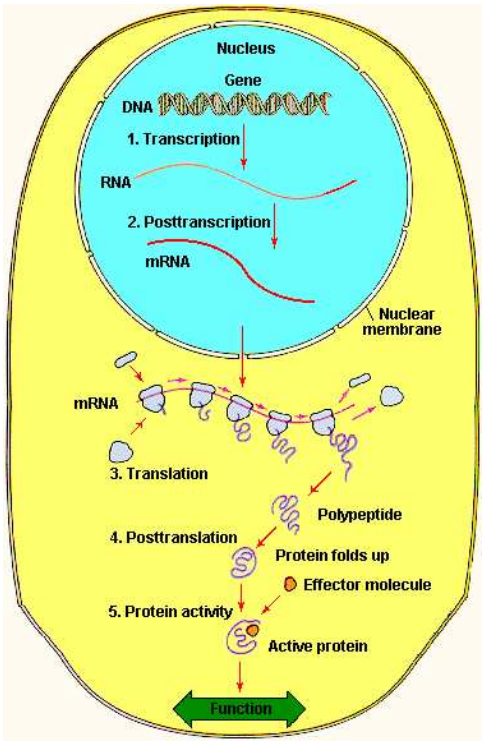


Figure 2.8: The transcription, translation and other intermediary steps leading from DNA to RNA to protein to biological function. The figure depicts the process as it happens in a eukaryotic cell. In prokaryotes, there is no nucleus, and the translation can occur simultaneously as the gene is being transcribed. (source: [161])

Translation is the process by which the information contained in the messenger RNA is used to synthesize polypeptides using in the ribosome machinery. Each of the 20 amino acids is specified by three nucleotides of the mRNA, called *codons* (Figure 2.9). Translation from the mRNA starts with an initiation codon (AUG), which also codes for methionine, and continues until one of the stop codons (UAG, UGA, or UAA) is found on the mRNA.

The *genetic code* specifies which amino acids are encoded by each codon. The genetic code has some redundancy (for example, both GAA and GAG code for glutamic acid). The accuracy of the genetic code is achieved by base complementarity between mRNA and the transfer

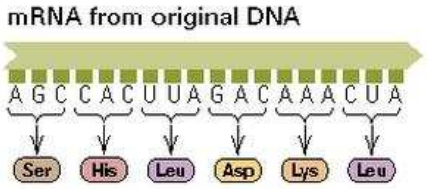


Figure 2.9: Triplets of mRNA nucleotides (codons) code for individual amino acids. (source: [161])

RNAs (tRNA), which are the carriers of the amino acids into the translation machinery. Once a polypeptide is synthesized, it may undergo post-translational modifications, such as glycosylation, and folds into its native structure.

Table 2.3: The genetic code. Each amino acid is specified by a particular combination of three nucleotides, called codons. Some amino acids are encoded by more than one codon. (source: [161])

FIRST POSITION	SECOND POSITION: U	SECOND POSITION: C	SECOND POSITION: A	SECOND POSITION: G	THIRD POSITION
U	phenyl-alanine	serine	tyrosine	cysteine	U
					C
	leucine		stop	stop	A
			stop	tryptophan	G
C	leucine	proline	histidine	arginine	U
					C
			glutamine		A
					G
A	isoleucine	threonine	asparagine	serine	U
					C
	* methionine		lysine	arginine	A
					G
G	valine	alanine	aspartic acid	glycine	U
					C
			glutamic acid		A
					G

* and start

This completes our overview of the molecular biology pertinent to the analysis of protein sequence and structure data. In the following sections, we review the related work on comparison, alignment, and search of protein sequence and structure data.

2.2 Sequence Alignment and Similarity Search

As discussed in Section 1.2.1, the optimal pairwise alignment of two protein sequences can be found using dynamic programming. In the context of searching for similar subsequences, the dynamic programming solution to finding subsequences in s that are within a certain distance r to the query sequence q runs in $O(mn)$ time and space, where m and n stand for the lengths of q and s [110, 77]. For large datasets, the basic dynamic programming approach becomes infeasible. The time and space requirements have been relaxed through heuristics in finding matches while sequentially scanning the dataset, or through preprocessing the dataset to create an appropriate index structure. Below, we briefly describe the sequential scan methods that use such heuristics, and then present a survey of the indexing methods.

2.2.1 Sequential Scan Methods

BLAST [2, 3] has been the favorite tool for biological homology searching since 1990. It uses a heuristic that assumes the presence of short exact matches between homologous sequences, and uses this assumption to quickly filter the database to identify the candidate sequences. The choice of the length of the pre-generated exact matches presents a tradeoff between the sensitivity of the search and time and memory requirements. BLAST first generates from the query, all the subsequences of a specified length k (typically 3 for protein sequences, and 11 for DNA sequences). Once these subsequences are generated, BLAST searches the database for exact matches to these substrings. The matches are then extended in both directions until the score falls below some threshold. For an alphabet of size σ, there are σ^k possible substrings of length k, which are called *probes*. BLAST keeps a pointer to the starting points of each of these substrings in the database to speed up the filtering phase. Increasing k increases the memory requirements, and decreases the sensitivity of the search, whereas decreasing k yields more false candidates from the filtering phase and slows down the computation.

Several improvements over BLAST have been proposed. Pattern Hunter [96] uses non-

consecutive symbols to detect the replacements in the sequence better. SENSEI [147] removes simple repeats and compactly encodes scoring tables for short segments to obtain better performance. The *Piers* method [25], uses randomly picked seeds to guide an inexact matching between short query segments, and achieves faster response without significant loss in sensitivity. There have also been attempts to apply suffix-trees and suffix arrays, which are popular structures for exact matching problems, to the task of similarity searching [69, 109, 23, 150, 101]. However, these methods generally demand large amounts of memory and disk usage, and are effective only when the number of mismatches is low.

2.2.2 Index-based Methods

The methods that preprocess the database to build a similarity-searchable index structure can be grouped into two broad categories: *distance-based indexing* and *spatial indexing*.

In distance-based indexing, the database sequences (or subsequences) are partitioned based on comparison with each other. A representative sequence (called a *pivot* or a *vantage point*) is chosen for each partition, and a tree is built by iterating the partitioning step. For a given query sequence, the tree is then traversed based on the distance to the pivot sequences, and the query is compared with each candidate sequence in the candidate partitions that the traversal terminates at. Employing multiple vantage point tree on a metric search space obtained from a metric model of amino acid substitution model [175] is shown to achieve better scalability than BLAST while maintaining comparable search accuracy [174]. A survey of distance-based indexing methods can be found in [151].

The spatial (vector space) indexing methods work in two steps: mapping the sequences into an appropriate feature space, and indexing the transformed sequences in this new feature space. For indexing, one can employ fine-tuned Spatial Access Methods, like the R^*-trees [10] or the z-ordering [117]. The challenge in indexing the sequence databases is, in general, mapping the sequences to an indexable vector space. The distance function defined in the new feature space has to guarantee to underestimate the distance defined between the sequences; otherwise, false-drops would occur during a querying process. Moreover, the new distance function has to be a close approximation to the original distance to obtain efficient filtering of irrelevant sequences.

In the MRS method [77], subsequences are generated from a database string by sliding a window of length $w = 2^i$ over the string. The *frequency vector* of a sequence window is defined as $f(s) = [n_1, n_2, ..., n_\sigma]$, where each n_i is the number of occurrences of the i^{th} alphabet character, α_i. A *frequency distance* is also defined that always underestimates the edit distance. Using this frequency transformation, each sequence is represented by its trail in σ dimensional space, formed by the locations of the mappings of the constituent windows. The trail is then subdivided into *Minimum Bounding Rectangles* (MBRs), which are indexed using R^*-trees. Moreover, in MRS, a wavelet-based transformation, and a corresponding *wavelet distance* are used to refine the lower bound distance, and a multi-resolution index structure is built to handle variable length queries.

In [120], the frequency and wavelet transformations use k-tuples rather than individual alphabet symbols. And in [148], a compressed multi-resolution index structure (CoMRI) is proposed, saving storage space, and resulting in faster searches.

Although these indexing methods do not give false-drops, the approximation of edit distance is not sufficiently tight, causing many false hits to be generated. Therefore, the methods are feasible only for near-exact matches, which is applicable in specific tasks, as in searching overlapping fragments in shotgun sequencing projects, determining the locations of ESTs in the genome, or cross-species comparisons between very closely related genomes. Searching for homologous proteins, on the other hand, requires more sensitive methods that can detect distant homologs that have sequence identity as low as 15%.

The spatial indexing methods we have surveyed focus primarily on DNA data where the alphabet size is small ($\sigma = 4$). The memory requirements of these methods grow in $O(\sigma^k)$ where k is the tuple size used in the transformation, and it remains to be seen if they are scalable to larger alphabet size ($\sigma = 20$) of the protein databases. A bigger problem these methods suffer is that they ignore the differences in inter-symbol costs. In the DNA databases, the differences in the costs of replacing one nucleotide type with another may be ignorable. A unit-cost edit distance measure can be used without any loss, especially considering that the distance measure in the transformed domain significantly underestimates the edit distance. On the other hand, the differences in costs of replacing amino-acids is not negligible (for example, in PAM250 amino-acid substitution matrix [33], the costs range from 0 to 25); thus, a unit-cost edit distance would be insufficient in modeling the distance between protein

sequences.

In order to overcome the the problems present in the currently available methods, Chapter 3 develops biologically accurate metric substitution matrices which lend themselves to use in more sensitive indexing of the protein sequences. Chapter 4 uses these matrices for alignment; and based on the edit distances to selected landmark sequences, embeds the sequences into a vector domain for efficient indexing.

2.3 Structure Alignment and Similarity Search

It is generally accepted that protein structures are better conserved through the evolution than sequences. Due to the biochemical similarities among the amino acids, a greater flexibility is present in the primary sequences of the proteins that share same function. On the other hand, proteins exert their function through their structure, and the mutations that cause structural changes often hamper the biological role of the protein. In recognition of the importance of structural information, a number of centers have been established in an effort to achieve high-throughput determination of protein structures [115]. Consequently, the number of available protein structures has been growing rapidly. As of April 2008, there are more than 50,000 structures deposited in the Protein Data Bank (PDB) [13]. Besides the traditional bioinformatics which mainly focuses on the sequence data, we are now in great need for tools and methods to index and retrieve structural patterns.

Due to the difficulty of describing and processing structural patterns, current work in management and analysis of the structure databases is still in its infancy stage. Most of the previous efforts focus on construction of a hierarchical classification for protein folds and families [108, 116, 66, 51]. When a new protein structure becomes available, its family membership is identified through exhaustive comparison with a representative of each family. Hence, an accurate and efficient retrieval scheme is still missing. It must be noted that the global fold similarities alone are not sufficient, and it is important to identify similarities in spatially localized active or binding sites. This forms the goal of Chapter 5, where we develop a method to identify local sites shared by a family of proteins. In Chapter 6, we propose a new integrated and effective approach that provides the ability to perform both search and comparison of the database proteins.

2.3.1 Pairwise Structure Alignment

Pairwise structure alignment seeks to find the correspondences of the residues between two proteins, and a translation/rotation matrix that superimposes the protein structures while minimizing the distance between corresponding residues, as measured by an error function (usually RMSD is used). While for a given set of correspondences the optimal superposition can be calculated very efficiently (linear in the length of the sequences, [75]), solving the correspondence and superposition problems simultaneously has been shown to be NP-complete [90]. For this reason, several heuristic approaches have been developed. These approaches often reduce the protein structures to some coordinate-independent space, so that they can be compared without requiring a detailed superposition.

DALI [65] represents each protein structure by its *distance matrix*, which is an NxN matrix listing the Euclidean distances between all pairs of residues in a protein of length N. Similar submatrices of size 6x6 are then searched between the distance matrices of two proteins. Submatrix matches are then reassembled using Monte Carlo simulation with the objective of maximizing the structural similarity of the reassembled alignment. DALI is used in an all-against-all comparison of proteins to produce the FSSP (Families of Structurally Similar Proteins) database [66].

The combinatorial extension (CE) [141] and MAtching Molecular Models Obtained from Theory (MAMMOTH) [119] methods also break each structure into short fragments. CE originally used the structural superposition and the inter-residue distances to measure the similarity of the fragments but has since been extended to include local environment properties such as secondary structure states, solvent exposure, and hydrogen bonding patterns. The matching pair of fragments between two proteins form the aligned fragment pairs (AFPs). The AFPs are assembled starting from the highest scoring AFP pair, and extending to the next highest scoring AFP that meets a given distance criteria, restricting the alignment to low gap sizes. MAMMOTH defines the similarity between fragments using a unit-vector RMS method [79], and calculates the final alignment based on these scores, using a hybrid local-global dynamic programming.

Unlike the fragment-based approaches DALI, CE, and MAMMOTH; SSAP [154] considers the residues individually and compares them using differences in the inter-residue distance

vectors between the residue under consideration and its nearest non-contiguous neighbors. Dynamic programming is applied to each pair of residue environments from two proteins to obtain a similarity score of their inter-residue distances. The scores obtained for individual residue pairs are then used in a second level of dynamic programming to obtain the final alignment path. The *double dynamic programming* approach used in SSAP is similar to the extension of the residue environment hits described in Chapter 6.

2.3.2 Structure Retrieval

While the pairwise structure alignment methods provide a comparison of two protein structures within seconds, an exhaustive scan of a large structure database using pairwise alignments becomes impractical. For this reason, a filter-and-refine approach is usually employed: perform a quick search of the database using coarse-level features to identify candidate structures, and apply pairwise structural alignment to only the top-scoring candidates.

The approaches to developing indexing methods for quick identification of similar structures can best be described in terms of the representation being used to capture the structural information, and the indexing method used on this representation for quick retrieval. ProGreSS [15] transforms the protein structure into a feature vector space of its curvature and torsion angles and of its sequence information. The space is partitioned into an equally spaced grid, and minimum bounding rectangles (MBRs) are extracted for each protein. The MBRs that lie close to the spline of a query protein are identified and a voting scheme is used to rank the protein hits.

[177] use distances and angles among the secondary structure elements (SSEs) and utilizes a hashing technique to identify similar structural cores composed of triples of SSEs in two proteins. 3D-Hit [122] generates clusters of short protein fragments based on the RMSD error between the fragments. The database proteins from a large database are then assigned to these clusters. A query protein is compared with each cluster and the database proteins from the clusters that are highly similar to the query protein are returned for further structural comparison.

[24] represents the secondary structure as a vector and extracts several features, such as SSE type, vector angle and center; and performs indexing on this vector representation using R*-

Tree. [22] applies a suffix tree to index the proteins based on the dihedral angles of the peptide bonds. The suffix tree approach favors exact matching of backbone segments that share highly similar dihedral angles, and is unable to provide flexibility in matching. [29] and [104] utilize geometric hashing to identify the triplets of atoms that share similar inter-residue distances with the query residues to identify all possible residue correspondences. Note that the geometric hashing technique usually identifies a huge number of false positives due to the lack of selectivity provided by triplets of atoms; the physical and geometrical constraints cause unrelated atoms to have similar inter-residue distances. For this reason, [104] implement the candidate hit evaluation in a massively parallel environment (more than 130,000 processors). [5] partitions distance matrix into contact regions of the secondary structure elements and uses geometric hashing to index the distance and angle between SSEs. Although the complexity is reduced by considering SSEs instead of the individual atoms or residues, one still gets many unrelated substructures sharing the same hash value, causing many false positives.

There have also been several recent attempts to reduce the structural information to a sequential representation so that sequence search tools can be used directly. Protein block expert (PBE) [160] uses 16 structural motifs as a structural alphabet, whereas 3D-BLAST [159] partitions the (κ, α) dihedral angles into a 23-letter alphabet, which is then used to convert the structures into one-dimensional sequences. A sequence alignment tool, such as [2] is then used to retrieve similar proteins. Note that, even though both of these methods provide good efficiency, they do not capture the structural topology of the protein. Furthermore, dissimilar structures in three-dimensional space may correspond to identical sequence representation under the given alphabets, causing high-scoring false hits.

In Chapter 6, we capture residue interactions and utilize metric indexing for efficient retrieval of the residue environments that share similar interactions, topology, secondary structure, and amino acid type. The residue environment hits are then extended to obtain high scoring segments (HSPs). Our approach achieves better accuracy than other indexing methods while requiring comparable search time. Furthermore, the generated HSPs already align structurally compatible residues, and are directly used in superposition, without having to resort to pairwise structure alignment methods. The structural alignments produced by our approach are of comparable or better quality when compared with the popular pairwise alignment tools.

2.3.3 Structural Motifs

While the structure retrieval and alignment methods are generally concerned with the similarity of the compared proteins as a whole, we remark that many proteins have a multi-functional nature, and the global similarities alone are not sufficient to identify functional similarities existing in distinct local domains. Inevitably, *local structural motifs* provide many functional clues for annotation of protein structres. Following this direction, the PROCAT system [166] builds a database of 3D motifs. However the motifs in the PROCAT database are manually annotated and only specific types of enzyme active sites and catalytic residues in enzymes are documented. Recently, much attention has been placed on the automatic discovery of more general motifs.

The 3D motifs have been modeled as graphs [146, 68], spatial patterns [73, 82], constraint-based templates [165, 8], and general purpose feature vectors [6, 94]. Search algorithms for identifying which of the motifs in the particular motif library are present in a new protein are based on graph theoretical algorithms [146, 68, 73, 8, 82], geometric hashing [165, 140] and others [142].

The motif discovery is the problem of finding common, recurrent local structures in space based on the specific model. In [68], a protein structure is modeled as a (weighted) graph and a motif is defined as a frequently occurring sub-graph in a set of graphs (structures). The number of potential subgraphs defined this way is unfortunately exponential, producing a huge search space. Some methods therefore consider only those local patterns centered at each residue or at some manually-chosen positions as potential motifs [73, 94], possibly missing motifs not centered around such positions. Furthermore, these methods usually miss relatively rare and novel motifs. There are also work on surveying enzyme binding sites [36, 17] based on geometric methods [37, 93]. In general, an automatic method that produces a concise yet complete coverage of the motif space is still missing.

The LFM-Pro method described in Chapter 5 samples the protein 3D space based on critical points (local maxima, minima, and saddle points) of a particular function. These local sites are enriched with the geometrical and biochemical features of the environment they contain. A motif is then defined as a local site common to a set of proteins that share the same function

or family classification. The success of LFM-Pro in capturing discriminative and functionally important local sites is demonstrated on a classification experiment and on case studies.

CHAPTER 3

AMINO ACID SUBSTITUTION MATRICES BASED ON 4-BODY DELAUNAY CONTACT PROFILES

3.1 Chapter Overview

[1]Sequence similarity search of proteins is one of the basic and most common steps followed in bioinformatics research and is used in making evolutionary, structural, and functional inferences. The quality of search and alignment of proteins depends crucially on the underlying amino-acid substitution matrix. We present a method for deriving amino acid substitution matrices from 4-body contact propensities of amino-acids in protein 3D structures. Unlike current popular methods, the approach does not rely on mutational analysis, evolutionary arguments, or alignment of protein sequences and structures. The alignment accuracy of derived matrices is illustrated using BAliBASE reference alignment set and found comparable to that of popular matrices from literature. Notably, the derived matrices perform the best among the metric matrices. The resulting matrices would find use especially in development of empirical potential energy functions and in distance based sequence indexing.

Supplementary Material: The substitution matrices are available and from `http://www.ceng.metu.edu.tr/~ahmet/bioinfo/distmat`

[1] The approach described in this chapter was published in the *Proc. of the IEEE Intl. Symp. on Bioinformatics & Bioengineering, BIBE-2007* [132]

3.2 Introduction

Alignment of protein sequences has been one of the most widely utilized tools of bioinformatics research [52]. Applications of sequence alignments and comparisons include finding homologous proteins, predicting protein structure or function, and defining the phylogeny of species.

Alignment score is defined as the sum of the individual scores of the aligned residues as looked up from a residue scoring matrix, and is used in database search for similar sequences. Optimal alignment of two sequences can be obtained by dynamic programming algorithm [113, 144]. The rapid increase in the size of protein sequence databases have prompted development of near-optimal heuristic approaches like BLAST [3] and FASTA [121].

The quality and significance of the database search results and sequence alignments depend strongly on the underlying residue scoring matrix and the gap cost function. For computational convenience, affine gap penalty is used in practice [56] and gap opening and gap extension penalties are determined by statistical optimization on a reference alignment set.

The popular scoring matrices are based on log-likelihood of residue substitutions obtained from frequencies of mutations observed in sequence alignment of similar proteins. The initial alignments were constructed either by hand [33], by automated alignments from large sequence databases [53] or by alignment of conserved blocks [62].

Structural superpositions have also served as the basis for alignment of sequences and counting substitutions [72]. Protein structures can be aligned even in the absence of significant sequence similarity. Substitution matrices derived from structural alignments are especially useful in detecting distantly related sequences and similarities that result from convergent evolution.

Other methods of obtaining residue exchangeability include evaluation of engineered mutations either by experimental assay studies [176], or by computational fitness functions such as based on force fields [35]. Physico-chemical properties such as hydrophobicity, volume, and conformational preferences have also been used as basis for similarity measures [57, 114].

In this study, we use the multi-body contact propensities of residues in protein three dimensional structures as the basis for amino-acid similarity. Amino-acids have previously been

found to have non-random multi-body contact preferences [143] and this property has been exploited in development of statistical pseudo-potentials to discriminate native and non-native protein conformations [86]. We use these non-random preferences to derive an amino-acid scoring matrix to be used in protein sequence alignments. We expect that this scoring matrix would be suitable for detecting remote homologs that share structural similarities. Moreover, the unique features of this matrix make it especially useful in development of contact-based empirical potential energy functions and in distance-based indexing of protein sequences.

3.3 Methods

Due to its objective and robust definition and well–defined geometric properties, Delaunay tessellation has been the method of choice for extracting multi-body contacts from protein structures [143]. The protein is modeled by a set of points representing the amino-acids. The region of space around each point that is closer to the enclosed point than any other point defines a Voronoi polyhedron. (See Figure 3.1). Delaunay tessellation is obtained by connecting points that share a Voronoi boundary. In 2D, each triangular area in the Delaunay tessellation defines a set of 3 points that are in contact. In 3D, each tetrahedra gives a set of 4-body contacts.

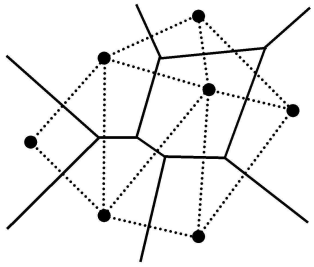


Figure 3.1: Delaunay tessellation (dashed lines) and Voronoi diagram (solid lines) of a set of points in 2D. In 3D, Delaunay tessellation would give space–filling tetrahedra.

There are several ways of representing amino acids of a given protein structure. Here, we use the most commonly used representations: location of alpha Carbon atom (CA), location of beta Carbon (CB), or the centroid ($CENT$) of the side-chain atoms. Glycine lacks a CB atom, so for Glycine, CA is used instead of CB. The Delaunay tessellation is computed using the

Quickhull algorithm [7].

For a given protein structure, the Delaunay tessellation results in a list of amino-acid quadruplets defining the 4-body contacts. We record the frequency of observing an amino acid type in contact with the remaining three amino acids in the quadruplet. This gives us a frequency matrix of size 20 by 8000, where each row stands for an amino acid type, and each column represents different combinations of the remaining three amino acids. We call each row of this matrix the *4-body contact profile* of the corresponding amino acid.

We postulate that the exchangeability of amino acids in three dimensional structures would be reflected in their Delaunay contact profiles. An amino acid substitution can thus be derived from the contact profiles matrix. We have used both the Euclidean distance (EUC) and Pearson's correlation (COR) measures between the rows of the contact profiles matrix in order to quantify the exchangeability of amino-acids. The Euclidean distance is defined as:

$$d_{a,b}^{EUC} = \sqrt{\sum_{i=1}^{8000} (A_i - B_i)^2} \tag{3.1}$$

where $d_{a,b}^{euc}$ is the calculated distance between amino acids a and b and where A_i and B_i are the i^{th} elements of the corresponding rows of the contact profile matrix. Similarly, the correlation distance is defined as:

$$d_{a,b}^{COR} = 1 - \frac{\sum_{i=1}^{8000} (A_i - \mu_A)(B_i - \mu_B)}{\sqrt{\sum_{i=1}^{8000} (A_i - \mu_A)^2 \sum_{i=1}^{8000} (B_i - \mu_B)^2}} \tag{3.2}$$

where μ_A denotes the mean value of the row A of the contact matrix. Each of these distances define a target 20 by 20 amino acid substitution matrix.

3.4 Experiments

PDBselect25 [63] representative dataset, which contains a non-redundant set of PDB structures with less than 25% mutual sequence identity, was used for the derivation of contact profiles and construction of substitution matrices. The downloaded version of PDBselect25 used in this study was compiled in January 2007 and contained 3080 proteins.

Using three types of amino acid representations, and two types of distance measures, a total of six substitution matrices were obtained. The compiled matrices were compared with 15

Table 3.1: Substitution matrices used for comparison.

Matrix name	Short name	Reference
PAM250 sequence alignment of similar proteins	PAM	[33]
BLOSUM30,40,50,62 sequence alignment of conserved blocks in related proteins	B30,40,50,62	[62]
GONNET exhaustive automated sequence alignments	GO	[53]
RISLER structural alignment of related proteins	RI	[127]
JOHNSON structure based sequence comparison	JO	[72]
MIYAZAWA base substitution – protein stability	MJ	[105]
NAOR structural alignment of spatially conserved substructural motifs	NA	[112]
REMOTEHOMO structural alignment of remote homologs	RE	[130]
ANALOGOUS structural alignment of analogous proteins	AN	[130]
COMBINED structural alignment of analogous and remote homologs	CO	[130]
SDM structurally equivalent residues of analogous proteins	SDM	[124]
HSDM structurally equivalent residues of homologous proteins	HSDM	[124]
CA-COR, CB-COR, CENT-COR correlation of Delaunay contact profiles from CA or CB atoms, or side-chain centers	CAC, CBC, CNC	present study
CA-EUC, CB-EUC, CENT-EUC Euclidean distance of Delaunay contact profiles from CA or CB atoms, or side-chain centers	CAE, CBE, CNE	present study
IDENTITY identity matrix	ID	present study

other matrices from the literature (see Table 3.1 for the list of matrices). For completeness, an identity matrix was also included. Comparison and analysis of matrices were performed via principal component analysis and hierarchical clustering. These methods have been noted to be sufficient to highlight the overall relationship between matrices [99].

Figure 3.2 displays a gray-scale depiction of matrix correlations based on sample correlation of their 400 elements. An unweighted average distance (UPGMA) clustering of matrices from these correlations is also derived (Figure 3.3). The type of amino acid representation does not have significant effect on the resulting matrix as can be observed from the high correlations among them. This is due to only a small fraction of the tetrahedrons differing among the tessellations obtained from different amino acid representations.

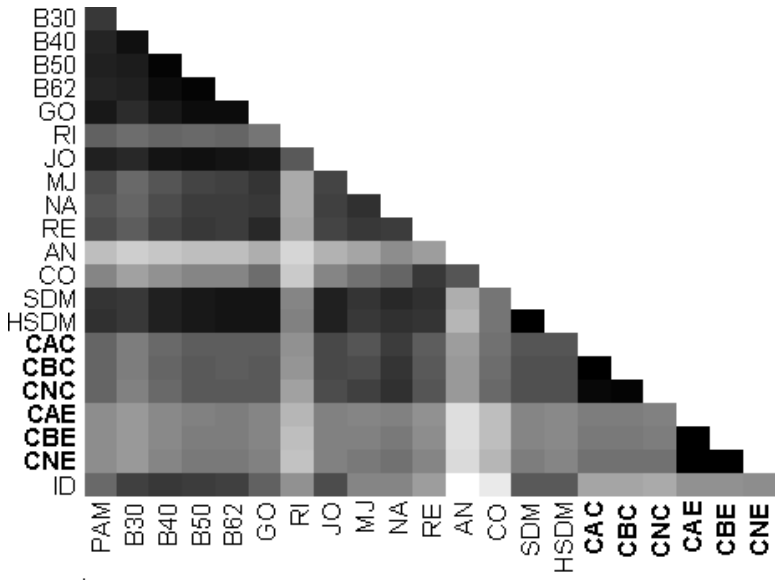


Figure 3.2: Correlation of matrices based on pairwise sample correlation of matrix elements. The higher the correlation between a pair of matrices, the darker the corresponding cell.

On the other hand, the choice of distance measure gives qualitatively different matrices. The Euclidean measure is sensitive to the background frequencies of amino acids in the initial protein structure dataset, and the derived matrices reflect this bias. Whereas, the correlation coefficient gives exchange values normalized for the background frequencies of amino acids.

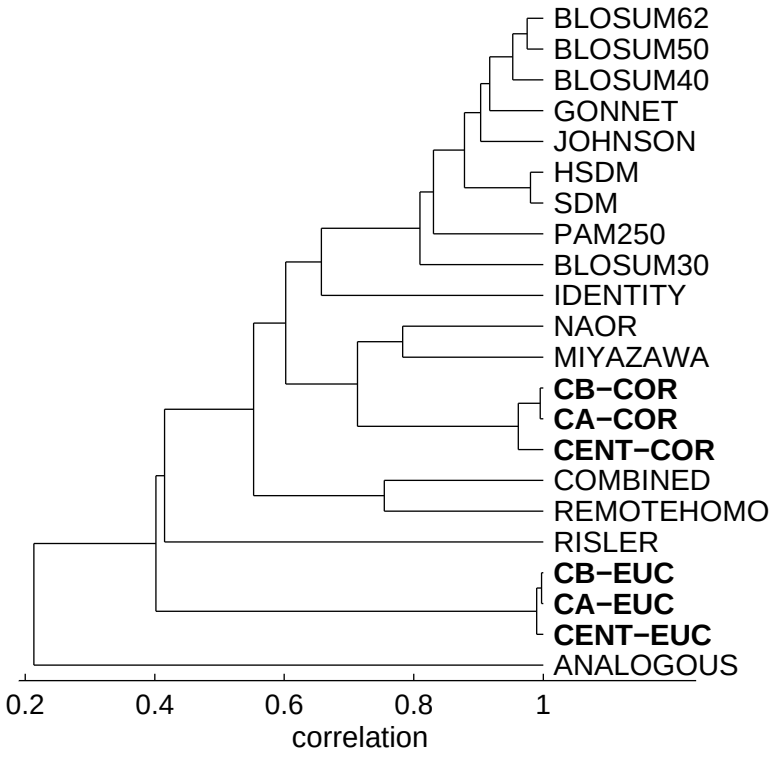


Figure 3.3: UPGMA clustering of matrices based on correlation of matrix elements.

The correlation set of matrices (CA-COR, CB-COR, and CENT-COR) are found to be closely correlated to NAOR [112] substitution matrix with an average correlation coefficient of 0.76. NAOR has been derived from amino acid interchanges observed at spatially, locally conserved regions in globally dissimilar and unrelated proteins. Although Delaunay tetrahedra is a more granular motif, we conjecture that the tetrahedra contacts derived in this study share common overall characteristics with the conserved substructural motifs studied by [112]. Note that, Delaunay tessellations have, in fact, been found useful in discovering locally conserved structural sites [131].

Unlike the matrices that use correlation coefficient as the distance measure, the Euclidean set of matrices (CA-EUC, CB-EUC, and CENT-EUC) did not show significant correlation with any other substitution matrix. We attribute this, again, to the inherent bias of the Euclidean measure to background amino acid frequencies, which is not present in the other matrices.

Analysis of the matrices at the amino acid level can help characterize the physico-chemical

properties underlying the amino acid exchanges. We observed that the exchange values defined in the *correlation* matrices displayed strong relation to the hydrophobicity of the amino acids. The substitution matrix CA-COR is represented as a projection on to the first two principal components (Figure 3.4). The first principal component is essentially a hydrophobicity scale with hydrophobic residues on the left, and hydrophilic and charged residues clustered on the right.

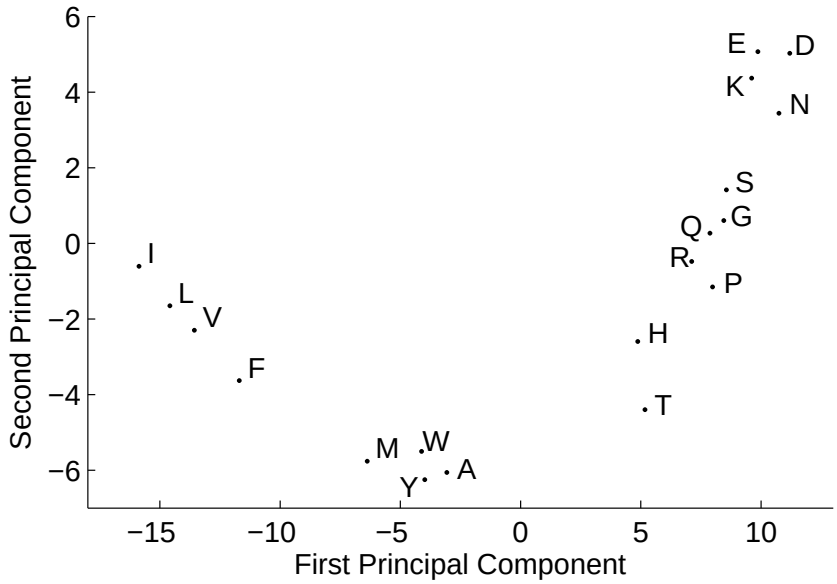


Figure 3.4: Principal component analysis of the matrix CA-COR. The first and second principal components account for the 72.7% and 24.7% of the variation in the matrix values. Cysteine residue with coordinates -18.2,20.2 is omitted from the figure for illustration purposes. The analysis for the other matrices can be found in the Supplementary Material.

The first eigenvector of the CA-COR matrix and the hydrophobicity scale of [41] are indeed highly correlated, with a coefficient of 0.93. The strong correlation with hydrophobicity scales is of no surprise; because protein folding and as a result, the Delaunay contacts are guided by hydrophobic interactions among amino acids residues.

In order to evaluate the sequence alignment performance of the matrices, we used BAliBASE [155, version 3] suite of reference alignments. Pairwise alignments for each multiple alignment were extracted to result in a total of 155,550 pairs. For each substitution matrix, pairwise

alignment of sequences is performed using Gotoh's algorithm [56] with affine gap penalties. The optimal gap penalties were used as found by [124]. For matrices where optimal gap penalties were not available, we used parameters interpolated from those of PAM250 matrix.

Table 3.2: Sequence alignment accuracy of matrices based on BaliBASE reference alignments. The BaliBASE subsets are ordered in increasing homology.

BAliBASE subset:	RV11	RV12	RV20	RV30	RV40	RV50	All
number of alignments:	760	2,275	47,497	74,836	22,855	7,327	155,550
substitution matrix	% correctly aligned residue pairs						
SDM	40.1	74.3	76.8	58.4	60.4	53.1	63.0
HSDM	40.3	74.3	76.8	59.0	61.6	55.2	62.6
GONNET	39.7	73.0	76.2	57.3	60.0	52.3	61.8
BLOSUM30	37.1	71.6	75.1	55.8	58.8	51.0	60.6
PAM250	36.6	71.8	75.3	55.2	58.9	50.2	60.3
MIYAZAWA	33.2	68.7	73.8	53.0	55.8	45.3	58.9
RISLER	33.9	69.2	73.6	52.7	56.0	45.5	58.7
NAOR	35.6	67.5	73.0	53.7	55.7	46.0	58.6
CA-COR	33.2	67.5	73.3	52.7	55.0	44.0	58.5
CB-COR	33.5	67.1	73.0	52.6	54.8	44.1	58.3
BLOSUM40	35.6	68.1	72.7	53.6	56.4	48.0	58.2
CENT-COR	33.1	66.2	72.5	52.3	54.5	43.9	57.9
CA-EUC	27.8	65.7	70.7	49.3	53.4	41.7	55.7
CB-EUC	27.9	65.7	70.7	49.4	53.4	41.8	55.7
CENT-EUC	28.9	65.9	71.0	49.9	53.8	42.7	56.0
BLOSUM50	33.1	63.4	69.5	50.9	54.0	45.4	54.9
REMOTEHOMO	31.6	62.9	69.1	49.8	52.8	42.6	54.9
BLOSUM62	34.2	64.6	69.9	51.5	55.8	48.2	54.8
JOHNSON	33.3	63.8	69.2	50.8	55.8	48.6	53.7
IDENTITY	22.0	29.2	32.4	27.3	43.3	34.7	22.5
COMBINED	15.9	21.1	17.3	15.1	32.1	19.1	19.1
ANALOGOUS	13.8	16.1	10.6	10.5	30.3	17.7	14.4

The performance of a matrix is defined as the percentage of the correctly aligned residues compared to the reference alignment. The summary results of the sequence aligments are tabulated in Table 3.2. The ranking of matrices obtained here for the BAliBASE dataset are comparable to those found by [124] on their smaller data set of 122 protein pairs; except for PAM250, which we found to have a higher performance ranking on the BAliBASE database.

The performance of the substitution matrices depend on the degree of similarity of the aligned sequences, with lower scores for sequences that have lower sequence identity. However, the ranking of matrix performances is found similar across different BAliBASE subsets. The correlation set of matrices perform slightly better than those derived using Euclidean distance measure. The performance of our derived matrices are comparable to that of other matrices.

CHAPTER 4

APPROXIMATE SIMILARITY SEARCH IN GENOMIC SEQUENCE DATABASES USING LANDMARK-GUIDED EMBEDDING

4.1 Chapter Overview

[1] Similarity search in sequence databases is of paramount importance in bioinformatics research. As the size of the genomic databases increases, similarity search of proteins in these databases becomes a bottle-neck in large-scale studies, calling for more efficient methods of content-based retrieval. In this study, we present a metric-preserving, landmark-guided embedding approach to represent sequences in the vector domain in order to allow efficient indexing and similarity search. We analyze various properties of the embedding and show that the approximation achieved by the embedded representation is sufficient to achieve biologically relevant results. The approximate representation is shown to provide several orders of magnitude speed-up in similarity search compared to the exact representation, while maintaining comparable search accuracy.

4.2 Introduction

With the advent of high-throughput sequencing methods, the genomic sequences have been accumulating at an ever increasing rate. GenBank, a central database of publicly available DNA sequences, has been doubling in size every 15 months [11]. Since the sequencing of

[1] The content of this chapter was published in the *Proc. of the IEEE 1st Intl. Workshop on Similarity Search and Applications (SISAP, an ICDE-2008 workshop)* [133]

Hemophilus influenza in 1995, close to 700 organisms have been completely sequenced and published, and there are currently more than 3000 ongoing genome sequencing projects [95].

Homology search over these genomic sequence databases is a crucial step in the inference of functional and evolutionary relationships among proteins. According to a survey, similarity search makes up 35% of the tasks in bioinformatics research [52]. The increase in database size and the demands of large-scale analysis have been the driving forces of several efforts to speed up the similarity search process. The most successful of these efforts have been based on fast retrieval and *stitching* of common short subsequences. The goal of this study is to develop more effective common subsequence retrieval methods without significant compromise to the sensitivity of the similarity search results.

BLAST [2], which is currently the popular tool for biological homology search, is based on a heuristic that assumes presence of short exact matches between homologous sequences. For a given subsequence length k, a hash table of all possible k-mers is used to map the subsequences in the database. For a new query sequence, all k-mers of the query are searched using the hash table for finding matching k-mers in the database. The k-mer hits of the database are then tested for extension to generate longer matching regions and obtain the final local alignments [144].

There have been several improvements over BLAST that achieve more efficient or sensitive identification of evolutionarily close k-mers. These improvements were obtained mainly by relaxing the "short exact matches" assumption of BLAST's heuristic approach. Pattern Hunter [96] uses non-consecutive residues to construct k-mers, detecting replacements in the sequence better. The Piers method [25] guides inexact matching of short query segments using randomly selected seeds, achieving faster response at the cost of a small degradation in sensitivity.

In contrast to BLAST-like methods that are based on hashing short sequences, there have been *indexed search* approaches to the sequence retrieval problem. Note that the sequences are not objects in a multi-dimensional Euclidean space, which makes the spatial access methods (SAMs) such as R-tree and its variants [60, 138, 10] inapplicable. This has prompted the application of metric indexing methods, which do not need the original objects to be represented in multi-dimensional space, but only require that the distance measure between objects be metric (i.e., satisfy symmetry, non-negativity, and triangle inequality properties). In metric

indexing, the relative distances of the sequences are used to organize and partition the data into a hierarchical structure based on the distances to representative sequences of the partitions at each level. The triangle inequality is then used to prune the search space during the traversal of the metric-tree while answering a similarity search query. A survey of the metric indexing based methods can be found in [151].

Due to the requirements on the distance measure, the metric indexing methods have only considered the basic edit distance measure, where an identity matrix is used as the residue substitution matrix [19]. The identity matrix may be appropriate for nucleotide sequences where the substitutability of the nucleotides is almost uniform. However, the identity matrix does not give biologically accurate results for protein sequences, where the similarities and differences among individual residues become biologically more significant. [134] considers modelling other substitution matrices as near-metric based on the maximum and minimum substitution values whereas [174] uses mPAM [175], a biologically more sensitive metric substitution matrix.

[174] uses a multiple vantage point (MVP) tree to index subsequences. MVP tree, like other vantage point trees, is built by means of a top-down recursive process and does not gracefully support insertions and deletions. M-tree of [30] maintains a height-balanced tree to overcome this problem. Despite having good performance in general metric indexing applications, M-trees still suffer having a large number of sequence comparisons in biological sequence search (see 162 for a comparison).

Reference-based indexing have also been applied to similarity search in biological databases. In [162], a variable number of reference sequences are assigned to each database sequence, and the distances to the reference points are used to avoid unnecessary distance calculations between query and database sequences. Even though the number of distance calculations is minimized in reference-based indexing, the search is performed sequentially (i.e., every single sequence in the database is tested), which does not scale well for larger database sizes.

One further problem of the metric search methods, as they have been employed so far, is that only the global similarity between sequences is considered (with [174] being an exception). In the biological domain, the global similarity has limited applicability, and requires that the sequences being compared be evolutionarily very close. The end-gaps, which are normally not penalized in the biological domain, are also not handled gracefully by these methods.

We note that the global similarity measure may find use in searching very similar proteins in whole-genome comparisons; however, it is far from being applicable to the general homology search problem, which ultimately relies on detection of locally conserved short subsequences.

In this study, we limit our focus to the k-mer search, which is the main step in biologically relevant local search of homologous sequences. We propose an approximate similarity search which is based on landmark-guided embedding of the k-mers. We map the k-mers of a sequence database to a vector space based on their distances to a reference set of k-mers (denoted as *landmarks*). The k-mers in the embedded space are then indexed using spatial access methods for fast similarity search.

The contributions of this study include: (1) hybridizing Fastmap [40] and LMDS [34] methods to achieve more robust and accurate sequence embedding, (2) providing an approximate vector representation of sequences, (3) showing that the embedded representation allows efficient and biologically relevant indexing and similarity search.

4.3 Methods

Throughout this presentation, we use q to denote an input query sequence of length m whose symbols are from an alphabet of size σ. The set of database sequences are denoted as $S = s_1, s_2, ..., s_N$ where N is the number of sequences in the database. We generate all k-mers from both database and query sequences using a sliding window over sequences with a step size of one symbol.

The edit distance between two sequences is defined as the minimum cost of edit operations (insert, delete, replace) that transform one sequence to the other. The cost of replacing an individual symbol to another is looked up from a substitution matrix M. The costs of insertions and deletions are generally provided as an optimized gap penalty parameter. Without loss of generality, we used the weighted Hamming distance instead of the general alignment distance between sequences in order to decrease the analysis time. Because the gap penalty is usually larger than mismatch scores, weighted Hamming distance is sufficient when comparing short k-mers (see [174] for a proof correctness).

Our goal is to represent the set of k-mers in a low-dimensional space while preserving the dis-

Table 4.1: Symbols used in this presentation and their definitions.

Symbol	Definition
σ	length of the alphabet
k	length of each k-mer subsequence
q	query sequence for which a similarity search is being performed
M	the substitution matrix that gives the costs of replacing symbols
d	the dimensionality of the space in which the k-mers are embedded
N	number of k-mers to be embedded
n	number of landmark points (sequences)
$D_{A,B}$	the distances of sequences in set A to those in set B
Δ	squared distances
D'	Euclidean distance in the embedded space

tances among them as much as possible. Note that the k-mers cannot be directly represented as points in multi-dimensional vector space, therefore the classical dimension reduction techniques that rely on presence of the original high-dimensional vector space are not applicable here. Moreover, in the context of similarity search, the mapping has to be easily extensible to new query objects without requiring re-embedding of the whole database. These requirements lead us to the landmark based methods that generate a metric-preserving embedding using distances to only a small selection of sequences.

The FastMap method by [40] uses an iterative embedding procedure where at each iteration, the data is embedded onto an axis formed by two data points and the projection of the data onto this axis is used as input for the next iteration. The landmark points at each step are chosen heuristically to be as distant as possible in order to account for the highest variance in the data distribution. FastMap relies on the assumption that the original space is a Euclidean space and makes use of the 'cosine law' for projection and embedding. This assumption causes the embedding to be unstable if the original space is not Euclidean.

Recently, [34] have proposed a scalable landmark-guided metric preserving embedding algorithm, LMDS, that shows better stability properties than FastMap. LMDS first designates a set of n objects as landmark points and applies classical MDS on $n \times n$ matrix $D_{n,n}$ of distances be-

tween pairs of landmarks to obtain an embedding in d-dimensional space. The classical MDS [156] computes the d largest positive eigenvalues λ, of the mean-centered inner-product matrix with the corresponding orthonormal set of eigenvectors v. The d-dimensional embedding vectors for the landmark points are given by the following matrix:

$$L_k = \begin{bmatrix} \sqrt{\lambda}_1 v_1^T \\ \sqrt{\lambda}_2 v_2^T \\ \cdots \\ \sqrt{\lambda}_d v_d^T \end{bmatrix} \tag{4.1}$$

For the rest of the data points, *distance-based triangulation* is applied to each point x using $\Delta_{x,n}$ vector of squared distances to the n landmark points. The embedding vector $\overrightarrow{x}$ is obtained using the pseudoinverse transpose $L_k^\sharp$ of the landmark embeddings by the formula:

$$\overrightarrow{x} = -\frac{1}{2} L_k^\sharp (\Delta_{x,n} - \Delta_{x,n}^\mu) \tag{4.2}$$

where $\Delta_{x,n}^\mu$ is the mean value of squared distances to the landmark points.

The quality of the embedding depends partially on the selection of initial landmark points. We use three different methods for designating the landmark points. $LMDS_{rand}$ randomly selects the landmarks from the original data points. $LMDS_{minmax}$ obtains a set of landmarks that are distant from each other by starting from a random landmark and heuristically adding new landmarks such as to maximize the minimum distance to the already selected landmarks (This heuristic is similar to the one proposed by [54]). In order to combine the landmark selection performance of Fatmap and stability of LMDS, we also propose $LMDS_{fastmap}$ method, which uses the same landmark points as found by the Fastmap method on the same dataset.

Once all the database k-mers are embedded into the vector space, the indexing and retrieval tasks can be delegated to spatial access methods. In the experiments section, we present the search speed results of using X-tree [12], however any of the spatial access methods can be used for this purpose. A query k-mer would be embedded into the same vector domain using its distances to the landmarks used in generating the embedding. Using the spatial method of choice, the mapped query can then be searched against the vector representations of the database k-mers.

4.4 Experiments

The performance of the embeddings is evaluated on synthetic and real datasets. In synthetic datasets, for a given alphabet size σ and subsequence length k, all k-mers were generated. The size of the synthetic datasets were limited to 10,000 sequences, and a random sampling from all possible sequences were performed if the number of k-mers $N = k^{\sigma}$ exceeded 10,000. An identity substitution matrix is used to calculate the distances between k-mers.

The real data was obtained from the yeast proteins dataset which is used to benchmark BLAST (`ftp.nsbi.nlm.nih.gov/pub/impala/blasttest`). The yeast dataset contains 6,341 protein sequences with a total of about 2.9 million residues. The dataset also contains a separate query set of 103 proteins ranging from 38 to 884 residues in length, whose true positive hits are determined by human experts. Note that the alphabet of the protein sequences has cardinality of 20 and is composed of amino-acid residue symbols.

To accurately model biologically relevant distances among sequences, we used CB-EUC substitution matrix by [132], which is a metric matrix with good sequence alignment performance. Note that according to [137], if a substitution matrix is metric, then the alignment distances of the sequences using this matrix also forms a metric.

For each embedding method and variations, we evaluated the quality of the embedded sequences using Sammon's *metric stress* measure E [135] which quantifies the error in the preservation of the original distances, with a value 0 indicating a lossless embedding:

$$E = \frac{1}{\sum_{i<j}^{n} D_{ij}} \sum_{i<j}^{n} \frac{(D_{ij} - D'_{ij})^2}{D_{ij}} \tag{4.3}$$

where D_{ij} is the original distance between kmers i and j, and D'_{ij} is the distance in the embedded space.

4.4.1 The dimensionality of the embedded space

There is an accuracy-performance trade-off on the number of dimensions to be used in the embedded space. As the number of embedding dimensions d is increased, the original data can be represented better, at a higher cost incurred on the similarity search in the embedded

space. Since a lossless embedding is not possible, one needs to empirically determine the dimensionality for a desired level of mapping accuracy.

Figure 4.1 and Figure 4.2 show the metric-stress and the correlation coefficient of the mapped distances with respect to the original distances. The original data is a synthetic set of sequences of length 5, where CB-EUC substitution matrix (alphabet size $\sigma = 20$) is used to calculate the original distances. (Qualitatively similar results were obtained for other k and σ values.) In order to obtain a fair comparison, the same number of landmark points required in Fastmap ($n = 2 \times d$) is used in the LMDS methods. The $LMDS_{maxmin}$ and $LMDS_{fastmap}$ methods show similar mapping accuracies, whereas $LMDS_{random}$ requires more dimensions to achieve the same level of accuracy. The Fastmap method performs similar to LMDS methods up to a certain number of dimensions, after which the numerical instability in the mapping accumulates and degrades the mapping accuracy.

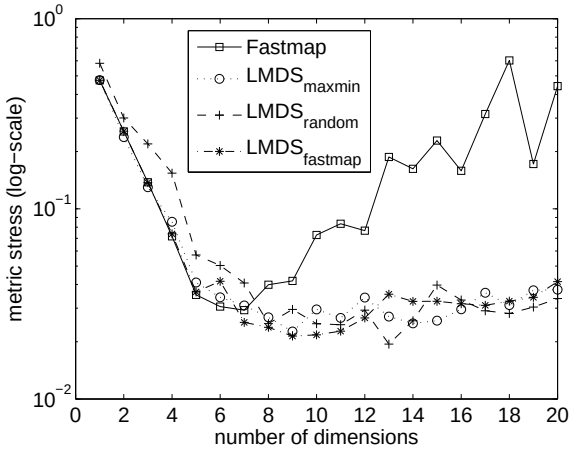


Figure 4.1: Metric stress of the embedding vs. target dimensionality (k=5).

Due to its numerical instability, Fastmap does not necessarily give a better embedding as the number of dimensions is increased. Despite this instability, we have observed an important merit of the Fastmap method. Namely, the number of dimensions beyond which Fastmap's accuracy degrades corresponds to the intrinsic Euclidean dimensionality of the original dataset. Notice that in Figures 4.1, the metric stress achieved by Fastmap at $d = 7$ is comparable to the metric stress achievable by the LMDS methods with higher dimensions. This observation has

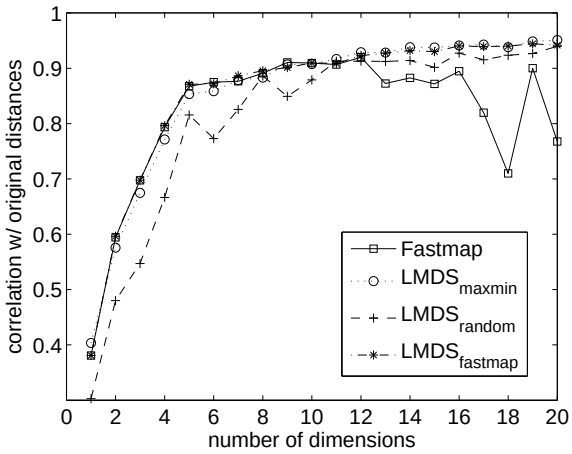


Figure 4.2: Correlation with the original distances vs. target dimensionality (k=5).

led us to determine the dimensionality of the target embedding space as this *breaking point* in Fastmap's mapping accuracy. This gives us a more well-defined definition than assessing the convergence of LMDS.

Defining the *breaking point dimensionality* of a given dataset has allowed us to analyze its dependence on the other parameters. As shown in Figure 4.3, the *breaking point dimensionality* of the dataset increases linearly with both the sequence length k and the alphabet size σ. An identity substitution matrix is used in the distance calculation in order to compare the alphabet size σ across datasets. Note that even though the alphabet size is 20 for proteins, amino-acid substitution matrices impose a clustering on the amino-acid types, which in effect reduces the intrinsic dimensionality of the dataset. The dimensionality of the dataset when the CB-EUC substitution matrix is used ranges between that of the datasets with $\sigma = 4$ and $\sigma = 5$. In fact, the principal component analysis of the CB-EUC matrix shows that the first 5 principal components account for the 98.2% of the variation in the matrix values.

4.4.2 Number of landmarks

In Fastmap, 2 landmarks are chosen for each dimension to provide an axis of projection for the data, which yields the total number of landmarks to be $n = 2 * d$. Whereas in the LMDS methods, the number of landmarks can be chosen arbitrarily, provided that $n \geq d + 1$. The effect of the number of landmarks on the mapping accuracy is shown in Figure 4.4. The

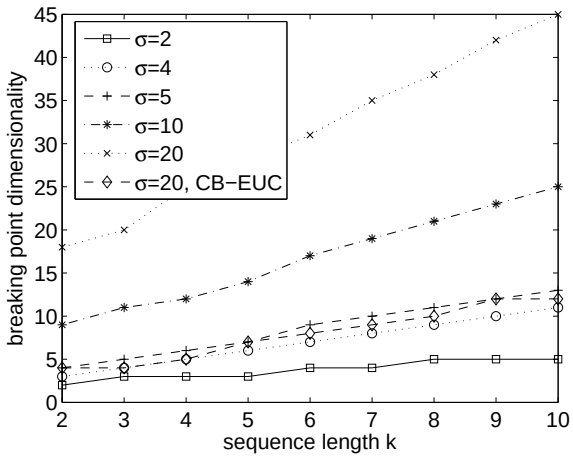


Figure 4.3: Dependency of dimensionality on sequence length and alphabet size. (k=5, d=7)

best landmark selection strategy is that of Fastmap, in terms of monotonically decreasing the metric stress. For $LMDS_{maxmin}$, each new landmark is not guaranteed to improve the mapping accuracy, because the landmark selection heuristic is only an approximation to the optimal selection. However, the LMDS methods do converge to an optimal mapping accuracy, if sufficiently large number of landmark points are used.

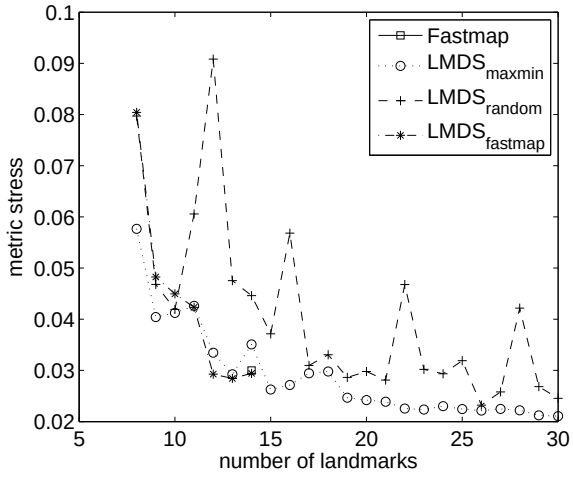


Figure 4.4: The effect of the number of landmarks on mapping accuracy. (k=5, d=7)

The number of landmarks affects the computational complexity of mapping the database to

the embedded space, and also of mapping new query sequences for similarity search in the embedded space. Even though LMDS methods can provide further improvement in the mapping accuracy, the number of landmarks would be limited by the amount of time one is willing to spend in mapping new sequences.

4.4.3 Similarity search performance

While metric-stress is a good indication of how well the distances among the original sequences are preserved in the embedded space, the similarity search accuracy within the embedded space still remains to be evaluated. In order to test the similarity search performance, we performed range queries on the yeast dataset using the separate set of query proteins and various distance thresholds. For a given query kmer q and distance threshold r, a *range query* in the original sequence space would return all the kmers in the database that are within r edit distance of the query kmer. Similarly, in the embedded vector space, all mapped objects that are within r Euclidean distance away from the image of the query q' are returned.

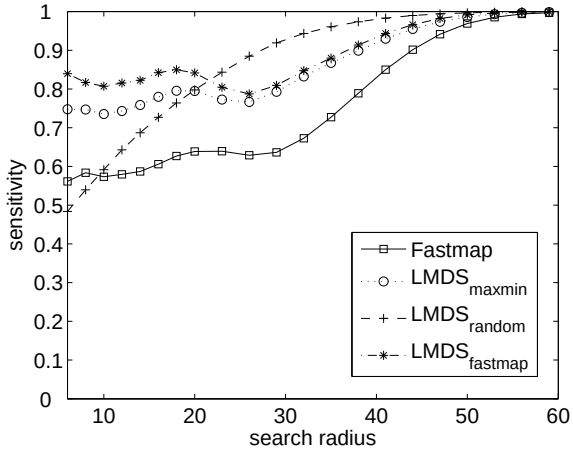


Figure 4.5: Sensitivity of kmer range search results. (k=6, d=8)

Figures 4.5 and 4.6 show the sensitivity (true positive rate) and specificity (1 - false positive rate) of the range queries for different mapping methods under various search radii r. The results are the averages of the queries performed for all kmers in the test query set. The approximation by Fastmap tends to overestimate the original edit distances, which yields less

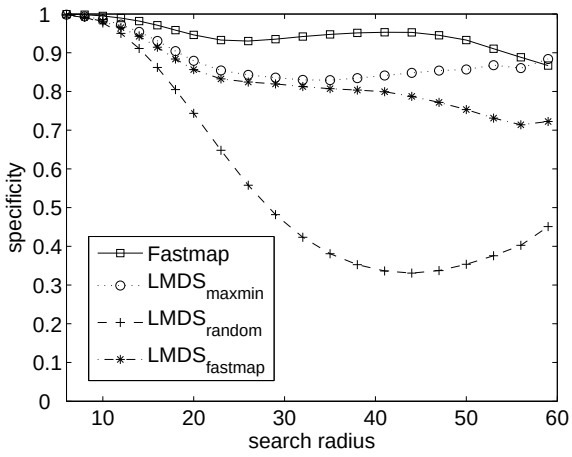


Figure 4.6: Specificity of kmer range search results. (k=6, d=8)

number of hits in the answer set, and thus higher specifity but lower sensitivity compared to other methods. $LMDS_{fastmap}$ combines the landmark selection algorithm of Fastmap with the stability of LMDS to yield the best sensitivity results while having comparable specificity with those of $LMDS_{maxmin}$ and Fastmap.

4.4.4 Homology search performance

It must be noted that in the context of homology search, a small distance threshold is sufficient to obtain biologically relevant range queries, because the homologous proteins are expected to share very similar subsequences. Moreover, the homology search procedure is particularly permissive to small errors in the approximation of kmer distances, because kmers from the homologous proteins missed by some of the query subsequences are compensated by other subsequences that correctly return the kmers of the homologous proteins.

For each of the 103 yeast query proteins, we generated all kmers and searched the yeast dataset for kmer hits. For each distance threshold, a search result is considered to be a true hit if at least one kmer of the query protein returns a kmer of the homologous proteins. Figures 4.7 and 4.8 show the homology search results using varying distance thresholds. For comparison, the results of an exact kmer search in the sequence space are also included. Notice that the results of the homology search are in accordance with the results of the kmer search in Figures 4.5 and 4.6. Namely, the methods that provide more sensitive kmer answer

sets also provide more sensitive homology search results.

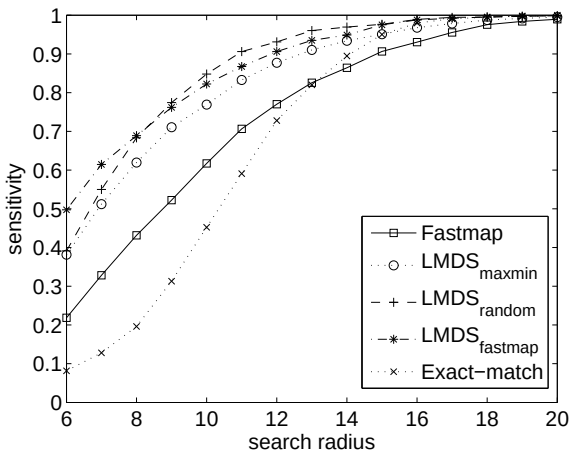


Figure 4.7: Sensitivity of the homology search on the yeast dataset. (k=6, d=8)

Even though for a given distance threshold, more kmers are returned per kmer search in the embedding methods compared to an exact kmer search (Figure 4.7, right), the embedding methods require a smaller search radius to achieve the same level of sensitivity (Figure 4.8, left). For instance, to achieve 90% sensitivity (i.e., to obtain 90% of the homologous proteins), the $LMDS_{maxmin}$ method returns $3.5\%_{000}$ of the kmers per kmer search, whereas an exact kmer search in the sequence space returns $4.1\%_{000}$ of the kmers. This is due to the fact that in the embedding methods, the approximation errors at lower distance thresholds cause the distances to some of the kmers of homologous proteins to be underestimated. These kmers are then returned in the answer set, whereas they would not be present in a range query in the original sequence space.

Note that smaller kmer lengths k while require a smaller search radius and provide more sensitive homology search, they incur higher false positive rates; whereas higher k values provide more specific results at the cost of sensitivity. $k = 6$ was found to be a good trade-off between sensitivity-specificity of homology search results on the yeast dataset. The relative performance of the methods were similar for other kmer lengths.

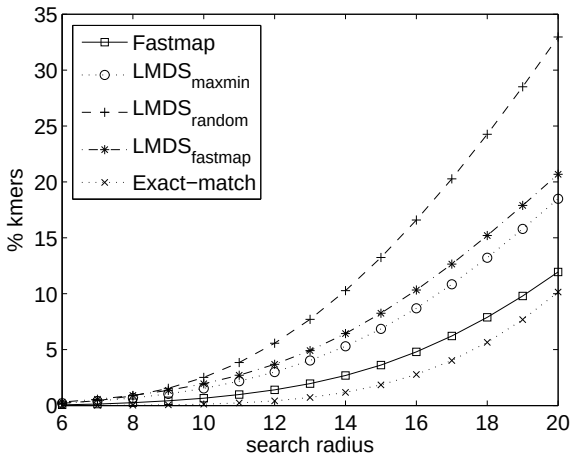


Figure 4.8: Database pruning performance of the homology search on the yeast dataset. (k=6, d=8)

4.4.5 Search time performance

Short subsequences are embedded under the premise that indexing and similarity search in a vector domain is more efficient than those in the sequence domain. In order to illustrate this, we performed indexing in both domains and compared the CPU times for range queries. We employed Slim-tree [157] *metric access method* (MAM) for indexing the sequences, and X-tree [12] *spatial access method* (SAM) for indexing the vector representations resulting from the $LMDS_{fastmap}$ method.

Figures 4.9 and 4.10 show the average query times for varying database sizes and search radii, respectively. Similarity search in the vector domain achieves approximately 500-fold speed-up over that in the original sequence domain. A search radius of 7 is used while varying the database sizes (left) and a database size of 100,000 is used while varying the search radii (right). A similar trend in search times were observed for other k, database size, and search radius values. While an exhaustive analysis and comparison of MAM and SAM methods are beyond the scope of this study, we note that the search speeds achieved by Slim-tree and X-tree are representative of those achievable by the currently available MAM and SAM methods.

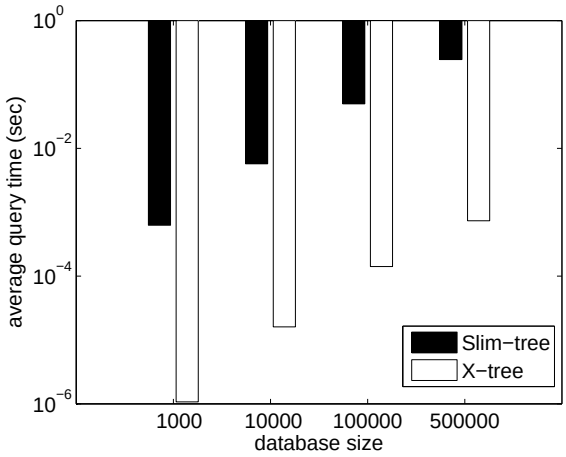


Figure 4.9: Average query time comparison (k=6, d=8)

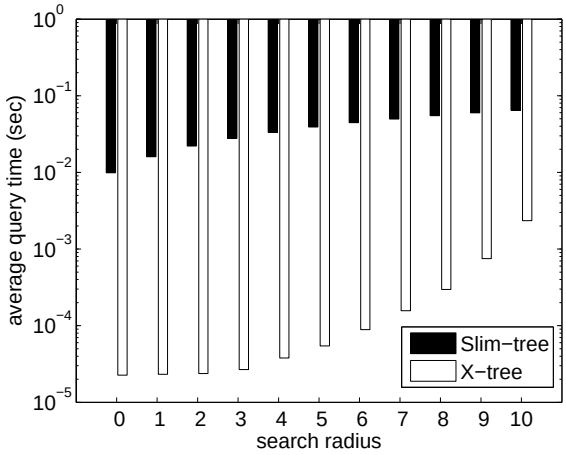


Figure 4.10: Average query time comparison (k=6, d=8)

CHAPTER 5

LFM-PRO: A TOOL FOR DETECTING SIGNIFICANT LOCAL STRUCTURAL SITES IN PROTEINS

5.1 Chapter Overview

[1]The rapidly growing protein structure repositories have opened up new opportunities for discovery and analysis of functional and evolutionary relationships among proteins. Detecting conserved structural sites that are unique to a protein family is of great value in identification of functionally important atoms and residues. Currently available methods are computationally expensive and fail to detect biologically significant local features.

We propose *LFM-Pro* (*Local Feature Mining in Proteins*) as a framework for automatically discovering family specific local sites and the features associated with these sites. Our method uses the distance field to backbone atoms to detect geometrically significant structural centers of the protein. A feature vector is generated from the geometrical and biochemical environment around these centers. These features are then scored using a statistical measure, for their ability to distinguish a family of proteins from a background set of unrelated proteins, and successful features are combined into a representative set for the protein family. The utility and success of LFM-Pro are demonstrated on Trypsin-like Serine Proteases family of proteins and on a challenging classification dataset via comparison with DALI. The results verify that our method is successful both in identifying the distinctive sites of a given family of proteins, and in classifying proteins using the extracted features.

The software and the datasets are freely available for academic research use at `http://`

[1] The study described in this chapter was published in the *Bioinformatics* journal in 2007 [131].

bioinfo.ceng.metu.edu.tr/Pub/LFMPro

5.2 Introduction

Rapidly growing protein structure repositories open up new possibilities for discovering functional and evolutionary relationships among proteins, and for elucidating the principles by which a certain structure produces an observed function. The increase in data size, however, also calls for more efficient and accurate methods of comparing proteins and identifying potential functional residues and binding sites.

The classical approaches of structural analysis have focused on global pairwise structural alignment of proteins to detect similarities [49], and help transfer of information about a well-known protein to unknown proteins that can be structurally aligned to it. The structural alignment methods, however, are computationally intensive and do not lend themselves to large-scale comparisons. Moreover, they miss remote homologies, especially when the proteins share only a local region.

Many proteins have a multi-domain nature, and the global similarities alone are not sufficient to identify functional similarities existing in distinct local domains. Inevitably, *local structural motifs* are often required for identification of biological function and homology relationships [152, 64, 153]. Manual identification of these regions require intensive genetic and molecular biology experimentation, which may take years of diligent studies. An automated method of detecting potential sites would thus be very much appreciated. We therefore focus, in this study, on automatic discovery of local sites of proteins which have distinguished structural and biochemical features, and may thereby have functional significance.

Previous approaches have assumed that such functional sites are already known [6, 166], and have focused on building a *description*, rather than *automatic detection* of these sites, with the hope of cataloguing these descriptions as structural motifs, so that unknown proteins could be annotated via comparison with these motifs. The *Local Feature Mining in Proteins* (*LFM-Pro*) framework proposed in this study starts with a group of proteins that share a certain function, and does not assume any prior knowledge about the location or nature of the functional sites. Through comparison of this group of proteins with a background set of unrelated proteins, it is able to detect sites that yield features unique to the family members.

Structural motif search is generally based on graph theoretical algorithms [146, 68, 73, 8, 82, 67], geometric hashing [165, 140] and others [142]. In order to discover motifs, these methods search for commonly recurrent local structures in space, based on their specific models. The graph theoretic approaches generally require exponential time in the number of the localities being matched. The computational bottle-neck of these approaches prevent effective automated detection of local motifs. More importantly, these methods analyze the protein at the *residue level*, and fail to handle substitutions of the amino-acids or displacements of the backbone. It has been shown that residues can adopt quite different conformations while managing to conserve the positions of their important functional atoms [166]. Therefore, an *efficient* method that can analyze the protein structures at the finer granularity of *atomic level* is needed.

5.3 Challenges and Directions

We focus on identification of local sites which are unique to a family of proteins sharing a certain structural or functional property. A site can be defined as a three dimensional location in the protein, and a local spatial neighborhood around this location having a certain structure or function [6]. In order to mine a protein dataset for possible functional sites, we are faced with three main challenges.

The first challenge is deciding on a data structure for sampling of the 3D distributions of the site locations and determining the size of their spatial neighborhood. For this purpose, a three dimensional grid has previously been utilized [55, 6]. Although grids offer computational advantages, the protein space has to be sampled in high resolution in order to capture micro-environments, which causes very large grids, defeating the purpose of using a grid-based distribution. Some methods therefore only consider local patterns centered at each residue or at some manually-chosen positions as potential motifs [73, 94], possibly missing motifs not centered around such positions. Furthermore, these methods usually miss relatively rare and novel motifs. An automatic method that produces a concise yet complete coverage of the motif space is still missing. The method we present in this paper is able to efficiently sample the motif space for identification of unique structural and functional local motifs. Our method relies on a novel computational geometry method for identification of topologically significant locations and also dynamically adjusts the size of the site based on the residues

surrounding the microenvironment.

The second challenge is the characterization of the microenvironment features. Presence of certain amino-acid types as the basic feature [164, 143, 107] does not provide a detailed characterization of the site, and may miss certain motifs because of the similarity and substitutability of amino acids. More detailed characterization of the microenvironment [6] consider properties such as hydrophobicity, mobility, and solvent accessibility which can capture the physico-chemical nature of the site at the cost of requiring more time for the computation of these properties. We have found that using the atom frequencies [92, 103] is a good tradeoff between accuracy and efficiency in characterizing the microenvironment for the purpose of local motif detection. Moreover, unlike previous studies, we also augment the feature vector to capture the topological information of the backbone surrounding the microenvironment.

The last main challenge is having an efficient and sensitive method for detecting common patterns. Determining which motifs are responsible for an observed function is a difficult task. Graph theoretic approaches try to find common subgraphs, but they are currently not scalable for large space of possible motifs, and they cannot easily handle noise in the data or substitution of residues. Statistical methods have been used [6] in characterization of the motif structure while comparing a group of known sites and non-sites, but these methods rely on *a priori* knowledge of the functional sites. Whereas, the method we present uses a data mining approach to discover distinguishing functional sites shared by a family of proteins without requiring prior knowledge of the location or nature of these sites. Moreover, it is robust to noisy patterns, and can handle incorrect initial classification of the data.

5.4 Methods

We first identify topologically significant local structural centers of each protein, by calculating the critical points of a particular distance field. A ball centered around each critical point defines the spatial neighborhood of these structural centers. Each critical point is then associated with topological and biochemical features of its spatial environment.

Figure 5.1 shows an overall flow-chart of the steps followed in LFM-Pro. For each protein, 1) the location of the critical points of distance field to backbone atoms are identified, 2) the critical points are filtered to remove nonpersistent or unimportant ones, 3) a feature vector

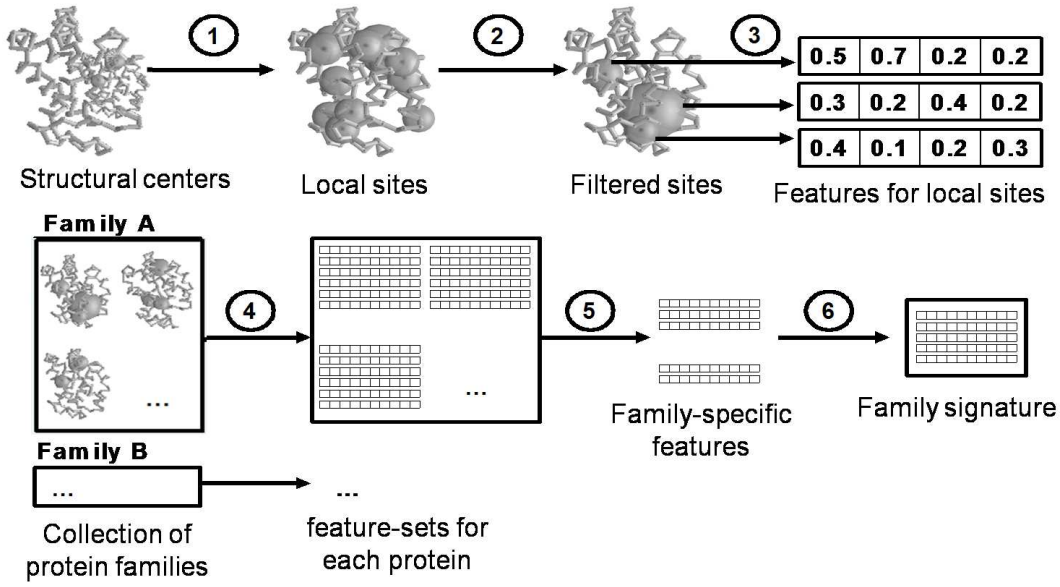


Figure 5.1: The general strategy of LFM-Pro.

that captures the topological and biochemical properties of its spatial neighborhood is associated with each critical point. 4) Feature vectors for the remaining critical points of each protein in the dataset are pooled and 5) those that are generated from family members are assessed for their ability to discriminate the family proteins from the rest of the dataset. 6) the critical points that display the best discriminating behavior in step 5 are combined into a representative feature set of the family.

Once we generate the feature vectors for each critical point of the proteins, a family of proteins are then searched for shared feature vectors. The aim here is to find critical points unique to a family; therefore, a set of shared feature vectors are chosen such that it is able to distinguish the members of the protein family from a background set of proteins that lack the properties and functions of interest. The group of critical points that are unique to a family are combined to obtain *a representative feature set* for the family. In the following subsections, each of these steps are described in detail.

5.4.1 Sampling of the Structural Centers

Given a protein P as the set of its alpha Carbon (C_α) atom centers $P = \{p_1, \ldots, p_n\}$, the distance function $\Phi_P : \mathbb{R}^3 \rightarrow \mathbb{R}$ w.r.t. P is defined as follows: $\Phi_P(x)$ is the nearest distance from x to any $p_i \in P$. Φ_P describes the influence of (the backbone atoms of) protein P to its neighboring space via the distance field. Intuitively, if two proteins have similar structure, they should have similar distance fields. In particular, if there are regions in space where proteins display similar local structural patterns, then they should have similar distance fields in and around that region as well.

We identify the potential motif centers by finding the critical points of this distance function. Formally, critical points of a smooth function g, are points with vanishing gradients. In our case, for a function defined over $\mathbb{R}^3$, there are four types of critical points: local minima, local maxima, and two types of saddle points. Note that, when distance to backbone atoms is used as function g, it turns out that the set of critical points of Φ_P is the set of intersection points between some Delaunay simplex (a point, edge, triangle, or tetrahedron) with its dual Voronoi elements (a polytope, face, edge, point, respectively), and can be computed in $O(n^2)$ time where $n = |P|$ [50].

Figure 5.2 shows the Delaunay tessellation (dashed lines) and Voronoi diagram (solid lines) of a set of points in 2D. Region enclosed by a Voronoi polyhedron is the area that is closest to the enclosed point than to any other point in the set. Delaunay tesselation is obtained by connecting points that share a boundary. In 3D, Delaunay tessellation would give space-filling tetrahedra. A circle (sphere) can be drawn whose center is a vertex of Voronoi diagram and which passes through the points in the corresponding Delaunay triangle (tetrahedra).

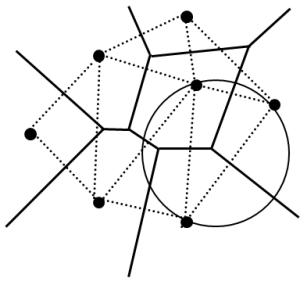
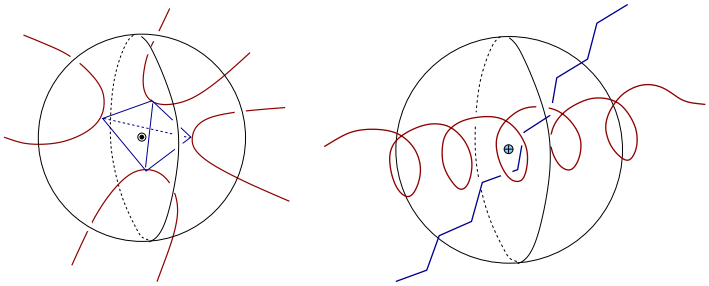


Figure 5.2: Delaunay tessellation and Voronoi diagram of a set of points in 2D.

We now collect Π as the set of critical points of the distance function. Some examples of structural motifs that such critical points can capture are illustrated in Figure 5.3. In Figure 5.3-a, four pieces of protein backbone come close in space, forming a contact as indicated by the tetrahedron in the middle. The double point is a local maximum of Φ. In Figure 5.3-b, the cross-point is a saddle point. Local spatial patterns can be captured by taking a ball centered at these critical points. The *spatial neighborhood* of a critical point is defined as the spherical region centered at the critical point, whose radius is its distance function value.



(a) (b)

Figure 5.3: Two types of motifs captured by critical points of the distance function. (a) A local maxima. (b) A saddle point.

Following the generation of all critical points of distance, we perform a filtering of these points to eliminate noise. The structural importance of the critical points were assigned using the topological persistence algorithm from [38], and those with small persistence were removed from Π. This topological method of removing noise is fundamentally different from those that employ clustering of neighboring points, in terms of the type of noise it removes. Roughly speaking, it measures the importance of a feature (critical point) by measuring how persistent this feature remains if the distance field is perturbed. Note that filtering based on persistence effectively eliminates the noise inherent in the crystallography methods used to obtain the atom coordinates. After the filtering step, the number of remaining critical points are roughly the same as the number of the amino acids in the protein.

5.4.2 Characterizing the Spatial Neighborhood

As a by-product of our structural center sampling method, we have a natural way to decide the neighborhood size, which is better than prefixing some threshold value. For the spatial neighborhood around each critical point, we associate a feature vector, based on both the structural and biochemical nature of the neighborhood. The structural features include: the persistence value of the critical point, the radius of the neighborhood, and the *writhing number*. The biochemical features we use are based on the frequency and location of the constituent atoms within the neighborhood.

The writhing number, or writhe, is originally used to measure the super-coiling phenomenon for a space curve, and has been used to characterize both DNA [47, 81, 149] and protein structures [91, 129]. We compute the writhe of those backbone pieces contained within the spatial neighborhood to measure their relative spatial arrangements.

In order to capture the biochemical nature of the spatial environment, we use the frequencies of each of the side-chain Carbon, Nitrogen, Oxygen, and Sulfur atoms within the spherical region. Furthermore, the location information of these atoms is captured by computing the center of mass for each atom type. Note that our framework can be easily extended to use physico-chemical properties such as hydrophobicity, solvent accessibility, Van der Waals radii, or mobility, which can capture more detailed information about the spatial environment [6]. However, we did not use such extended features in this study, because of the computational cost they incurred.

5.4.3 Mining for a Representative Feature Set

Each protein p_i now has a set $\Pi = \{c_1, \ldots c_n\}$ of feature vectors generated from its important critical points. Let $F = \{p_1, \ldots p_m\}$ denote a family of proteins that are known to share a common structural or functional property. And let the set G denote the rest of the proteins in the dataset. We wish to determine the critical points that are unique to family F, and assess their ability to discriminate the proteins within the family from the rest of the proteins. Note that the algorithm to detect family-specific critical points has to allow changes in the values of the feature vectors. We utilized a distance-based approach for this purpose.

The dissimilarity $d(c_i, c_j)$ of any given two critical points can be defined in terms of an appropriate distance function between their corresponding feature vectors. We observed that a simple Euclidean distance measure on normalized feature vectors was sufficient in detecting family specific structural centers. A *weighted*–Euclidean distance, that can highlight varying contributions of the individual environment features could also be designed by optimizing the weights against an objective function.

When comparing a critical point c_x to a protein p, we take the distance of c_x to its closest match in p as defined with the distance function:

$$d(c_x, p) = min\{d(c_x, c_1), \ldots, d(c_x, c_n)\} \tag{5.1}$$

where $c_1, \ldots, c_n$ are the critical points of the protein p. Intuitively, if a critical point c_x is part of a protein p, one would expect a very small value for $d(c_x, p)$.

For each candidate critical point c_x of the proteins in the family F, we calculate its distance to all the proteins in the dataset. For an ideal discriminative critical point, the distances to the proteins in F would be clustered at a minimal, whereas the distances to the rest of the proteins, G, would take upon higher values. We modeled this intuition by defining the *discrimination score s* of a critical point as follows:

$$s(c_x) = \frac{\mu(c_x, G)}{(1 + \mu(c_x, F)) * (1 + \kappa(c_x, F, G))} \tag{5.2}$$

where $\mu(c_x, F)$ is the average distance of c_x to proteins in the family F,

$$\mu(c_x, F) = avg(d(c_x, p \in F)) \tag{5.3}$$

and κ is the number of background proteins that have a distance smaller than the maximum within-family distance $d^*(c_x, F) = max(d(c_x, p \in F))$.

$$\kappa(c_x, F, G) = count(d(c_x, p \in G) \leq d^*(c_x, F)) \tag{5.4}$$

In Equation 5.2, $\mu(c_x, F)$ and $\mu(c_x, G)$ ensure that those critical points that have small within-family distance and high out-of-family distance get higher discrimination scores. The average distances alone, however, do not guarantee a clear separation of the family proteins from the rest. The term κ favors those critical points that can cluster the family proteins with minimal number of out-of-family proteins. In other words, μ works to select features common to

family, while κ works to avoid features that cannot discriminate family proteins from the rest. Each term in the denominator is padded with 1 for numerical stability.

Using the discrimination scores, we obtain a set of critical points ranked by the scores reflecting how representative they are for a given family F. We refer the collection of the critical point features with their associated scores as the *representative feature set* of the family.

5.4.4 Classification Modeling.

Let $\Pi = \{c_1, \ldots c_n\}$ be the representative feature set of family F, with corresponding discriminative scores $S = \{s_1, \ldots s_n\}$ and maximum within-family distances $D^* = \{d_1^*, \ldots d_n^*\}$. The *membership score* of a new protein p to the family F is calculated as follows:

$$\psi(p, F) = \frac{1}{n} \sum_{i=1\ldots n} s_i \frac{d_i^* - d(c_i, p)}{d(c_i, p)} \tag{5.5}$$

The membership score ψ, is dominated by the matching features that have small distance and high representative scores. The numerator term in the summation in Equation 5.5 provides a threshold logic based on the maximum within-family distances d^*. Those features that match the protein with a distance smaller than d^* contribute positively in the membership score, whereas those that have a greater distance are penalized in the scoring. The overall membership score reflects how well a protein matches a representative feature set. In a multi-family classification scheme, the membership score $\psi(p, F)$ can be used to assign the protein p to the closest family.

5.5 Results

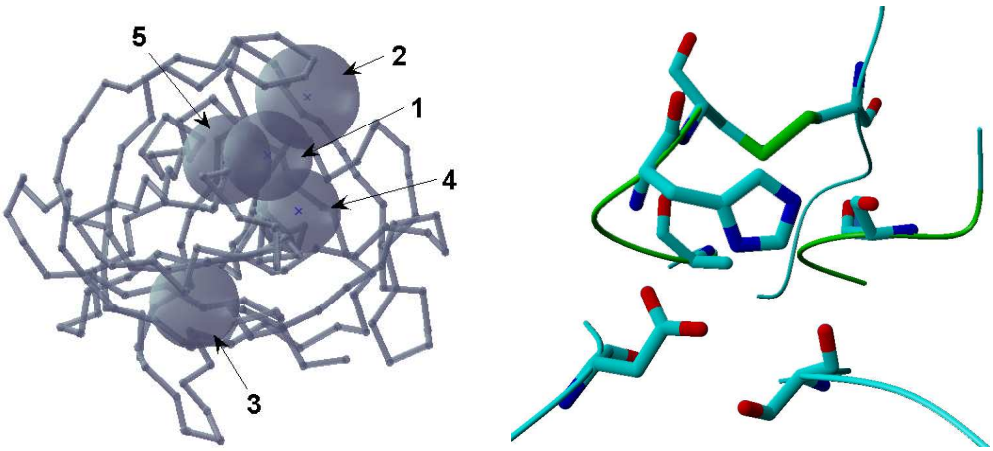
5.5.1 Experimental Setup

All the experiments were conducted on a single processor *Pentium 4* PC with 2.8 GHz CPU and 1 GB main memory. The selection of centers via determination of critical centers of the distance function was implemented in Python and C, using CGAL [27] computational geometry library; the feature extraction and mining methods were developed under Matlab environment [98].

The proteins used in this study were selected from the representative ASTRAL [20] dataset of SCOP 1.69 [108] with less than 40% sequence homology. There were a total of 7,237 entries in the ASTRAL dataset. The one-time-only generation of critical points and their corresponding feature vectors took 49 seconds (38 sec. for critical points, and 11 sec. for features) on the average per protein.

5.5.2 Mining Functional Sites

The success of LFM-Pro could be assessed by applying it to protein families that have well-defined functional sites, and investigating whether the sites detected by LFM-Pro match the known functional sites in these proteins. Serine Proteases are the most studied family of proteins, in the context of structural motif extraction [6, 166, 103, 68, 67]. We follow the tradition and also use Serine Proteases for this study. The proteins were selected from the SCOP superfamily (b.47.1.*) "trypsin-like serine proteases," here on referred as the **SP** superfamily and included both prokaryotic (PSP: 10 SCOP entries) and eukaryotic (ESP: 19 SCOP entries) proteins, which share the same catalytic site.



(a) (b)

Figure 5.4: Top scoring sites in Alpha-lytic protein (1ssx). The features were obtained by mining SP dataset against a random set of 200 background proteins. *Left:* Features 1,2,4,5 span the neighborhood of the catalytic triad, whereas feature 3 contains a distant disulfide bridge Cys189-Cys220. *Right:* A closer look into the catalytic region spanned by features 1,2,4,5 is given in. The residues whose side-chain atoms are contained within these sites are shown.

The local site mining for the SP family took 30 seconds to complete. Note that, with the same number of localities to compare, the subgraph mining methods may take several days to complete [67]. Figure 5.4 shows the mapping of the top scoring features on Alpha-lytic protein (1ssx). The top sites obtained by the feature mining algorithm corresponded to the catalytic triad site of the Serine Proteases. The atoms within the immediate neighborhood of the catalytic triad have relatively conserved positions, which is successfully picked up by the mining algorithm. The highest scoring site contained atoms of the residues Ser195, His57, Asp102, Ser214 and Ala55. The residues Ser195-His57-Asp102 form the charge relay system responsible for the hydrolytic cleavage of the appropriate substrate. Ser214 has also been found to be highly conserved in SP [166]. We also observed that Ala55 is conserved in SP and we speculate that Ala55 keeps the catalytic triad in its relative orientation via Van der Waals interactions.

The third highest scoring site includes the disulfide bridge Cys189-Cys220, which is distant to the catalytic site, but is nevertheless conserved across Serine Proteases. This disulfide bond keeps the backbone such that Ser195 and Ser214 can remain in close proximity. The next highest scoring site is another disulfide bridge, Cys42-Cys58, which helps keep the His57 and Ala55 residues within the catalytic site.

5.5.3 Selection of the Background Proteins.

One interesting question is whether the use of a background set of proteins is really necessary, i.e., whether it would be possible to detect the functional sites by just finding features common to a family of proteins, without comparison to unrelated proteins. Figure 5.5 illustrates the effect of the size and nature of the background class of proteins on the detection of functional site in SP. The rank of the first feature that map to the catalytic triad site is used as the basis of evaluation.

We expected that the performance of the algorithm would improve with increasing number of out-family proteins used. As the size of the background set is increased, the contribution of $\mu(c_x, F)$ term in Equation 5.2 decreases, which translates into *distinguishing* features ranking higher than *common* features. Figure 5.5 shows that for each type of background set of proteins we used, the algorithm was able to detect the functional site, when given a sufficiently

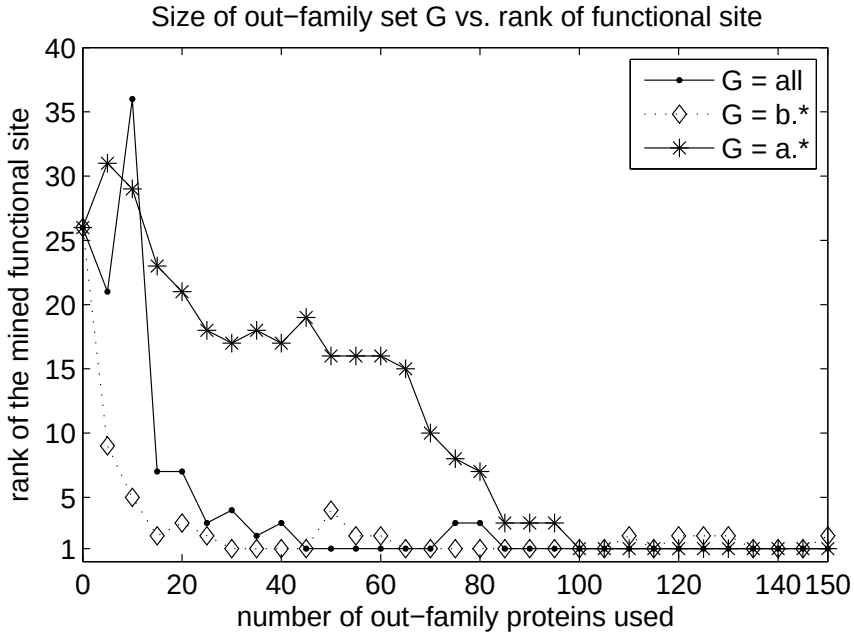


Figure 5.5: The effect of the size of the background set G on detection of the functional site. Results are shown for mining SP dataset against selection of proteins using three sets of proteins: all proteins, only b.* all-beta class, or only a.* all-alpha class. The size of G is shown up to 150 proteins for illustration purposes; the rank of the mined functional site did not change beyond 150 proteins.

large number of background proteins.

Furthermore, Figure 5.5 demonstrates that using proteins that share structural features with the family under investigation increases the accuracy of the mining.

When random out-family members were selected from b.* SCOP class of all-beta proteins, the functional triad site is detected among the top-scoring sites, even with only a few out-family proteins. Whereas, significantly more proteins are needed in the out-family set if one uses a.* SCOP class of all-alpha proteins, which share little structural fold similarity with SP. This observation is attributed to the fact that proteins that share structural folds with the investigated family can better prune out insignificant scaffold sites and enhance detection of unique sites.

The set of background proteins needed to obtain the most desirable feature-mining results would depend on the specific family being studied. Even though all available proteins can be used as the background set G, it may be desirable to reduce the size of G for efficiency

purposes. As a general guideline, we recommend the use of proteins that share the same structural folds, but are missing the target function of interest.

5.5.4 Selection of Family Proteins.

While seeking features that are distinguishing from unrelated proteins, we also seek that these features be common across the family. For this reason, appropriate selection of the family proteins plays an important role in detection of functional sites. Figure 5.6 demonstrates the effect of composition and size of the family proteins on detection of the catalytic triad. The region of the catalytic triad is more conserved in Eukaryotic proteins, giving the functional site a higher score. When PSP and ESP proteins are combined (SP), the family set would contain an evolutionarily more diverse set and the algorithm can attribute lower scores to those sites that are unique only to either of these two families, and highlight the functional site that is shared by both protein families.

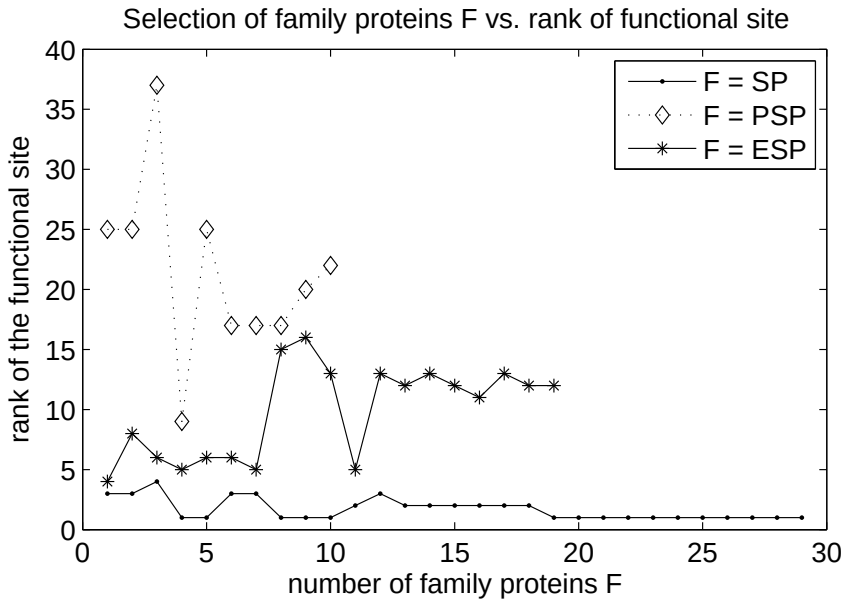


Figure 5.6: The effect of the size and composition of the family set F on detection of the functional site. The background set G for this experiment is composed of 200 randomly selected proteins from the b.* SCOP class of all-beta proteins.

Appropriate composition of the family proteins was more effective in mining for the functional site than simply increasing the size of the family. In fact, increasing the number of proteins did not give the catalytic triad significantly higher scores in PSP or ESP families. For PSP and ESP families, the high scoring features involved the sites that represent the hydrophobic cores and loops in the secondary structure. These spatial regions show greater variation across proteins, and are detected as representative of the family when a smaller family set is used.

5.5.5 Binary Classification

Having the functional site as the top-scoring feature is definitely desirable, but cannot be expected for any chosen family of proteins. There may be other high-scoring sites that are conserved and unique, but not defined as functional sites by the biologists. These sites are still of interest to biologists. Therefore, evaluating the LFM-Pro solely on the basis of detecting a set of known functional sites may not be appropriate. The success of LFM-Pro in extracting discriminative features can further be evaluated under a classification scheme.

In order to investigate the classification capabilities of LFM-Pro, we used a dataset that was previously utilized under a binary classification scheme [68], summarized in Table 5.1 (The complete list of proteins can be found on the supplementary web page). The first dataset (C_1) includes two families from different SCOP classes: nuclear receptor ligand-binding domain proteins (NB, 16 proteins) from all-alpha class, and the prokaryotic serine protease family (PSP, 10 proteins) from all-beta class. The second dataset (C_2) uses ESP (19 proteins) and PSP families which belong to the same superfamily. Note that PSP and ESP were used together above in the functional-site mining experiments. Whereas, the goal in this section is to evaluate the discrimination power of the representative feature sets for clearly distinct families (C_1) and closely related families (C_2). The proteins were selected from the *Culled-PDB* list [168] with less than 60% identity.

For families in datasets C_1 and C_2, the feature sets were extracted and scored as described above, and these representative feature sets were used for binary classification of proteins. The subgraph mining approach in [68] have achieved perfect accuracy for C_1 dataset, where the two families are from different SCOP classes, but had 5% classification error for the

Table 5.1: Protein families used for binary classification experiment.

Dataset	Family I	size	Family II	size
C_1	NB	16	PSP	10
C_2	PSP	10	ESP	19

C_2 dataset, in which the two families belong to the same superfamily. LFM-Pro classifies the proteins in both of these datasets with 100% accuracy, when all the extracted features were used in classification (Table 5.2). The methods Delaunay Tesselation (DT) and Almost Delaunay (AD) are from subgraph mining approach in [68]; results for the AD entry are given for a range of allowable perturbation values ($\epsilon = 0.1 - 0.75$). The fourth column shows the number of features that have *discrimination power* above 0.75, as defined by the authors; and the number of features required to obtain maximum accuracy in LFM-Pro. Accuracy is defined as the fraction of correct predictions measured by five-fold cross validation.

We attribute the success of LFM-Pro, in comparison with the graph mining approaches, to the fact that it can accommodate amino acid substitutions and displacements in the backbone, and focuses on the individual atoms within a spatial neighborhood rather than the coarser level information about location of CA atom of the amino acid residues.

Table 5.2: Binary classification performance.

Dataset	Method	Features	Dist.Feat	Accuracy
C_1	DT	20,646	934	100%
	AD	23,130–37,394	1,093–1,674	96–100 %
	LFM-Pro	5,282	1	100%
C_2	DT	15,895	20	95%
	AD	18,491–32,569	29–36	93–95 %
	LFM-Pro	2,180	139	100%

In LFM-Pro, each feature in the representative feature set contributes according to its corresponding score, which guarantees that the features that are not as discriminative as the top scoring features do not distort the classification, but only fine-tune it. However, it may be

desirable for efficiency and maintenance purposes, to keep only a small fraction of the top-scoring features for classification. Figure 5.7 shows the accuracy achieved using different number of features. Even though perfect accuracy was achieved in C_1 dataset using a single feature; the classification was more stable when more than 20 features are used. Considerably more features were required to distinguish the closely related families in the C_2 dataset.

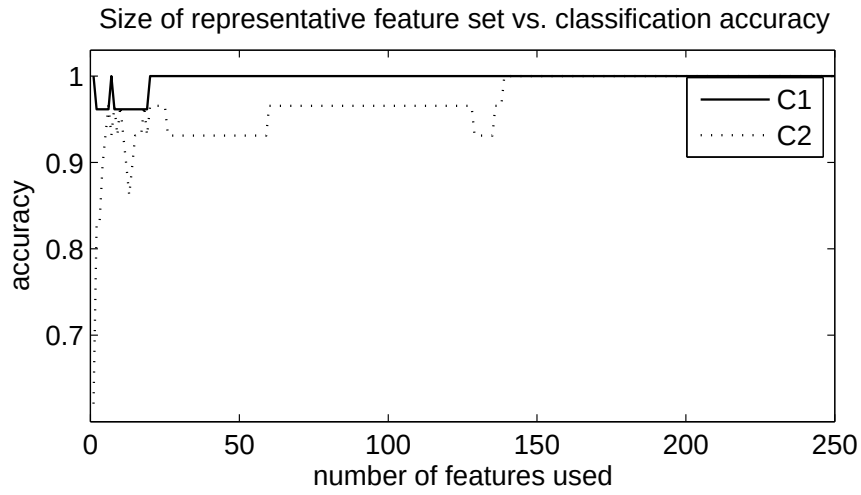


Figure 5.7: Number of features used in the representative feature set versus accuracy of the classification. The accuracy of using up to 250 features is shown here for illustration purposes, the accuracy value did not change beyond 250 features.

5.5.6 Multi-class Classification

In order to further validate our method, we performed a multi-class classification experiment on a more challenging dataset. Namely, the new entries introduced in SCOP 1.69 were classified based on family representations generated from SCOP 1.67. For both SCOP versions, ASTRAL dataset with less than 40% were used. The proteins or families that were re-classified in 1.69 and families that contain a training set less than 5 members were ignored. The final dataset contained 90 families with a total of 1,056 training proteins from SCOP 1.67 and 157 test proteins that were newly added in SCOP 1.69.

For comparison, the test proteins were also classified based on pairwise DALI [65] scores,

such that a query protein is assigned to the family of the protein with highest pairwise Z score. The results of multi-class classification experiment are tabulated in Table 5.2. The restriction of 40% homology in the dataset makes it particularly challenging. Moreover, an increase in the number of families result in higher number of false positives . DALI could only classify 31.2% of the test proteins correctly, whereas LFMPro obtained a classification accuracy of 37.58%.

Table 5.3: Multi-class classification accuracy. The training set is from SCOP 1.67 and test set is the newly added proteins in SCOP 1.69. The last row assumes that an oracle chooses the correct classification given by either method.

Method	Training Accuracy	Test Accuracy
DALI	100%	31.21%
LFMPro	100%	37.58%
DALI and LFMPro	100%	56.05%

Note that the proteins classified correctly by LFMPro are disjoint from those classified correctly by DALI. Combining DALI and LFMPro results and assuming an oracle to decide which one to use for a give protein, 56.05% accuracy is possible. Therefore, a classifier combining the output of these complementary methods would achieve higher accuracy, which is among our future research goals.

CHAPTER 6

INTEGRATED SEARCH AND ALIGNMENT OF PROTEIN STRUCTURES

6.1 Chapter Overview

[1] Identification and comparison of similar three dimensional (3D) protein structures has become an even greater challenge in the face of the rapidly growing structure databases. Here we introduce Vorometric, a new method that provides efficient search and alignment of a query protein against a database of protein structures. Voronoi contacts of the protein residues are enriched with the secondary structure information and a metric substitution matrix is developed to allow efficient indexing. The contact hits obtained from a distance-based indexing method are extended to obtain high scoring segment pairs, which are then used to generate structural alignments.

The benefits of Vorometric are demonstrated in several tasks including structure alignment, similarity search, and protein classification. In each of these tasks, Vorometric performs comparable or better than the popular structure search or alignment tools. The experimental results show that Vorometric is effective in retrieving similar protein structures, producing high quality structure alignments and identifying cross-fold similarities.

Availability: Vorometric is available as a web service at `http://bio.cse.ohio-state.edu/Vorometric`

[1] The study described in this chapter is currently under submission to the *Bioinformatics* journal.

6.2 Introduction

A tremendous amount of sequence and structure data is being produced with the motivation of deriving biological insights through comparison and analysis of similarities, differences, and interactions among biological macromolecules. Whereas the sequence comparison methods are generally sufficient for comparing proteins that share a high level of similarity, structure comparison becomes essential in discerning more distant evolutionary relationships. Moreover, the spatial organization of the protein residues provides stronger clues into the biochemical function of the proteins than can be derived from sequence information alone.

Pairwise structure alignment is the basic step for comparing protein structures. Finding the optimal alignment has been proven to be NP-hard [90], and several heuristics have been employed in the structure alignment tools DALI [65], CE [141], and MAMMOTH [119]. The rapidly increasing size of the protein databases, however, has made exhaustive pairwise structure alignment infeasible.

To overcome the difficulties presented by the database size, several strategies that aim to quickly identify relevant protein structures have recently been proposed. These strategies can best be summarized in terms of the choice of protein representation and the indexing method utilized for fast searching. ProGreSS [15] maps windows of protein backbone to a feature vector space using the curvature and torsion angles and the amino acid type information, and performs spatial indexing in this feature space. Protdex2 [5] represents the protein as a set of feature vectors of the inter-SSE contact regions and uses an inverted-file index for searching. Yakusa [26] describes the protein structure as a sequence of its backbone dihedral α angles and uses a method analogous to BLAST for searching blocks of this sequence. 3D-BLAST [159] clusters the κ and α angles to reduce the description to an alphabet and constructs a BLOSUM-like substitution matrix for this backbone angle alphabet, so that BLAST algorithm can be used without any modifications.

Currently available protein structure search methods provide database filtering, but defer a detailed structural alignment to further analysis by external alignment methods. More importantly, they focus on finding proteins that share similar overall topology or secondary structure composition, and are not sensitive to detect residue-level non-local interactions. Such non-local interactions are especially important in detecting functionally or evolutionarily sig-

nificant similarities among proteins that span multiple structural folds [45, 21].

In this study, we propose Vorometric as an integrated approach to both search and alignment tasks. We collect residue interactions from the protein structures using Voronoi tesselation and build a database of these *residue environments*. For a query protein, similar residue environments are retrieved from the database and extended to obtain high scoring segment pairs (HSPs), which are then used for structural superposition. We have developed a sensitive metric substitution matrix for accurate comparison of both amino acid and secondary structure information of related residue environments. Whereas an exhaustive search of similar residue environments in the database is prohibitive, our metric matrix has made distance-based indexing possible so that similar environments can be retrieved very efficiently. To the best of our knowledge, Vorometric is the first study employing distance-based indexing to protein structure data. The main benefits of our approach can be summarized as follows:

- The correspondences obtained from search and extension of residue environments endorse integrated and accurate structural superpositions, so that further structural alignment by external programs is no longer necessary.
- Unlike other protein structure search methods that at best capture the inter-SSE contacts, Vorometric provides contact sensitivity at the residue level.
- The hit & extend methodology inherently detects local, flexible structure alignments, a feature not commonly available in pairwise structure alignment methods.

We demonstrate the advantages and limitations of Vorometric using both quantitative performance evaluation on large scale datasets and on several detailed case studies. The experimental results show that Vorometric outperforms other structure search tools, and at the same time, yields high-quality structural alignments that are comparable or better than those produced by other structure alignment tools.

6.3 Methods

Vorometric is built on the observation that the residues in structurally similar protein structures share similar residue-residue contact interactions. We capture these interactions using

Voronoi tesselation and represent the contacts as a sequential string of residues. We incorporate both the amino acid type and secondary structure information into this representation. The contacts from all the proteins are then compiled in a database and metric-indexing is used for fast similarity search in this database. For a query protein, the contacts that are similar to those formed by its residues are searched in the database, and hits are extended for structural alignment. In the next few sections, we describe each of these steps in detail.

6.3.1 Representing the residue environments

Voronoi tesselation has previously been proposed as an effective method for extracting multi-body contacts from protein structures [86], and have been successfully utilized in packing analysis [126], protein folding [48], structure alignment, and structural motif mining [131]. [71] observed that structurally related proteins share common Voronoi contacts and used this observation to systematically match compatible tetrahedrons by shape, volume, and backbone topology in order to obtain candidate seeds for structure alignment. [128] and [18] use a different representation of the Voronoi contacts to obtain a sequential representation which allows direct use of dynamic programming. [128] measure the compatibility of the contacts through discretization of the Voronoi edge lengths, whereas [18] use another level of dynamic programming to compare the residue contacts.

We acknowledge that a sequential representation of the residue environments is very effective for their comparison, and utilize a similar representation in this study. We use the location of C_{alpha} atoms to represent the amino acids of a protein structure as a set of points in 3D space. The region of space around each point closer to the enclosed point than any other point defines a Voronoi polyhedron (See Figure 6.1). Delaunay tessellation is obtained by connecting the points that share a Voronoi boundary. For each residue, we define the set of all of its Delaunay neighbors, ordered by their sequence number along the backbone, as its *environment* (also denoted as *contact string*). This definition of contacts encodes much of the geometric proximity information and provides an abstract description of the underlying geometry. The length of the contact strings, generated from a large structure dataset, has an average of 11.6 and a maximum of 23 elements.

We encode the amino acid type and the secondary structure assignment of the residues in each

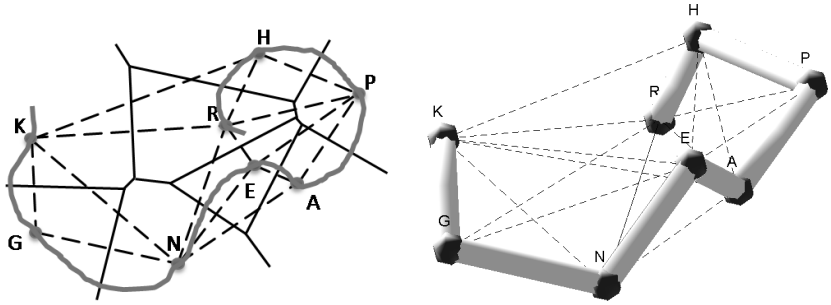


Figure 6.1: Delaunay tessellation (dashed lines) of a set of points in 2D and 3D. The Voronoi diagram is shown for only 2D (solid lines). The 2D curve represents a projection of the 3D backbone segment from beta2-microglobulin domain (3hla). The residue names are shown next to the C_α atoms.

contact string. For instance, the contact string for the second, Histidine residue in Figure 6.1 (3D) is: $R_C H^*_C P_C A_C E_C K_C$, where the secondary structure state is given in subscript notation, and the central residue is marked with an asterisk "*". We use DSSP [76] to obtain the secondary structure assignment for each residue and consider only the three primary states: alpha helix (H), beta sheet (E), and turns (C).

6.3.2 Comparison of the contact strings

An SSE-enriched *distance matrix* (described below) is used to compare individual elements of two contact strings (e.g., comparing a helix forming Arginine, R_H, with a beta-sheet forming Asparagine, N_E). The optimal alignment that minimizes the edit distance between two SSE-enriched strings with respect to the distance matrix can be obtained using the classical dynamic programming method by [113]. Since the contact strings are relatively short, global alignment with linear gap penalty provides sufficient accuracy in comparing contact strings.

Note that, even though most of the related proteins share similar Voronoi contacts, slight differences in the backbone configurations, or insertion and deletion of backbone segments may induce significantly different Voronoi contacts. Even for the same protein, the inherent noise in the X-ray crystallography or NMR spectroscopy techniques may result in different Voronoi contacts. Furthermore, when two related proteins are aligned around their backbone segments that share similar Voronoi contacts, the structural divergence at the boundaries of these segments can cause the residues flanking the aligned segments to have significantly

different Voronoi contacts. These boundary residues may be identical and may superimpose well, however, they would be penalized based on the contacts they form, and may not be included as part of the aligned segments.

In order to increase the robustness of the comparison measure, we consider the contacts and the central residues separately when aligning two contact strings. The distance between two contact strings E and F is defined as follows:

$$D(E,F) = d(E^{-},F^{-}) + \eta\, d(E^{0},F^{0}) + d(E^{+},F^{+}) \tag{6.1}$$

where d is the edit distance between two SSE-enriched sequences and η is a parameter used to adjust the importance of the similarity of the central residues (^{0}) compared to that of the contact residues preceding ($^-$) and following ($^+$) the central residues. Note that the edit distance between the central residues is simply a lookup in the distance matrix and does not require dynamic programming.

6.3.3 Metric SSE-enriched distance matrix

[137] has proved that if a metric distance matrix is used in the global alignment, then the resulting edit distance also forms a metric. There has been a number of efforts to construct metric amino-acid distance matrices [179]. On the other hand, a metric matrix that captures both the amino acid and SSE information is not available. Using an identity matrix is an obvious solution; however, the identity matrix is not sensitive to detect similarities between different types of amino acids.

We construct a 60x60 SSE-enriched distance matrix (M) using a weighted combination of a metric amino-acid distance matrix (N) that we have previously derived from 4-body Delaunay contact profiles of amino-acids [132] and a metric SSE exchange matrix (K) derived from an SSE similarity matrix [167] using the inter-row distance method [179]. The elements of M are defined as follows:

$$M(\langle a,s\rangle,\langle b,t\rangle) = w_1 N(a,b) + w_2 K(s_1,s_2) \tag{6.2}$$

where a and b are types of amino acids, s and t are the SSE states, and w_1,w_2 are positive weighing parameters to adjust the contributions of amino acid types and SSE states.

A distance matrix (or function) f is metric if the following properties are satisfied for any three elements x, y, and z:

1. *Positivity:* $f(x, y) \geq 0$
2. *Identity:* $f(x, y) = 0$ iff $x = y$
3. *Symmetry:* $f(x, y) = f(y, x)$
4. *Triangle Inequality:* $f(x, y) \leq f(x, z) + f(y, z)$

Now, we show that M, which is a weighted combination of the metric matrices N and K is also metric.

1. The weights and matrices in Eq. 6.2 are all positive, which makes M to be positive.
2. If $M(\langle a, s\rangle, \langle b, t\rangle) = 0$, then $N(a, b) = 0$ and $K(s, t) = 0$ from Eq. 6.2. Moreover, $a = b$, $s = t$ because N and K satisfy *identity*. Then, it follows that $\langle a, s\rangle = \langle b, t\rangle$. The reverse condition is also true using the same premises.
3. $w_1 N(a, b) + w_2 K(s, t) = w_1 N(b, a) + w_2 K(t, s)$ because both N and K are symmetric, therefore M is also symmetric.
4. $M(\langle a, s\rangle, \langle b, t\rangle) + M(\langle b, t\rangle, \langle c, u\rangle)$
   $= w_1(N(a, b) + N(b, c)) + w_2(K(s, t) + K(t, u))$
   $\geq w_1 N(a, c) + w_2 K(s, u) = M(\langle a, s\rangle, \langle c, u\rangle)$, therefore M also satisfies triangle inequality.

Note that, the distance function D defined for the contact strings is similar to M, in that it is also composed of a weighted combination of functions that are metric. According to the properties shown above, both M and D are metric.

6.3.4 Indexing and searching contact strings

Having a metric distance function D to compare contact strings allows us to utilize distance-based indexing for efficient retrieval. The main idea in distance-based indexing is to organize and partition the data into a hierarchical structure based on distances to representative elements of the partitions at each level. A partition whose representative entry is too dissimilar

to a query can then be pruned using the triangle inequality, without having to examine the rest of the entries in that partition. This allows an efficient and focused search over the data for entries similar to the query. (Please refer to [151] for a survey of distance-based indexing methods.) While any metric indexing method can be used to index and search the contact strings, we have implemented the Slim-tree method [157] which achieves sufficient time and memory performance for the large datasets used in this study.

We extract the contact strings from all of the protein structures in a dataset, and index them with respect to the distance function D. For a given query protein structure, we extract its contact strings, and search for similar entries in the database that are within the range δ from the query contact strings. The parameter δ specifies a threshold on the similarity of the contact strings being searched. A loose threshold would capture the contact strings of all protein structures that are similar to the query but may also result in many false positive hits. Whereas, a tight threshold would seek only the proteins that share highly conserved structural cores with the query.

6.3.5 Generating HSPs

The pseudocode for generating high scoring segment pairs (HSPs) from the contact string hits is outlined in Algorithm 1. The hits obtained for the individual residues of the query protein are first grouped based on which database proteins they belong to. These hits (also called *seeds*) correspond to a pair of residues, one from the query and one from the database protein, and are represented by the individual cells of the dynamic programming table as illustrated in Figure 6.2. Please note that the substitution score of each residue pair is defined by the similarity of their contact strings, so the hit extension phase is, in fact, a 2nd level of dynamic programming.

The extension heuristic employed for each seed is similar to that of BLAST sequence search tool [3] in that we also construct gapped local alignments in both forward and backward directions and only consider the cells in the dynamic programming table whose score falls no more than a fraction of the best score yet found. However, we introduce several notable enhancements over the basic method that increase the efficiency while maintaining the same level of sensitivity.

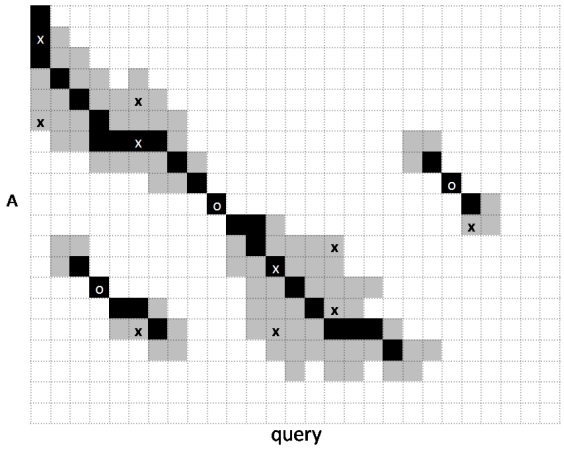


Figure 6.2: Illustration of the hit extension phase to obtain HSPs from the contact string hits from a database protein *A*. The seeds being extended are marked with "o", and those that are pruned are marked with "x". The gray area represents the cells that are explored by the dynamic programming and the black cells form the alignment paths of the HSPs.

The hits to a protein *A* are sorted based on their distances to the query contact strings, such that the more similar hits, which are more likely to be part of the final HSPs, are explored first. Naturally, we would expect many seeds on the alignment path of an HSP; extending each of these seeds would be redundant because they would find the same HSP path. We therefore skip the seeds whose residue pairings have already been explored by the extension of the previous seeds. This heuristic effectively eliminates about 42% of the seeds from consideration (based on randomized searches on the ASTRAL-25 database). Furthermore, to overcome the problem of generating many short HSPs, we merge a new HSP if its alignment path intersects with that of a previously generated HSP and if the merging produces a higher score than the individual HSP scores. This strategy results in merging 7% of the HSPs, which otherwise would have been generated as separate, shorter alignment segments.

6.3.6 Structure superposition

The residue correspondences defined by the HSP alignments are used to obtain a structural superposition of the query and the related database proteins. Finding the optimal transformation that minimizes the root mean square deviation (RMSD) between two structural alignments can be computed very fast: linear in the size of the proteins [75]. Following the iterative

Algorithm 1: Generate HSPs from contact string hits

Input: the contact string hits from the database

Output: $HSPs$: high scoring segment pairs

$HSPs \leftarrow []$;

foreach *protein A that has contact string hits* **do**
  $H \leftarrow$ sort hits to A by their distance to query strings;
  foreach *hit* $h \in H$ **do**
    if *h is already explored in dynamic programming table* **then**
      continue;
    $hsp \leftarrow$ `ExtendHit` (h);
    if *hsp can be merged into a previous* $hsp' \in HSPs$ **then**
      $hsp' \leftarrow$ `MergeHSP`(hsp', hsp);
    else if `Score`$(hsp) \geq \gamma$ **then**
      add hsp to $HSPs$;

optimization procedure commonly employed by the structure alignment tools, we derive a new set of correspondences from the superposition by finding the local alignment that minimizes the total distance of the aligned residues, and then repeat the iteration. The procedure is repeated until the transformation matrix no longer changes. Because the initial correspondences defined by the HSP alignments already optimize the structural compatibility of the aligned residues, the algorithm converges fast; in only a few iterations.

6.3.7 Parameter optimization

Parameters used in Vorometric are optimized on an independent training set using the Nelder-Mead simplex method [88]. The objective function used for the optimization was the geometric mean of the precision and recall values of the results returned by Vorometric and the TM-score [178] of the structural alignments between the queries and the resulting proteins.

The training set used for optimization was taken from the representative ASTRAL v1.73 database with 25% sequence identity [28]. We removed all the families that were used as queries in the evaluation of the Vorometric reported below, and kept remaining families that had at least 10 domains. From 13 such families, we randomly selected 10 members and assigned one of them to be the query and compiled the rest into a dataset. The training data is

available from the supplementary web site.

6.4 Experimental Results

Since Vorometric is proposed as a protein structure database search tool that at the same time produces high quality structure alignments, we compare its performance with that of both pairwise structure alignment and database search tools. In the next few sections, we first demonstrate that the structural alignments produced by Vorometric are in fact comparable or better than those of other pairwise structural alignment tools. We then show on large-scale experiments, that the structures in the database that are similar to a query protein are retrieved correctly, using the SCOP classifications [108] as the gold standard.

6.4.1 Quality of the structural alignments

In order to evaluate the quality of the structural alignments generated by Vorometric, we used the *ten difficult pairs* of protein structures that have previously been used to evaluate structural alignment methods [43]. A difficult pair is defined as a structurally similar pair that has a low sequence similarity and that had proven difficult to align with the available methods. For each pair, we use one of the proteins as query to search against the database composed solely of the other protein, and report the top-scoring HSP alignment. We compare the structural alignments produced by Vorometric with those by other popular structural alignment tools. The comparison is made using the RMSD deviation between the superimposed structures, the percentage of the query protein aligned (%N), and the TM-score [178]. The summary of the alignments is given in Table 6.2); detailed comparison for each pair of proteins can be found in Table 6.1. TM-score is a normalized measure of structural similarity (ranging from 0 to 1, with 1 being a perfect superposition) that simultaneously assesses the distance between the aligned residues and the length of the alignment, and has shown to agree with the results of human expert visual assessments.

Table 6.1: Detailed comparison of alignment quality on 10 difficult pairs.

			CE			SSAP			DaliLite			Vorolign*			Vorometric		
		%identity	rmsd	%N	TM	rmsd	%N	TM	rmsd	%N	TM	rmsd	%N	TM	rmsd	%N	TM
1ubq	1fxia	7	3.82	0.84	0.49	4.02	0.88	0.49	2.69	0.80	0.53	2.16	0.42	0.40	2.48	0.82	0.56
1ten	3hhrb	19	1.90	0.97	0.80	1.88	0.96	0.81	1.91	0.96	0.79	—	—	—	1.74	0.97	0.82
3hlab	2rhe	4	3.38	0.85	0.51	4.99	0.85	0.45	3.03	0.76	0.51	2.18	0.39	0.43	3.15	0.84	0.54
1paz	2azaa	12	2.86	0.70	0.52	3.41	0.73	0.53	2.46	0.68	0.53	2.26	0.59	0.63	2.73	0.71	0.54
1mola	1cewi	15	2.34	0.86	0.66	2.46	0.87	0.66	2.26	0.86	0.67	1.99	0.76	0.70	2.12	0.86	0.68
2rhe	1cid	11	2.91	0.85	0.64	3.72	0.90	0.64	3.02	0.83	0.63	1.95	0.58	0.62	2.79	0.87	0.67
1ede	1crl	5	3.85	0.71	0.57	9.25	0.86	0.46	3.43	0.67	0.56	3.22	0.34	0.44	4.86	0.76	0.56
2sim	1nsba	8	2.97	0.72	0.63	5.42	0.84	0.65	3.28	0.76	0.65	2.63	0.52	0.71	3.67	0.81	0.68
2gmfa	1bgeb	13	4.64	0.90	0.52	5.36	0.97	0.56	3.20	0.77	0.56	2.17	0.46	0.50	3.86	0.91	0.61
4fgf	1tie	10	3.04	0.94	0.70	3.23	0.96	0.69	2.88	0.90	0.70	2.00	0.59	0.63	2.79	0.94	0.72
average		10	3.17	0.83	0.60	4.37	0.88	0.59	2.82	0.80	0.61	2.28	0.52	0.56	3.02	0.85	0.65

* Vorolign reports alignments for multiple substitution matrices; here we use the SM-THREADER matrix [35], which gives the best results.

Table 6.2: Comparison of alignment quality on 10 difficult pairs.

method	RMSD (Å)	%N (% query aligned)	quality (TM-score)
CE	3.17	83.4	0.60
SSAP	4.37	**88.1**	0.59
DaliLite	2.82	80.0	0.61
Vorolign†	**2.28**	51.7	0.56
Vorometric	3.02	84.8	**0.65**

† Vorolign reports alignments for multiple substitution matrices; here we use the SM-THREADER matrix [35], which gives the best results.

CE and DaliLite give comparable coverage and RMSD values, whereas SSAP produces slightly longer alignments with significantly worse RMSD values. For instance, the alignment produced by SSAP for the 1ede-1crl pair had the worst RMSD (9.25Å) among all the alignments. 1ede and 1crl belong to the Alpha/Beta Hydrolases superfamily and are relatively large proteins (310 and 534 residues, respectively), having 8 beta strands wrapped around by 11 alpha helices. SSAP relies on aligning residues that share similar inter-residue distances; the high number of contacts formed by the residues at the core of these proteins makes their alignment difficult by SSAP.

Vorometric produces better alignments than any other method as measured by the TM-score. The coverage of the alignments by Vorometric are as long as those of SSAP's, while Vorometric at the same time achieves the best average RMSD, when compared with CE, SSAP, and DaliLite.

Vorolign [18], which is a pairwise alignment method also based on Voronoi contacts, generates the shortest alignments (about 30% smaller than Vorometric) and consequently, achieves better RMSD. However, the average alignment quality evaluated by TM-score is poorer than the other methods. Furthermore, Vorolign fails to generate an alignment for the 1ten-3hhrb pair. Both 1ten and 3hhrb are in the Fibronectin type III family; 1ten is composed of only one domain of the Immunoglobulin-like beta-sandwich fold, whereas 3hhrb contains two such domains, one of which aligns well with 1ten (see Figure 6.3). We attribute Vorolign's failure to its sensitivity to differences in residue contacts introduced by the additional domain in 3hhrb.

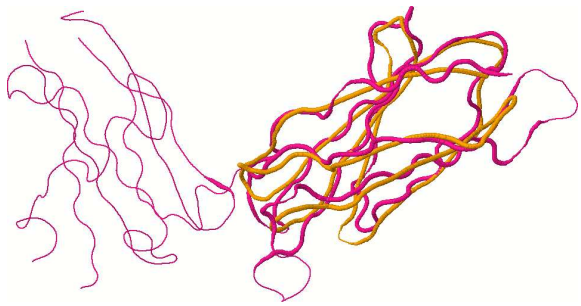


Figure 6.3: Structural alignment produced by Vorometric for 1ten (orange) and 3hhrb (pink). Aligned regions are shown thicker.

6.4.2 Database search for similar proteins

Large-scale comparison of different structure alignment or search methods is in itself a serious undertaking which is neither straightforward, nor completely fair, because each such method uses different databases and accuracy measures (see [83] for a comprehensive evaluation). Furthermore, some methods are made available only as a web service, which makes large-scale experimentation with newly crafted datasets impossible, if not prohibitive. For these reasons, we use the same dataset used by [5] and [159], and compare our results with those reported by them.

The dataset consists of 34,055 proteins which cover about 90% of the ASTRAL database. 108 queries are selected from medium-size families and have less than 40% sequence homology to each other. The precision of the results for different recall levels is shown in Figure 6.4 and summarized in Table 6.3. Even when the hits returned by Vorometric are ranked according to their raw HSP alignment scores (Vorometric-raw), the accuracy is better than other search methods and is comparable to that of detailed pairwise structure alignment methods CE and MAMMOTH, which indicates that the contact string representation and comparison used by Vorometric accurately captures the structural compatibility of the residues. When the results are ranked by their superposition TM-scores, Vorometric-TM achieves higher accuracy than any other method; giving slightly worse accuracy than MAMMOTH only above the 95% recall level.

Please note that CE [141] and MAMMOTH [118] are pairwise structure *alignment* methods, and for each query, they exhaustively scan the entire database. On the other hand, 3D-BLAST,

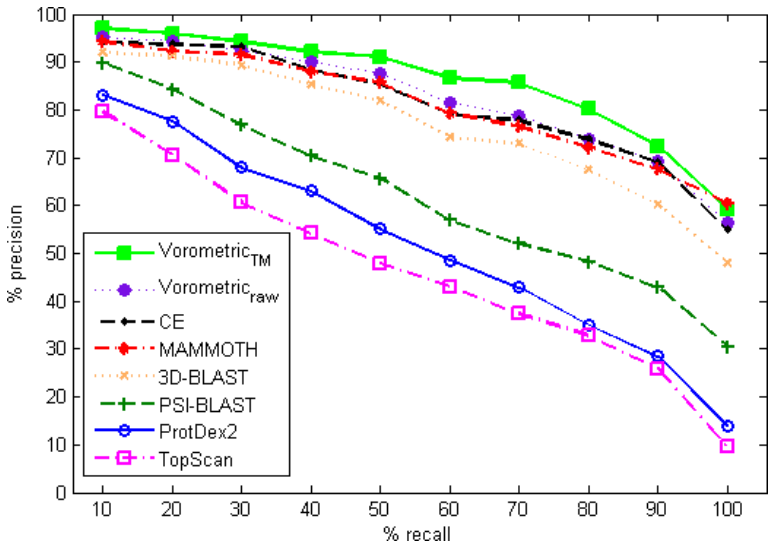


Figure 6.4: Average precision-recall curves for 108 queries on the database of 34,055 proteins.

ProtDex2, and TopScan [97] are structure *search* methods which are proposed as fast filters for similar structures, and do not produce structural superpositions. PSI-BLAST [3] is a sequence profile search method, which interestingly performs better and faster than ProtDex2 and TopScan. Vorometric achieves the best precision while running in a comparable time scale as the other database search methods, and additionally produces detailed structural superpositions for the returned hits.

6.4.3 Protein Classification

Another task that is closely related to the structure similarity search is to identify the structural or functional class of a protein via comparison with already annotated set of protein structures. In order to evaluate the classification performance, we used the dataset previously used by [18], where the difference set between SCOP v1.67 and v1.65 are queried against the ASTRAL-25 v1.65 containing 4,358 proteins. The classification performance is measured as the percentage of the 979 query proteins correctly classified when compared with their actual classifications in SCOP v1.67.

Although more elaborate voting schemes are possible for this task, the most commonly employed strategy is to assume that the query has the same class as the top-1 hit returned from

Table 6.3: Average precision and running times on the database of 34,055 proteins. Average precision is calculated as the mean of precision values for different recall levels. The time results for Vorometric are based on returning top 100 hits, performed on a Pentium 2.6 GHz personal computer. Vorometric-raw does not include the time spent for optimization of structural superposition, whereas Vorometric-TM does. The times for CE, MAMMOTH, 3D-BLAST, and PSI-BLAST are approximate values interpolated from [159] using the running times of CE as basis of comparison.

	avg. precision (%)	time per query	superposition
Vorometric-TM	**82.9**	51 sec	**yes**
Vorometric-raw	79.7	44 sec	no
CE	80.9	14 hours	**yes**
MAMMOTH	80.8	1.6 hours	**yes**
3D-BLAST	76.2	14 sec	no
PSI-BLAST	61.8	**8 sec**	no

a database search. In order to provide a fair comparison, we also use the top-1 hit for assignment. Vorometric-TM achieves the best classification accuracy in Family and Superfamily levels (Table 6.4), and only slightly worse accuracy than Vorolign at the Fold level. Note that the average structural divergence between the queries and their top hits are less for this dataset than that of the ten difficult pairs discussed above, which results in less pronounced differences in the alignment qualities. Nevertheless, Vorometric-TM produces longer alignments, while maintaining similar TM-score alignment quality.

Vorometric-raw, which uses the raw HSP alignment scores and does not generate structural superimpositions has similar classification accuracy as Vorolign and CE. SSEA [44] uses alignment of secondary structure elements to search the database, whereas BLAST is based on local alignment of primary sequences. The classification by these two database search methods are significantly worse than other methods. Please note that due to time constraints, [18] use SSEA to prefilter the database and use only the top 250 proteins to perform detailed pairwise structure alignment by Vorolign and CE. On the contrary, the integrated approach we employ in Vorometric relieves the dependence on pre-filtering the database with a coarse-level retrieval method.

A number of the misclassifications by Vorometric were due to low quality of the query entries. One of the extreme cases is 1oau:I; 85% of whose residues were not located in the X-ray experiment. A more subtle example is the 1r1g:A short-chain of scorpion neurotoxin,

Table 6.4: Classification of ASTRAL v1.65 - v1.67 difference set. Vorolign and CE scan only the top 250 proteins returned by SSEA. The classification accuracy and the structural alignment metrics are based on top-hit assignments and alignments.

	Family	Superfam	Fold	TM	%N	rmsd
Vorometric-TM	**90.7**	**94.9**	97.6	0.74	**87.2**	2.43
Vorometric-raw	85.9	91.2	97.0	—	—	—
Vorolign	86.4	92.4	**97.7**	0.74	76.3	**1.9**
CE	84.6	91.9	94.1	**0.77**	78.2	1.95
SSEA	60.8	68.9	75.6	—	—	—
BLAST	48.9	52.5	52.8	—	—	—

whose few missing residues cause the structural alignment with 1aho:A domain, a long-chain scorpion toxin of the same superfamily (TM-score 0.62, 28% sequence identity), better than that with the correct family member 1jlz:A (TM-score 0.28, **48%** identity).

A large fraction of the other misclassifications was due to the cross-fold similarities, especially in highly conserved domains such as the Immunoglobulin-like beta-sandwich, Zinc-finger, and OB-fold. It must also be noted that the SCOP classifications are based on not only structural similarity, but also functional and sequence similarity considerations, and even on the dimerization state of the proteins (e.g., the distinction between c.3.1.1 and c.3.1.5 families). As such, even though Vorolign places the structurally most similar protein as the top hit, it can be evaluated as a misclassification. For instance, 1urf:A of the a.2.6.1 family is structurally aligned better with 1lrz:A1 of the a.2.7.4 family, instead of 1cxz:B of the a.2.6.1 family. In most such cases, the correct family member were among the top few hits, and a simple analysis of the sequence homology was able to identify it correctly, indicating that a more elaborate strategy considering the top-k hits can be developed for highly accurate and fully automated classification of proteins.

6.4.4 Cross-fold similarities

We remark that there is no obvious or unambiguous way of clustering the proteins into discrete groups, and a significant number of overlaps will inevitably exist between proteins that are treated as unrelated based on hierarchical classification schemes [84]. While the ability to replicate these classifications demonstrate the performance of the structure search and align-

ment methods and is useful in functional annotation, we believe that the ability to identify the cross-fold similarities is also as important in identifying more distant evolutionary and functional relationships that may help understand the biochemical mechanisms of the particular biological function. While a systematic and exhaustive analysis of such cases is beyond the scope of this study, here we present a few examples to demonstrate that Vorometric is able to identify such relationships.

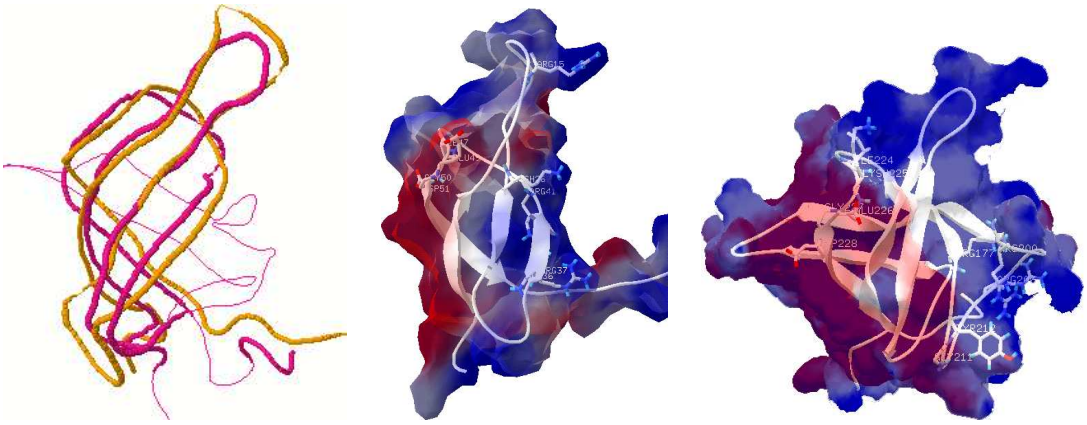
When Vorometric is used to query the first Ferrodoxin domain of the small subunit of FDH (1h0h:B, d.58.1.5), the second high scoring hit is the Immunoglobulin-binding domain of protein L (1hz6:A, d.15.7.1) (Figure 6.5a). The similarity betweeen these two proteins have previously been used to put forth a mechanism of structural drift during evolution [85]. Other significant cross-fold similarities were found between Beta-D-xylosidase (d1uhva1, b.71.1.2) and Chondroitin ABC lyase I (d1hn0a3, b.24.1.1), and between Sucrose phosphorylase (d1r7aa1, b.71.1.1) and Acidic mitochondrial matrix protein p32 (1p32:A, d.25.1.1).

(a) 1hoh:B-1hz6:A (b) 1lmr:A-1h59:B

Figure 6.5: Examples of cross-fold similarities.

We have also observed that many of the toxins share a significant structural similarity with proteins whose normal function is critical for the organism. The similarity between the Assasin bug toxin AD01 (1lmr:A, g.3.6.3) and Human Insulin-like growth factor-binding protein-5 (1h59:B, g.3.9.1) shown in Figure 6.5b; between the Chinese scorpion neurotoxin (1r1g:A, g.3.7.2) and Human transcription initiation factor TAF(II)18 (1bh9:A, a.22.1.3), and between short-chain scorpion Cobatoxin 1 (1pjva:A, g.3.7.2) and Human Methylation-dependent transcriptional repressor MBD1/PCM1 (1owt:A, d.230.3.1) are only some of such instances. We believe that a detailed analysis of these similarities may provide insight into

the biochemical mechanism of the toxins, and that of the respective proteins they mimic.



(a) 1ne3:A-1d1n:A (b) 1ne3:A (c) 1d1n:A

Figure 6.6: Ribosomal protein S28e (1ne3) and translation initiation factor IF2/eIF5b (1d1n:A).

Perhaps the most striking similarity we have discovered is that between the Ribosomal protein S28e (1ne3:A, b.40.4.5) and the translation initiation factor IF2/eIF5b (1d1n:A, b.43.3.1). While there is a significant body of work comparing IF2 with the translation elongation factors EF-tu and EF-G, it has hitherto not been compared with S28. It has been established that the initiation and elongation factors bind aminoacyl-tRNA, carry it to the ribosome, and detach from the ribosome after a conformational change caused by GTP hydrolysis, leaving the aminoacyl-tRNA attached at the A-site [4, 102]. On the other hand, the function of the ribosomal protein S28 is not characterized, although it has been conjectured to bind RNA, based on the analogy of its surface to the OB-fold proteins [173]. The Vorometric for S28e finds IF2 as a significant hit; their structure alignment (Figure 6.5c-e) shows a conserved [RK]EGD motif which provides a negatively charged site on both proteins. A comparison of their surface electrostatic potentials reveals a large, positively charged, Arginine-rich face on both proteins. The structure and surface similarities suggest that the ribosomal protein S28 may be responsible for taking over the aminoacyl-tRNA from the IF2 as it is being detached from the ribosome, and support the codon-anticodon binding as the peptidyl-tRNA is translocated from A-site to the P-site.

CHAPTER 7

DISCUSSION AND FUTURE RESEARCH

7.1 Contact-profile based amino acid substitution matrices

In Chapter 3, 4-body Delaunay contact profiles of amino acid residues were generated from a non-redundant set of protein structures and these contact profiles were then used to derive amino acid substitution matrices. We have investigated the effects of using different amino acid representations and different contact profile distance measures on the resulting matrices. The matrices derived using correlation measure were closely related to the NAOR [112] substitution matrix which is derived from amino acid substitutions observed at locally conserved but globally unrelated protein structures. Furthermore, principal component analysis of these matrices showed a strong correlation with hydrophobicity scale of amino acids, which inherently guides the contacts formed in the protein structures.

Alignment accuracies of the matrices have been illustrated using the BAliBASE multiple alignment dataset as reference. To the best of our knowledge, this is the first study to compare the alignment accuracies of the popular matrices on the comprehensive BAliBASE dataset. The performance of the matrices derived from Delaunay contacts were comparable to that of other matrices. It is interesting to see that the matrices we have derived, which do not rely on any evolutionary arguments or on observed substitution rates, can perform as such.

Multiple scoring matrices can be used to increase the reliability and significance of sequence alignments [46]. Availability of matrices based on different principals is important also for providing for the needs of specialized problems. Our studies using the Delaunay profile matrices in fold recognition are currently underway.

In applications where the distance, rather than similarity, between sequences is relavant, the similarity matrix is converted to a dissimilarity matrix by subtracting each matrix element from the maximum value in the matrix. Unfortunately, the matrices commonly used in practice fail to meet the *identity* condition of being a metric distance function, due to unequal values along the diagonal of the matrix. This results in positive distance values of a sequence to itself, which is undesirable in distance-based similarity measures. A unique feature of the Delaunay similarity matrices is the satisfaction of the *identity* condition in their dual dissimilarity matrices. This makes them suitable especially in distance-based indexing of protein databases for fast retrieval of similar sequences.

When Euclidean distance measure is used to compare the contact profiles, the resulting substitution matrices form a metric distance function that besides the identity property, satisfy the *triangle inequality* property. We have shown that these metric matrices perform sufficiently well in the sequence alignment tasks, which demonstrates that they are able to capture the interchangeability of amino acids accurately. The metric matrices produced in this study give the best alignment accuracy when compared with other available metric matrices. The biological sensitivity provided by these metric matrices is further exploited for efficient search of both protein sequence and structure data, in Chapters 4 and 6.

Although the matrices generated in this study happened to provide sufficient accuracy, they were not generated with the goal of giving the best alignment quality. Perhaps a more rigorous approach to obtaining biologically meaningful metric matrices is to define the task as an optimization problem with the objective of obtaining the best alignment quality on a reference set. This would involve a large optimization procedure involving both the substitution matrix elements and other alignment-related parameters (such as the gap penalties); forming a large search space and requiring super-computing facilities to sample and explore the search space for the optimal set of parameters.

7.2 Approximate sequence similarity search using landmark-guided embedding

In Chapter 4, we have proposed an approximation to similarity search in sequence databases by embedding the sequences in a vector space based on their distances to selected landmark

sequences. We have demonstrated that similarity search in the embedded space can be performed several orders of magnitude faster than that in the original sequence space, without significant loss in the accuracy of the search results. Fastmap and LMDS methods with various landmark selection heuristics are investigated for their embedding and similarity search accuracy.

While the Fastmap method provides a good heuristic for landmark selection, its numerical instability causes degradation in the mapping accuracy. Moreover, Fastmap tends to overestimate the original distances compared to LMDS methods, causing a lower sensitivity in similarity search results. We have proposed $LMDS_{fastmap}$ method which uses the landmarks generated by the Fastmap, yet provides stability in the mapping, yielding better performance in mapping and similarity search. The mapping accuracy achieved by the embedding methods can be further improved using higher dimensionality in the embedding space, or using a larger number of landmark sequences. We have presented a systematic comparison of the performance on synthetic and real sequence datasets.

In this study, we have mainly focused on the *kmer search*, which constitutes a significant initial step in the general homology search problem. These short subsequences can then be extended and stitched to obtain the final sequence alignments. We note that the efficiency of the embedding and the indexing will further depend on the subsequence extension algorithm used. We refer the reader to [59] for details of these algorithms.

We expect the vector representation of sequences to have applications beyond similarity search. For instance, a vector domain simplifies the representation of a group of sequences by their mean vector, which otherwise is not readily available in the sequence domain. We are currently investigating the use of such abstract representations in sequence clustering and multiple sequence alignment applications.

The landmark embedding methods presented here can also be applied to content-based retrieval in other domains such as image and multimedia databases. In such applications, the original space is a high-dimensional space formed by various features extracted from the database objects. The landmark-guided embedding would provide a dimensionality reduction and allow efficient similarity search. Furthermore, the similarity search in these applications are especially tolerant of the approximation errors incurred by the embedding, because the original features and the classification of objects are already very subjective, and approximate

results are considered satisfactory.

7.3 Mining for local structural sites

In Chapter 5, we have presented a data-mining based framework, *Local Feature Mining in Proteins* (*LFM-Pro*), whereby topologically and biochemically conserved regions of a protein family could be automatically discovered. We have demonstrated the success of the method on Serine Protease family of proteins and also on two binary classification datasets. The sites unique to a family of proteins were identified via comparison to a background set of proteins. We have confirmed that the sites detected by our method conforms with the previously reported functional sites. When a background set of proteins is not provided, LFM-Pro scores the local sites based on how common they are across the family proteins.

LFM-Pro gives the most desirable site-mining results when the family being studied contains proteins that are evolutionarily distant but share the same site of interest, and when the background family is chosen to contain proteins that share the same structural folds with the family being studied. The objective of maximizing the discriminative scores can be used to determine the optimal size of the background set in feature mining, and the optimal number of features in classification.

LFM-Pro uses feature vectors associated with local neighborhoods that provides comprehensive sampling of the protein space. One of the major advantages of a feature-based approach is the computational efficiency; because the time-consuming graph matching or structural alignment steps are no longer required. Moreover, the feature vectors can be stored in an index structure optimized for range queries, which would further improve the efficiency of the algorithm. A custom filtering step to remove features related to trivial secondary structures can also be performed to reduce the number of candidate features, which would further increase the efficiency of the algorithm.

The framework presented in this study is easily extensible to more sophisticated feature extraction and scoring schemes. One may, for example, augment the features presented here with physico-chemical features such as hydrophobicity, solvent accessibility, or mobility. It would also be interesting to investigate critical points of other function fields, such as force fields or electrostatic potential. Note that we utilized a simple unweighted Euclidean distance

function for measuring the dissimilarity between feature vectors, and it was our experience that the algorithm allowed imperfect distance functions. However, fine-tuning the weights of the spatial features may be desirable in order to highlight the contributions of each feature in the representation of local sites. The weights of the distance function can be automatically optimized with the objective of maximizing the discriminative scores of the representative set. We have provided in the software distribution of LFM-Pro, a *simulated annealing* approach for such fine-tuning.

Using *local* structural and biochemical features as opposed to structural alignment of proteins, can potentially yield in identification of very distant evolutionary relationships, and can help discern the function of yet uncharacterized proteins. Local sites of the proteins resist evolutionary modifications if they perform an important biological function, whereas the rest of the protein simply provides a scaffold and is more prone to modifications through mutation, insertion, deletion, and duplication events. Therefore, related proteins can share a common evolutionary ancestry or a common biological function, which may only be identifiable through comparison of these local sites.

Inference of remote homology is also a key step in evolutionary-based cataloguing of all available protein structures. Assigning a new protein to unique positions in the classification scheme becomes impossible when the homology is not detectable. Using LFM-Pro, it is possible to identify a distinguishing representative feature set for each family, and to quickly assign a new protein to one (or more, for multi-domain proteins) of these families. For instance, using the representative feature set generated by LFM-Pro for Globins family of proteins, we were able to discover proteins 1uby, 1gai, and 1xis to have similar distinctive sites as the Globins. These three proteins were not previously classified to have structural or functional similarities with Globins; however, a multiple alignment revealed that they could indeed be significantly aligned with Globins, confirming the detection by LFM-Pro.

Effective discovery of functional local motifs would have tremendous impact in bioscience research, and would find applications in areas such as multiple structural alignment, protein modeling, drug design and targeting. As a future work, we plan to undertake a large-scale, systematic study where we would extract representative feature sets for all SCOP families, and provide them as a publicly available motif database. The feature vectors extracted from the proteins also lend themselves for an unsupervised learning method where unique functional

sites could be automatically discovered without any prior family-membership information.

While identification of the conserved sites and their constituent residues provides a valuable information, we note that the biologist often wants to know the exact residue correspondences from different proteins that share the same conserved site. While such a post-processing does not affect the analysis performed for our approach, it would greatly enhance its usability and the interpretation of the results by the biologists. The task of finding such residue correspondences is the subject of multiple sequence and structure alignment problems, for which there are numerous tools available, such as CLUSTALW [111] and T-Coffee [123] for sequences and CE-MC [58] and Multiprot [139] for structures.

7.4 Integrated Search and Alignment of Protein Structures

In Chapter 6, we focus on protein structure search and comparison problems. We employ a *hit & extend* methodology which first identifies residues that share similar contacts, and then performs alignment-extension using these residues as seeds. While Vorometric is presented as a specific implementation, it directs to a more general, extensible framework of structural search and alignment. Particularly, different substitution matrices or distance functions that incorporate geometrical or biochemical nature of the residue environments can be developed and used in Vorometric without any changes to the rest of the algorithm, provided that they satisfy metric properties, or permit other efficient indexing strategies. The extension phase of Vorometric can also incorporate other filters for candidate evaluation, or other structural compatibility functions, such as filtering by volume or surface accessibility of the contact environments. Similarly, structural superposition of the seed contact environments can be performed to measure structural compatibility more accurately, before proceeding with the extension phase. These additional filters would reduce the number of candidate seeds that need to be considered for extension, increasing the speed of the overall algorithm.

The pairwise structural alignment methods hitherto proposed rely on coarse-level filtering methods to scan the database of protein structures for candidates that are worthy of alignment. Unlike previous methods, Vorometric is introduced as a fast protein structure database search and alignment tool that uses the same sensitive representation of residue interactions for both identifying similar proteins and generating high-quality structural alignments. The heuristic

that structurally similar proteins share similar residue interactions is exploited through a metric comparison of these interactions which has allowed efficient distance-based indexing and retrieval of related proteins.

The additional accuracy achieved by Vorometric does not incur significant time and memory requirements. The whole index structure for the large dataset of 34,055 proteins needs less than 600 MB, and is kept in the main memory for fast access. The speed achieved by the distance-based indexing method is complemented by the hit-extension strategy, which allows fast exploration of the search space by effectively pruning redundant or unpromising hits. The search of a query protein against a large database takes less than a minute, including detailed superposition of the retrieved proteins.

Evaluation of Vorometric on large-scale datasets shows that it provides the accuracy of pairwise structural alignment tools and the speed of database search tools. Vorometric performs better than other methods on the database search and classification tasks and produces longer, high quality structure alignments, relieving the dependence on separate structural alignment tools. Finally, Vorometric successfully identifies cross-fold similarities between proteins so that distant evolutionary and functional relationships can be discerned. Our future research focus involves applying the ideas developed in Vorometric to the problems of structural motif discovery and multiple structure alignment.

7.5 Conclusion

One of the main issues that remains to be investigated is the definition of residue contacts. While several contact definitions have been explored in the literature under different contexts, a comprehensive evaluation of different contact definitions in capturing biologically important residue interactions is still missing. Delaunay tessellation used in our studies provides a natural and well-formed definition that associates nearby residues and captures the local geometry around each residue. However, Delaunay tessellation is not robust to noise or differences in the structural data, and the contacts defined by the Delaunay tessellation are not necessarily biologically interacting residues. A more elaborate definition of contacts that considers spatial and physico-chemical interactions between characteristic atoms on the amino acid side-chains may yield biologically more meaningful results.

We note that similar to the shift from sequence to structure as a functionality more informative type of data, bioinformatics research will soon experience a shift of interest from structure to protein *surface* analysis, which provides a representation that is more amenable to how proteins interact and function. While a number of recent efforts [16] have taken up the task of providing methods for surface comparison and analysis, we still need robust, efficient, intuitive, and publicly available implementations for protein surface analysis. We believe that the application of residue-contacts based approaches developed in this thesis to the protein surface search and comparison problems would complement the protein function and homology information provided by the sequence and structure analysis methods.

Finally, we remark that the approaches and tools developed in this thesis for protein sequence and structure analysis are nowhere near final; and like any other available method, they are bound to evolve to better fit the needs of the biological researchers. We believe that making their implementations accessible to the wider community as we did, will help to further identify their strengths and weaknesses, and guide toward further extension and development.

REFERENCES

[1] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. *Molecular Biology of the Cell, Fourth Edition*. Garland, 2002.

[2] S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.

[3] S. Altschul, T. Madden, A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*, 25:3389–3402, 1997.

[4] G. R. Andersen, P. Nissen, and J. Nyborg. Elongation factors in protein biosynthesis. *Trends in Biochemical Sciences*, 28(8):434–441, 2003.

[5] Z. Aung and K. Tan. Rapid 3D protein structure database searching using information retrieval techniques. *Bioinformatics*, 20:1045–1052, 2004.

[6] S. C. Bagley and R. B. Altman. Characterizing the microenvironment surrounding protein sites. *Protein Sci*, 4:622–635, 1995.

[7] C. Barber, D. Dobkin, and H. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483, 1996.

[8] J. A. Barker and J. M. Thornton. An algorithm for constraint-based structural template matching: application to 3D templates with statistical analysis. *Bioinformatics*, 19(13):1644–1649, 2003.

[9] J. Battey, E. Jordan, D. Cox, and W. Dove. An action plan for mouse genomics. *Nature Genet.*, 21:73–75, 1999.

[10] N. Beckmann, H. Kriegel, R. Schneider, and B. Seeger. The R*-tree: an efficient and robust accessmethod for points and rectangles. *ACM SIGMOD*, pages 322–331, 1990.

[11] D. A. Benson, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and D. L. Wheeler. Genbank: update. *Nucleic Acids Res.*, 32, Jan 1:D23–26, 2004.

[12] S. Berchtold, D. A. Keim, and H.-P. Kriegel. The X-tree: An index structure for high-dimensional data. In *VLDB*, pages 28–39, 1996.

[13] H. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. Shindyalov, and P. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28:235–242, 2000.

[14] A. Bernal, U. Ear, and N. Kyrpides. Genomes OnLine Database (GOLD): a monitor of genome projects world-wide. *NAR*, 29:126–127, 2001.

[15] A. Bhattacharya, T. Can, T. Kahveci, A. K. Singh, and Y.-F. Wang. Progress: Simultaneous searching of protein databases by sequence and structure. *Pacific Symposium on Biocomputing*, 9:264–275, 2004.

[16] T. A. Binkowski, A. Joachimiak, and J. Liang. Protein surface analysis for function annotation in high-throughput structural genomics pipeline. *Protein Science*, 14:2972–2981, 2005.

[17] T. A. Binkowski, S. Naghibzadeg, and J. Liang. Castp: computed atlas of surface topography of proteins. *Nucleic Acid Research*, 31:3352–3355, 2003.

[18] F. Birzele, J. E. Gewehr, G. Csaba, and R. Zimmer. Vorolign: fast structural alignment using Voronoi contacts. *Bioinformatics*, 23(2):e205–e211, 2007.

[19] T. Bozkaya and M. Ozsoyoglu. Indexing large metric spaces for similarity search queries. *ACM Transactions on Database System*, 24(3):361–404, 1999.

[20] S. E. Brenner, P. Koehl, and M. Levitt. The astral compendium for sequence and structure analysis. *Nucleic Acids Research*, 28:254–256, 2000.

[21] S. Brown, J. Gerlt, J. Seffernick, and P. Babbitt. A gold standard set of mechanistically diverse enzyme superfamilies. *Genome Biology*, 7(1):R8, 2006.

[22] A. Buchner and H. Taubig. A fast method for motif detection and searching in a protein structure database. Technical Report TUM-I0314, Computer Science Dept., TU Munchen, 2003.

[23] S. Burkhardt, A. Crauser, P. Ferragina, H. P. Lenhof, and M. Vingron. q-gram based database searching using a suffix array (quasar). *In Int. Conf. RECOMB, Lyon*, April 1999.

[24] O. Camoglu, T. Kahveci, and A. K. Singh. Psi: indexing protein structures for fast similarity search. *Bioinformatics*, 19:181–183, 2003.

[25] X. Cao, S. C. Li, B. C. Ooi, and A. K. H. Tung. Piers: An efficient model for similarity search in dna sequence databases. *SIGMOD Record*, 33(2), June 2004.

[26] M. Carpentier, S. Brouillet, and J. Pothier. Yakusa: a fast structural database scanning method. *Proteins*, 61:137–151, 2005.

[27] CGAL. The cgal project-release 3.1, 2006. `http://www.cgal.org/`, accessed 26-June-2008.

[28] J. Chandonia, G. Hon, N. Walker, L. L. Conte, P. Koehl, M. Levitt, and S. Brenner. The ASTRAL compendium in 2004. *Nucleic Acids Research*, 32:189–192, 2004.

[29] L. Chen, R. Oughtred, H. M. Berman, , and J. Westbrook. Targetdb: a target registration database for structural genomics projects. *Bioinformatics*, 20(16):2860–2862, 2004.

[30] P. Ciaccia, M. Patella, and P. Zezula. M-tree: An efficient access method for similarity search in metric spaces. *Proc. Intl. Conf. on Very Large Databases (VLDB)*, pages 426–435, 1997.

[31] A. M. Cohen and W. R. Hersh. A survey of current work in biomedical text mining. *Brief Bioinform*, 6(1):57–71, 2005.

[32] F. Crick. On Protein Synthesis. *Symp. Soc. Exp. Biol.*, XII:139–163, 1958.

[33] M. O. Dayhoff, R. M. Schwartz, and B. Orcutt. A model of evolutionary change in proteins. *In Atlas of Protein Sequence and Structure 5(3) M.O. Dayhoff (ed.), National Biomedical Research Foundation, Washington.*, pages 345–352, 1978.

[34] V. de Silva and J. B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. *Proc. NIPS*, 15:721–728, 2003.

[35] Z. Dosztanyi and A. Torda. Amino acid similarity matrices based on force fields. *Bioinformatics*, 17:686–699, 2001.

[36] J. Dundas, Z. Ouyang, J. Tseng, A. Binkowski, Y. Turpaz, and J. Liang. CASTp: computer atlas of surface topography of proteins with structural and topographical mapping of functionally annotated residues. *Nucleic Acids Researc*, 34:116–118, 2006.

[37] H. Edelsbrunner, M. A. Facello, and J. Liang. On the definiton and the construction of pockets in macromolecules. *Discrete Applied Mathematics*, 88(1–3):83–102, 1998.

[38] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. *Discrete Comput. Geom.*, 28:511–533, 2002.

[39] P. Edman. Preparation of phenyl thiohydantoins from some natural amino acids. *Acta Chem. Scand.*, pages 277–282, 1950.

[40] C. Faloutsos and K.-I. Lin. FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, pages 163–174, 1995.

[41] G. Fasman. *Prediction of Protein Structure and the Principles of Protein Conformation*. Plenum, New York, 1989. Table XVII.

[42] R. Finn, J. Mistry, B. Schuster-Böckler, S. Griffiths-Jones, V. Hollich, T. Lassmann, S. Moxon, M. Marshall, A. Khanna, R. Durbin, S. Eddy, E. Sonnhammer, and A. Bateman. Pfam: clans, web tools and services. *Nucleic Acids Research Database Issue*, 34:D247–D251, 2006.

[43] D. Fischer, A. Elofsson, D. Rice, and D. Eisenberg. Assessing the performance of fold recognition methods by means of a comprehensive benchmark. *Pacific Symposium on Biocomputing*, pages 300–318, 1996.

[44] P. Fontana, E. Bindewald, S. Toppo, R. Velasco, G. Valle, and S. C. E. Tosatto. The SSEA server for protein secondary structure alignment. *Bioinformatics*, 21(3):393–395, 2005.

[45] I. Friedberg and A. Godzik. Connecting the protein structure universe by using sparse recurring fragments. *Structure*, 13(8):1213–1224, 2005.

[46] F. Frommlet, A. Futschik, and M. Bogdan. On the significance of sequence alignments when using multiple scoring matrices. *Bioinformatics*, 20(6):881–887, 2004.

[47] F. B. Fuller. Decomposition of the linking number of a closed ribbon: a problem from molecular biology. In *Proc. Natl. Acad. Sci. USA*, volume 75, pages 3557–3561, 1978.

[48] H. Gan, A. Tropsha, and T. Schlick. Lattice protein folding with two and four-body statistical potentials. *Proteins*, 43:161–174, 2001.

[49] J. F. Gibrat, T. Madej, and S. H. Bryant. Surprising similarities in structure comparison. *Curr. Opin. Struct. Biol.*, 6(3):377–385, 1996.

[50] J. Giesen and M. John. The flow complex: A data structure for geometric modeling. *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 285–294, 2003.

[51] D. R. Gilbert, D. R. Westhead, N. Nagano, and J. M. Thornton. Motif-based searching in TOPS protein topology database. *Bioinformatics*, 5(4):317–326, 1999.

[52] R. S. C. Goble, P. Baker, and A. Brass. A classification of tasks in bioinformatics. *Bioinformatics*, 17:180–188, 2001.

[53] G. H. Gonnet, M. A. Cohen, and S. A. Benner. Exhaustive matching of the entire protein sequence database. *Science*, 256:1443–1445, 1992.

[54] T. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.

[55] P. Goodford. A computational procedure for determining energetically favorable binding sites on biologically important macromolecules. *J Med Chem*, 28:849–857, 1985.

[56] O. Gotoh. Optimal sequence alignment allowing for long gaps. *Bull. Math. Biol.*, 52:359–373, 1990.

[57] R. Grantham. Amino acid difference formula to help explain protein evolution. *Science*, 185:862–864, 1974.

[58] C. Guda, E. D. Scheeff, P. E. Bourne, and I. N. Shindyalov. A new algorithm for the alignment of multiple protein structures using Monte Carlo optimization. *Proceedings of the Pacific Symposium on Biocomputing*, 6:275–286, 2001.

[59] D. Gusfield. *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*. Press Syndicate of the University of Cambridge, USA, 1997.

[60] A. Guttman. R-trees: A dynamic index structure for spatial searching. *ACM SIGMOD*, pages 419–429, 1984.

[61] H. Hegyi and M. Gerstein. The relationship between protein structure and function: a comprehensive survey with application to the yeast genome. *J Mol Biol*, 288:147–64, 1999.

[62] S. Henikoff and J. Henikoff. Amino acid substitution matrices from protein blocks. *Proc. Natl Acad. Sci. USA*, 89:10915–10919, 1992.

[63] U. Hobohm and C. Sander. Enlarged representative set of protein structures. *Protein Science*, 3:522–524, 1994.

[64] T. C. Hodgman. The elucidation of protein function by sequence motif analysis. *Computer Appl. in the Biosci. (CABIOS)*, 5:1–13, 1989.

[65] L. Holm and C. Sander. Protein structure comparison by alignment of distance matrices. *J. Mol. Biol.*, 233:123–138, 1993.

[66] L. Holm and C. Sander. Dali/FSSP classification of three-dimensional protein folds. *Nucleic Acids Res*, 25(1):231–234, 1997.

[67] J. Huan, D. Bandyopadhyay, W. Wang, J. Snoeyink, J. Prins, and A. Tropsha. Comparing graph representations of protein structure for mining family-specific residue-based packing motifs. *Journal of Computational Biology*, 12:6:657–71, 2005.

[68] J. Huan, W. Wang, D. Bandyopadhyay, J. Snoeyink, J. Prins, and A. Tropsha. Mining protein family specific residue packing patterns from protein structure graphs. *Proc. of 8th Ann. Intl. Conf. on Research in Comp. Molecular Bio. (RECOMB)*, pages 308–15, 2004.

[69] E. Hunt, M. P. Atkinson, and R. W. Irving. A database index to large biological sequences. *In International Journal on VLDB, Roma, Italy*, pages 139–148, September 2001.

[70] L. Hunter. *Molecular biology for computer scientists*, chapter 1, pages 1–46. AAAI Press, 1993.

[71] V. A. Ilyin, A. Abyzov, and C. M. Leslin. Structural alignment of proteins by a novel TOPOFIT method, as a superimposition of common volumes at a topomax point. *Protein Sci.*, 13:1865–1874, 2004.

[72] M. Johnson and J. Overington. A structural basis for sequence comparisons. an evaluation of scoring methodologies. *J. Mol. Biol.*, 233:716–738, 1992.

[73] I. Jonassen, I. Eidhammer, D. Conklin, and W. R. Taylor. Structure motif discovery and mining the PDB. *Bioinformatics*, 18(2):362–367, 2001.

[74] M. Justice. Mouse ENU mutagenesis. *Hum. Mol. Genet.*, 8:1955–1963, 1999.

[75] W. Kabsch. A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Crystallogr.*, A34:827–828, 1978.

[76] W. Kabsch and C. Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, pages 2577–637, 1983.

[77] T. Kahveci and A. Singh. An efficient index structure for string databases. *In VLDB*, pages 351–360, 2001.

[78] S. Kawashima, P. Pokarowski, M. Pokarowska, A. Kolinski, T. Katayama, and M. Kanehisa. AAindex: amino acid index database, progress report 2008. *Nucleic Acids Res.*, 36:D202–D205, 2008.

[79] K. Kedem, L. Chew, and R. Elber. Unit-vector rms (urms) as a tool to analyze molecular dynamics trajectories. *Proteins*, 37:554–564, 1999.

[80] J. C. Kendrew, G. Bodo, H. M. Dintzis, R. G. Parrish, H. Wyckoff, and D. C. Phillips. A Three-Dimensional Model of the Myoglobin Molecule Obtained by X-Ray Analysis. *Nature*, 181(4610):662–666, 1958.

[81] K. Klenin and J. Langowski. Computation of writhe in modeling of supercoiled DNA. *Biopolymers*, 54:307 – 317, 2000.

[82] G. J. Kleywegt. Recognition of spatial motifs in protein structures. *J Mol Biol*, 285(4):1887–1897, 1999.

[83] R. Kolodny, P. Koehl, and M. Levitt. Comprehensive evaluation of protein structure alignment methods: Scoring by geometric measures. *J. Mol. Biol.*, 346:1173–1188, 2005.

[84] R. Kolodny, D. Petrey, and B. Honig. Protein structure comparison: implications for the nature of fold space, and structure and function prediction. *Current Opinion in Structural Biology*, 16(3):393–398, 2006.

[85] S. S. Krishna and N. V. Grishin. Structural drift: a possible path to protein fold change. *Bioinformatics*, 21(8):1308–1310, 2005.

[86] B. Krishnamoorthy and A. Tropsha. Development of a four-body statistical pseudo-potential to discriminate native from non-native protein conformations. *Bioinformatics*, 18(12):1540–1548, 2003.

[87] N. Kyrpides. Genomes OnLine Database (GOLD): a monitor of complete and ongoing genome projects world wide. *Bioinformatics*, 15:773–774, 1999.

[88] J. C. Lagarias, J. Reeds, M. H. Wright, and P. E. Wright. Convergence properties of the Nelder-Mead simplex method in low dimensions. *SIAM Journal of Optimization*, 9(1):112–147, 1998.

[89] E. Lander and D. Botstein. Mapping mendelian factors underlying quantitative traits using RFLP linkage maps. *Genetics*, 121:185–199, 1989.

[90] R. Lathrop. The protein threading problem with sequence amino acid interaction preferences is NP-complete. *Protein Eng.*, pages 1059–1068, 1994.

[91] M. Levitt. Protein folding by restrained energy minimization and molecular dynamics. *J. Mol. Biol.*, 170:723–764, 1983.

[92] H. Li and S. Parthasarathy. Automatically deriving multi-level protein structures through data mining. In *HiPC Workshop on Bioinformatics and Computational Biology*, 2001.

[93] J. Liang, H. Edelsbrunner, and C. Woodward. Anatomy of protein pockets and cavities: measurement of binding site geometry and implication for ligand design. *Protein Science*, 7:1884–1897, 1998.

[94] M. P. Liang, D. R. Banatao, T. E. Klein, and D. L. Brutlag. Webfeature: An interactive web tool for identifying and visualizing functional sites on macromolecular structures. *Nucleic Acids Res*, 31:3324–3327, 2003.

[95] K. Liolios, K. Mavrommatis, N. Tavernarakis, and N. Kyrpides. The Genomes On Line Database (GOLD) in 2007: status of genomic and metagenomic projects and their associated metadata. *NAR Database issue*, 2008. in press.

[96] B. Ma, J. Tromp, and M. Li. Patternhunter: faster and more sensitive homology search. *Bioinformatics*, 18:440–445, 2002.

[97] A. Martin. The ups and downs of protein topology: rapid comparison of protein structure. *Protein Eng.*, 13:829–837, 2000.

[98] MATLAB. Matlab user's guide. *The MathWorks, Inc., Natick, MA 01760*, 1992.

[99] A. May. Towards more meaningful hierarchical classification of amino acid scoring matrices. *Protein Eng.*, 12:707–712, 1999.

[100] V. McKusick. *Mendelian Inheritance in Man: A Catalog of Human Genes and Genetic Disorders.* Johns Hopkins University Press, 12 edition.

[101] C. Meek, J. M. Patel, and S. Kasetty. Oasis: An online and accurate technique for local-alignment searches on biological sequences. *In Proc. 2003 Int. Conf. Very Large Data Bases (VLDB*, pages 910–921, 2003.

[102] S. Meunier, R. Spurio, M. Czisch, R. Wechselberger, M. Guenneugues, C. O. Gualerzi, , and R. Boelens. Structure of the fMet-tRNAfMet-binding domain of B.stearothermophilus initiation factor IF2. *The EMBO Journal*, 19:1918–1926, 2000.

[103] M. Milik, S. Szalma, and K. Olszewski. Common structural cliques: a tool for protein structure and function analysis. *Protein Engineering*, 16:8:543–52, 2003.

[104] T. Milledge, G. Zheng, T. Mullins, and G. Narasimhan. Sblast: Structural basic local alignment searching tools using geometric hashing. *Proceedings of the 7th IEEE International Conference on Bioinformatics and Bioengineering*, pages 1343–1347, 2007.

[105] S. Miyazawa and R. Jernigan. A new substitution matrix for protein sequence searches based on contact frequencies in protein structures. *Protein Eng.*, 6:267–278, 1993.

[106] G. T. Montelione, D. Zheng, Y. J. Huang, K. C. Gunsalus, and T. Szyperski. Protein NMR spectroscopy in structural genomics. *Nat Struct Mol Biol*, 7:982–985, 2000.

[107] P. Munson and R. Singh. Statistical significance of hierarchical multi-body potentials based on delaunay tessellation and their application in sequence-structure alignment. *Protein Sci.*, 6:1467–1481, 1997.

[108] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.*, 247:536–540, 1995.

[109] S. Muthukrishnan and S. Sahinalp. Approximate nearest neighbors and sequence comparison with block operation. *In STOC, Portland, Or,*, 2000.

[110] E. Myers. An O(ND) difference algorithm and its variations. *Algorithmica*, pages 251–266, 1986.

[111] E. W. Myers and W. Miller. Optimal alignments in linear space. *Comput. Applic. Biosci.*, 4:11–17, 1988.

[112] D. Naor, D. Fischer, R. Jernigan, H. Wolfson, and R. Nussinov. Amino acid pair interchanges at spatially conserved locations. *J. of Mol. Biol.*, 256(5):924–938, 1996.

[113] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol Biol.*, 48:443, 1970.

[114] K. Niefind and D. Schomburg. Amino acid simiarity coefficients for protein modeling and sequence alignment derived from main-chain folding angles. *J. Mol. Biol.*, 219:481–497, 1991.

[115] J. C. Norvell and A. Z. Machalek. Structural genomics programs at the US National Institute of General Medical Science. *Nat Struct Biol*, 7 Suppl:931–932, 2000.

[116] C. A. Orengo, A. D. Michie, S. Jones, D. T. Jones, M. B. Swindells, and J. M. Thornton. CATH- A hierarchic classification of protein domain structures. *Structure*, 5(8):1093 –1108, 1997.

[117] J. Orenstein. Spatial query processing in an object oriented database system. *Proc. ACM SIGMOD*, pages 326–336, May 1986.

[118] A. Ortiz, C. Strauss, and O. Olmea. Mammoth (matching molecular models obtained from theory): an automated method for model comparison. *Protein Sci.*, 11:2606–2621, 2002.

[119] A. R. Ortiz, C. E. Strauss, and O. Olmea. Mammoth (matching molecular models obtained from theory): An automated method for model comparison. *Protein Sci*, 11(11):2606–2621, 2002.

[120] O. Ozturk and H. Ferhatosmanoglu. Effective indexing and filtering for similarity search in large biosequence datasbases. In *(BIBE'03)In Third IEEE Symposium on Biolnfonnatics and BioEngineering*, pages 359–366, 2003.

[121] W. A. Pearson. Rapid and sensitive sequence comparison with fastp and fasta. *in Methods in Enzymology ed. R. Doolittle Academic Press, San Diego 183*, pages 63–98, 1990.

[122] D. Plewczynski, J. Pas, M. von Grotthuss, , and L. Rychlewski. 3d-hit: fast structural comparison of proteins. *Appl. Bioinformatics*, 1(4):233–235, 2002.

[123] O. Poirot, E. OToole, and C. Notredame. Tcoffee@igs: A web server for computing, evaluating and combining multiple sequence alignments. *Nucleic Acids Res.*, 31(13):3503–3506, 2003.

[124] A. Prlic, F. Domingues, and M. Sippl. Structure-derived substitution matrices for alignment of distantly related sequences. *Protein Eng.*, 13:545–550, 2000.

[125] G. Ramsay. DNA chips: State-of-the art. *Nature Biotechnology*, 16:40–44, 1998.

[126] F. Richards. The interpretation of protein structures: total volume, group volume distributions and packing density. *J. Mol. Biol.*, 82:1–14, 1974.

[127] J. Risler, M. Delorme, H. Delacroix, and A. Henaut. Amino acid substitutions in structurally related proteins. a pattern recognition approach. determination of a new and efficient scoring matrix. *J. Mol. Biol.*, 204:1019–1029, 1988.

[128] J. Roach, S. Sharma, M. Kapustina, and C. J. Carter. Structure alignment via Delaunay tetrahedralization. *Proteins*, 60:66–81, 2005.

[129] P. Rogen and B. Fain. Automatic classification of protein structure by using Gauss integrals. *Proc Natl Acad Sci U S A*, 100(1):119–124, 2003.

[130] R. Russell, M. Saqi, R. Sayle, P. Bates, , and M. Sternberg. Recognition of analogous and homologous protein folds: Analysis of sequence and structure conservation. *Journal of Molecular Biology*, 269:423–439, 1997.

[131] A. Sacan, O. Ozturk, H. Ferhatosmanoglu, and Y. Wang. LFM-pro: A tool for detecting significant local structural sites in proteins. *Bioinformatics*, 23(6):709–716, 2007.

[132] A. Sacan and I. H. Toroslu. Amino acid substitution matrices based on 4-body Delaunay contact profiles. *IEEE 7th Intl Symp on Bioinformatics and Bioengineering (IEEE-BIBE2007)*, pages 796–802, 2007.

[133] A. Sacan and I. H. Toroslu. Approximate similarity search in genomic sequence databases using landmark-guided embedding. In *Proc. of the IEEE 1st Intl. Workshop on Similarity Search and Applications (SISAP, an ICDE-2008 workshop) April 11-12, 2008, Cancun, Mexico.*, pages 43–50, 2008.

[134] S. Sahinalp, M. Tasan, J. Macker, and Z. Ozsoyoglu. Distance based indexing for string proximity search. *IEEE Data Engineering Conference*, pages 125–137, 2003.

[135] J. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, C18:401–409, 1969.

[136] M. Schena, D. Shalon, R. Davis, and P. Brown. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 270:467–470, 1995.

[137] P. Sellers. On the theory and computation of evolutionary distances. *J. Appl. Math. (SIAM)*, 26:787–793, 1974.

[138] T. Sellis, N. Roussopoulos, and C. Faloutsos. The R^+-tree: A dynamic index for multimensional objects. *13th VLDB*, pages 507–518, 1987.

[139] M. Shatsky, R. Nussinov, and H. J. Wolfson. A method for simultaneous alignment of multiple protein structures. *Proteins: Structure, Function, and Bioinformatics*, 56(1):143–156, 2004.

[140] M. Shatsky, A. Shulman-Peleg, R. Nussinov, and H. J. Wolfson. Recognition of binding patterns common to a set of protein structure. *Lecture Notes in Computer Science*, 3500:440 – 455, 2005.

[141] I. N. Shindyalov and P. E. Bourne. Protein structure alignment by incremental combinatorial extension (CE) of optimal path. *Protein Engineering*, 11(9):739–747, 1998.

[142] R. Singh and M. Saha. Identifying structural motifs in proteins. In *Pac Symp Biocomput*, pages 228–239, 2003.

[143] R. K. Singh and A. Tropsha. Delaunay tessellation of proteins: Four body nearest neighbor propensities of amino acid residues. *J. Comput. Biol.*, 3:213–221, 1996.

[144] T. Smith and M. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.

[145] O. D. Sparkman. *Mass spectrometry desk reference*. Global View Pub., 2000.

[146] R. V. Spriggs, P. J. Argymiuk, and P. Willett. Searching for patterns of amino acids in 3D protein structures. *J Chem Inf Comput Sci*, 43(2):412–421, 2003.

[147] D. J. States and P. Agarwal. Compact encoding strategies for dna sequence similarity search. *In ISMB*, 1996.

[148] H. Sun, O. Ozturk, and H. Ferhatosmanoglu. Comri: A compressed multi-resolution index structure for sequence similarity queries. In *Proc. Computational Systems Bioinformatics (CSB'03)*, pages 553–558, 2003.

[149] D. Swigon, B. D. Coleman, and I. Tobias. The elastic rod model for DNA and its application to the tertiary structure of dna minicircles in mononucleosomes. *Biophysical Journal*, 74:2515–2530, 1998.

[150] Z. Tan, X. Cao, B. C. Ooi, and A. K. H. Tung. The ed-tree: an index for large dna sequence databases. *In Proc. 15th International Conference on Scientific and Statistical Database Management. Jul. 9–11 2003 Cambridge, MA, U.S.A.*, pages 151–160, 2003.

[151] M. Taskin and Z. M. Ozsoyoglu. Improvements in distance-based indexing. *Proceedings of the 16th International Conference on Scientific and Statistical Database Management, SSDBM'04*, pages 161–170, 2004.

[152] W. Taylor. Pattern matching in protein sequence comparison and structure prediction. *Protein Eng*, 2:77–86, 1988.

[153] W. Taylor and D. Jones. Templates, consensus patterns and motifs. *Current opinion in structural biology*, 1:327–323, 1991.

[154] W. Taylor and C. Orengo. Protein structure alignment. *J. Mol. Biol.*, 208(1):1–22, 1989.

[155] J. Thompson, F. Plewniak, and O. Poch. Balibase: a benchmark alignment database for the evaluation of multiple alignment programs. *Bioinformatics*, 15(1):87–88, 1999.

[156] W. S. Torgerson. *Theory and Methods of Scaling*. New York: Wiley, 1958.

[157] J. C. Traina, A. J. M. Traina, B. Seeger, and C. Faloutsos. Slim-trees: High performance metric trees minimizing overlap between nodes. In *Proc. of the 7th Intl. Conf. on Extending Database Techn.*, pages 51–65, 2000.

[158] A. Tramontano. *The Ten Most Wanted Solutions in Protein Bioinformatics*. CRC Press, University of Rome, Italy, May 2005.

[159] C.-H. Tung, J.-W. Huang, and J.-M. Yang. Kappa-alpha plot derived structural alphabet and BLOSUM-like substitution matrix for rapid search of protein structure database. *Genome Biology*, 8:R31.1–R31.16, 2007.

[160] M. Tyagi, P. Sharma, C. S. Swamy, F. Cadet, N. Srinivasan, A. G. de Brevern, , and B. Offmann. Protein block expert (pbe): a web-based protein structure analysis server using a structural alphabet. *Nucl. Acids. Res.*, 34:W119–W123, 2006.

[161] D. W. Ussery. Genetics lecture notes, 1998. `http://www.cbs.dtu.dk/staff/dave/roanoke/genetics980320f.htm`, accessed 26-June-2008.

[162] J. Venkateswaran, D. Lachwani, T. Kahveci, and C. Jermaine. Reference-based indexing of sequence databases. *VLDB 2006*, 2006.

[163] G. Vriend. Bioinformatics i: Introduction to bioinformatics, 2008. `http://www.cmbi.ru.nl/gvteach/bioinformatica1/index.shtml`, accessed 26-June-2008.

[164] H. Wako and T. Yamato. Novel method to detect a motif of local structures in different protein conformations. *Protein Engineering*, 11:981–990, 1998.

[165] A. C. Wallace, N. Borkakoti, and J. M. Thornton. TESS: a geometric hashing algorithm for deriving 3D coordinate templates for searching structural databases. Application to enzyme active sites. *Protein Sci*, 6:2308–2323, 1997.

[166] A. C. Wallace, R. A. Laskowski, and J. M. Thornton. Derivation of 3D coordinate templates for searching structural databases: application to Ser-His-Asp catalytic triads in the serine proteinases and lipases. *Protein Sci*, 5(6):1001–1013, 1996.

[167] A. Wallqvist, Y. Fukunishi, L. R. Murphy, A. Fadel, and R. M. Levy. Iterative sequence/secondary structure search for protein homologs. *Bioinformatics*, 16:988–1002, 2000.

[168] G. Wang and R. Dunbrack. Pisces: a protein sequence culling server. *Bioinformatics*, 19:1589–1591, 2003.

[169] Wikipedia. Amino acid, 2008. http://en.wikipedia.org/w/index.php?title=Amino_acid&oldid=221651281, accessed 26-June-2008.

[170] Wikipedia. Dna, 2008. http://en.wikipedia.org/w/index.php?title=DNA&oldid=220250555, accessed 26-June-2008.

[171] Wikipedia. Flavodoxin fold, 2008. http://en.wikipedia.org/w/index.php?title=Flavodoxin_fold&oldid=197793078, accessed 26-June-2008.

[172] Wikipedia. Globin fold, 2008. http://en.wikipedia.org/w/index.php?title=Globin_fold&oldid=197725454, accessed 26-June-2008.

[173] B. Wu, A. Yee, A. Pineda-Lucena, A. Semesi, T. A. Ramelot, J. R. Cort, J.-W. Jung, A. Edwards, W. Lee, M. Kennedy, and C. H. Arrowsmith. Solution structure of ribosomal protein S28E from Methanobacterium thermoautotrophicum. *Protein Science*, 12:2831–2837, 2003.

[174] W. Xu, R. Mao, S. Wang, and D. P. Miranker. On integrating peptide sequence analysis and relational distance-based indexing. In *BIBE '06: Proceedings of the Sixth IEEE Symposium on BionInformatics and BioEngineering (BIBE'06)*, pages 27–34, 2006.

[175] W. Xu and D. P. Miranker. A metric model of amino acid substitution. *Bioinformatics*, pages 1214–1221, 2004.

[176] L. Y. Yampolsky and A. Stoltzfus. The exchangeability of amino acids in proteins. *Genetics*, 170:1459–1472, 2005.

[177] M. M. Young, A. G. Skillman, and I. D. Kuntz. A rapid method for exploring the protein structure universe. *Proteins*, 34(3):317–32, 1999.

[178] Y. Zhang and J. Skolnick. Scoring function for automated assessment of protein structure template quality. *Proteins*, 57:702–710, 2004.

[179] E. Zintzaras. A comparison of amino acid distance measures using procrustes analysis. *Computers in Biology and Medicine*, 29(5):283–288, 1999.

VITA

PERSONAL INFORMATION

Surname, Name: Saçan, Ahmet
Nationality: Turkish (TC)
Date and Place of Birth: 1 January 1979, Batman
Marital Status: married, with 1 daughter
Phone: +90 312 210 55 93
Fax: +90 312 210 55 44
email: ahmet@ceng.metu.edu.tr

EDUCATION

Degree	Institution	Year
Ph.D. in Computer Eng.	Middle East Technical University	2008
B.Sc. in Computer Sci.	University of Michigan, Ann Arbor	2001
B.Sc. in Cell & Molecular Bio.	University of Michigan, Ann Arbor	2001
High School	Ankara Fen Lisesi	1996

WORK EXPERIENCE

Year	Place	Title
2007-2008	The Ohio State University Dept. of Computer Sci. & Eng.	Visiting Scholar
2002-2007	METU Dept. of Computer Eng.	Research Assistant
2004-2007	METU IDEA Online Education Program	System Admin & Web Developer
2000-2001	University of Michigan	Academic Computer Consultant

PUBLICATIONS

1. Ahmet Sacan, I. Hakki Toroslu, and Hakan Ferhatosmanoglu. Integrated Search and Alignment of Protein Structures. *Bioinformatics*, (submitted).

2. Ahmet Sacan, I. Hakki Toroslu, and Hakan Ferhatosmanoglu. Distance-based Indexing of Residue Contacts for Protein Structure Retrieval and Alignment. *IEEE 8th International Symposium on Bioinformatics & Bioengineering (IEEE-BIBE2008)*, (accepted).

3. Ahmet Sacan, Nilgun Ferhatosmanoglu, and Hakan Ferhatosmanoglu. MaD: An Online Search Tool and Repository for Near–Optimal Microarray Experimental Designs. *BMC Bioinformatics*, (submitted).

4. Hong Sun, Ahmet Sacan, Hakan Ferhatosmanoglu, and Yusu Wang. Smolign: A Spatial Motifs Based Multiple Protein Structures Alignment Method. *Bioinformatics*, (under submission).

5. Hong Sun, Ahmet Sacan, Hakan Ferhatosmanoglu, and Yusu Wang. Smolign: A Spatial Motifs Based Multiple Protein Structures Alignment Method (poster presentation). *Ohio Collaborative Conference on Bioinformatics* (OCCBIO), Toledo, OH, 2008.

6. Ahmet Sacan, Hakan Ferhatosmanoglu, and Huseyin Coskun. CellTrack: An Open-Source Software for Cell Tracking and Motility Analysis. *Bioinformatics*, 24(14):1647-1649, 2008.

7. Ahmet Sacan and I. Hakki Toroslu. Approximate Similarity Search in Genomic Sequence Databases using Landmark-Guided Embedding. *1st International Workshop on Similarity Search and Applications (SISAP, an ICDE-2008 workshop)*, April 11-12, 2008, Cancún, Mexico.

8. Fatih Altiparmak, Ali Tosun, Hakan Ferhatosmanoglu, and Ahmet Sacan. Automated Data Discovery in Similarity Score Queries. *International Conference of Database Systems for Advanced Applications (DASFAA 2008), Lecture Notes in Computer Science*, India, 2008.

9. Ahmet Sacan and I. Hakki Toroslu. Amino acid substitution matrices based on 4-body Delaunay contact profiles. *IEEE 7th International Symposium on Bioinformatics & Bioengineering (IEEE-BIBE2007)*, 796-801, 2007.

10. Ahmet Sacan, Ozgur Ozturk, Hakan Ferhatosmanoglu, and Yusu Wang. LFM-Pro: A Tool for Detecting Significant Local Structural Sites in Proteins. *Bioinformatics*, 23(6):709-716, 2007.

11. Ahmet Sacan, Ozgur Ozturk, Hakan Ferhatosmanoglu, and Yusu Wang. LFM-Pro: a tool for mining family-specific sites in protein structure databases (Poster presentation). *Third Midwest Database Research Symposium*, Apr 2006.

12. Ahmet Sacan. Mapping Protein Sequences and Structures to Feature Spaces for Efficient Indexing. *Proceedings of the International Symposium on Health Informatics and Bioinformatics* (HIBIT 2005), 2005.