2005

Detailed analysis of protein sequence entropy, hydrophobicity, and flexibility

William Yeh
San Jose State University

Follow this and additional works at: https://scholarworks.sjsu.edu/etd_theses

Recommended Citation

Yeh, William, "Detailed analysis of protein sequence entropy, hydrophobicity, and flexibility" (2005).
Master's Theses. 2824.

DOI: <https://doi.org/10.31979/etd.gq9z-q4wd>
https://scholarworks.sjsu.edu/etd_theses/2824

This Thesis is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Theses by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.

NOTE TO USERS

This reproduction is the best copy available.

UMI[®]

DETAILED ANALYSIS OF PROTEIN SEQUENCE ENTROPY,
HYDROPHOBICITY, AND FLEXIBILITY

A Thesis

Presented to

The Faculty of the Department of General Engineering
San Jose State University

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science

by

William Yeh

December 2005

UMI Number: 1432442

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI Microform 1432442

Copyright 2006 by ProQuest Information and Learning Company.

All rights reserved. This microform edition is protected against unauthorized copying under Title 17, United States Code.

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

© 2005

William Yeh

ALL RIGHTS RESERVED

APPROVED FOR THE DEPARTMENT OF GENERAL ENGINEERING

Dr. Brooke Lustig, Department of Chemistry

Dr. Melanie McNeil, Department of Chemical Engineering

Dr. Elaine Collins, Department of Chemistry

APPROVED FOR THE UNIVERSITY

ABSTRACT

DETAILED ANALYSIS OF PROTEIN SEQUENCE ENTROPY, HYDROPHOBICITY, AND FLEXIBILITY

by William Yeh

Complete understanding of the connection between protein structure and protein sequence has been a major goal in bioinformatics. A strong correlation between sequence entropy, hydrophobicity, and flexibility has been shown in recent studies, but with anomalous regions of very high and low flexibility that needed more exploration. Using a set of 130 proteins and their aligning sequences, direct observations revealed high counts of small residues in the anomalous, low flexibility region, and high counts of gaps in the anomalous, high flexibility region. Additional investigations on the effect of packing radius and % bit score cutoff showed that 6 to 9 Å radii gave the most reasonable sequence entropy data, and bit score cutoffs have no significant effect on the general trend of the sequence entropy.

ACKNOWLEDGEMENTS

Dr. Brooke Lustig (Research Director)

Dr. Melanie McNeil

Dr. Elaine Collins

Dr. Michael Jennings

Haihong Liao

My family and friends

TABLE OF CONTENTS

	Page
List of Tables	ix
List of Figures	x
Chapter	
1. Introduction	01
2. Background	04
Proteins and Amino Acids	04
Protein Structures	06
Three-Dimensional Structure Determination Methods	07
NMR Analysis	07
X-Ray Crystallography Analysis	08
Important Key Definitions	08
Sequence Alignment	11
BLAST (Basic Local Alignment Search Tool)	12
3. Literature Reviews	15
Sequence-Based Protein Structure Prediction	15
Sequence Entropy	16
Flexibility	17
Hydrophobicity	19
Implications of Residue Types and Gaps in Proteins	20
Relating Sequence Entropy, Flexibility, and Hydrophobicity	21

4. Hypothesis and Objectives	27
Hypothesis	27
Objectives	27
5. Materials and Methods	28
Materials	28
Methods	28
Collecting and Combining Data	29
Residue Counting	30
Data Analysis	31
6. Results and Discussion	33
Results	33
Discussion	43
Optimum Percentage Bit Score and Identity Cutoffs	43
Optimum Packing Radius	45
Fraction of Strongly Hydrophobic Residues	46
Fraction of Small Residues	47
Fraction of Gaps	47
7. Conclusions	49
References	51
Appendix A PERL Codes Used	55
Appendix B Fraction of non-strongly hydrophobic residues versus 1/density, using bit score 40	74

Appendix B1	Fraction of non-strongly hydrophobic residues versus 1/density, for radius 9 to 12 Å	75
Appendix B2	Fraction of non-strongly hydrophobic residues versus 1/density, for radius 6 to 9 Å	76
Appendix B3	Fraction of non-strongly hydrophobic residues and average sequence entropy versus 1/density, for radius 6 to 12, using bit score 40	77
Appendix C	Average Sequence Entropy using different Bit Score and Percentage Identity Cutoffs, with a packing radius of 9 Å	78
Appendix D	Fraction of small residues and average sequence entropy versus 1/density, for radius 6 to 12, using bit score 40	79
Appendix E	Fraction of gaps and average sequence entropy versus 1/density, for radius 6 to 12, using bit score 40	80

LIST OF TABLES

		Page
Table 1	The list of the Protein Data Bank identities of the 130 query proteins used	28
Table 2	The mean, median, and standard deviation of the frequency distributions of C_{α} packing density, calculated using a packing radius of 6 to 12 Å	33

LIST OF FIGURES

		Page
Figure 1	The amino acid structure and formation of the peptide bond	04
Figure 2	The 20 types of amino acids	05
Figure 3	Protein structures – Primary, Secondary, Tertiary, Quaternary	07
Figure 4	An example of a simple sequence alignment	12
Figure 5a	Query Sequence for protein 1CRC	13
Figure 5b	BLAST alignment output for 1CRC	13
Figure 6	Single average entropy, double average entropy, and Gerstein-Altman single averaged entropy against flexibility	24
Figure 7	Average sequence entropy, hydrophobicity, and flexibility	25
Figure 8	The flowchart of the different steps taken in this study	29
Figure 9	Residue frequency distributions over packing density for 6 to 12 Å packing radius	34
Figure 10	The graph of average sequence entropy over 1/density, for the packing radius of 6 to 9 Å and 9 to 12 Å, using a 40% bit score cutoff	35
Figure 11	Normalized average sequence entropy over 1/density, using different bit score cutoffs	37
Figure 12	Fraction of non-strongly hydrophobic residues and average sequence entropy versus inverse density	40

Figure 13	Fraction of non-strongly hydrophobic residues and average hydrophobicity versus inverse density	41
Figure 14	Fraction of small residues (A i.e. alanine, G i.e. glycine) and average sequence entropy over inverse packing density (9 Å, 40% bit score)	42
Figure 15	Fraction of gaps (-) and average sequence entropy over inverse packing density (9 Å, 40% bit score)	44

CHAPTER 1

Introduction

Proteins are a class of biological molecules that are responsible for the functions that occur in the cells of organisms. The building blocks of proteins are 20 commonly found amino acids, where each amino acid and its properties are distinguished by the side-chain it contains (Mount, 2001). The different combinations of amino acids derived from different proteins, give rise to the diversity of proteins' functions such as enzymatic activity, signal transduction, immune response, storage, and transport. The cataloguing of protein function is critical to understanding the function of biological systems (Eisenberg, Marcotte, Xenarios & Yeates, 2000). Based on this knowledge of protein function, drug design and diagnostics become more possible in their implementation.

The sequence of a protein is responsible for its structure, and the structure of a protein is directly related to its function. This is especially important in the area of drug discovery because reliable prediction of a protein structure from sequence can potentially save both money and time.

Protein or nucleic acid sequences are now routinely determined. This has created many large public online databases such as GENBANK (National Center of Biotechnology Information [NCBI], 2004), which have made this information easily accessible. Now, to be able to determine of the structure of a protein from its sequence is of primary interest to the bioinformatics community. NMR (Nuclear Magnetic Resonance) and X-ray crystallography can accurately make out the structure of a protein;

however the protein must crystallize well or be soluble at a required concentration for the experiments. Even if the criteria are met, the cost and time expended are usually not desirable (Skolnick, Fetrow & Kolinski, 2000). To have methods that can predict the structure from the amino acid sequence would, in turn, be an attractive approach.

There are a number of algorithms that can perform sequence-structure predictions, however, prediction is significantly lacking when compared to X-ray and even NMR methods (Baker & Sali, 2001). The major reason why this prediction cannot be achieved efficiently and effectively is the incomplete understanding of how protein sequences relate to structures. Knowledge, such as how proteins fold, the forces, bonds or energies that are involved, the role of individual atoms in each of the amino acids and its side chain plays, the external factors that influence the folding such as the post-translational modifications, and most importantly the role of solvent, are not completely available.

In understanding the connection between structure and sequence, it is common to study the patterns of hydrophobic residues in the protein, as it is the major driving force of globular protein folding (Dill et al., 1995). The calculation of sequence entropy for a protein is another approach to understanding structure and sequence. This method has been applied to a large set of proteins, where the sequence entropy and hydrophobicity were found to correlate with local residue flexibility (Liao, Yeh, Chiang, Jernigan & Lustig, 2005).

However, from these studies, anomalous regions of very high (greater than 26 C_{α} per 9 Å radius) and very low (lesser than 5 C_{α} per 9 Å radius) density were not completely characterized. These regions represent less than 1% of the total number of

residue positions. In this research, the objectives will be to study in more detail these two regions in comparison to the existing analyzed regions, by examining the presence of strongly hydrophobic residues, small residues, and gaps. The effects of packing radii and bit score cutoffs will also be examined. Further studies in these areas will increase the understanding of sequence entropy in relation to protein structure.

CHAPTER 2

Background

Proteins and Amino Acids

Amino acids, or residues, are the basic units of proteins. The general structure of an amino acid is shown in Figure 1. An amino acid consists of an amino group ($-\text{NH}_2$), a carboxyl group ($-\text{COOH}$), a hydrogen atom ($-\text{H}$), and a side-chain ($-\text{R}$), all attached to a carbon atom called the alpha-carbon (C_α). The side-chain, also called the R-group, determines the properties of the amino acid, whether it is polar, non-polar, aromatic, aliphatic, acidic or basic. To link the amino acids together, peptide bonds are formed between the amino group of one residue and the carboxyl group of another residue, through the loss of a water molecule. The resulting polymer of the 20 main types of amino acids is called a polypeptide. A protein is a biological molecule that consists of one or more polypeptides that has a specific structure and function. Figure 2 shows the different types of amino acids, their structural formulas, and their hydrophobicity.

Figure 1. The amino acid structure and formation of the peptide bond (Concept 2, n.d.; Concept 4, n.d.).

Amino Acids			Hydrophobicity (kcal/mol)		
Name	Abbrev.	Structural Formula	Hopp-Woods	Engelman-Steitz	Sharp-Honig
Alanine	Ala A	$\text{CH}_3\text{-CH(NH}_2\text{)-COOH}$	-0.50	-1.60	2.03
Cysteine	Cys C	$\text{HS-CH}_2\text{-CH(NH}_2\text{)-COOH}$	-1.00	-2.00	4.19
Aspartic Acid	Asp D	$\text{HOOC-CH}_2\text{-CH(NH}_2\text{)-COOH}$	3.00	9.20	1.09
Glutamic Acid	Glu E	$\text{HOOC-CH}_2\text{-CH}_2\text{-CH(NH}_2\text{)-COOH}$	3.00	8.20	1.78
Phenylalanine	Phe F		-2.50	-3.70	6.03
Glycine	Gly G	$\text{H-CH(NH}_2\text{)-COOH}$	0.00	-1.00	1.01
Histidine	His H		-0.50	3.00	2.95
Isoleucine	Ile I	$\text{H}_3\text{C-CH}_2\text{-CH(CH}_3\text{)-CH(NH}_2\text{)-COOH}$	-1.80	-3.10	5.87
Lysine	Lys K	$\text{H}_2\text{N-(CH}_2\text{)}_4\text{-CH(NH}_2\text{)-COOH}$	3.00	8.80	2.01
Leucine	Leu L	$\text{H}_3\text{C-CH(CH}_3\text{)-CH}_2\text{-CH(NH}_2\text{)-COOH}$	-1.80	-2.80	5.89
Methionine	Met M	$\text{H}_3\text{C-S-(CH}_2\text{)}_2\text{-CH(NH}_2\text{)-COOH}$	-1.30	-3.40	4.80
Asparagine	Asn N	$\text{H}_2\text{N-C(=O)-CH}_2\text{-CH(NH}_2\text{)-COOH}$	0.20	4.80	1.40
Proline	Pro P		0.00	0.20	3.51
Glutamine	Gln Q	$\text{H}_2\text{N-C(=O)-CH}_2\text{-CH}_2\text{-CH(NH}_2\text{)-COOH}$	0.20	4.10	2.48
Arginine	Arg R	$\text{HN(CH}_2\text{NH}_2\text{)-CH}_2\text{-CH}_2\text{-CH(NH}_2\text{)-COOH}$	3.00	12.30	1.78
Serine	Ser S	$\text{HO-CH}_2\text{-CH(NH}_2\text{)-COOH}$	0.30	-0.60	1.59
Threonine	Thr T	$\text{H}_3\text{C-CH(OH)-CH(NH}_2\text{)-COOH}$	-0.40	-1.20	2.53
Valine	Val V	$\text{H}_3\text{C-CH(CH}_3\text{)-CH(NH}_2\text{)-COOH}$	-1.50	-2.60	4.51
Tryptophan	Trp W		-3.40	-1.90	7.14
Tyrosine	Tyr Y		-2.30	0.70	4.90

Figure 2. The 20 types of amino acids (King, 2003). Note strongly hydrophobic residues in bold (Poupon & Mornon, 1999).

Even though there are several types of amino acids, the ones of interest in this study are small and hydrophobic ones. The *small residues* refer to the amino acids that have smaller side-chains. Glycine and alanine are considered small residues because they only have a hydrogen atom (-H) and a methyl group (-CH₃) as their R-groups, respectively. The *hydrophobic residues* are the amino acids with R-groups have non-polar characteristics, such as valine, leucine, and methionine. They avoid interaction with aqueous environments and tend to bury themselves inside the protein.

Protein Structures

As residues link together to form a protein, this polypeptide chain immediately begins to fold into very specialized patterns. Figure 3 shows the four levels of structures in proteins. The primary structure is the basic polypeptide chain of amino acids. The secondary structure consists of two main types – the α -helix and the β -pleated sheets. These are formed due to hydrogen bond formed between neighboring backbone atoms. When numerous hydrogen bonds are present in the structure, they produce a stabilizing effect.

Tertiary structure is formed when secondary structures are folded together in a stable three-dimensional configuration by various forces. The main driving force for tertiary structures is the hydrophobic effect (Dill et al., 1995), where the hydrophobic residues are packed into the structure, away from the surrounding solvent. Other forces that contribute to stabilizing the tertiary structures are hydrogen bonds, van der Waals forces, and disulfide bridges. Finally, quaternary structure is made up of two or more subunits of polypeptides, stabilized by similar forces used in tertiary structures.

Figure 3. Protein structures – Primary, Secondary, Tertiary, and Quaternary (Horton, Moran, Ochs, Rawn, & Scrimgeour, 2002).

Three-Dimensional Structure Determination Methods

NMR Analysis

NMR is a phenomenon that occurs to a nucleus of an atom due to a property of the subatomic particles called the spin (Norton, 1990). The spin can be thought of as a magnetic moment that responds to its environment. When it is placed in an external magnetic field, the particle will try to align itself with the field, alternating from high-energy state to low-energy state. The transition of the particle from the two energy states

will cause the absorption or emission of energy, which can be detected and interpreted by a spectroscope into structural information.

X-Ray Crystallography Analysis

To visualize an object, its size has to be greater than half the wavelength of the visible light, i.e. $7 \times 10^{-5} \text{ cm}$ to $4 \times 10^{-5} \text{ cm}$ (Holmes & Blow, 1965). However, the wavelength of visible light is much greater than an atom ($\sim 10^{-8} \text{ cm}$), making it 'invisible' even with the help of a microscope. X-rays have a much smaller wavelength ($10^{-7} \text{ cm} \sim 10^{-9} \text{ cm}$), making them suitable for 'detecting' atoms. To determine the atomic positions of a molecule using X-ray analysis, the molecule has to be crystallized first. X-rays are diffracted into specific patterns by the electrons in the proteins. Using complex mathematical equations, these patterns are then interpreted by a computer to give the coordinates of the atoms by their electron density distributions.

Important Key Definitions

Hydrophobicity of an amino acid determines the degree of its exclusion by water molecules, due to the non-polar property of the side chains of the amino acid. This is a characteristic that is conserved during evolution (Poupon & Mornon, 1999). A major driving force that influences how a protein folds, is the hydrophobicity of its amino acids (Dill et al., 1995). The need to protect the hydrophobic amino acids from the water molecules in the surrounding causes them to be buried into the protein structure, with the non-hydrophobic amino acids typically left on the outer regions of a globular protein. This phenomenon, also termed as hydrophobic effect, gives the protein stability in

structure. Hydrophobicity can be measured by several approaches, and each involves a particular hydrophobicity scale.

Entropy is the measure of the disorder or randomness in a system. This measurement is also being used in the prediction of certain features of protein structure from sequence. A generalized expression (Kono & Saven, 2001) of this entropy is

$$S = -\sum_{(\text{protein states})} W_{(\text{protein state})} \ln W_{(\text{protein state})} \quad (1)$$

where protein state refers to all the variables that are needed to describe the state of a folded protein, and $W_{(\text{protein state})}$ represents the probability of being in that particular state. To be more explicit, this entropy expression can be modified to look at the frequency of an amino acid at its specific position. This is called the ‘sequence entropy’ and is expressed as,

$$S_k = -\sum_{j=1,20} P_{jk} \ln P_{jk} \quad (2)$$

where P_{jk} represent the probability of amino acid j , appearing in protein sequence position k , and j ranges from 1 to 20 (Valdar, 2002). Sequence entropy is derived from Shannon entropy, a central representation of the Information Theory.

Sequence entropy essentially also measures the variability of the sequence of interest, by looking at all of the sequence positions and summing the total variability at each position. Sequence variability is an important parameter in understanding protein

structure, as it had been shown to correspond to the types of fold that can occur (Larson, England, Desjarlais & Pande, 2002).

Flexibility of a residue position indicates the amount of motion that is allowed at that position. This property allows a protein to bind to more than one type of substrate, but it needs to be balanced with a certain amount of rigidity for the protein's stability (Jaenicke, 2000). It has been shown in previous modeling studies of NMR data that the greater the flexibility at a sequence position, the more able it is to allow mutation to occur (Hsieh, Collins, Blomquist & Lustig, 2002). Previous studies also showed that flexibility, as measured by both X-ray and NMR, could be estimated by calculating the inverse of the packing density (Bahar, Wallqvist, Covell & Jernigan, 1998). This is because when a residue is tightly packed, it becomes harder for the residue to move. Another approach uses the atomic displacement parameter, or the B-factor, as a common measure of flexibility. The B-factors of the atoms in a protein are obtained from X-ray crystallography studies. They represent the amount of thermal motions and static disorders of atoms (Smith, Radivojac, Obradovic, Dunker & Zhu, 2003). Smith and coworkers modified the early estimates of B-factors, to give better predictive values of residue flexibility.

Packing density of an amino acid determines how well that amino acid is surrounded by neighboring residues. As mentioned earlier, packing density is used as an estimation of the flexibility at a protein sequence position. One method to calculate this value is by counting the number of alpha-carbons (C_{α}) that surround the C_{α} of the residue of interest, in a spherical volume of a given radius, such that

$$D(i, j) = \sqrt{(x(i) - x(j))^2 + ((y(i) - y(j))^2 + (z(i) - z(j))^2)} \quad (3)$$

where $D(i, j)$ is the distance between $C_{\alpha i}$ and $C_{\alpha j}$, and is to be compared to the given radius.

Gaps are spaces introduced into an alignment to compensate for insertions and deletions in one sequence relative to another (NCBI, 2005). Gaps are not residues, however they are treated as residues in the calculations of this research. These gaps are commonly caused by point mutations in the nucleic acids.

Sequence Alignment

Sequence alignment is the basis for both database searching and multiple sequence alignment tools. It is the process where two or more sequences are compared side by side, and arranged by only adding gaps in the sequences, so that similar patterns in the sequences can be matched together under the same columns. This process is done under certain given criteria and parameters, depending on the purpose of the alignment and the relationship of the sequences under observation. A simplified example of a sequence alignment is shown in Figure 4. The query and the subject sequence are aligned when the two sequences give the best match, after the introduction of gaps that are represented by the dash symbol (-). The most basic algorithm for aligning two sequences involves highly recursive algorithms that exhaustively calculate the best alignment. This is called dynamic programming and it can be used to look for global or local similarities in aligned sequences (Needleman & Wunsch, 1970; Smith & Waterman, 1981).

Sequence Alignment	
Query Sequence	V C C N K Y L S K R
Subject Sequence	V - C N - - L S K -

Figure 4. An example of a simple sequence alignment.

Global alignment is done by aligning two or more sequences and looking for similarities in the aligned sequences as a whole. It gives the optimal scoring alignment. However, most biological significant sites on proteins are well-aligned sub-regions of the entire sequence, therefore the local alignment is developed (Mount, 2001). Local alignment is derived from the global alignment algorithm. Instead of looking for similarities ‘globally’, it looks for regional similarities.

BLAST (Basic Local Alignment Search Tool)

The most widely used sequence alignment algorithm is the BLAST (Altschul, Gish, Miller, Myers & Lipman, 1990). BLASTp, based on the BLAST algorithm, is a protein database-searching program that takes a query sequence and looks for similar sequences in a database. A typical BLASTp input query sequence is shown in Figure 5a. ‘1CRC’ represents the PDB identification code for the protein ‘cytochrome C’. The letter ‘A’ indicates that it is ‘chain A’ of the protein. One of the many results output of a BLASTp search using the query sequence in Figure 5a, is shown in Figure 5b. In this figure, the different measures of similarity of the query sequence and the result sequence are indicated, along with their aligned sequences.

The ‘Score’ is the bit score of the alignment, which indicated how good is the alignment. The bit score is calculated using a user-selected substitution matrix and is

normalized so that it can be compared with results derived from other substitution matrices. 'Expect' is the expect value, which describes the probability of the alignment occurring randomly. The lower the expect value, the more significant is the matching to the subject sequence. Lastly, 'Identities' indicates the number of exact matches in this alignment, while 'Positives' represents the number of non-identical but characteristically similar matches in the alignment, also called conservative substitutions.

```
>1CRC:A CYTOCHROME C
XGDVEKGGKIFVQKCAQCHTVEKGGKHKGTGPNLHGLFGRKTGQAPGFTYTDANKNKGITW
KEETLMEYLENPKKYIPGTMIFAGIKKKTEREDLIAYLKATNE
```

Figure 5a. Query sequence for protein 1CRC.

```
┌>gi|3930766|qb|AAC80552.1| cytochrome c [Tigriopus californicus]
  gi|3930764|qb|AAC80551.1| cytochrome c [Tigriopus californicus]
  gi|3930762|qb|AAC80550.1| cytochrome c [Tigriopus californicus]
    Length = 105

Score = 177 bits (449), Expect = 7e-44
Identities = 77/103 (74%), Positives = 93/103 (90%)

Query: 2 GDVEKGGKIFVQKCAQCHTVEKGGKHKGTGPNLHGLFGRKTGQAPGFTYTDANKNKGITWK 61
        GD++KGGKIFVQKC QCHT+E GGKHK GPNLHG++GR+TG+A G++YTDANK+KG+TW
Sbjct: 2 GDIDKGGKIFVQKCTQCHTIEAGGKHKVGPNLHGMVGRQTGKAAGYSYTDANKSKGVTWN 61

Query: 62 EETLMEYLENPKKYIPGTMIFAGIKKKTEREDLIAYLKATN 104
        EETL YL NPKKYIPGTM+FAG+KKK +REDLIAYLK A++
Sbjct: 62 EETLDIYLTNPKKYIPGTMVFAGLKKKGREDLIAYLKSASS 104
```

Figure 5b. BLAST alignment output for 1CRC.

Due to the size of the protein and DNA databases, to perform an exhaustive search of a database by a query sequence is problematic (Nicholas, Deerfield II & Ropelewski, 2000). Therefore, unlike the dynamic programming methods by Needleman-Wunsch and Smith-Waterman, BLAST has shown success in characterizing alignments efficiently as a heuristic algorithm (Altschul et al., 1990; Altschul et al., 1997). This means that it progresses along empirical paths by using rules of thumb to

reach a solution. Heuristics are usually faster than dynamic programming but they do not necessarily produce the optimal solutions. There are also improved versions of BLAST, such as Gapped-BLAST, which is three-times faster than the original, and PSI-BLAST, which is able to detect more evolutionarily distant sequences. BLAST can be found on the NCBI (National Center for Biotechnology Information) website.

CHAPTER 3

Literature Review

Sequence-Based Protein Structure Prediction

Protein structure prediction has not achieved the consistent and high-resolution results associated with the three-dimensional structure determination methods of X-ray and NMR (Baker & Sali, 2001). For this thesis research, the primarily methods used, focus on sequence comparison and variability. Although high-resolution protein structure can be experimentally determined, it is not always practical. Structure prediction based on only modeling of the energetics and other physical parameters of proteins is still a not well-defined approach. Typically, they are included in the category of ab initio methods, predicting structures directly from sequence without relying on any known sequence-structure information (Skolnick et al., 2000). This class of methods assumes that proteins in their native form always exist in a minimal free energy state. The ab initio methods involve determining all possible structural conformations of a given sequence, and identifying which conformations are low in free energy.

The second class, which is sequence-based, consists of sequence alignment and threading methods, using templates for detecting similarities in unknown sequences and structures (Skolnick et al., 2000). Sequence alignment methods such as homology structure building, first find template sequences with known structure that are similar to the original (i.e. query) sequence, and then use them as a template to build a model. Threading methods model sequences that are more distantly related. Instead of aligning

sequence with sequence, threading methods align sequence to structure, and then each alignment is examined to see if the residues in the sequence are spatially and chemically compatible with the structure (Skolnick et al., 2000, and Baker & Sali, 2001). Threading is able to detect sequences with similar folds but have no sequence similarities. In this project, the approach involves applying methods that also use measures of sequence conservation to explore features of three-dimensional structure.

Sequence Entropy

Information theory was used to find the information content in nucleic acid sequences from which important sites were identified (Schneider, Stormo, Gold, & Ehrenfeucht, 1986). In this study, they indirectly measured the amount of information, using the degree of ‘uncertainty’ determined by Shannon entropy. An increase in the amount of information corresponds to a decrease in uncertainty. The result showed that at binding sites of control regions such as initiation codon and promoters, sequence entropy values were found to be lower than other randomly selected regions.

Sequence entropy has been applied to protein calculations (Larson et al., 2002). First, they utilized common structural features for a set of 253 proteins obtained from Protein Data Bank to generate hundreds of suitable minimum-energy sequences from each of these features (Protein Data Bank [PDB], 2005). Then, the sequence entropy values were calculated using these sequences. They observed that structures in proteins that have similar overall folds tend to have similar entropy values.

Three differently derived entropy values were calculated from a set of 10 proteins (Koehl & Levitt, 2002). S_{des} was calculated from sequence alignments generated

using a large set of sequences, designed from structural and other physical conformations. S_{str} is entropy calculated directly from structural alignments of naturally occurring protein sequences, and S_{seq} is calculated from sequence alignments of 5 iterative PSI-BLAST results. They found that for most of the proteins these three values correspond well, especially S_{str} and S_{des} , even though they are assumed to be independent calculations.

Shannon entropy, a measure of diversity, has been used for scoring positional conservation in nucleic acid and amino acid sequences (Valdar, 2002). Conservation in protein residues is often seen as an indication of important protein structures and sites. The three studies reviewed have shown correlation between sequence entropy and structure in DNA (Schneider et al., 1986) and proteins (Larson et al., 2002; Koehl & Levitt, 2002). More recently, sequence entropy was calculated by the Lustig group, for a large set of aligned sequence residues and their corresponding query protein packing densities (Liao et al., 2005). Their results showed significant correlations that will be mentioned in a later section.

Flexibility

Lustig and coworkers explored the entropic effect on binding peptides associated with loops and bulges in RNAs, using Gaussian Network Model (GNM) and lattice calculations (Lustig, Bahar & Jernigan, 1998). It was observed that binding of peptides to RNA was enhanced by RNA loop size. This was contrary to existing models based on chain statistical mechanics. In subsequent studies, they also found a connection between flexibility and sequence entropy, using RMSD and lattice modeling on NMR-derived peptide structures (Hsieh et al., 2002). Although the correlations between sequence and

configurational entropy were modeled using an RNA-peptide complex, a complete quantification of that relationship in proteins still remains.

Flexibility was investigated by applying experimental hydrogen exchange (HX) data of protein folds using the GNM (Bahar et al., 1998). The HX data of a protein had been shown to monitor a protein's conformational fluctuations. They found that the entropic component of free energy around a deformation of a protein at particular residue components to be a degree of C^α packing. This defines a relationship between local flexibility and configurational entropy.

The relative flexibility of protein backbones can be determined by conformational backbone entropy (Meirovitch & Hendrickson, 1997). Flexibility was determined by the average B-factor values of the backbone alpha-carbons, while the backbone entropy data were obtained from a molecular dynamic trajectory, using the local state method. Halle explored the hypothesis that B-factors can be predicted using packing density only (Halle, 2002). A local density model (LDM) was applied to a set of 38 protein crystal structures, and the predicted B-factor values of the proteins were calculated. The LDM model is derived from the B-factor equation, by the application of a set of approximations. When comparing the predicted value to the experimental values, it was also demonstrated that the B-factor is inversely proportional to the number of non-covalent neighbor atoms within a local volume of the protein's alpha-carbons.

Different measurements and estimations of flexibility, such as sequence entropy, conformational entropy, B-factors, and packing density, are interrelated (Lustig et al., 1998; Bahar et al., 1998; Halle, 2002). Recently, a connection was made between

flexibility, packing density and sequence entropy (Liao et al., 2005). They approximated flexibility using packing density with a 9 Å packing radius and correlating it to the sequence entropy.

Hydrophobicity

It is generally accepted that a main global driving force for protein folding is due to the association of non-polar amino acids to the surrounding water molecules, rather than hydrogen bonds within a protein (Dill et al., 1995). The polar nature of the surrounding water molecules causes the non-polar amino acids to be pushed into the protein core. Hydrophobicity is often conserved during evolution (Poupon & Mornon, 1999). Strongly hydrophobic residues (V, I, L, F, M, Y, W) are associated with buried residues in proteins.

The hydrophobic effect of the amino acids in a particular solvent was shown using an energy minimization calculation (Levitt, 1976). Each amino acid side-chain was given an estimate of its solvent interaction energy from its solubility in water and ethanol. Levitt found that one of the major energy terms that can greatly affect the results of the protein folding simulation was the solvent energy term. Hopp and Woods refined Levitt's scale with respect to experimental observations involving antigenic determinants (Hopp & Woods, 1981). Antigenic determinants are the parts of an antigen to which an antibody binds. They are normally found to be at the surface regions of a protein where they are highly exposed to the solvent. Thus, antigenic determinants are usually composed of residues that are hydrophilic and charged.

Engelman-Steitz parameters involve identifying non-polar transbilayer helices from membrane protein sequences, estimating the hydrophobicity by calculating both the hydrophobic and hydrophilic component of the free energy (Engelman, Steitz & Goldmann, 1986). The hydrophobic part was modeled by calculating the total surface area of amino acid side-chains that are in contact with water. The hydrophilic part involved the parameterization of polar interaction arising from hydrogen bonding.

Sharp-Honig parameters revised an existing hydrophobicity scale (Fauchere & Pliska, 1983) with respect to side-chain size, based on octanol to water transfer energies (Sharp, Nicholls, Friedman & Honig, 1991). The three scales, Engelman-Steitz, Hopp-Woods, and Sharp-Honig, involve differently derived hydrophobicity parameters that represent the range and diversity of such calculations. Figure 2 shows the hydrophobicity values of these three scales.

Implications of Residue Types and Gaps in Proteins

Because each residue has individual properties, certain types of residues are observed to occur more frequently in certain regions of a protein. Using sequence alignment on the sequences of a set of similar proteins can often reveal the regions of consensus residues, whether they are charged residues, small sized residues, hydrophobic residues, or even gaps (Waterman, 1986). Likewise, by identifying where a certain type of residue occurs in a protein sequence, it is possible to infer the likely three-dimensional position of that residue in the protein structure.

Gaps occur more frequently in random coil regions, while residue variability is higher at helical ends, outer strands of beta sheets, or loops between major secondary

structures (Chothia & Lesk, 1986; Sibanda & Thornton, 1993; Rufino, Donate, Canard & Blundell, 1997). Using the calculations of occluded surface, small residues (A, G) were found to have high packing values, while charged residues (D, E, R, H, K) have low packing values, in both membrane proteins and soluble proteins (Eilers, Shekar, Shieh, Smith & Fleming, 2000). Hydrophobic residues (I, L, M, V) have high packing values in soluble proteins, and low packing values in membrane proteins, since hydrophobic effect is not a determining factor in the folding of membrane proteins.

Small residues are more common than charged ones in shorter gaps, and non-hydrophobic residues are more common than hydrophobic ones in longer gaps (Wrabl & Grishin, 2004). Gaps were also found to occur very infrequently in beta strands. They suggested that strands are hydrophobic and responsible for facilitating pairing in beta sheets, making them less able to accommodate insertions and deletions.

Relating Sequence Entropy, Flexibility, and Hydrophobicity

The relationship between sequence variability, hydrophobicity, and protein flexibility is actively being studied. Patterns of their relationships are often observed with respect to sequence entropy as a measure of sequence variability. Sequence entropy was used to discover correlations between protein flexibility and hydrophobicity (Liao et al., 2005). A detailed description of their method includes:

1. Selected a set of 130 unique, well-characterized and diverse, single and multi-chained proteins from Protein Data Bank (PDB, 2005). A set of 130 proteins gave sufficient amount of data for this initial study, but more proteins would

be included in the future. Files (.pdb files) that contain important information of each protein were downloaded.

2. Sequence alignments of each of the 130 proteins were obtained by using BLASTP (NCBI, 2004). The resulting alignments were generated mostly from using default parameters, with exception of a 40% bit score cutoffs.
3. For each of the 130 proteins, a 9 Å radius C_{α} packing density was calculated from the number of C_{α} found in a spherical volume of said radius. The distance between two C_{α} was determined from the atomic coordinates obtained from the .pdb files. The 9 Å cutoff was determined from a set of 7 to 11 Å radii for a limited set of query proteins.
4. The inverse packing density of each protein's residues was easily calculated by finding the reciprocal of the residue's C_{α} packing density.
5. Sequence entropy S_k was calculated for each residue position, using equation 2 (see page 9) on the sequence alignments obtained from BLASTP. Gaps were not treated as residues, and therefore excluded from the calculation.
6. The sequence entropy values for residue positions of all 130 proteins are then averaged according to the inverse packing density of that same position. This is called the "single average entropy."
7. The "double average entropy" can be calculated by averaging the sequence entropy values within individual proteins first, and then averaged again between the full set of proteins.

8. The ‘Gerstein-Altman single average entropy’ was calculated in similar manner as step 6, but using the ‘Gerstein-Altman sequence entropy’ instead of S_k . The Gerstein-Altman sequence entropy was calculated by subtracting S_k with the random entropy S^R for each protein. S^R can be determined from Equation 4 (Gerstein & Altman, 1995), shown below.

$$S^R = -\sum_{j=1,20} P_j \ln P_j \quad (4)$$

where P_j represent the probability of amino acid j , appearing in the whole protein sequence alignment, and j ranges from 1 to 20.

9. The plot of single and double average entropy, and the Gerstein-Altman single average entropy against inverse density is shown in Figure 6.
10. Hydrophobicity of the residues of the query proteins was calculated by using three different hydrophobicity scales: Hopp-Woods, Engelman-Steitz, and Sharp-Honig. The same single averaging method was used to determine the average hydrophobicity for each inverse density value.
11. Figure 7 shows that the sequence entropy and hydrophobicity correlates with inverse packing density.

The correlation plot of Figure 6 involves two main regions, high and low density (Liao et al., 2005). Region I corresponds to the strongly correlated behavior for inverse packing density from 0.040 to 0.083 (25 and 12 C_α), where 74.9% of the 130 proteins’ residues fall in this region. A linear correlation was derived to be $y=12.35x-0.20$, with a

correlation coefficient of 0.997 and $P < 0.001$. Region II corresponds to the inverse packing density of 0.091 to 0.167 (11 to 6 C_α), where 24.4% of the total residues fall in this region. This correlation is consistent with residues that are very flexible and thus have optimal sequence entropy.

Figure 6. Single average entropy, double average entropy, and Gerstein-Altman single averaged entropy against flexibility (Liao et al., 2005).

The single and double average entropy curves are essentially identical, except for their standard deviations. The double averaging calculation greatly reduced the standard deviations values. Since the results of single and double average entropy are almost identical, the similarity of the two types of averaging indicates that such averaging is consistent. The Gerstein-Altman single average entropy curve, calculated by considering

the possible effect of random entropy for each protein, was also found to be comparable to the single and double averaged curves.

Figure 7. Average sequence entropy, hydrophobicity, and flexibility (Liao et al., 2005).

According to the results, at tightly packed regions of a protein, average sequence entropy increases almost linearly as flexibility increases (Liao et al., 2005). However, at the lower density regions, average sequence entropy stays relatively constant while flexibility changes. The same calculations were also performed on the 130 query proteins using hydrophobicity data. The hydrophobicity curve matched closely with the sequence entropy curve, indicating that there is a direct correlation between the two.

The research results showed well-behaved results for packing density of 25 to 6 C_{α} (Liao et al., 2005). However, flanking regions appear less predictable. Less than 1% of the total protein residues fell under those two anomalous regions. Further studies on these two extreme regions will provide a more complete understanding of the relationship between sequence entropy, packing density and hydrophobicity.

CHAPTER 4

Hypothesis and Objectives

Hypothesis

The hypothesis of this study is that the correlations of sequence entropy and hydrophobicity when plotted against inverse packing density (estimate of flexibility) show significant similarity, where small residues and gaps are found in excess for regions of low and high flexibility, respectively.

Objectives

The main objective of this project is to directly examine the residue characteristics in the regions of very high and very low flexibility. This includes the following specific tasks:

1. To study the effects of packing radii and BLAST cutoff criteria, using the full set of 130 proteins.
2. To see if the lower flexibility regions contain a larger fraction of small (glycine, alanine) residues, and the higher flexibility regions contain a larger fraction of gaps.
3. To calculate directly the percentage of strongly hydrophobic residues in all flexibility regions and compare it with sequence entropy, as well as other calculations derived from the hydrophobicity scales.

CHAPTER 5

Materials and Methods

Materials

1. Diverse and well-characterized 130 protein sequences shown in Table 1 (PDB, 2005) and alignments (BLASTp) (Liao et al., 2005). (.txt)
2. Inverse density calculation data for each residue (Liao et al., 2005). (.txt,.xls)
3. Sequence entropy values (Liao et al., 2005). (.txt)
4. Perl programs to parse the sequence data (alignments, entropy, density) and to analyze. (See Appendix A for Perl program codes.)
5. Microsoft Excel for merging and analyzing data.

Table 1. The list of the Protein Data Bank identities of the 130 query proteins used.

1A1I	1AAT	1AHN	1ANG	1AW9	1BG3	1BXQ	1D6M	1E5M	2CTS	3EST	4DFR	5RUB
1A1S	1AB4	1AI2	1A05	1AYE	1BIA	1BYT	1DAJ	1EBV	2LBP	3GBP	4MDH	6LDH
1A32	1ACB	1AK2	1A0B	1AYL	1BIT	1CBQ	1DCS	1EEH	2LDX	3GRS	4PEP	6XIA
1A3C	1ADD	1AKO	1AQO	1AYX	1BLZ	1CEX	1DHS	1HGU	2LIV	3PFK	4TNC	7API
1A3S	1ADI	1ALB	1AQH	1AZI	1BN6	1CJX	1DHT	1LZ1	2PRK	3PGK	5ACN	7CAT
1A4B	1AE4	1ALC	1ATP	1BA3	1BO6	1CK6	1DIN	1OMD	2RN2	3PGM	5CHA	8ADH
1A59	1AF3	1ALN	1AV5	1BC2	1BOH	1CRC	1DMR	1RBP	2TAA	3PSG	5CPA	8ATC
1A5Z	1AGM	1AMN	1AV6	1BF2	1BSI	1CRM	1E1K	1RHD	3BLM	3RN3	5CPV	8DFR
1A6F	1AGX	1AMP	1AV7	1BFD	1BT3	1CRZ	1E3H	1TON	3CLA	3RP2	5CTS	9PAP
1A6Q	1AHA	1AN9	1AW5	1BGO	1BUL	1CSR	1E3Q	2ACT	3CNA	4APE	5LDH	9WGA

Methods

The raw sequence data were mostly obtained from the Lustig group (Liao et al., 2005). Simple data merging, parsing, and detailed data analysis were performed using Perl programs and Microsoft Excel. Figure 8 shows a flowchart of how the steps of this study are connected.

Figure 8. The flowchart of the different steps taken in this study.

Collecting and Combining Data

Since the sequence data for each protein used were separated in various files, it was more efficient for analysis to combine these data into one file per protein (Liao et al., 2005). A Perl program was written to perform the following collecting and combining data step:

1. Opened the .txt files containing sequence alignments, sequence entropy, and inverse density for one protein.

2. Read all the lines that contain relevant data and wrote them to one output file per protein. For example, “3psg.c9” is the combined file’s name for the data of the 3psg protein, obtained using the 9 Å packing radius.
3. Repeated steps 1 and 2 for the whole set of 130 proteins.
4. Each line of the output files contained: protein name, query residue, residue position, density at residue position, sequence entropy at residue position, sequence alignment result from BLASTp at residue position.

The merged files were manually checked for consistency using Microsoft Excel, making sure that the sequence entropy, packing density, residue, and sequence alignment data are correct for each residue position of each protein. Lustig and coworkers had manually truncated some of the protein input files (Liao et al., 2005), due to the unclear and missing data at certain regions of these proteins.

Residue Counting

The analyses consisted of counting different residue types at each density value for all the proteins. However, the size and complexity of the original output files required additional programming in Perl in order to process the data:

1. Opened an output file from the ‘collecting and combining data’ step (e.g., 3psg.c9).
2. Counters were set for density 0 to 60. When a residue of interest was detected at a residue position, the corresponding counter was incremented by one.
3. Residues of interest were small residues (A, G), strongly hydrophobic residues (V, I, L, F, M, Y, W), and gaps. Gaps were treated as residues for simplicity.

4. Repeated 1 to 3 for the whole set of proteins, which includes the 130 query proteins and the aligned proteins obtained from BLASTp.
5. The total number of residues for each density was counted.
6. Outputted the count results as a summary file (e.g. result_count.txt).

Data Analysis

The first objective was to examine the effects of radii of packing and BLASTp parameters for the set of the 130 query proteins. The tasks completed:

1. Selected the .pdb files for the 130 query proteins.
2. Modified Liao's Perl program to calculate packing density values for packing radii of 7 through 11 Å (Liao et al., 2005).
3. Selected Liao's BLAST output .txt files for the chosen proteins.
4. Modified Liao's Perl program to extract outputs using different bit score cutoffs, corresponding to 20% through 80% of highest bit score.
5. Plotted results in Microsoft Excel.
6. Evaluated sequence entropy versus inverse packing density calculations for the different packing radii and bit score cutoffs.

The second and third objectives involved looking at residue characteristics in detail, especially the very large and very small inverse density values. Microsoft Excel was mainly used for plotting graphs and analyzing data of the following:

1. Fraction strongly hydrophobic residues versus inverse density. This was compared with graph of sequence entropy versus density.

2. Fraction of glycine and alanine residues versus inverse density. This was used to check if smaller residues are more predominant in densely packed regions.
3. Fraction of gapped residues versus inverse density. From this analysis, the output was expected to show that highly flexible regions have more gaps. Here, the fraction of gaps is calculated out of all the possible number of sequence positions, gapped plus aligned. Note that this denominator is not the same as the total number of residues used to calculate sequence entropy.

These fraction residue data were also partitioned into sections according to the sequence entropy curve. This will be done to examine each region's characteristics.

CHAPTER 6

Results and Discussion

Results

The seven residue frequency distributions for the 130 query proteins obtained using 6 to 12 Å packing radius, with respect to C_α packing density, are shown in Figure 9. There are a total of 42,253 query residues, but each packing radius gives a different value for mean, median, and standard deviation, as shown in Table 2. The distribution curve becomes steeper and narrower when packing radius becomes lesser than 9 Å, and becomes flatter and wider as the C_α packing radius increases. The most normal-like distributions are those of radius 6, 7, 8, and 9 Å, since their data points are more normally distributed.

Table 2. The mean, median, and standard deviation of the frequency distributions of C_α packing density, calculated using a packing radius of 6 to 12 Å.

C_α Packing Radius (Å)	6	7	8	9	10	11	12
Mean C_α Packing Density	6.297	8.474	10.219	13.02	16.789	21.335	25.966
Median Packing Density	6	8	10	12	16	20	25
Standard Deviation	1.735	2.311	2.983	4.352	6.408	9.188	11.995

The average sequence entropy values of each residue position of the 130 query proteins were calculated from the alignment sequences using BLASTp. A 40% bit score cutoff was initially chosen because of its utility in previous calculations done by the Lustig group (Liao et al., 2005). The average entropy was then calculated for packing radius 6 to 12 Å and plotted against the inverse packing density, as shown in Figure 10A and 10B. Figure 10 shows that the apparent scatter of these sequence entropy plots

Figure 9. Residue frequency distributions over packing density for 6 to 12 Å packing radius. Figures 9A to 9G show the frequency distributions of all the query residues as a function of the C_{α} packing density for 6 to 12 Å packing radius. These query residues are obtained from the 130 query proteins. The query proteins consist of 42,252 residues total, with mean packing densities of 6.297, 8.474, 10.219, 13.020, 16.789, 21.335, and 25.966, medians of 6, 8, 10, 12, 16, 20, and 25, and standard deviations of 1.735, 2.311, 2.983, 4.352, 6.408, 9.188, and 11.995, for packing radii 6, 7, 8, 9, 10, 11, and 12 Å, respectively.

Figure 10. The graph of average sequence entropy over 1/density, for the packing radius of 6 to 9 Å and 9 to 12 Å, using a 40% bit score cutoff. Average sequence entropy of the 130 query proteins plotted against inverse packing density. The average sequence entropy (ordinate) is calculated for each inverse packing density, by averaging the entropy values of all the residues within a density interval of one. The protein sequences used to calculate sequence entropy are selected using a 40% bit score cutoff. The inverse packing density (abscissa) of each residue is determined from the C_{α} coordinates supplied by the PDB protein files, using the specified range of packing radii. (A) Packing radius of 6, 7, 8, and 9 Å were used. (B) Packing radius of 9, 10, 11, and 12 Å were used.

indicates the 9 Å cutoff is the best choice. In the case of the 9 Å C_{α} packing radius, there are two regions that stand out. The first region is a steep slope ranging from packing density 12 to 25, while the second region is a flat slope ranging from density 26 to 30.

A general trend can be seen in Figure 10A and 10B. As the packing radius increases, the average entropy curve shifts towards the left and its slope grows steeper. These changes in the plots were crosschecked with the fraction of non-strongly hydrophobic residues (A, C, D, E, G, H, K, N, O, P, R, S, T). The plot of fraction non-strongly hydrophobic residues, using a 40% bit score cutoff and packing radius 6 to 12 Å (see Appendixes B1, B2, and B3), indicates that the average sequence entropy and hydrophobicity are correlated for each choice of packing radius and maintain similar relative plots. So in a scale of relative packing density and flexibility, there is no fundamental difference.

To see the effect of different bit score cutoffs, Figure 11 shows the different normalized average sequence entropy plots for 9 Å packing radius, using 20%, 40%, 60%, and 80% bit score cutoffs. The bit score cutoffs serve as a filter for the hundreds of sequence alignments produced using BLASTp. Qualified sequences for each protein have alignment bit scores higher than 20%, 40%, 60%, or 80% of the highest bit score of that protein. 20% bit score cutoff gives a total of 10,540 sequences, while 40%, 60%, and 80% give 7,242, 3,951, and 1,886 sequences, respectively. Normalization of these four plots was necessary in order to remove the differences in their magnitude and simply compare their trends. To normalize an average entropy value, the difference between the entropy of interest and the lowest entropy value was divided by the difference between

Figure 11. Normalized average sequence entropy over 1/density, using different bit score cutoffs. Normalized average sequence entropy of the 130 query proteins plotted against inverse packing density of a 9 Å radius, using different bit score cutoffs. The average entropy (ordinate) is calculated as described in Figure 10. The bit score cutoffs used for filtering out the sequences for calculate the entropy, were 20%, 40%, 60%, and 80% of the highest bit score from each BLASTp result.

the highest and lowest entropy value. Even though the plot for the 80% bit score cutoff seems to be the least smooth, the normalized average entropy plots are generally overlapping each other. Percentage identities of the BLASTp alignment sequences were also used as a filter to reconfirm the result from the bit score cutoffs, as both determine the protein set's degree of homology. The 20%, 40%, 60%, and 80% identities were used for this calculation (see Appendix C). Results of the percentage identities showed that other than a slight overall difference in magnitude, for each average sequence entropy plot, the curves are very similar. The significance of the cutoffs is to determine what sequence set is being used for calculation.

A high percentage bit score cutoff generally indicates that a BLASTp result represents a set of highly homology and closely related sequences. This can greatly reduce the size of the sequence set, causing each calculated result to strictly represent a specific protein. Likewise, a BLASTp result with a low cutoff will give a bigger set of proteins, but may include too many unrelated sequences. Since this sequence entropy versus flexibility study is to give a general correlation for all proteins, the cutoff that is selected needs to include as many homologous sequences as possible, and at the same time, include little to none unrelated sequences. The 40% bit score cutoff was selected as reasonable, balancing homology with diversity of sequence variability (Park, Teichmann, Hubbard & Chothia, 1997; Brenner, Chothia & Hubbard, 1998).

In a previous study using the 130 query proteins, the average sequence entropy appears closely related to the average hydrophobicity plot of three different scales (Liao et al., 2005). To double-check this correlation between sequence entropy and

hydrophobicity, the fraction of non-strongly hydrophobic residues for each packing density was calculated using 9 Å packing radius and 40% bit score cutoff. Figure 12 shows the superposition of the fraction of non-strongly hydrophobic residues and the average sequence entropy, as a function of inverse density. The fraction of non-strongly hydrophobic residues is calculated by dividing the number of non-strongly hydrophobic residues with the total number of residues for each packing density value. This calculation was done using two different sets of residues. One set is from the 130 query protein sequences only, while the other set is from the 7242 alignment sequences. Since the difference appears to be insignificant, as both curves show the same pattern over the entire region, subsequent fraction of residues calculations were made using the residues from the 7242 alignment sequences.

Following the previous works in the Lustig group (Liao et al., 2005), the average hydrophobicity of all aligned residues was calculated using Hopp-Woods, Sharp-Honig, and Engelman-Steitz hydrophobicity scales. Figure 13 shows an overlay of the average hydrophobicity plot of these three scales and the fraction of non-strongly hydrophobic residues. The two plots show high similarity throughout the range of flexibility, as described by inverse density, suggesting that the fraction of non-strongly hydrophobic residues can also be used to correlate with sequence entropy.

Similar analyses were made for small-sized residues and gaps. The fraction of small residues such as glycine and alanine, and average sequence entropy are plotted versus inverse packing density, as shown in Figure 14. From observation, the fraction of small residues increases with flexibility in very tightly packed regions (16 C_α and above),

Figure 12. Fraction of non-strongly hydrophobic residues and average sequence entropy versus inverse density. Comparison of average sequence entropy and fraction of non-strongly hydrophobic residues (ordinate), with respect to inverse C_{α} packing density (abscissa) of 9 Å radius and 40% bit score cutoff. Average sequence entropy is calculated for each inverse packing density, by averaging the entropy values of all the residues within a density interval of one. Fraction of non-strongly hydrophobic residues for each inverse packing density is determined by dividing the number of non-strongly hydrophobic residues in an inverse packing density (all residues that are not F, I, L, M, V, W, Y) with the total number of residues that fall in the same inverse packing density (Poupon & Mornon, 1999).

Figure 13. Fraction of non-strongly hydrophobic residues and average hydrophobicity versus inverse density. Average hydrophobicity is calculated by averaging three different hydrophobicity scales: Hopp-Woods, Sharp-Honig, and Engelman-Steitz.

Figure 14. Fraction of small residues (A i.e. alanine, G i.e. glycine) and average sequence entropy over inverse packing density (9 Å, 40% bit score). Comparison of average sequence entropy and fraction of small-sized residues (ordinate), with respect to inverse C_{α} packing density (abscissa) of 9 Å radius and 40% bit score cutoff. Average sequence entropy is calculated for each inverse packing density, by averaging the entropy values of all the residues within a density interval of one. Fraction of small residues for each inverse packing density is determined by dividing the number of small residues (A, G) in an inverse packing density with the total number of residues that fall in the same inverse packing density. The fraction of residue axis (left ordinate) is in reversed order, so that it can be compared to the average sequence entropy plot.

and becomes constant in the more flexible regions (below 16 C_α). Figure 15 shows the graph of the fraction of gaps. The fraction of gaps increases as flexibility increases. The increase of the fraction of gaps goes past the flattening of sequence entropy that begins at 11 C_α packing density. In extremely flexible regions of 3 and 4 C_α packing density, the fraction of gaps jumps up dramatically. However, it should be noted that this region accounts for only 2% of the query residues. All fractions of residues calculations were made using residues obtained from 9 Å packing radius for query proteins and 40% bit score cutoff for aligned residues.

Discussion

Optimum Percentage Bit Score and Identity Cutoffs

The percentage bit score cutoffs and the percentage identity cutoffs are both methods to determine and filter the level of homology of BLASTp sequence alignment outputs. The normalized average entropy calculations using bit score cutoffs of 20%, 40%, 60%, and 80%, showed that the cutoffs do not seem to have a significant effect on average entropy. The effect of percentage identity cutoffs was also observed side by side with the bit score cutoffs in Appendix C. At the same percentage cutoffs, both methods yield very similar results and patterns. Due to the vagueness of a clear optimum cutoff value, the 40% bit score cutoff was selected as the cutoff value. Past supporting work has indicated that 30% identity signifies homology for sequence alignments of 150 residues or greater, while 40% identity indicates homology for alignments of 70 residues (Brenner et al., 1998). Since the number of sequence alignments for majority of the proteins is

Figure 15. Fraction of gaps (-) and average sequence entropy over inverse packing density (9 Å, 40% bit score). Comparison of average sequence entropy and fraction of gaps (ordinate), with respect to inverse C_{α} packing density (abscissa) of 9 Å radius and 40% bit score cutoff. Average sequence entropy is calculated for each inverse packing density, by averaging the entropy values of all the residues within a density interval of one. Fraction of gaps for each inverse packing density is determined by dividing the number of gaps in an inverse packing density with the total number of residues that fall in the same inverse packing density.

already selected at a default value of 100, selecting 40% identity is a reasonable cutoff to include as many homologous sequences as possible (Park et al., 1997). Judging from a comparison of entropy values derived from bit score cutoffs and identity cutoffs, the choice of the 40% bit score cutoff is slightly more conservative than the 40% identity cutoff.

Optimum Packing Radius

From the residue frequency distributions alone, it is easy to decide that packing radii of 10, 11, and 12 Å are less suitable, because radius of 6 to 9 Å gives better-behaved distributions. To decide between packing radii 6, 7, 8, and 9 Å, plots of average entropy and fraction of non-strongly hydrophobic residues were considered. The least apparent scatter of the remaining plots of sequence entropy versus inverse density involves packing density of 9 Å. In addition, the relative placement of sequence entropy and hydrophobicity with respect to an inverse density for two particular packing radii shows consistency. In the case for all the plots of packing radii, the major features of the sequence entropy plots can be identified.

Past supporting work has indicated an upper boundary of 14.9 Å and a lower boundary of 4 Å for packing radius, suggesting that 6 to 9 Å is an acceptable radius range (Miyazawa & Jernigan, 1985; Covell, 1994). The most significant range of C_{α} radii is around 8 to 9 Å (R. L. Jernigan, personal communication, March 27, 2001). The two pairs of sequence entropy and fraction of non-strongly hydrophobic residues plots for 8 and 9 Å give an appropriate range of the corresponding inverse density. The range of pairs of plots does not appear to be significant, since the estimated distance between each

pair of plots is about 0.015 inverse units ($\text{\AA}^{-1}$). Conceivably, there is no single optimal packing radius, but rather a minimal range of radii. For the purpose of our study, 9 $\text{\AA}$ was selected.

Fraction of Strongly Hydrophobic Residues

The general tendency of a hydrophobic residue is to fold itself into the core of the protein where it can avoid the aqueous environment. The fraction of non-strongly hydrophobic residues shows four distinct regions. The two main regions of sequence entropy for 9 $\text{\AA}$ C_α packing radius have C_α packing densities of 19 to 12 and 11 to 6, which contain approximately 45% and 40% of the total number of aligned residues, respectively. The two anomalous regions are at packing densities less than 6 C_α , and greater than 19 C_α , which includes around 2% and 13% of the total aligned residues, respectively. The former value of 2% is fairly consistent with fraction of query residues (Liao et al., 2005). However our high density anomalous region is calculated starting at a lower density of 19 C_α . It turns out that our calculation of the fraction of query residues is about the same as the aligned ones.

Of the two main regions, the fraction of non-strongly hydrophobic residues initially increases as the density decreases from 19 C_α to 12 C_α . This is intuitive if one considers that strongly hydrophobic residues generally pack themselves into the core of the proteins to avoid the surrounding water molecules. However, when the packing density decreases below 12 C_α , the fraction of non-strongly hydrophobic residues begins to plateau shows behavior consistent with residues being optimally flexible and likely to be adjacent to solvent. In the packing density regions of higher than 19 C_α , the fraction

of non-strongly hydrophobic residues increases dramatically as density increases. This trend holds true for all tested packing radii. Even though this region contains only about 13% of the total number of aligned residues, the result suggests that in very highly packed regions of a protein, hydrophobic effect alone is not sufficient for strong packing.

Fraction of Small Residues

Small residues were found to be in higher ratios in densely packed areas within a protein (Eilers et al., 2000). In the region of C_{α} packing density 16 and higher, the fraction of small residues increases as density increases. This holds true for even the densest regions of 30 C_{α} . This higher ratio of substituted alanine and glycine suggests that small sized residues are preferred in areas that are tightly packed. This preference in denser regions in proteins may be due to the small size of these smaller residues' side chains, which can be more effectively packed to obtain higher packing density. When compared to the fraction of non-strongly hydrophobic plot, the increase in fraction of small residues seems in tandem. This seems to reinforce the idea that residue size is as significant as hydrophobicity in the densely packed regions of proteins. As density drops below 16 C_{α} , the fraction of small sized aligned residues becomes constant at about 15%. This suggests that below a certain density, small residues become less significant in their contributions to protein packing, as opposed to hydrophilic residues.

Fraction of Gaps

The presence of gaps usually involves some type of mutation, whether it is an insertion or deletion. The ability to accommodate mutations was investigated by the Lustig group and found to be correlated to the enhanced flexibility of the local residue

positions (Hsieh et al., 2002). Regions with enhanced flexibility are often at the surface of the proteins, helical ends, outer strands of beta sheets, or loops between major secondary structures (Chothia & Lesk, 1986; Sibanda & Thornton, 1993; Rufino et al., 1997). At very highly packed regions, or regions of very low flexibility, the presence of gaps is extremely low. As density decreases, flexibility increases, and therefore, the number of gaps also increases. It should be noted that, with packing density of 5 C_{α} and higher, the increase in the fraction of gaps is relatively small. However, when packing density is below 5 C_{α} , where the region is extremely flexible, the ratio of gaps increases significantly to about 50%. Although, this region only has less than 1% of the total number of aligned residues, this sudden jump in the fraction of gaps may still be significant because it indicates that a third to half of the residues in these regions are gaps.

CHAPTER 7

Conclusions

Studying the effects of packing radius and percentage bit score cutoff, led to two conclusions. First, the packing radius has an effect on the frequency distributions of the amino acid residues. Packing radius of 9 Å was determined to be an appropriate one for the purpose of our research, based on the well-behaved and more normal frequency distributions and sequence entropy plots. Secondly, percentage bit score cutoff has no distinct effect on the overall data. Choosing to use 40% of the highest bit score as a cutoff for BLASTp results is also found to be reasonable, since correlated behavior of sequence entropy and hydrophobicity remains effectively the same for all packing radii.

Observations in the fraction of gaps and small residues show consistency with the hypothesis, which states that the correlations of sequence entropy, packing density, and hydrophobicity hold true for extreme regions of flexibility. Here, there are indications of larger fractions of small residues and strongly hydrophobic residues in the low flexibility region, and a higher count of gaps in the high flexibility region. The presence of gaps are fairly infrequent in most of the range of C^α packing densities, only increasing slowly as flexibility increases. However, gaps become predominant in extremely low-density regions, suggesting that mutations can occur easily in very flexible areas of proteins. Small residues like alanine and glycine are important contributors to highly packed regions, due to their size. Their influence becomes less significant when packing density falls below 16 C^α for our chosen 9 Å packing radius, where their occurrences stabilize.

The fraction of non-strongly hydrophobic residues follows the average sequence entropy trend closely, with a distinct linear region that increases proportionally with flexibility and a flat region that stays at a constant fraction of residues. At extremely high-density regions, containing 13% of the total number of aligned and query residues, the fraction of strongly hydrophobic residues decreases, as the number of small residues increases.

These observations in the trends of residue types and gaps in relation to flexibility are consistent to what is expected. This gives a certain level of confidence to the protein residue data through out the whole range of flexibility, and especially in the extreme regions. Additional studies will be required to explain the anomalous trend in the sequence entropy in the low and high flexibility regions of proteins. To have the ability to determine a protein's structure from its sequence is an important goal in bioinformatics and for many pharmaceutical companies. In the bigger picture, such knowledge will not only save resources on designing proteins and drugs, but also help to fight diseases more effectively. Although this study does not completely answer the protein-folding problem, it does contribute to our overall learning of this field.

References

- Altschul, S. F., Gish, W., Miller, W., Myers, E., & Lipman, D. J. (1990). Basic local alignment search tool. *J. Mol. Biol.*, 215, 403-410.
- Altschul, S. F., Madden, T. L., Saffer, A. A., Zhang, J., Zhang, Z., Miller, W., et al. (1997). Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucl. Acids Res.*, 25, 3389-3402.
- Bahar, I., Wallqvist, A., Covell, D. G., & Jernigan, R. L. (1998). Correlation between native-state hydrogen exchange and cooperative residue fluctuations from a simple model. *Biochemistry*, 37, 1067-1075.
- Baker, D., & Sali, A. (2001). Protein structure prediction and structural genomics. *Science*, 294, 93-96.
- Brenner, S. E., Chothia, C., & Hubbard, T. J. P. (1998). Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships. *Proc. Natl. Acad. Sci. USA*, 95, 6073-6078.
- Chothia, C., & Lesk, A. M. (1986). The relation between the divergence of sequence and structure in proteins. *EMBO J*, 5, 823-826.
- Concept 2: Protein Structure: Amino Acid Building Blocks.* (n.d.). Retrieved July 7, 2005, from http://www.phschool.com/science/biology_place/biocoach/translation/prstaabb.html
- Concept 4: The Peptide Bond.* (n.d.). Retrieved July 7, 2005, from http://www.phschool.com/science/biology_place/biocoach/translation/pepb.html
- Covell, D. G. (1994). Lattice model simulations of polypeptide chain folding. *J. Mol. Biol.*, 235, 1032-1043.
- Dill, K., Bromberg, S., Yue, K., Feibig, K., Yee, D., Thomas, P., et al. (1995). Principles of protein folding - a perspective from simple exact models. *Protein Sci.*, 4, 561-602.
- Eilers, M., Shekar, S. C., Shieh, T., Smith, S. O., & Fleming, P. J. (2000). Internal packing of helical membrane proteins. *Proc. Natl. Acad. Sci. USA*, 97, 5796-5801.
- Eisenberg, D., Marcotte, E. M., Xenarios, I., & Yeates, T. O. (2000). Protein function in the post-genomic era. *Nature*, 405, 823-826.

- Engleman, D. M., Steitz, T. A., & Goldmann, A. (1986). Identifying nonpolar transbilayer helices in amino acid sequences of membrane proteins. *Annu. Rev. Biophys. Chem.*, 115, 321-353.
- Fauchere, J. L., & Pliska, V. (1983). Hydrophobic Parameters p of amino acid side chains from the partitioning of N-acetyl-amino-acid amides. *Eur. J. Med. Chem.*, 18, 369-375.
- Gerstein, M., & Altman, R. B. (1995). Average core structures and variability measures for protein families: application to the immunoglobulins. *J. Mol. Biol.*, 251, 161-175.
- Halle, B. (2002). Flexibility and packing in proteins. *Proc. Natl. Acad. Sci. USA*, 99, 1274-1279.
- Holmes, K. C., & Blow, D. M. (1965). The use of x-ray diffraction in the study of protein and nucleic acid structure. *Methods Biochem. Anal.*, 13, 113-239.
- Hopp T. P., & Woods K. R. (1981). Prediction of protein antigenic determinants from amino acid sequences. *Proc. Nat. Acad. Sci. USA*, 78, 3824-3828.
- Horton H. R., Moran L. A., Ochs R. S., Rawn J. D., & Scrimgeour K. G. (2002). *Principles of Biochemistry Third Edition. [Electronic version]*. Retrieved July 7, 2005, from http://cwx.prenhall.com/horton/medialib/media_portfolio/text_images/FG04_01.JPG
- Hsieh, M., Collins, E. D., Blomquist, T., & Lustig, B. (2002). Flexibility of BIV TAR-Tat: Models of peptide binding. *J. Biomol. Struct. Dyn.*, 20, 243-251.
- Jaenicke, R. (2000). Stability and stabilization of globular proteins in solution. *J. Biotechnol.*, 79, 193-203.
- King, M. W. (2003). *Chemical Nature of the Amino Acids*. Retrieved July 7, 2005, from <http://web.indstate.edu/thcme/mwking/amino-acids.html>
- Koehl, P., & Levitt, M. (2002). Protein topology and stability define the space of allowed sequences. *Proc. Natl. Acad. Sci. USA*, 99, 1280-1285.
- Kono, H., & Saven, J. G. (2001). Statistical theory for protein combinatorial libraries. packing interactions, backbone flexibility, and the sequence variability of a main-chain structure. *J. Mol. Biol.*, 306, 607-628.

- Larson, S. M., England, J. L., Desjarlais, J. R., & Pande, V. S. (2002). Thoroughly sampling sequence space: Large-scale protein design of structural ensembles. *Protein Sci.*, *11*, 2804-2813.
- Levitt, M. (1976). A simplified representation of protein conformations for rapid simulation of protein folding. *J. Mol. Biol.*, *104*, 59-107.
- Liao, H., Yeh, W., Chiang, D., Jernigan, R. L., & Lustig, B. (2005). Protein sequence entropy is closely related to packing density and hydrophobicity. *Prot. Engr.*, *18*, 59-64.
- Lustig, B., Bahar, I., & Jernigan, R. L. (1998). RNA bulge entropies in the unbound state correlate with peptide binding strengths for HIV-1 and BIV TAR RNA because of improved conformational access. *Nucl. Acids Res.*, *26*, 5212-5217.
- Meirovitch, H., & Hendrickson, T. F. (1997). Backbone entropy of loops as a measure of their flexibility: Application to a Ras protein simulated by molecular dynamics. *Proteins*, *29*, 127-140.
- Miyazawa, S., & Jernigan, R. L. (1985). Estimation of effective interresidue contact energies from protein crystal structures: Quasi-chemical approximation. *Macromolecules*, *18*, 534-552.
- Mount, D. W. (2001). *Bioinformatics: Sequence and Genome Analysis*. Cold Spring Harbor, New York: Cold Spring Harbor Press.
- National Center for Biotechnology Information (2005): <http://www.ncbi.nlm.nih.gov/>
- Needleman, S. B., & Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, *48*, 443-453.
- Nicholas Jr., H. B., Deerfield II, D. W., & Ropelewski, A. J. (2000). Strategies for searching sequence databases. *Biotechniques*, *26*, 1174-1191.
- Norton, R. S. (1990). Nuclear magnetic resonance (NMR) spectroscopy: applications to protein structure and engineering. *Aust. J. Biotechnol.*, *4*, 114-120.
- Park, J., Teichmann, S. A., Hubbard, T., & Chothia, C. (1997). Intermediate sequences increase the detection of homology between sequences. *J. Mol. Biol.*, *273*, 349-354.
- Poupon, A., & Mornon, J. P. (1999). "Topohydrophobic positions" as key markers of globular protein folds. *Theor. Chem. Acc.*, *101*, 2-8.

Protein Data Bank (2005): <http://www.rcsb.org.pdb/>

- Rufino, S. D., Donate, L. E., Canard, L. H., & Blundell, T. L. (1997). Predicting the conformational class of short and medium size loops connecting regular secondary structures: application to comparative modelling. *J. Mol. Biol.*, 267, 352-367.
- Schneider, T. D., Stormo, G. D., Gold, L., & Ehrenfeucht, A. (1986). Information content of binding sites on nucleotide sequences. *J. Mol. Biol.*, 188, 415-431.
- Sharp, K. A., Nicholls, A., Friedman, R., & Honig, B. (1991). Extracting hydrophobic free energies from experimental data: relationship to protein folding and theoretical models. *Biochemistry*, 30, 9686-9697.
- Sibanda, B. L., & Thornton, J. M. (1993). Accommodating sequence changes in β -hairpins in proteins. *J. Mol. Biol.*, 229, 428-447.
- Skolnick, J., Fetrow, S. F., & Kolinski, A. (2000). Structural genomics and its importance for gene function analysis. *Nature Biotechnol.*, 18, 283-287.
- Smith, D. K., Radivojac, P., Obradovic, Z., & Dunker, A. K., Zhu, G. (2003). Improved amino acid flexibility parameters. *Protein Sci.*, 12, 1060-1072.
- Smith, T. F., & Waterman, M. S. (1981). Identification of common molecular subsequences. *J. Mol. Biol.*, 147, 195-197.
- Valdar, W. S. J. (2002). Scoring residue conservation. *Proteins*, 48, 227-241.
- Waterman, M. (1986). Multiple sequence alignment by consensus. *Nucl. Acids Res.*, 14, 9095-9102.
- Wrabl, J. O., & Grishin, N. V. (2004). Gaps in structurally similar proteins: Towards improvement of multiple sequence alignment. *Proteins*, 54, 71-87.

Appendix A

PERL Codes Used

```
#-----
# Author: William Yeh
# Date: March 28, 2005
# Purpose: Takes PDB structure file in ASCII text format.
# File: pdb2denMOD.pl (Modified from Liao's.)
#-----

# Updates:
# 3/28/05 William Yeh - Reads straight from directory of PDB files.
#                               Calculates for packing density 6 to 12.
#

# User Specified Variables to Control Analysis and Output
@TabValues = (6, 7, 8, 9, 10, 11, 12); # For each Value, calc's #dist <= Value
# Must be increasing in value.
@TabPrint = (0, 1, 2, 3, 4, 5, 6); # Defines which @TabValues printed
@DebugIndices = (1, 17, 'last'); # Debug-print calcs for these AA's
# These are sequential values starting
# with 1 and ends with no. of total aa's.

# Initialize Amino Acid 3-letter to 1-letter associative list
%AADictionary = (
  'GLY' => 'G', 'ALA' => 'A', 'VAL' => 'V', 'LEU' => 'L',
  'ILE' => 'I', 'MET' => 'M', 'PRO' => 'P', 'PHE' => 'F',
  'TRP' => 'W', 'SER' => 'S', 'THR' => 'T', 'ASN' => 'N',
  'GLN' => 'Q', 'TYR' => 'Y', 'CYS' => 'C', 'LYS' => 'K',
  'ARG' => 'R', 'HIS' => 'H', 'ASP' => 'D', 'GLU' => 'E');

$P_directory = "PDB_files";
opendir (DIRECTORY, $P_directory);

while (defined($P_filename = readdir(DIRECTORY))) {
  if ($P_filename != "." || $P_filename != "..") {

    # Initialize all variables
    $Name = ""; # PDB name (extracted from HEADER line), in lower case
    $Line = "";
    @AtomLine = (); # Temp var for current ATOM line as list
    @AASeqRes = (); # 3-letter form from SEQRES, for checking
    @AA = (); # 3-letter form from ATOM statements
    @AA1 = (); # 1-letter form (translated)
    @Tag = (); # Unused for now
    @Distance = (); # Calc'd distances
    $AAreft = "";
    $AAreft1 = 'A';
    $Indexref = 0;
    $Tagref = "";
    $AACount = 0; # Number of aa's
    $Xref = 0;
    $Yref = 0;
    $Zref = 0;
    $i = 0;
    $j = 0;
    $k = 0;
    @X = ();
    @Y = ();
    @Z = ();
    @XVect = ();
    @YVect = ();
    @ZVect = (); # Temp var for storing vector of aa being compared
```

```

@TabCount = (); # TabCount tracks no. < each @TabValues
@TabDensity= (); # Calc density from TabCount & TabValues
@PrintLine = (); # Temp var holds line for printing
$Count = 0;

open ( IN , "$p_directory/$p_filename") or die "Cannot open input files for read". "\n";

$i = 0; # Indices for AA's
while( <IN> ) {

    $Line = $_; chomp($Line); # Save current line

    # ===== Extract Amino Acid seq from SEQRES statements =====
    if ($Line =~ /^SEQRES +/) { # If SEQRES, extract aa residues
        $Line =~ s/^SEQRES +[\d]+ +[\d]+ +//; # Remove prefix
        $Line =~ s/[0-9A-Z]+ +[\d]+ *$//; # Remove suffix
        push(@AASeqRes, split(/ +/, $Line)); # Append onto AASeqRes list
    }

    # ===== Extract alpha-C (x,y,z) from ATOM statements =====
    elsif ($Line =~ /^ATOM +[\d]+ +CA +/) { # Find alpha-Carbon ATOM lines
        $i += 1; # aa index for tracking only
        @AtomLine = split(/ +/, $Line);
        ($AAref, $Indexref, $Xref, $Yref, $Zref) = @AtomLine[3,4,5,6,7];
        if(length($AAref) eq 4) {
            $AArefl = substr($AAref,0,1);
            $AAref = substr($AAref,1,3);
        }
        $Tagref = $Indexref; # Store index in Tag

        if ( $AADictionary{$AAref} ne " " && $AArefl eq 'A'
            && $Indexref ne 'B' && $Indexref ne 'C' ) { # ONLY EXTRACT CHAIN

            push( @AA , $AAref);
            push( @AA1, $AADictionary{$AAref});
            push( @X , $Xref);
            push( @Y , $Yref);
            push( @Z , $Zref);
            push( @Tag, $Tagref);

        }

        $AArefl = 'A';
    }

    elsif ($Line =~ /^HEADER +/) { # Extract PDB name from HEADER line
        @AtomLine = split(/ +/, $Line);
        $Name = substr($Line,62,4);
        $Name = lc($Name);
    }

};

$AACount = 1 + $#AA;
@TabValues = sort { $a <=> $b } @TabValues; # Make sure it's ascending

# Output Filename...
open ( OUT , '>'. "$Name". ".den") or die "Cannot open out_pdbden.txt for write.\n";

# ===== Print File Header =====
printf OUT "I NAME= ". $Name . "\n";
printf OUT "I NO_RESIDUES= ". $AACount . "\n";
$PrintLine = "I SEQ= ". join(" ", @AA1) . "\n";
printf OUT $PrintLine;

### !!!!!ADDED STUFF!!!!!! ###
$TempFromSEQRES = "";
@TempFromSEQRES = ();
@AASeqRes1 = ();

```

```

while (@AASeqRes != NULL) {
    $tPop = pop(@AASeqRes);
    push(@TempFromSEQRES, $AADictionary{$tPop});
}
while (@TempFromSEQRES != NULL) {
    $tPop = pop(@TempFromSEQRES);
    push(@AASeqRes1, $tPop);
}
$TempFromSEQRES = "I2SEQ=" . join(" ", @AASeqRes1) . "\n";
printf OUT $TempFromSEQRES;

# I SEQ = sequence with the coordinate.
# I2SEQ = the 'complete' sequence.
printf OUT "#". "\n";

# ===== Calculate distances and tabulate =====
# Note that $i is the aa location, and $j is used to scan to build vects.
for ($i = 0; $i < $AACount; $i++) {

    for ($j = 0; $j < $AACount; $j++) {
        @Distance[$j] = sqrt( (($X[$j] - $X[$i])**2)
                               + (($Y[$j] - $Y[$i])**2)
                               + (($Z[$j] - $Z[$i])**2) );
    };

    # ===== Sort and tabulate according to distance
    for ($j = 0; $j <= $#TabValues; $j++) {
        $Count = 0;
        for ($k = 0; $k <= $#Distance; $k++) {
            if ($Distance[$k] <= $TabValues[$j]) {
                $Count++;
            };
        }
        $TabCount[$j] = $Count;
        $TabDensity[$j] = 1000 * $Count / ((4.0/3.0)*3.14159
            * ($TabValues[$j]**3)); # Compute density
    }

    # ===== Print to output (and debug) file =====
    $PrintLine = "D "
    . $Name
    . " ". sprintf("%03d", 1+$i)
    . " ". sprintf("%1s ", $AA1[$i]);

    foreach $i (@TabPrint) { # Output count C()
        $PrintLine = $PrintLine
        . "C(" . $TabValues[$i] . ") = "
        . sprintf("% 3d", $TabCount[$i]) . " ";
    };
    printf OUT $PrintLine . "\n";
};
close(IN);
}
close(OUT);

```

```

#-----
# Author: William Yeh
# Date: March 28, 2005
# Purpose: Takes BLAST output file in ASCII test format, using Bit Score and/or % Identity cutoffs.
# File: pdb2denMOD.pl (Modified from Liao's.)
#-----

```

Specify User Parameters


```

$IdenPercentMin = 0.10;    # Min value of Identity% score for qualified seq
                           # ($IdenPercentMin is another approach of BLAST results cut-off. Not used for this study)
$HomologMin    = 1;       # Min value of non-deleted homologs for entropy calc
                           # ($HomologMin is needed for entropy calc, so an error won't occur when dividing by 0)
$ScoreMin      = 100;     # Min value of match score
$BitCutOff     = 40;

$P_directory = "blast_all";
opendir (DIRECTORY, $P_directory);
while (defined($P_filename = readdir(DIRECTORY))) {
    if ($P_filename != "." || $P_filename != "..") {

        # Initialize all variables
        $PDBName    = "";    # Passed param; used to tag residues in output only
        $Line       = "";
        @LineSplit  = ();
        $ScoreBits  = 0;
        $Expect     = 0;
        $IdenPercent = 0;
        $PositivePercent = 0;
        $QuerySeq   = "";    # Original Query Seq (from 1st pair)
        $QuerySeqTemp = "";  # Query Seq in pair for manipulation
        $SubjectSeqTemp = ""; # Subject Seq in pair for manipulation
        $QueryOffset = 0;    # Initial seq offset (always 1 for 1st seq)
        $LengthDiff = 0;    # Temp vars ...
        $QueryOffDummy = 0;
        $Dummy       = "";
        $Dummy2      = "";
        @SeqList     = ();
        $LineOut     = "";
        %EntropyCount = ();
        $EntropyCountTot = 0;
        $Prob        = 0;
        $Entropy     = 0;

        # Setup files for writing and reading
        open ( IN , "$P_directory/$P_filename") or die "Cannot open input files for read". "\n";
        $PDBName = substr($P_filename,0,4);    # Get PDB protein name as 1st parameter
        open ( OUT , ">". $PDBName . "_" . $BitCutOff . ".ent") or die "Cannot open output file for write.\n";
        open ( OUTD , ">". $PDBName . "_" . $BitCutOff . ".dbg") or die "Cannot open debug file for write.\n";

        while( <IN> ) {
            $Line = $_ ; chomp($Line);          # Save current line

            # -----Extracting BitScore, Expected Value, Identity, Positives-----
            while ( $Line =~ /^ Score =/ ) {    # Find next set of Query/Sbjct
                $QueryOffset = 0;                # Reset values
                $QuerySeqTemp = "";
                $SubjectSeqTemp = "";
                @LineSplit = split(/ +/, $Line);
                $ScoreBits = $LineSplit[3];
                $Expect = $LineSplit[8];        # Extract Expect value

                if ($Expect =~ /^e/) { $Expect = "1". $Expect }; # If format "e-xxx" add '1' prefix

                $Line = <IN>; chomp($Line);      # Save next line
                @LineSplit = split(/ +/, $Line);
                $IdenPercent = $LineSplit[4];    # Extract Identity%, stripping
                $IdenPercent =~ s/[()%,]//g;     # strip out of "(xxx%)," format
                $IdenPercent = $IdenPercent / 100.0;
                $PositPercent = $LineSplit[8];   # Same with Positive%
                $PositPercent =~ s/[()%,]//g;
                $PositPercent = $PositPercent / 100.0;

                $Line = <IN>;                    # Skip next line (should be blank)
                $Line = <IN>;
            }
        }
    }
}

```

```

# -----Extracting Query Sequences, Subject Sequences, Query Offsets-----
while ( $Line =~ /^Query / ) {      # Find 1st Query line in set
    ($Dummy, $QueryOffDummy, $Line, $Dummy2)
        = split(/ +/, $Line); # Separate into fields

    if ($QueryOffset == 0) {          # Keep 1st $QueryOffset
        $QueryOffset = $QueryOffDummy
    };

    $QuerySeqTemp = $QuerySeqTemp.$Line; # Combine running seq
    $Line = <IN>;                        # Throw away 2nd line
    $Line = <IN>; chomp($Line);          # Save 3rd = Sbjct line
    $Line =~ s/^Sbjct \d+ +//;          # Strip seq prefix
    $Line =~ s/ *\d+.*//;               # Strip suffix
    $SubjectSeqTemp = $SubjectSeqTemp.$Line; # Combine running seq

    $Line = <IN>;                        # Throw away 2nd line
    $Line = <IN>; chomp($Line);          # Next line (another Query?)
};

if ( $QuerySeq eq "" ) {              # Very 1st Query is saved
    $QuerySeq = $QuerySeqTemp;
    $QuerySeq =~ s/X/x/g; # convert special X,N chars to lower case
    $QuerySeq =~ s/-//g; # also, extract insertions ('-')

    $ScoreMin = $ScoreBits*$BitCutOff/100;

    printf OUTD "# === Original Sequence (from 1st match)\n";
    printf OUTD $QuerySeq . "\n\n";
    printf OUTD "MaxScore = ";
    printf OUTD $ScoreBits;
    printf OUTD ", MinScore = ";
    printf OUTD $ScoreMin . "\n\n";

    # Storing Sequences for Entropy calculations
    push(@SeqList, $QuerySeq."\n");
};

# ===== WRITE DEBUG FILE =====
printf OUTD "# === NEW MATCH PAIR  Offset = " . $QueryOffset . "\n";
printf OUTD $QuerySeqTemp . "\n"; # Write out complete seq
printf OUTD $SubjectSeqTemp . "\n\n";

# ===== PROCESS SEQUENCES =====
# Fill out prefix offset
$QuerySeqTemp = substr($QuerySeq, 0, -1+$QueryOffset) . $QuerySeqTemp;
$SubjectSeqTemp = ("-" x (-1+$QueryOffset)) . $SubjectSeqTemp;

printf OUTD "# === Matched pair with prefix & suffix filled\n";
printf OUTD $QuerySeqTemp . "\n"; # Write out complete seq
printf OUTD $SubjectSeqTemp . "\n\n";

# Find & delete insertions
for ($i = -1+length($QuerySeqTemp); $i >= 0; $i += -1) {
    if ( substr($QuerySeqTemp, $i, 1) eq "-" ) {
        substr($QuerySeqTemp, $i, 1) = "";
        substr($SubjectSeqTemp, $i, 1) = "";
    };
};

# Fill out suffix if necessary
$LengthDiff = length($QuerySeq) - length($SubjectSeqTemp);
if ( $LengthDiff > 0 ) {
    $QuerySeqTemp = $QuerySeqTemp . substr($QuerySeq, - $Length);
    $SubjectSeqTemp = $SubjectSeqTemp . ("-" x $LengthDiff);
};

```

```

};

# === QUALIFY THIS SEQUENCE ===
# Not-qualified = lower case (ALTER WHEN NEEDED)
if ($ScoreBits < $ScoreMin) {
    $SubjectSeqTemp = lc($SubjectSeqTemp);
};
# if ($IdenPercent < $IdenPercentMin) {
#     $SubjectSeqTemp = lc($SubjectSeqTemp);
# };

# === WRITE OUT SEQUENCE ===
printf OUTD "# === Matched pair with insertions omitted\n";
printf OUTD $QuerySeqTemp . "\n"; # Write out complete seq
printf OUTD $SubjectSeqTemp . "\n\n";

# === REMOVE UNQUALIFIED SEQS FROM FINAL ENTROPY OUTPUT ===
if ($ScoreBits >= $ScoreMin) {
    push(@SeqList, $SubjectSeqTemp . "\n");
};

};
close(IN);

for ($i = 0; $i <= -2+length( @SeqList[0] ); $i += 1) { # Why is the -2 value there?
    $LineOut = "";
    for ($j = 1; $j <= $#SeqList; $j += 1) {
        $LineOut = $LineOut . substr( @SeqList[$j], $i, 1); # Cycling thru seq j, at pos i
    };

    # ===== COMPUTE ENTROPY =====
    $Line = $LineOut;
    $Line =~ s/[^\A-Z]//g; # Delete anything not capital (IMPORTANT!!!!)
    @LineSplit = split("", $Line);
    %EntropyCount = ();
    $EntropyCountTot = 0;

    foreach $i (@LineSplit) {
        $EntropyCount{$i}++;
        $EntropyCountTot++;
    };

    @AllEntValues = sort(values(%EntropyCount));
    if ($EntropyCountTot >= $HomologMin) {
        $Entropy = 0;
        for ($j = 0; $j <= $#AllEntValues; $j++) {
            $Prob = $AllEntValues[$j] / $EntropyCountTot;
            $Entropy = $Entropy - ($Prob * ( log($Prob)/log(2) ));
        }

        # Debug Code
        printf OUTD "@\n"
        ." Entropy= " . $Entropy
        ." EntropyCount= " . join(" ", %EntropyCount)
        ." EntropyCountTot= " . $EntropyCountTot
        ." AllEntValues= " . join(" ", @AllEntValues)
        . "\n";
    }
    else {
        $Entropy = -1; # Flag as error -- too few homologs
    };

    $LineOut = "E= " . sprintf("%.3f", $Entropy) . " A= " . $LineOut;

    # ===== CREATE OUTPUT LINE PREFIX =====
    $LineHeader = "D " # Start line format eg "D 1agm_001_A"

```

```

        . $PDBName."_"
        . sprintf("%03d", 1+$i)
        . "_" . substr(@SeqList[0], $i, 1);

        printf OUT $LineHeader ." ". $LineOut . "\n";
    };

    close(OUTD);
    close(OUT);
};
};

```

```

#-----
# Author: William Yeh
# Date: Jan 06, 2005
# Purpose: Combining Density files with Entropy/Alignment files.
# 1. Reads entropy/alignment data from alignment files.
# 2. Reads density data from density files. Density = 9 to 12.
# 3. Merge the density data with the entropy and alignment data.
# 4. Give combined file a new filename.
# File: CombDenEnt.pl
#-----

# Modified May 31, 2005
# Check if the residues from entropy data and density data are in consistent
# Typically density files have more lines than entropy files.
# Set constraint so that entropy files determines the maximum number of lines

# LOOPING DENSITY
$ent_cutoff = 80; # Cutoff for Bit-score or Identity
$den_max = 12;
$den_min = 6;
$den_inc = 1;

for ($den_loop=$den_min; $den_loop<=$den_max; $den_loop+=$den_inc) {

    # INITIALIZING VARIABLES
    $den_extract = $den_loop;
    $dir_den = "DensityFiles";
    $dir_ent = "EntropyFiles_". $ent_cutoff; # For Bit-score Cutoffs
    $sdir_ent = "EntropyFiles_". $ent_cutoff."id"; # For Identity Cutoffs
    $sdir_out = "CombineFiles_d". $den_extract;
    $den_CaCount = 0;
    $den_queryRes = "";
    $bool_fndDen = 0;
    $checkline = "";
    $file_ent = "";
    $file_den = "";
    $ct_files = 0;
    $p_id = ""; # protein ID
    $p_size = 0; # loop size for output
    $p_size1 = 0; # size of protein (entropy file)
    $p_size2 = 0; # size of protein (density file)
    @file_array = (); # splitting line to parts
    @file_array1 = (); # splitting line to parts
    @file_array2 = (); # splitting line to parts
    @file_array3 = (); # splitting line to parts
    $density = 0; # packing density of residue
    $original1 = 0; # original residue from entropy file
    $original2 = 0; # original residue from density file
    $position = 0; # residue position
    @array_ent = ();
    @array_den = ();
    $k1 = 0;

```

```

$K2      = 0;
$P       = 0;
$Q       = 0;

# DIRECTORY AND FILE HANDLE NAMES
# (DIR_ENT)
# (DIR_DEN)
# (FILE_CMB)
# (FILE_ENT)
# (FILE_DEN)

#opendir (DIR_OUT, $dir_out);
opendir (DIR_DEN, $dir_den);
opendir (DIR_ENT, $dir_ent);
open (FILE_ERR, ">>error_log.txt");
$ct_files = 1;

# OPEN ENTROPY/ALIGNMENT DIRECTORY
while (defined($file_ent = readdir(DIR_ENT))) {
    if ($file_ent != "." || $file_ent != "..") {

        # OPEN ENTROPY/ALIGNMENT FILES
        open (FILE_ENT, "$dir_ent/$file_ent") or die "Can't open entropy file: $!";

        # GET PROTEIN NAME
        $p_id = substr($file_ent,0,4);
        $file_den = $p_id.".den";

        # WRITE ENTROPY/ALIGNMENT FILE CONTENT TO MEMORY
        while (<FILE_ENT>) {
            chomp;
            $p_size1++;
            $array_ent[$k1++] = $_;
            #@file_array1 = split(/ /,$_);
            #$original = substr($file_array1[1],9,1);
            #$position = substr($file_array[0],5,3);
        }

        # OPEN CORRESPONDING DENSITY FILE
        open (FILE_DEN, "$dir_den/$file_den") or die "Can't open density file $file_den: $!";

        # WRITE DENSITY FILE CONTENT TO MEMORY
        while (<FILE_DEN>) {
            chomp;
            @file_array2 = split(/\(|\)|\=|\s+/, $_);
            $checkline = $file_array2[0];          # check for 'D' as the first char
            if ($checkline eq 'D') {
                $p_size2++;
                $den_queryRes = substr($file_array2[1],9,1);
                for ($p=3;$p<=@file_array2;$p+=5) {
                    if ($file_array2[$p] == $den_extract) {
                        $den_CaCount = $file_array2[$p+3];
                        $bool_fndDen = 1;
                    }
                }
                if ($bool_fndDen == 1) {
                    $array_den[$k2++] = $den_CaCount." ".$den_queryRes."t";
                }
                elsif ($bool_fndDen == 0) {
                    print "density value not found for $file_array[1]\n";
                }
            }
        }

        # PRINT TO OUTPUT FILE
        open (FILE_CMB, ">>$p_id"."c"."$den_extract");
    }
}

```

```

        if ($p_size1 != $p_size2) {
            print FILE_ERR
                "$p_id size inconsistent! file_ent=$p_size1, file_den=$p_size2\n";
            if ($p_size1 > $p_size2) { $p_size = $p_size2; }
            else { $p_size = $p_size1; }
        }
        else {
            $p_size = $p_size1;
        }

        for ($q=0;$q<$p_size;$q++) {
            @file_array1 = split(/ /,$array_ent[$q]);
            $original1 = substr($file_array1[1],9,1);
            $original1 = uc($original1);
            @file_array3 = split(/ /,$array_den[$q]);
            $original2 = substr($file_array3[1],0,1);
            $original2 = uc($original2);

            if ($original1 ne $original2) {
                print FILE_ERR
                    "Inconsistent residues for $p_id at pos $q $original1.$original2.\n";
            }

            print FILE_CMB "$array_den[$q]\t$array_ent[$q]\n";
        }

        # RESET VARIABLES
        $den_CaCount = 0;
        $den_queryRes = "";
        $bool_fndDen = "";
        $checkline = "";
        $file_ent = "";
        $file_den = "";
        $p_id = "";
        $p_size = 0;
        $p_size1 = 0;
        $p_size2 = 0;
        $original1 = "";
        $original2 = "";
        @file_array1 = ();
        @file_array2 = ();
        @file_array3 = ();
        @array_ent = ();
        @array_den = ();
        $k1 = 0;
        $k2 = 0;
        $p = 0;
        $q = 0;

        close (FILE_CMB);
        close (FILE_DEN);
        close (FILE_ENT);
    }

    }

    close (FILE_ERR);
    closedir (DIR_DEN);
    closedir (DIR_ENT);
    print "\nEnd of directory\n";
}

```

```

#-----
# Author: William Yeh
# Date: July 12, 2005
# Purpose: Calculate various data for analysis purposes.

```

```

# 1. Count the number of original residues by density for each protein.
# 2. Count the number of alignments for each protein.
# 3. Calculate the total entropy by density of each protein by density.
# File: ResidueAnalysis.pl
#-----

# Updates:
# 03/28/05      Alter program to count from alignments, the number of -
#               1. strongly hydrophobic residues.
#               2. alanine and glycine.
#               3. gaps.

$min_bit = 20;
$max_bit = 80;
for ($y=$min_bit; $y<=$max_bit; $y+20) {

    $min_rad = 6;
    $max_rad = 12;
    for ($x=$min_rad; $x<=$max_rad; $x++) {

# INITIALIZING VARIABLES
$rad_extract = $x;
$bit_cutoff = 20;
$max_density = 60;
$dir_input = $bit_cutoff."CombinedD".$rad_extract;
$file_output = "check_all_".$bit_cutoff.".txt";
$file_input = "";
$ct_files = 1;
$i = 0;
@qry_by_dens = ();      # number of residues by density (only query residues)
@res_by_dens = ();      # number of residues by density (include alignment)
@gap_by_dens = ();      # gaps or -
@shp_by_dens = ();      # strongly hydrophobic residues contain FILMVWY
@sml_by_dens = ();      # small residues contain Ala, Gly
@snh_by_dens = ();      # small residues contain DEKR
@ent_by_dens = ();      # sum of entropy by density
@file_array = ();
$pid = "";              # protein ID
$den = "";              # packing density of residue
$ori_1 = "";            # original residue from density file
$ori_2 = "";            # original residue from entropy file
$position = "";         # residue position
$entropy = 0;           # residue entropy
$alignment = "";        # alignment at position
$al_length = 0;         # alignment length

# VARIABLES FOR QUERY SEQUENCE
@shp_by_dens_qr = ();    # strongly hydrophobic residues contain FILMVWY
@sml_by_dens_qr = ();    # small residues contain Ala, Gly
@snh_by_dens_qr = ();    # small residues contain DEKR

# VARIABLES FOR EACH PROTEIN
@pid_by_prot = ();       # protein name, PDB id
@qry_by_prot = ();       # number of residues by density (only query residues)
@res_by_prot = ();       # number of residues by density (include alignment)
@gap_by_prot = ();       # gaps or -
@shp_by_prot = ();       # strongly hydrophobic residues contain FILMVWY
@sml_by_prot = ();       # small residues contain Ala, Gly
@snh_by_prot = ();       # small residues contain DEKR
@ent_by_prot = ();       # sum of entropy by density

#-----
# DIRECTORY AND FILE HANDLE NAMES
# (DIR_INPUT)
# (RESULT_ALL)
# (FILE_INPUT)

```

```

# (FILE_OUTPUT)

opendir (DIR_INPUT, $dir_input);
open (RESULT_ALL, ">>$file_output");

print "begin";
# PRINTING TITLE LINES
print RESULT_ALL "Total_residue_number_includes_alignments\n";
print RESULT_ALL "Packing_radius_".$rad_extract."\n";

#-----
while (defined($file_input = readdir(DIR_INPUT))) {
    if ($file_input != "." || $file_input != "..")
    {
        open (FILE_INPUT, "$dir_input/$file_input") or die "Can't open input.txt: $!";

        # GET PROTEIN NAME
        $p_id = substr($file_input,0,4);

        while (<FILE_INPUT>)
        {
            # GET PARAMETERS
            chomp;
            @file_array = split(/s+|_|./,$_);
            $den      = $file_array[0];
            $ori_1    = $file_array[1];
            $ori_2    = $file_array[5];
            $position  = $file_array[4];
            $entropy   = $file_array[7];
            $alignment = $file_array[9];
            $al_length = length($alignment);

            # STORING PARAM ACCORDING TO DENSITIES, USING QUERY RESIDUES ONLY
            $shp_by_dens_qr[$den] += ($ori_2 =~ tr/FILMVWYfilmvwy//);
            $sml_by_dens_qr[$den] += ($ori_2 =~ tr/AGag//);
            $snh_by_dens_qr[$den] += ($ori_2 =~ tr/DEKRdekr//);

            # STORING PARAM ACCORDING TO DENSITIES, USING QUERY AND ALIGNMENTS
            $qry_by_dens[$den] ++;
            $res_by_dens[$den] += $al_length;
            $gap_by_dens[$den] += ($alignment =~ tr/a-zA-Z//c);
            $shp_by_dens[$den] += ($alignment =~ tr/FILMVWYfilmvwy//);
            $sml_by_dens[$den] += ($alignment =~ tr/AGag//);
            $snh_by_dens[$den] += ($alignment =~ tr/DEKRdekr//);
            $ent_by_dens[$den] += $entropy;

            # STORING PARAMETERS ACCORDING TO PROTS, USING QUERY & ALIGNMENTS
            $qry_by_prot[$ct_files] ++;
            $gap_by_prot[$ct_files] += ($alignment =~ tr/a-zA-Z//c);
            $shp_by_prot[$ct_files] += ($alignment =~ tr/FILMVWYfilmvwy//);
            $sml_by_prot[$ct_files] += ($alignment =~ tr/AGag//);
            $snh_by_prot[$ct_files] += ($alignment =~ tr/DEKRdekr//);
            $ent_by_prot[$ct_files] += $entropy;

            $pid_by_prot[$ct_files] = $p_id;
            $res_by_prot[$ct_files] = $al_length;

            # RESET VARIABLES
            @file_array = ();
            $den          = "";
            $ori_1        = "";
            $ori_2        = "";
            $position     = "";
            $entropy      = 0;
            $alignment    = "";
            $al_length    = 0;
        }
    }
}

```



```

    }
    $ct_files ++;
    close (FILE_INPUT);
}

closedir (DIR_INPUT);

### PRINT OUTPUT "DEN\tSHP\tSML\n";
print RESULT_ALL "DEN\tSHP\tSML\n";
for ($i=0; $i<=$max_density; $i++)
{
    print RESULT_ALL "$i\t";
    print RESULT_ALL "$shp_by_dens_qr[$i]\t";
    print RESULT_ALL "$sml_by_dens_qr[$i]\t";
    print RESULT_ALL "\n";
}

### PRINT OUTPUT "\nDEN\tQRY\tTOT\tGAP\tSHP\tSML\tENT\n";
print RESULT_ALL "\nDEN\tQRY\tTOT\tGAP\tSHP\tSML\tENT\n";
for ($i=0; $i<=$max_density; $i++)
{
    print RESULT_ALL "$i\t";
    print RESULT_ALL "$qry_by_dens[$i]\t";
    print RESULT_ALL "$res_by_dens[$i]\t";
    print RESULT_ALL "$gap_by_dens[$i]\t";
    print RESULT_ALL "$shp_by_dens[$i]\t";
    print RESULT_ALL "$sml_by_dens[$i]\t";
    printf RESULT_ALL ("%0.3f\t", $ent_by_dens[$i]);
    print RESULT_ALL "\n";
}

### PRINT OUTPUT "\nPROT\tQRY\tTOT\tGAP\tSHP\tSML\tENT\n";
print RESULT_ALL "\nPROT\tQRY\tTOT\tGAP\tSHP\tSML\tENT\n";
for ($i=1; $i<=$ct_files; $i++)
{
    print RESULT_ALL "$pid_by_prot[$i]\t";
    print RESULT_ALL "$qry_by_prot[$i]\t";
    print RESULT_ALL "$res_by_prot[$i]\t";
    print RESULT_ALL "$gap_by_prot[$i]\t";
    print RESULT_ALL "$shp_by_prot[$i]\t";
    print RESULT_ALL "$sml_by_prot[$i]\t";
    printf RESULT_ALL ("%0.3f\t", $ent_by_prot[$i]);
    print RESULT_ALL "\n";
}

close (RESULT_ALL);

}

}

```

```

#-----
# Author: William Yeh
# Date: August 11, 2005
# Purpose: Calculate hydrophobicity, using scales
# File: Hydrophobicity.pl
#-----

$min_rad = 6;      #<<<<<<----- CHANGE HERE!
$max_rad = 12;     #<<<<<<----- CHANGE HERE!
for ($x=$min_rad; $x<=$max_rad; $x++) {

# INITIALIZING VARIABLES

```

```

$rad_extract = $x;
$bit_cutoff = ""; #<<<<<<----- CHANGE HERE!
$max_density = 60;
$dir_input = $bit_cutoff."CombinedD".$rad_extract;
$file_output = "hpb_result_r".$bit_cutoff.".txt";
$file_input = "";
$ct_files = 1;
$i = 0;

@qry_by_dens = (); # number of residues by density (only query residues)
@res_by_dens = (); # number of residues by density (include alignment)
@shp_by_dens = (); # strongly hydrophobic residues contain FILMVWY
@snh_by_dens = (); # strongly non-hydrophobic residues contain DEKR
@oth_by_dens = (); # other residues
@ent_by_dens = (); # sum of entropy by density

@hopp_woods = (); # hydrophobicity using hopp-woods scales
@engelman_steitz = (); # hydrophobicity using engelman-steitz scales
@sharp_honig = (); # hydrophobicity using sharp-honig scales
$hw = 0;
$es = 0;
$sh = 0;
$arr_len = 0;
$arr_elem = 0;
@arr_align = ();

@file_array = ();
$pid = ""; # protein ID
$den = ""; # packing density of residue
$ori_1 = ""; # original residue from density file
$ori_2 = ""; # original residue from entropy file
$position = ""; # residue position
$entropy = 0; # residue entropy
$alignment = ""; # alignment at position
$al_length = 0; # alignment length

# VARIABLES FOR QUERY SEQUENCE
@shp_by_dens_qr = (); # strongly hydrophobic residues contain FILMVWY
@snh_by_dens_qr = (); # strongly nonhydrophobic residues contain DEKR
@hopp_woods_qr = (); # hydrophobicity using hopp-woods scales
@engelman_steitz_qr = (); # hydrophobicity using engelman-steitz scales
@sharp_honig_qr = (); # hydrophobicity using sharp-honig scales

#-----
# DIRECTORY AND FILE HANDLE NAMES
# (DIR_INPUT)
# (RESULT_ALL)
# (FILE_INPUT)
# (FILE_OUTPUT)

opendir (DIR_INPUT, $dir_input);
open (RESULT_ALL, ">>$file_output");

print "begin";
# PRINTING TITLE LINES
print RESULT_ALL "Total_residue_number_includes_alignments\n";
print RESULT_ALL "Packing_radius_"$rad_extract."\n";

#-----
while (defined($file_input = readdir(DIR_INPUT))) {
    if ($file_input != "." || $file_input != "..")
    {
        open (FILE_INPUT, "$dir_input/$file_input") or die "Can't open input.txt: $!";

        # GET PROTEIN NAME
        $pid = substr($file_input,0,4);
    }
}

```

```

while (<FILE_INPUT>)
{
    # GET PARAMETERS
    chomp;
    @file_array = split(/s+|_|,$_);
    $den      = $file_array[0];
    $ori_1    = $file_array[1];
    $ori_2    = $file_array[5];
    $position  = $file_array[4];
    $entropy   = $file_array[7];
    $alignment = $file_array[9];
    $al_length = length($alignment);

    # Rest - G, A, P, C
    # Polar / hydrophilic - N, Q, S, T, K, R, H, D, E (CY)
    # Non-polar / hydrophobic - V, L, I, Y, F, W, M (AGPC)

    # STORING PARAMS ACCORDING TO DENSITIES, USING QUERY RESIDUES ONLY
    $shp_by_dens_qr[$den] += ($ori_2 =~ tr/FILMVWYfilmvwy//);
    $snh_by_dens_qr[$den] += ($ori_2 =~ tr/NQSTKRHDEnqstkrhde//);
    $oth_by_dens_qr[$den] += ($ori_2 =~ tr/GAPCgapc//);

    # HOPP-WOODS, ENGELMAN-STEITZ, SHARP-HONIG CALCUATIONS
    $hopp_woods_qr[$den] += hopp_woods($ori_2);
    $engelman_steitz_qr[$den] += engelman_steitz($ori_2);
    $sharp_honig_qr[$den] -= sharp_honig($ori_2);

    # STORING PARAMS ACCORDING TO DENSITIES, USING QUERY & ALIGNMENTS
    $qry_by_dens[$den] ++;
    $res_by_dens[$den] += $al_length;
    $ent_by_dens[$den] += $entropy;
    $shp_by_dens[$den] += ($alignment =~ tr/FILMVWYfilmvwy//);
    $snh_by_dens[$den] += ($alignment =~ tr/NQSTKRHDEnqstkrhde//);
    $oth_by_dens[$den] += ($alignment =~ tr/GAPCgapc//);

    # HOPP-WOODS, ENGELMAN-STEITZ, SHARP-HONIG CALCUATIONS
    @arr_align = split(/,$alignment);
    foreach $arr_elem ( @arr_align ) {
        # $hopp_woods[$den] += &hopp_woods($arr_elem);
        # $engelman_steitz[$den] += &engelman_steitz($arr_elem);
        # $sharp_honig[$den] -= &sharp_honig($arr_elem);
        $hw += &hopp_woods($arr_elem);
        $es += &engelman_steitz($arr_elem);
        $sh -= &sharp_honig($arr_elem);
        if ($arr_elem ne '-') {$sarr_len++;}
    }
    $hopp_woods[$den] += $hw/$sarr_len;
    $engelman_steitz[$den] += $es/$sarr_len;
    $sharp_honig[$den] -= $sh/$sarr_len;

    # RESET VARIABLES
    @file_array = ();
    $den      = "";
    $ori_1    = "";
    $ori_2    = "";
    $position  = "";
    $entropy   = 0;
    $alignment = "";
    $al_length = 0;
    $hw       = 0;
    $es       = 0;
    $sh       = 0;
    $sarr_len = 0;
    $sarr_elem = 0;
    @arr_align = ();

```

```

    }
    $ct_files ++;
    close (FILE_INPUT);
}

closedir (DIR_INPUT);

#-----
### PRINT OUTPUT "DEN\tSHP\tSNH\tOTH\tH_WtE_S\tS_H\n";
print RESULT_ALL "DEN\tSHP\tSNH\tOTH\tH_WtE_S\tS_H\n";
for ($i=0; $i<=$max_density; $i++)
{
    print RESULT_ALL "$i\t";
    print RESULT_ALL "$shp_by_dens_qr[$i]\t";
    print RESULT_ALL "$snh_by_dens_qr[$i]\t";
    print RESULT_ALL "$oth_by_dens_qr[$i]\t";
    print RESULT_ALL "$hopp_woods_qr[$i]\t";
    print RESULT_ALL "$engelman_steitz_qr[$i]\t";
    print RESULT_ALL "$sharp_honig_qr[$i]\t";
    print RESULT_ALL "\n";
}

### PRINT OUTPUT "\nDEN\tQRY\tTOT\tSHP\tSNH\tOTH\tH_WtE_S\tS_H\tENT\n";
print RESULT_ALL "\nDEN\tQRY\tTOT\tSHP\tSNH\tOTH\tH_WtE_S\tS_H\tENT\n";
for ($i=0; $i<=$max_density; $i++)
{
    print RESULT_ALL "$i\t";
    print RESULT_ALL "$qry_by_dens[$i]\t";
    print RESULT_ALL "$res_by_dens[$i]\t";
    print RESULT_ALL "$shp_by_dens[$i]\t";
    print RESULT_ALL "$snh_by_dens[$i]\t";
    print RESULT_ALL "$oth_by_dens[$i]\t";
    print RESULT_ALL "$hopp_woods[$i]\t";
    print RESULT_ALL "$engelman_steitz[$i]\t";
    print RESULT_ALL "$sharp_honig[$i]\t";
    printf RESULT_ALL ("%3ft", $ent_by_dens[$i]);
    print RESULT_ALL "\n";
}

close (RESULT_ALL);
}

#-----
##### Subroutine hopp-woods #####
sub hopp_woods {
    if ($_[0] =~ /A/i) {return -0.50;}    if ($_[0] =~ /C/i) {return -1.00;}
    if ($_[0] =~ /D/i) {return 3.00;}    if ($_[0] =~ /E/i) {return 3.00;}
    if ($_[0] =~ /F/i) {return -2.50;}    if ($_[0] =~ /G/i) {return 0.00;}
    if ($_[0] =~ /H/i) {return -0.50;}    if ($_[0] =~ /I/i) {return -1.80;}
    if ($_[0] =~ /K/i) {return 3.00;}    if ($_[0] =~ /L/i) {return -1.80;}
    if ($_[0] =~ /M/i) {return -1.30;}    if ($_[0] =~ /N/i) {return 0.20;}
    if ($_[0] =~ /P/i) {return 0.00;}    if ($_[0] =~ /Q/i) {return 0.20;}
    if ($_[0] =~ /R/i) {return 3.00;}    if ($_[0] =~ /S/i) {return 0.30;}
    if ($_[0] =~ /T/i) {return -0.40;}    if ($_[0] =~ /V/i) {return -0.15;}
    if ($_[0] =~ /W/i) {return -3.40;}    if ($_[0] =~ /Y/i) {return -2.30;}
    if ($_[0] =~ /-/i) {return 0;}
}

##### Subroutine engelman-steitz #####
sub engelman_steitz {
    if ($_[0] =~ /A/i) {return -1.60;}    if ($_[0] =~ /C/i) {return -2.00;}
    if ($_[0] =~ /D/i) {return 9.20;}    if ($_[0] =~ /E/i) {return 8.20;}
    if ($_[0] =~ /F/i) {return -3.70;}    if ($_[0] =~ /G/i) {return -1.00;}
    if ($_[0] =~ /H/i) {return 3.00;}    if ($_[0] =~ /I/i) {return -3.10;}
}

```

```

    if ($_[0] =~ /K/i) {return 8.80;}    if ($_[0] =~ /L/i) {return -2.80;}
    if ($_[0] =~ /M/i) {return -3.40;}    if ($_[0] =~ /N/i) {return 4.80;}
    if ($_[0] =~ /P/i) {return 0.20;}    if ($_[0] =~ /Q/i) {return 4.10;}
    if ($_[0] =~ /R/i) {return 12.30;}    if ($_[0] =~ /S/i) {return -0.60;}
    if ($_[0] =~ /T/i) {return -1.20;}    if ($_[0] =~ /V/i) {return -2.60;}
    if ($_[0] =~ /W/i) {return -1.90;}    if ($_[0] =~ /Y/i) {return 0.70;}
    if ($_[0] =~ /-/i) {return 0;}
}

##### Subroutine sharp-honig #####
sub sharp_honig {
    if ($_[0] =~ /A/i) {return 2.03;}    if ($_[0] =~ /C/i) {return 4.19;}
    if ($_[0] =~ /D/i) {return 1.09;}    if ($_[0] =~ /E/i) {return 1.78;}
    if ($_[0] =~ /F/i) {return 6.03;}    if ($_[0] =~ /G/i) {return 1.01;}
    if ($_[0] =~ /H/i) {return 2.95;}    if ($_[0] =~ /I/i) {return 5.87;}
    if ($_[0] =~ /K/i) {return 2.01;}    if ($_[0] =~ /L/i) {return 5.89;}
    if ($_[0] =~ /M/i) {return 4.80;}    if ($_[0] =~ /N/i) {return 1.40;}
    if ($_[0] =~ /P/i) {return 3.51;}    if ($_[0] =~ /Q/i) {return 2.48;}
    if ($_[0] =~ /R/i) {return 1.78;}    if ($_[0] =~ /S/i) {return 1.59;}
    if ($_[0] =~ /T/i) {return 2.53;}    if ($_[0] =~ /V/i) {return 4.51;}
    if ($_[0] =~ /W/i) {return 7.14;}    if ($_[0] =~ /Y/i) {return 4.90;}
    if ($_[0] =~ /-/i) {return 0;}
}

#-----
# Author: William Yeh
# Date: August 31, 2005
# Purpose: Calculate Statistics of Total Residues.
#           Mean, Median, Standard Deviation.
# File: stats.pl
#-----

$min_bit = 20;
$max_bit = 80;
for ($y=$min_bit; $y<=$max_bit; $y+20) {

    $min_rad = 6;    #<<<<<<----- CHANGE HERE!
    $max_rad = 12;    #<<<<<<----- CHANGE HERE!
    for ($x=$min_rad; $x<=$max_rad; $x++) {

        # INITIALIZING VARIABLES
        $rad_extract = $x;
        $bit_cutoff = $y;    #<<<<<<----- CHANGE HERE!
        $max_density = 60;
        $dir_input = $bit_cutoff."CombinedD".$rad_extract;
        $file_output = "stats_".$bit_cutoff.".txt";
        $file_input = "";
        $ct_files = 1;
        $i = 0;
        @file_array = ();
        $p_id = "";    # protein ID
        $den = "";    # packing density of residue
        $ori_1 = "";    # original residue from density file
        $ori_2 = "";    # original residue from entropy file
        $position = "";    # residue position
        $entropy = 0;    # residue entropy
        $alignment = "";    # alignment at position
        $al_length = 0;    # alignment length
        $mean_value = 0;    # mean value of density
        $stdv_value = 0;    # standard deviation value of density
        $medi_value = 0;    # median value

        # ARRAY: FOR STORING ALL DENSITY VALUES, ACCORDING TO PACKING RADIUS
        @dens_values = ();    # array that stores all the density values
    }
}

```

```

#-----
# DIRECTORY AND FILE HANDLE NAMES
# (DIR_INPUT)
# (RESULT_ALL)
# (FILE_INPUT)
# (FILE_OUTPUT)

opendir (DIR_INPUT, $dir_input);
open (RESULT_ALL, ">>$file_output");

print "begin";
# PRINTING TITLE LINES
print RESULT_ALL "Total_residue_number_includes_alignments\n";
print RESULT_ALL "Packing_radius_".$rad_extract." Bitscore_cutoff_". $bit_cutoff." \n";

#-----
while (defined($file_input = readdir(DIR_INPUT))) {
    if ($file_input != "." || $file_input != "..")
    {
        open (FILE_INPUT, "$dir_input/$file_input") or die "Can't open input.txt: $!";

        # GET PROTEIN NAME
        $p_id = substr($file_input,0,4);

        while (<FILE_INPUT>)
        {
            # GET PARAMETERS
            chomp;
            @file_array = split(/s+|_|,$_);
            $den = $file_array[0];
            $ori_1 = $file_array[1];
            $ori_2 = $file_array[5];
            $position = $file_array[4];
            $entropy = $file_array[7];
            $alignment = $file_array[9];
            $al_length = length($alignment);

            # STORING DENSITY VALUES
            push (@dens_values, $den);

            # RESET VARIABLES
            @file_array = ();
            $den = "";
            $ori_1 = "";
            $ori_2 = "";
            $position = "";
            $entropy = 0;
            $alignment = "";
            $al_length = 0;
        }
        $ct_files ++;
        close (FILE_INPUT);
    }
}
closedir (DIR_INPUT);

#-----
### CALCULATE MEAN AND STANDARD DEVIATION
$mean_value = Calculate_Mean(@dens_values);
$stdv_value = Calculate_StdDev(@dens_values);
$medi_value = Calculate_Median(@dens_values);

#-----
### PRINT OUTPUT "DEN\tSHP\tSML\n";
print RESULT_ALL "Mean: $mean_value\n";

```

```

print RESULT_ALL "Standard deviation: $stdv_value\n";
print RESULT_ALL "Median: $medi_value\n";
print RESULT_ALL "\n";

close (RESULT_ALL);
}
}
#-----

##### Subroutine Calculate_Mean #####
# Given an array of values, calculate and return the mean.

sub Calculate_Mean {
    my( $mean ); # the mean of the values, calculated by this subroutine
    my( $n );    # the number of values
    my( $sum );  # the sum of the values
    my( $x );    # set to each value

    # Initialize variables.
    $n = 0;
    $sum = 0;

    foreach $x ( @_ ) {
        $sum += $x; # Add each value to the growing sum.
        $n++;      # Increment the count of the number of values.
    }

    if ( $n != 0 ) { $mean = $sum/$n; } # Calculate the mean.
    return $mean; # Return the value of the mean.
}

##### Subroutine Calculate_StdDev #####
# Given an array of values, calculate & return the standard deviation.
# variance = (sumSq-((sum * sum)/n))/(n-1)
# std.dev. = sqrt(variance)
# sumSq is the sum of the squares of the values

sub Calculate_StdDev {
    my( $n );      # the number of values
    my( $stddev ); # the standard deviation,
    my( $sum );    # the sum of the values
    my( $sumSq );  # the sum of the squares of the values
    my( $variance ); # the variance, calculated by this subroutine
    my( $x );      # set to each value

    # Initialize variables.
    $n = 0;
    $sum = 0;
    $sumOfSquares = 0;

    foreach $x ( @_ ) {
        $sum += $x; # Add the value to the growing sum.
        $n++;      # Increment $n, the number of values.
        $sumSq += $x * $x; # Add the square of the value to the growing sum of the squares.
    }

    $variance = ( $sumSq - ( ( $sum * $sum ) / $n ) ) / ( $n - 1 );
    $stddev = sqrt( $variance );
    return $stddev;
}

##### Subroutine Calculate_Median #####
# Given an array of values, calculate & return the Median.

sub Calculate_Median {
    my( @sorted );

```

```

my( $median );
my( $a_size );

# Initialize variables.
@sorted = {};
$median = 0;
@sorted = sort { $a <=> $b } @_;
$a_size = scalar(@sorted);

if ($a_size%2 == 1) { # ODD
    $median = $sorted[(($#sorted)/2)];
    return $median;
}
else { # EVEN
    $median = $sorted[int(($#sorted/2)]
        + (($sorted[int(($#sorted/2)+1] - $sorted[int(($#sorted/2)])]/2);
    return $median;
}
}

```


Appendix B

Fraction of non-strongly hydrophobic residues versus $1/\text{density}$, using bit score 40.

Appendix B1

Fraction of non-strongly hydrophobic residues versus $1/\text{density}$, for radius 9 to 12 Å.

Appendix B2

Fraction of non-strongly hydrophobic residues versus $1/\text{density}$, for radius 6 to 9 Å.

Appendix B3

Fraction of non-strongly hydrophobic residues and average sequence entropy versus $1/\text{density}$, for radius 6 to 12, using bit score 40.

Appendix C

Average Sequence Entropy using different Bit Score and Percentage Identity Cutoffs,
with a packing radius of 9 Å.

Appendix D

Fraction of small residues and average sequence entropy versus 1/density, for radius 6 to 12, using bit score 40.

Appendix E

Fraction of gaps and average sequence entropy versus $1/\text{density}$, for radius 6 to 12, using bit score 40.