Fall 2012

Utilization of Protein Tertiary Contacts to Improve Protein Structure Prediction Using Sequence Homology

Trung Thanh Nguyen
San Jose State University

UTILIZATION OF PROTEIN TERTIARY CONTACTS TO IMPROVE PROTEIN STRUCTURE PREDICTION USING SEQUENCE HOMOLOGY

A Thesis

Presented to

The Faculty of the Department of Chemistry

San Jose State University

In Partial Fulfillment

of the Requirements for the Degree

Masters of Science

by

Trung Thanh Nguyen

December 2012

The Designated Thesis Committee Approves the Thesis Titled

UTILIZATION OF PROTEIN TERTIARY CONTACTS TO IMPROVE PROTEIN STRUCTURE PREDICTION USING SEQUENCE HOMOLOGY

by

Trung Thanh Nguyen

APPROVED FOR THE DEPARTMENT OF CHEMISTRY

SAN JOSE STATE UNIVERSITY

December 2012

Dr. Brooke Lustig	Department of Chemistry
Dr. Elaine D. Collins	Department of Chemistry
Dr. Roger H. Terrill	Department of Chemistry

ABSTRACT

UTILIZATION OF PROTEIN TERTIARY CONTACTS TO IMPROVE PROTEIN STRUCTURE PREDICTION USING SEQUENCE HOMOLOGY

by Trung Thanh Nguyen

The structure of a protein ultimately determines its function; therefore, knowledge of three-dimensional structure is essential for understanding its function and mechanism of action. The two most common methods for determining protein structure are x-ray crystallography and Nuclear Magnetic Resonance (NMR) spectroscopy. These methods are quite successful but can be very time-intensive and costly. An alternative method is protein structure prediction, where structure is computationally predicted from amino acid sequence. As opposed to x-ray crystallography and NMR spectroscopy, protein structure prediction is not encumbered by potential experimental problems. In this research, we attempted to determine if certain protein structure features, known as tertiary contacts, can improve the prediction of protein three-dimensional structure. By calculating and analyzing sequence homology and related values, it was shown that tertiary contacts, which typically are long-range amino acid interactions separated by at least 10 amino acids in sequence length, generally have lower pair averaged sequence homology-based values. From our calculations we were able to create a prediction filter based on our known literature-derived tertiary contacts of whether amino acid residues are buried or on the surface of a protein. From our tertiary contact prediction filter, it was shown that approximately 80% of the amino acid residues in our protein learning set were correctly filtered to be on the surface of a protein. These results imply that tertiary contacts are more conserved, densely packed, and less likely to be on the surface of a protein. From the tertiary contact prediction filter, we hope that tertiary contacts can be utilized in conjunction with other prediction approaches to more accurately predict where amino acids may be located in a protein.

ACKNOWLEDGEMENTS

There are three main people whom I would like to offer my gratitude for helping me throughout my tenure at the Department of Chemistry at San Jose State University. First and foremost, I would like to thank Dr. Brooke Lustig, my research advisor, for mentoring me throughout my research endeavors and the writing of my Masters thesis. He was able to provide me with his patience, knowledge, and expertise to guide me through this experience. Most importantly, he has given me an invaluable tool of becoming a more scientifically minded person, which will help me throughout my life. I would also like to thank my committee members, Dr. Roger H. Terrill and Dr. Elaine D. Collins, for taking the time to review my thesis. I especially would like to thank Dr. Roger H. Terrill for giving me motivation and encouragement throughout this process. Lastly, I would like to thank my fellow peers of the SJSU Department of Chemistry graduate programs. I thank everyone for their support, encouragement, and great personalities and for making this experience very positive.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Importance of Proteins

Proteins are one of the most important biological macromolecules found in nature. They are an essential player in almost all biological processes for living organisms and participate in virtually every process within the cell. Many proteins are enzymes that play roles in catalyzing biochemical reactions and are vital to metabolism. Proteins also have structural and mechanical functions, such as actin and myosin in muscle and the proteins in the cytoskeleton that support the structure of the cell. Proteins are also important in cell signaling, immune responses, the cell cycle, oxygen transport, and in the maintenance of chemical potential across cell membranes (Moret and Zebende, 2007). Proteins are essential to so many biological processes, it is necessary to understand the nature of how they work.

The way amino acids interact with one another determines the structure of a protein. A protein's amino acid sequence is called its primary structure. Proteins can also be classified by secondary structure, tertiary structure, and quaternary structure. Secondary structure refers to the sub-structures that are formed by local inter-residue interactions of the amino acids with one another. The most common secondary structures are alpha helices and beta sheets. Tertiary structure is the complete three-dimensional

structure of a single protein molecule, as defined by the atomic coordinates. Quaternary structure is an assembly of large polypeptide subunits. The structure of a protein ultimately determines the protein's function; therefore, knowledge of even basic elements of three-dimensional structure is useful for understanding a protein's function and mechanism of action (Richardson and Barlow, 1999).

1.2 Protein Structure Determination

Two of the most common methods in determining the structure of a protein are x-ray crystallography and Nuclear Magnetic Resonance (NMR) spectroscopy, both of which can provide information at the atomic level. In x-ray crystallography, x-rays are beamed at protein crystals causing the beam of light to diffract in many specific directions. With the resulting angles and intensities of the diffracted beams, a three-dimensional picture of the density of electrons from within the crystals can be produced and interpreted. X-ray crystallography is the gold standard for determining the structure of proteins. Today, x-ray crystallography is routinely utilized by many researchers and scientists to determine how drugs interact with certain parts of a protein's structure.

NMR spectroscopy is another very popular method for determining protein structure, second only to x-ray crystallography. In NMR spectroscopy, a protein sample is placed in a magnetic field where certain intrinsic magnetic properties are utilized to determine physical and chemical properties of the protein in question. NMR

spectroscopy is often useful in the direct characterization of particular non-covalent bond interactions.

Although x-ray crystallography and NMR spectroscopy have been very successful in determining protein structure, these methods have limitations and downfalls. One of the biggest problems in x-ray crystallography is obtaining protein crystals, which are due to inherent limitations and difficulties (Dale et al., 2003). For example, membrane proteins are very difficult to crystallize because they must be isolated first. The isolation process can interfere with crystallization. The major limitation in NMR spectroscopy is that it is typically restricted to smaller proteins (Romero et al., 2001). Lastly, both of these methods are generally very slow and costly, often taking many months or years of laboratory work to complete (Montelione and Anderson, 1999).

1.3 Protein Structure Prediction

An alternative method to x-ray crystallography and NMR spectroscopy in determining protein structure is to predict the structure from the amino acid sequence. In protein structure prediction, a protein's three-dimensional structure, the secondary, tertiary, and even possibly quaternary structure, are predicted solely from its primary structure.

Protein structure prediction is mainly performed computationally using large databases and libraries. The databases and libraries contain vast amounts of information pertaining to nucleotide sequences, protein sequences, protein sequence patterns or

motifs, macromolecular three-dimensional structure, gene expression data, metabolic pathways, and more. A main goal of bioinformatics, the study of biology with the utilization of the techniques of computer science and information technology, is the prediction of protein structure.

There are many methods that are used to predict protein structure, and these methods are often classified by the type of structure that is being predicted: the secondary, tertiary, or quaternary structure of the protein. In secondary structure prediction, certain regions of the protein that facilitate and stabilize tertiary structure can be predicted. These regions include alpha-helices, beta-sheets, and turns/loops as well as solvent accessible regions, tertiary contacts, and other interactions that occur between the amino acids. Some secondary structure prediction methods are machine learning methods that include Neural Networks and Support Vector Machines (Wagner et al., 2005). In tertiary prediction methods, a protein's three-dimensional structure is predicted by *ab initio* modeling or comparative protein modeling. *Ab initio* modeling predicts the protein's structure from the sequence alone, without relying on the similarity of known structures, while comparative protein modeling or threading relies on prior knowledge of similarity among sequence and known structures (Bonneau and Baker, 2001). Quaternary protein structure prediction attempts to recognize and catalog physical interactions between pairs or groups of proteins, which can be used to understand intracellular signaling and other biochemical processes.

There are many benefits from using protein structure prediction methods. As opposed to x-ray crystallography and NMR spectroscopy, protein structure prediction

more quickly acquires results and is relatively cost efficient. Protein structure prediction is one the most important goals pursued in bioinformatics research and is highly important in medicine and biotechnology.

1.4 Present Research

In this thesis, the sequence entropy, packing density, and relative solvent accessibility (RSA) were utilized to gain insights into a certain protein structural feature, hereinafter referred to as a *tertiary contact*, and to determine if these tertiary contacts can serve in characterizing protein structure, especially in terms of whether amino acids are buried or on the surface of a protein. Most importantly, the methods section will contain a meticulous and detailed walkthrough of how to apply the protein sets to the computational programs. Lastly, the future studies section will propose some suggestions on continuing this research such as expanding the protein set or modifying the computer programs.

Chapter 2

Background

2.1 Bioinformatics

In its most basic of definitions, bioinformatics is the study of biology using computers as a tool. The majority of the biology studied in bioinformatics deals with the sequences of nucleotides and proteins and also protein structure and characterization of protein domains. The computational tools and techniques utilized to study bioinformatics include algorithms, databases and information systems, web technologies, artificial intelligence, information and computation theory, software engineering, data mining, and modeling and simulation. Since there are many other ways to study biology using computers that are not bioinformatics related (e.g. three-dimensional protein structure determinations from x-ray crystallographic data relying on computer analysis predates the field of bioinformatics), a more specific way to define bioinformatics is the application of computational tools and techniques to the management and analysis of biological data (Tisdal, 2001).

2.2 Protein Sequences and Databases

Much of the biological data bioinformatics research involves, as stated before, are nucleotide and protein sequences. For this research we are looking at protein sequences. In proteins, there are 20 standard amino acids. These twenty amino acids are encoded by the universal genetic code and each amino acid is designated a name that suits the amino acid depending on its individual chemical composition and a 3-letter and 1-letter abbreviation chosen by the IUPAC committee. In bioinformatics research, the 1-letter abbreviations are typically used to identify a protein's amino acid sequence.

In 1991, there were about 12,000 proteins of known sequences totaling more then three million amino acid residues (Sander and Schneider, 1991). As of 2012, there are almost 80,000 known protein primary sequences, and the number will constantly be increasing (Rutgers, the State University of New Jersey and San Diego Supercomputer Center (SDSC) and Skaggs School of Pharmacy and Pharmaceutical Sciences, 2012).

With so many known protein sequences, the next logical step is the creation of protein sequence databases and libraries. The objective of a database is to search and distinguish sequences (nucleotide or protein) related to the query by some model (e.g., evolution) from unrelated sequences (Nicholas et al., 2000). One of the more popular portals for bioinformatics databases, and also one used for this research, is provided by NCBI (National Center for Biotechnology Information), a government-funded website and a branch of the National Institute of Health (NIH). NCBI houses genome sequencing data in GenBank, biomedical research articles in PubMed as well as other information

relevant to bioinformatics. The NCBI website also houses BLAST, or Basic Local Alignment Search Tool, which is an algorithm for comparing nucleotide or amino acid sequences and will be discussed later in detail.

Another popular bioinformatics database is the Protein Data Bank or PDB. The PDB mainly contains data on proteins such as three-dimensional structural data elucidated from x-ray diffraction, x-ray crystallography, NMR spectroscopy, electron microscopy, and other methods. The PDB also contains primary protein sequences, and most importantly the information is contained in files that can be downloaded and are of standard use in bioinformatics research. These files include: FASTA Sequence (.txt) which is a text based format for representing nucleotide and protein sequences, PDB File (.pdb) which contain textual file information on three-dimensional structures of molecules held in the Protein Data Bank, and mmcif File (.cif) which involves standard text and contains crystallographic information. For this research, FASTA Sequence, PDB File, and mmcif File are used. On a last note, the proteins that are stored in the PDB each have their own unique .pdb file classifier or PDB ID. The classifier is usually four characters as numbers and letters, where the last character is typically a protein chain identifier.

2.3 Sequence Homology and Sequence Alignment

The advent of bioinformatics databases has opened up the door for many opportunities in protein research. Major research areas include sequence analysis,

computational evolutionary biology, and protein structure prediction. One main feature that these research areas have in common is that they involve comparing protein sequences. Comparing protein sequences is the essence of sequence homology.

Sequence homology is when sequences, protein or nucleotide, have some kind of relatedness or similarity through a common evolutionary ancestor (Nicholas et al., 2000). Being similar through a common evolutionary ancestor can mean that sequences that are homologous to one another can have the same structure and/or the same function. This knowledge of homologous sequences is quite valuable seeing that structure is usually a precursor to function (Rodionov and Blundell, 1998). If one were to expand on this presumption, then proteins with unknown sequence, structure, and function can be predicted from proteins with known sequence, structure, and function. Sequence homology has been able to accurately predict the structures of thousands of proteins (Rost and Sander, 1994), and today it is one of the most accurate methods and is utilized in many aspects of protein structure prediction (Chen et al., 2004).

To determine whether two or more proteins are homologous to one another the usual technique that is employed is sequence alignment. Sequence alignment is a way of arranging protein sequences to identify regions of similarity that may be a consequence of functional, structural, or evolutionary relationships between the sequences. Sequences are usually written in rows arranged so that aligned residues appear in successive columns. Gaps, represented by dashes (-), are inserted into the sequence so that similar patterns within the aligned sequences can be matched together under the same column (DeSantis et al., 2006).

There are two main computational methods in which sequences are aligned: global alignment and local alignment. Global alignment attempts to align sequences by including the entire length of all sequences and is most useful when the sequences are similar and of equal size (Nicholas et al., 2000). Local alignment on the other hand takes the entire length of the sequences but only aligns and identifies certain regions that are suspected to contain some sort of homology. Local alignment techniques are often preferable and are believed to be a better choice then global alignment. For this research, the local alignment algorithm that is used is BLAST.

Employing sequence alignment has proven to be among the most successful computational methodologies for protein structure prediction (Bramucci et al., 2012). Evolutionary information derived from sequence alignment has been shown to improve significantly the accuracy of secondary structure prediction (Adamczak et al., 2005). It also is generally accepted that the utilization of sequence alignment brings about a gain in protein secondary structure prediction accuracy of 6-8% (Frishman and Argos, 1997). Much research has shown that sequence alignment is a very valuable tool for protein structure prediction.

2.4 Basic Local Alignment Search Tool or BLAST

BLAST or Basic Local Alignment Search Tool is a computer algorithm that is used to compare and align biological sequence information such as protein amino acid sequences or DNA/RNA nucleotide sequences. To run the BLAST program, a user

inputs a sequence of interest, or a query sequence, and BLAST then searches a database of known sequences to align and compare with the query sequence. The method that BLAST uses to align sequences is referred to as a heuristic method meaning that BLAST aligns sequences similar to techniques based on experience, rule of thumb, or trial and error. For example, BLAST finds homologous sequences by locating short matches between two sequences. Then, if BLAST finds matches, it begins to make local alignments.

The heuristic method is opposed to extensive methods such as the Needleman-Wunsch (Needleman and Wunsch, 1970) and Smith-Waterman (Smith and Waterman, 1981) sequence alignment algorithms. The Needleman-Wunsch algorithm is a global alignment algorithm while the Smith-Waterman algorithm is a local alignment algorithm. Both of these algorithms differ from BLAST in that the methods they use to align sequences are based on dynamic programming. The concepts of dynamic programming are outside the scope of this thesis but what is known is that dynamic programming is mathematically rigorous and computationally demanding (Nicholas et al., 2000). Also, the computational complexity of dynamic programming can be impractical if many or long sequences are involved. Although BLAST's heuristic method of sequence alignment is not as accurate and comprehensive as the dynamic programming methods, BLAST is faster, more efficient, and less computationally intensive.

Another thing about BLAST is that it is based on the Smith-Waterman algorithm and as stated earlier the Smith-Waterman algorithm is a local alignment algorithm, similar to BLAST, but uses dynamic programming instead of heuristics. It has also been

shown that the Smith-Waterman local alignment algorithm is the most effective of the database searching dynamic programming algorithms for finding similar sequences (Nicholas et al., 2000). BLAST being based on the Smith-Waterman algorithm and using a heuristic method for aligning sequences, has the best of both worlds as a sequence alignment algorithm by being as effective as the Smith-Waterman algorithm but also being faster and more efficient by using a heuristic method. Lastly, there are a couple of different BLAST programs depending on the type of biological sequence data being compared. For this research the BLAST program being used is protein-protein BLAST or blastp. Note blastp, when given a protein query, returns the most similar protein sequences from the protein database that the user specifies.

2.5 Sequence Entropy

The aligned protein sequences that blastp generates is the raw data that is used for this research. Within the aligned protein sequences is a wealth of information that can be manipulated and deciphered to gain knowledge about protein secondary, tertiary, and quaternary structure, protein amino acid evolution, and selective amino acid mutation. One of the first applications applied to the data of aligned protein sequences is the calculation of sequence entropy.

Sequence entropy, in terms of protein sequences, is the degree of conservation (or variability) of each amino acid residue at that single position in the whole primary protein sequence across all homologous proteins (Gerstein and Altman, 1995). In our case, ‘all

homologous proteins' are all the sequences that are aligned to the protein in question. Sequence entropy gives a quantitative measure of how each amino acid can change or stay conserved within a protein sequence giving insights to how important said amino acid is to overall protein structure.

Here, sequence entropy is based on a concept of information theory called Shannon entropy. Shannon entropy is an often-used measure of diversity (Valdar, 2002) and in terms of a protein sequence is a measure of variability at a particular amino acid position. Shannon entropy is mathematically defined as:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_b p(x_i)$$

where *p* denotes the probability mass function of *X*, and *b* is the base of the logarithm used. When rewritten to relate to sequence entropy, the equation is defined as:

$$S_k = -\sum_{j=1,20} P_{jk} * \log_2 P_{jk}$$

where the probability *Pjk* at any sequence position *k* is obtained from the frequency of an amino acid type *j* at sequence position *k* for all the aligned residues.

The application of sequence entropy has been very successful in many other types of research. The calculation of sequence entropy has been found to provide a better contrast between whether amino acid residues are in the core or the rim of a protein (Elcock and McCammon, 2001; Liao et al., 2005). Sequence entropy has also been said

to be a sophisticated, intuitive, and statistical measure that accords well with the estimated sequence variability (Rodionov and Blundell, 1998). Other researchers have also employed sequence entropy algorithms for protein secondary structure prediction (Shenkin et al., 1991). It is clear that continued exploration of sequence entropy is critical to a better understanding of protein sequence and structure.

2.6 Packing Density

Another application that can be applied to the aligned protein sequences is the calculation of amino acid packing density. Packing density is the ratio of the summed atom volumes to the molecular volume (Rose, 1993) and is defined as the amount of space that is occupied within the van der Waals envelope of the molecule divided by the total volume of space that contains the molecule (Liang and Dill, 2001). Packing density determines how closely a single amino acid is surrounded by its neighboring amino acids.

To measure the packing density for a single amino acid residue, the alpha carbon (C-alpha) of said amino acid is designated as the center of a spherical volume. The distance between the center C-alpha and other adjacent and neighboring C-alpha's of other amino acids is calculated. This calculation is performed within a chosen radius of the sphere. The distance between any two C-alpha amino acids is given by the equation:

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

where dist(i, j) is the distance between C-alpha i and C-alpha j and x, y, and z are the C-alpha coordinates at the position. The number of C-alpha's within the sphere of a chosen radius is the packing density of the center C-alpha. The packing density of a protein core described in terms of atom packing has been investigated as a criterion for amino acid residue substitution and conservation (Rodionov and Blundell, 1998). The high packing densities seen in globular proteins have been interpreted to mean that protein conformation is linked to internal packing (Rose, 1993; Ting and Jernigan, 2002).

2.7 Relative Surface Accessibility

The last calculation that is applied to the aligned protein sequences is the calculation of Relative Surface Accessibility or RSA. RSA of an amino acid residue indicates its level of burial (or exposure) in a protein core or surface (Adamczak et al., 2004). RSA is a key property of amino acid residues (Ahmad et al., 2003), and the prediction of amino acid RSA helps us to understand the three-dimensional structure and function of proteins (Wang et al., 2007).

For this research, amino acid RSA is calculated by using a program called NACCESS (http://www.bioinf.manchester.ac.uk/naccess/). NACCESS is a program that calculates the atomic and residue accessibilities for both proteins and nucleic acids from a PDB file format. NACCESS calculates the accessible surface for each atom and also provides an average surface accessibility value per amino acid residue by rolling a probe of a given size around a protein surface (Hubbard and Thornton, 1993). The utilization

of NACCESS is quite beneficial in the calculation of RSA and can lead to numerous insights into protein structure.

RSA of an amino acid residue can be used as an effective local fingerprint of the overall topology and packing of a protein, allowing the improvement of protein secondary structure prediction (Adamczak et al., 2005). RSA of amino acid residues can also improve the accuracy of predicting three-dimensional structures of proteins, especially ones without homology to other protein structures (Ahmad et al., 2003). In general, the prediction of RSA can aid in the elucidating the relationship between amino acid sequence and protein structure (Naderi-Manesh et al., 2001).

2.8 Protein Tertiary Contacts

Here we are specifically studying and investigating whether protein tertiary contacts can aid in the prediction of protein secondary and tertiary structure. A protein tertiary contact is defined as a pair of amino acid long range interactions (Kallblad and Dean, 2004) that are separated by at least 10 residues in the protein primary sequence with at least one of their atomic distances less than the sum of the van der Waals radii of the two atoms plus 1.0 Å (Kim and Park, 2004).

From much literature research it has been shown that the interactions between tertiary contacts and secondary structure types (alpha helices and beta sheets) in proteins have valuable implications for the prediction of three-dimensional structure or tertiary structure (Kallblad and Dean, 2004). In general, secondary structure is inherently

unstable and its stability is enhanced by tertiary interactions (Daggett and Fersht, 2003). Tertiary contacts are very important in the stabilization and prediction of protein structure.

It has also been shown that studying tertiary contacts in the context of sequence entropy, packing density, and RSA can be useful in protein structure prediction. Pairs of residues associated with tertiary contacts show a tendency to be better conserved and more densely packed then regular protein residues (Do, S.; Lustig, B. San Jose State University. Unpublished work, 2010). RSA is related to tertiary contact interactions between residues that are far apart in sequence but close in three-dimensional space (Kim and Park, 2004), and once RSA is understood protein tertiary contacts can be predicted with much ease.

For our research, we performed sequence entropy, packing density, and RSA calculations on tertiary contacts that were found for a subset of 75 proteins taken from the 268 protein set of Lustig and coworkers (Liao et al., 2005; Mishra, 2010). After the values for the tertiary contacts were accumulated, further analysis was performed specific to tertiary contact data and will be discussed later. From the analysis of the tertiary contact values one may be able to acquire insights into the nature of protein amino acid residues as pertaining to protein structure prediction, especially whether said residues are buried or on the surface of a protein.

Chapter 3

Methods

3.1 268 Protein Set

The protein data set used for this research is a combination of two other protein data sets from Lustig and coworkers (Liao et al., 2005; Mishra 2010). From the combination of these two protein data sets a total of 268 proteins were chosen; only proteins with structure determined by x-ray crystallography methods were selected. The 268 protein set was culled for PDB chain identifiers that had a sequence percentage identity of $\leq$ 25%, a structural resolution between 0.0 – 2.5 Å, an R-factor of $\leq$ 0.3, and a sequence length between 40 and 10,000. The culling process was performed by a protein sequence culling server called PISCES, which can cull a list of user-provided PDB chain identifiers according to user-input criteria such as sequence identity and other structural qualities (Wang and Dunbrack, 2003). From culling the combination of both protein sets, we ended up with a diverse protein set of 268 proteins consisting of monomeric, homodimeric, and heterodimeric proteins.

3.2 Tertiary Contacts Protein Set

The 268 protein set was further selected to include only proteins that had tertiary contacts. An extensive literature search was performed to find tertiary contacts for each protein of the 268 protein set. From a literature search (Tu, V. T.; Le, T.; Arora, S.; Lustig, B. San Jose State University. Unpublished work, 2010), the amino acid residue, amino acid primary structure sequence position, and type of tertiary contacts were documented. The major types of tertiary contacts that were found were hydrogen bonds, hydrophobic interactions, ionic interactions, polar interactions, salt bridges, and disulfide bonds. Out of the 268 proteins it was found that 102 proteins have tertiary contacts (Table A.1 in Appendix A). From this set of 102 proteins we excluded tertiary contacts that were hydrophobic interactions due to the intrinsic difficulties hydrophobic interactions present for the prediction of whether an amino acid residue is buried or on the surface of a protein (Do, S.; Mishra, R.; Lakkaraju, H.; Dee, J.; Kantardjieff, K.; Lustig, B. San Jose State University. Unpublished work, 2010). This led us to a final total of 75 proteins that had tertiary contacts that were acceptable for further analysis (Table A.2 in Appendix A).

3.3 PDB Files

For each of the 75 proteins, we downloaded files necessary for our research from the RCSB PDB website, http://www.rcsb.org/pdb/home/home.do. These files are the

FASTA Sequence and the mmCIF File. FASTA Sequence has the filename extension of .txt and is a text file that contains a protein's amino acid sequence. The mmCIF File has the filename extension of .cif and contains atom coordinate data for the entire protein. The atom coordinate data is used to calculate the packing density of each amino acid of a protein. These files were acquired by individually entering each of the 75 proteins PDB ID into the search bar.

3.4 Running BLAST (blastp)

To perform a sequence alignment search using blastp, the amino acid sequence that was downloaded from the FASTA Sequence file was copied into the blastp search box (National Center for Biotechnology Information, 2012). Default settings were used except for "Max target sequences," which had to be changed from 100 to 10000. The blastp algorithm was then run with the amino acid sequence from the FASTA Sequence file as the query sequence. The results were saved into a plain text format with the filename extension of .txt, by noting "Formatting Options" of "Plain text," and "Reformat." The blastp .txt results were downloaded and ready for input in the Perl programs. Note PSI-BLAST (Rose et al., 2011) multiple sequence alignments (Lau, R.; Lustig, B. San Jose State University. Unpublished work, 2010) have shown less utility then blastp in the calculation of sequence homology-based parameters.

3.5 Extracting Sequence Homology-Based Data Using Perl

One of main issues when using Perl for bioinformatics research is the lack of precise and detailed documentation of the applications of the Perl scripts. Here provided is an the computational protocol to run the Perl scripts that extract protein sequence homology-based data.

First, here is a list of the Perl scripts (Mishra, 2010) in order of usage:

1. bst2entMOD2.pl or Radhika-6pointentropy.pl
2. extract_fractanalysis_entropy_aggr.pl
3. ftp-script-1.pl
4. cif2den.pl and Chainselectivecif2den.pl
5. extract_individualfractentropy_density_aggr.pl
6. calculate_aggr_per_protein.pl
7. double_agg_forPlot.pl
8. listNoAlignments.pl
9. No_of_res_count.pl
10. Bitscorelistno_ofsubject.pl

A full list of the Perl scripts used for this research is in Appendix B.

3.5.1 bst2entMOD2.pl

bst2entMOD2.pl is a Perl script that was created by in 2002 and modified in 2008 to include the chain name of the protein and a variable bitscore cutoff. *bst2entMOD2.pl* is a program that takes the blastp .txt results and outputs sequence entropy values from these .txt files. To run *bst2entMOD2.pl* natively the blastp .txt input files must be in a directory named "nblast_all". Once all files and directories are in place, *bst2entMOD2.pl* is run and the sequence entropy values for each query protein were outputted in a file for each protein blastp .txt file. All sequence entropy files were then be placed in a user defined directory.

3.5.2 extract_fractanalysis_entropy_aggr.pl

Once *bst2entMOD2.pl* outputs the sequence entropy files (.ent), *extract_fractanalysis_entropy_aggr.pl* takes the sequence entropy files and outputs files containing other sequence homology-based values for other homology-based parameters such as fraction of residues that are strongly hydrophobic, fraction of residues that are small, fraction of residues that are non-strongly hydrophobic, fraction of residues that are gaps, and also the sequence entropy values of each protein. In our case, *extract_fractanalysis_entropy_aggr.pl* was run under the Windows MS-DOS command prompt, but it can be run under other operating systems such as UNIX (as all other Perl scripts). Figure 3.1 shows a sample screenshot of the Windows MS-DOS command

prompt running a relevant Perl script. To run the program natively, the sequence entropy files (.ent) and the program itself should be in the same directory, where all the sequence entropy files (.ent) are in a user defined directory. Now under the Windows MS-DOS command prompt you simply input the name of the program (*extract_fractanalysis_entropy_aggr.pl*) followed by a space and then the name of the directory that all the sequence entropy files are located in. This will then output .fract files that contain fractional data. All these .fract files can then be placed in a user defined directory.

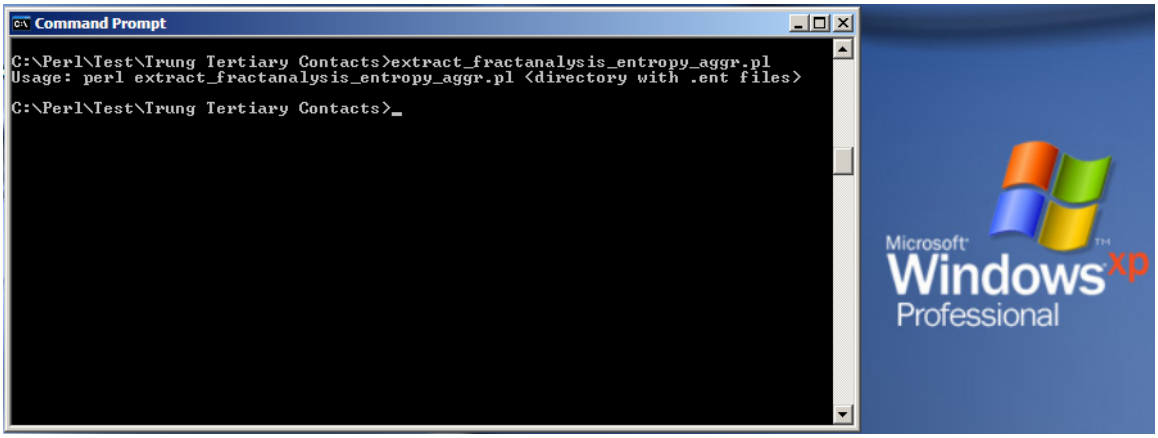

Figure 3.1: Sample screenshot of Windows MS-DOS command prompt running the Perl Script *extract_fractanalysis_entropy_aggr.pl*. Here the name of the program (*extract_fractanalysis_entropy_aggr.pl*) was inputted but omitted was the name of the directory with the sequence entropy files (.ent). Perl outputs directions on what is necessary for the program to run which is the name of the directory with the sequence entropy files (.ent).

3.5.3 ftp-script-1.pl

The next homology-based value that was to be calculated was packing density which is defined as the number of C-alpha atoms falling within the radius of 9 Å around a residue of interest. Residue packing density is important because it provides an estimate of how well an amino acid is surrounded by neighboring residues. There are three Perl scripts that work together to extract the residue packing density data, and these scripts are *ftp-script-1.pl, cif2den.pl,* and *Chainselectivecif2den.pl*. The Perl script *ftp-script-1.pl* was used to download the mmCIF files for all 268 proteins, which can save a lot of time.

3.5.4 cif2den.pl and Chainselectivecif2den.pl

Once *ftp-script-1.pl* downloads all the mmCIF files from the RCSB Protein Data Bank, *cif2den.pl* and *Chainselectivecif2den.pl* then extracts the residue packing density data from the mmCIF files. First off, for these scripts to work correctly the file *SeqIDlearningset268.txt* must be in the same directory as the Perl scripts. This file can be found in Appendix C. Also, the mmCIF files must be stored in a directory that is named "CIF_files" which must also be in the same directory as the Perl scripts.

To extract residue packing density data from the mmCIF files one by one, *cif2den.pl* is run under the Windows MS-DOS command prompt. To run *cif2den.pl* correctly, the file name was input followed by a space, then the chain name of the protein followed by a space, and lastly the name of mmCIF file (xxxx.cif). This will then extract

residue packing density data and output a .den file, which can be viewed by Excel or Notepad, or any other text viewer.

To extract the packing density values of the whole set of 268 proteins at once, the Perl script *Chainselectivecif2den.pl* was utilized. This program was simply run under the Windows MS-DOS command prompt, with *SeqIDlearningset268.txt* and mmCIF files (in "CIF_files" directory) in the same directory, and the program outputs all the .den files of the all 268 set of proteins. The .den files used for this research were directly taken from previously performed research (Mishra, 2010).

3.5.5 Data alignment and extract_individualfractentropy_density_aggr.pl

Some problems that occur with the outputs of the sequence entropy/fractional calculations and residue packing density data is that the data values of each file of the same protein sometimes do not line up correctly. For example, when viewing the .den and .fract files for the same protein with a text editor such as Notepad, the data for one file might start on the first line, while the data for the other file might start on the second.

A possible reason why this issue arises is because there is a difference between the amino acid sequence positions of the RCSB Protein Data Bank FASTA Sequence format and the FASTA format from the NCBI website. What has been seen is that for some proteins, the amino sequence positions do not line up exactly. Usually seen for the NCBI FASTA formatted amino acid sequence, there is sometimes one extra character in the beginning of the sequence that shifts the whole sequence one position when compared

to the RCSB FASTA Sequence format. This discrepancy between the two formats of the amino acid sequence can affect the outputs of the blastp .txt files since the FASTA format that is used is from RCSB whereas the BLAST program is run under NCBI. This inconsistency between the two FASTA formats not only can create problems with the blastp .txt files but also the other data files, i.e. .ent and .fract. However, the Perl script *extract_individualfractentropy_density_aggr.pl* is used to bypass these issues.

The script *extract_individualfractentropy_density_aggr.pl* takes both the .den files and the .fract files and outputs an aggregate set of data in one file. The outputted file contains the sequence entropy data, the fractional parameters calculations, and the residue packing density data where the values for each residue are now lined up correctly. With the correctly aligned data, analysis of the values can be performed to show trends of the sequence homology-based parameters.

To run *extract_individualfractentropy_density_aggr.pl*, the Windows MS-DOS command prompt was followed by the inputting of the name of the program followed by a space, then the name of the directory with the .fract files followed by a space, and lastly the name of the directory with the packing density (.den) files. The program then outputs .txt files with aggregate data.

3.5.6 calculate_aggr_per_protein.pl and double_agg_forPlot.pl

The Perl script *calculate_aggr_per_protein.pl* further analyzed the aligned sequence using the homology-based parameter values of *extract_individualfractentropy_density_aggr.pl*. It calculated the single average of sequence homology-based parameters at each packing density position. To run this Perl script, the name of program under the Windows MS-DOS command prompt was followed by a space, then the name of the directory with all aligned sequence homology-based files followed by a space, and then the name of the directory where the files are to be outputted, which must be created before running the program. The program then outputs .txt files of the calculations.

The script *double_agg_forPlot.pl* compiled outputs of *calculate_aggr_per_protein.pl* into one single file. To run the script, the name of the program was input at the Windows MS-DOS command prompt followed by a space, then the name of the directory with the aggregate files followed by a space, and the name of the output file. The output file was a text file, for example .txt.

3.5.7 listNoAlignments.pl, No_of_res_count.pl, and Bitscorelistno_ofsubject.pl

listNoAlignments.pl is a Perl script that outputs a file that contains the frequency of query proteins versus the number of alignments. This data was used to create a histogram. To run the program, the name of the program was inputted at the Windows

MS-DOS command prompt followed by a space, then the name of the directory with the blastp .txt files followed by a space, and then the name of the output file, which was a text file.

No_of_res_count.pl output a file that contained data on the frequency of query proteins versus length of query proteins. The data is also used to create a histogram. To run the program, the name of the program was inputted at the Windows MS-DOS command prompt followed by a space, and then the name of the directory with all the packing density (.den) files followed by a space, and then the name of the output file preferably a text file with the .txt extension.

Bitscorelistno_ofsubject.pl was the last script that is used to extract protein sequence homology-based data and this script outputs a file that list the frequency of subject proteins at a certain BLAST bit score. To run this program the name of the program was inputted at the Windows MS-DOS command prompt followed by a space, then the name of the directory with the blastp .txt files followed by a space, and then the name of the output file with .txt as the extension.

Figure 3.2 shows a schematic flowchart detailing the Perl-based procedures in order of usage. Displayed are the names of each Perl script and the files that are inputted or outputted for each script. File outputs were then validated independently.

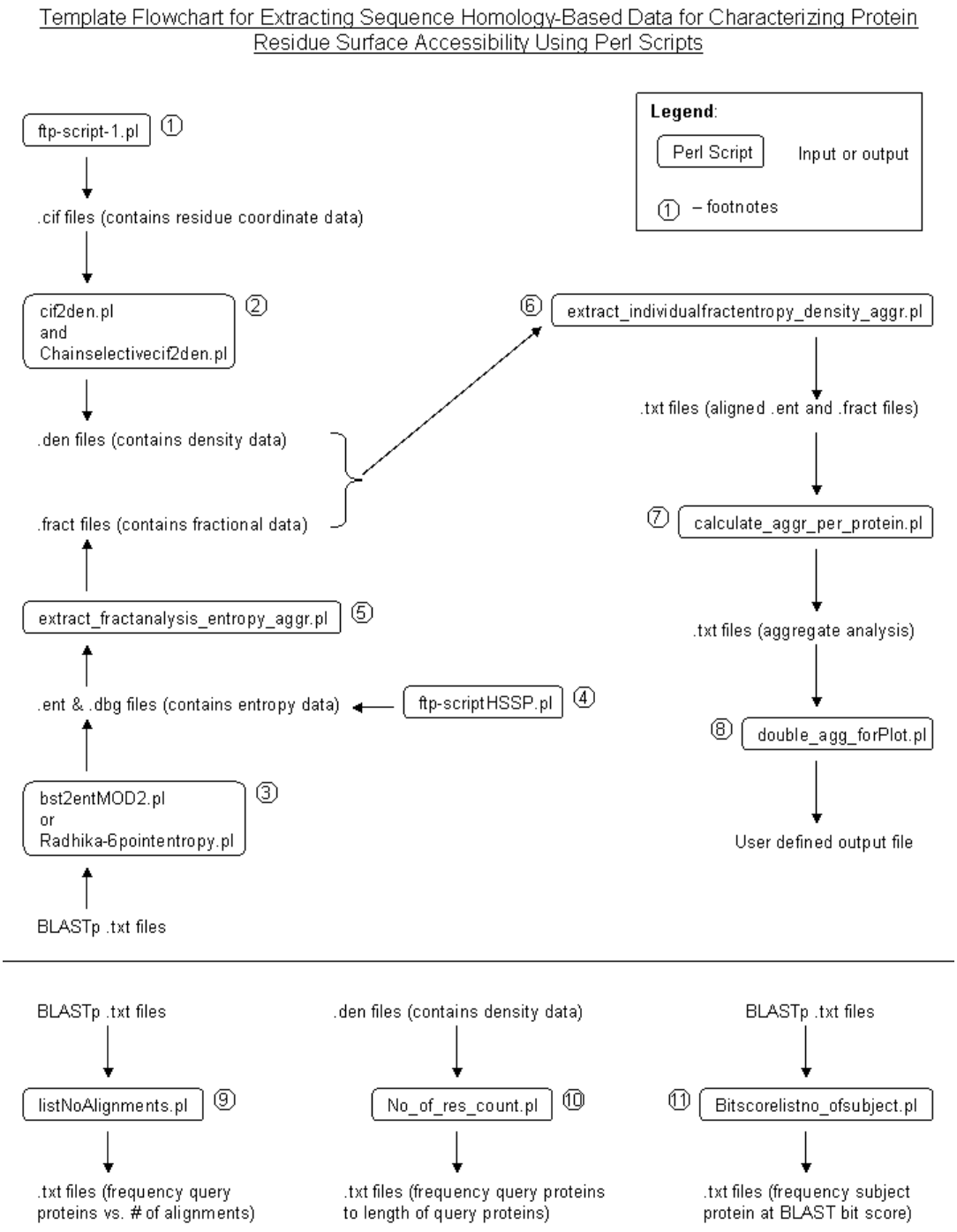

Figure 3.2: Schematic flowchart detailing methods on extracting sequence homology-based data using Perl. The flowchart contains names of each Perl script use for this research (complete Perl script can be found in Appendix B) and list what type of input or output each Perl script requires.

3.6 Averaging Tertiary Contacts and Learning Sets

Since tertiary contacts are pairs of amino acids residues, one can represent their sequence entropy, packing density, and RSA values also as a pair where the values of both of the residues are added together and then averaged. What is to be seen is whether this averaging method validates tertiary contacts as residues with generally lower sequence entropy and greater packing density since tertiary contacts are known to be better conserved and more densely packed.

For one of the learning sets, the sequence entropy, packing density, and RSA value of every possible pair of amino acids were also averaged. To do this, an *m x m* matrix was composed where *m* is the particular value for the amino acid residue in the order of the protein sequence, and the elements of the matrix are the averaged values of the pairs of sequence entropy, packing density, or RSA values. After the averaging was performed, each row of the matrix, starting with the first, were aligned one beneath the other where further analysis was to be performed. Each *m x m* calculation was appropriately calculated by a standard spreadsheet (i.e. Microsoft Excel 2003). Specifically for all-pair averaged amino acids, sequence entropy, packing density, and RSA values are also averaged. Non-averaged values can then be compared.

For the final learning set of tertiary contacts, we filtered out tertiary contacts that were separated by at least 10 amino acids in the primary protein sequence (Kim and Park, 2004). This final learning set of tertiary contacts is used to create the tertiary contact

threshold value and the full set of tertiary contacts is another learning set to be tested against.

3.7 Sequence Entropy versus Inverse Packing Density and RSA

Sequence entropy values were plotted against both inverse packing density and RSA to determine if there were any trends between the homology-based values. Inverse packing density and RSA were also averaged at each position that corresponded with the averaging of sequence entropy values. There are a total of four different learning sets: non-averaged values refer to the lack of averaging for any particular residue, all-pair averaged values refer to averaging among all possible pairs of residues, pair averaged tertiary contact values, and 10-separated pair averaged tertiary contact values, which are tertiary contacts that are separated by at least 10 amino acids. For these four learning sets of values, plots of sequence entropy versus inverse packing density and sequence entropy versus RSA were created and analyzed for trends and patterns that potentially give insights into protein structure and tertiary contacts.

3.8 Tertiary Contact Threshold, RSA Threshold, and Packing Density Threshold

From the 10-separated pair averaged tertiary contact values, a tertiary contact threshold value was determined from the sequence entropy values to potentially be utilized as a prediction filter of whether amino acid residues are buried or on the surface

of a protein. To do this, the requirement was for 95% of the lowest sequence entropy values from the 10-separated pair averaged tertiary contact values to be correctly identified as being buried. The resulting tertiary contact threshold value was applied to the sequence entropy values of the last two learning sets (excluding non-averaged values) i.e. all-pair averaged values and the pair averaged tertiary contact values. Other thresholds involving packing density and RSA values are also to be applied. The packing density thresholds were consistent with the non-averaged correlation plot of sequence entropy versus inverse packing density. An RSA value of less than or equal to 20.0 indicates that an amino acid residue is buried within a protein (Carugo, 2000). Also examined was a packing density threshold, where a packing density value of less than 11 denotes an amino acid residue that is consistent with being found on the surface and greater than or equal to 11 is an amino acid residue that is buried. The packing density threshold is applied only to the corresponding packing density values of the complete population of RSA values where the tertiary contact threshold filter was applied first to the corresponding sequence entropy values of said RSA values.

Chapter 4

Results

4.1 Characterization of 75 Protein Set of Tertiary Contacts

Table 4.1 lists the PDB ID's of the 75 proteins with known tertiary contacts along with protein chain identifier, number of query residues, and number of alignments for each protein. This table is a subset from the 268 protein set (Mishra, 2010).

Table 4.1: 75 protein set with known tertiary contacts. Listed for each protein is the protein chain identifier, number of query residues, and number of alignments.

PDB ID	Chain	# Query Residues	# Alignments	PDB ID	Chain	# Query Residues	# Alignments
1A2K	A	127	401	1CRC	A	105	1012
1A32	A	88	982	1DCS	A	311	636
1A48	A	306	1001	1DHT	A	327	1001
1A4I	A	301	1001	1DIN	A	236	986
1A6Q	A	382	1064	1E5M	A	416	1000
1ADE	A	431	1000	1EEH	A	437	1008
1AF3	A	196	513	1RBP	A	182	289
1AG9	A	175	747	2ACY	A	98	796
1AK4	C	145	1000	2G3P	A	225	219
1AMK	A	251	1000	2HDH	A	293	1195
1AMP	A	291	1001	2ILK	A	160	240
1AMU	A	563	2309	2JEL	P	85	1017
1AOB	A	265	1007	2LIV	A	344	1004
1AQ6	A	253	1000	2OHX	A	374	1005
1ATL	A	202	1005	2RN2	A	155	1000
1AUO	A	218	915	2SCP	A	174	447
1AW7	A	194	86	2SHP	A	525	1577
1AW9	A	216	1000	2SIC	I	107	55
1B3A	A	67	607	2SQC	A	631	970

1B5E	A	246	563	2TCT	A	207	1002
1B67	A	68	312	2TGI	A	112	1000
1B8A	A	438	1122	3GRS	A	478	1001
1BAM	A	213	15	3PFK	A	319	1222
1BBH	A	131	162	3PMG	A	561	1004
1BD0	A	388	1000	3RN3	A	124	626
1BEA	A	127	340	3SDH	A	146	633
1BF2	A	750	1029	3SGB	E	185	349
1BIA	A	321	1000	3SGB	I	56	572
1BIQ	A	375	1001	4DFR	A	159	1000
1BJW	A	382	1000	4HTC	I	65	35
1BMD	A	327	1002	5CPA	A	307	1075
1BRS	A	110	106	5CSM	A	256	87
1BRW	A	433	849	6LDH	A	330	1001
1BT3	A	345	1046	8ATC	A	310	1000
1BXQ	A	323	1004	8PTI	A	58	1543
1CB0	A	283	989	9PAP	A	212	1001
1CEX	A	214	272	9WGA	A	171	1556
1CJX	A	357	615				

Frequency distribution plots were constructed for the number of query residues, the number of alignments, and the BLAST bit score values for each protein of the 75 protein set with known tertiary contacts. There are a total of 19744 query residues for the 75 protein chains which aligned to 62230 subject proteins from the BLAST database and had a total of 107520 BLAST bit score values. Figure 4.1A, Figure 4.1B, and Figure 4.1C displays the frequency distribution plots for the frequency of the number of query residues, the number of alignments, and the BLAST bit score values, respectively, for each protein of the 75 protein set.

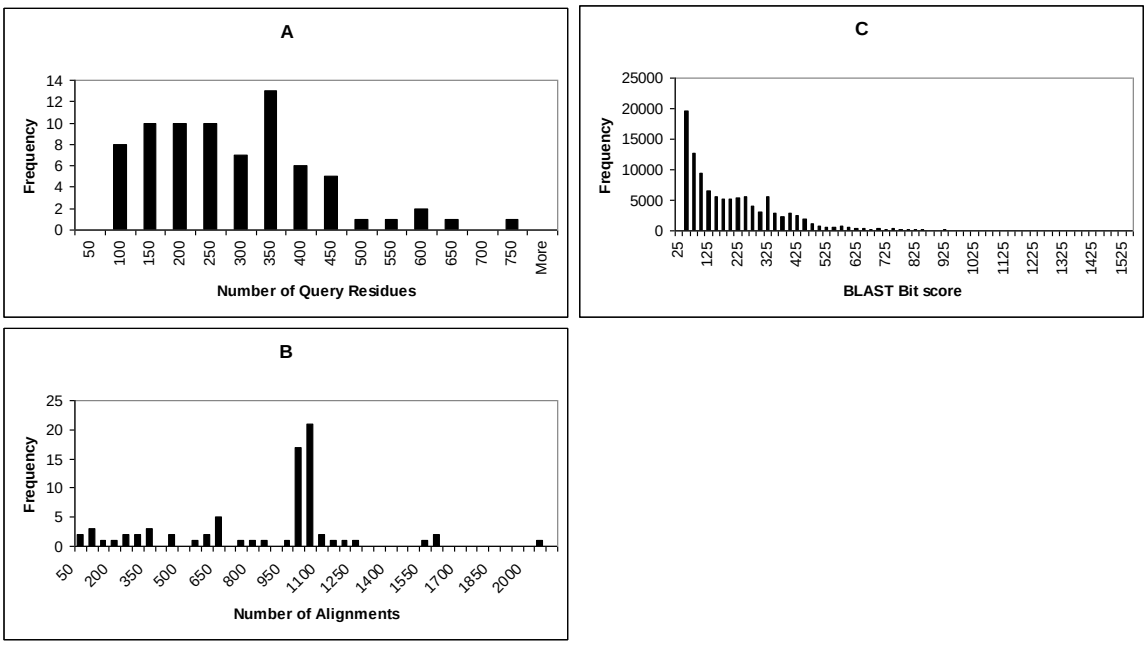

Figure 4.1: Frequency distribution plots for the characterization of the 75 protein set with known tertiary contacts. Total of 75 protein chains with 19744 query residues and 62230 aligned subject protein sequences were used for these calculations. A. Frequency distribution plot for each protein of the 75 protein set with respect to the number of query residues. B. Frequency distribution plot for each protein of the 75 protein set with respect to the number of alignments. C. Frequency distribution plot for each protein of the 75 protein set with respect to BLAST bit score values.

The frequency distribution plot for the 75 protein set with respect to the number of query residues (Figure 4.1A) is weighted towards the lower number of query residues with approximately 80% of the number of query residues ranging from 50 – 350 with a total range of 50 – 750. The apparent maximum for this distribution is number of query residues value of 350. The number of alignments associated with the 75 protein set range from 15 for 1BAMA to 2309 for 1AMUA. Approximately half (50.7%) of the proteins from the 75 protein set have alignments between 1000 and 1050 which is roughly the midpoint of the overall number of alignments ranging from 15 – 2309. The apparent

maximum for the frequency distribution of the number of alignments corresponds to 1050 alignments (Figure 4.1B). The frequency distribution of the aligned subject sequences with respect to BLAST bit score resulted in a right skewed distribution with the bit score values ranging from 0 – 1500 where 41873 (39%) of the bit score values falls between 0 to 100 (Figure 4.1C). These frequency distributions for the 75 protein set are comparable to the frequency distributions for the complete 268 protein set (Mishra, 2010).

A detailed characterization of the tertiary contacts for each of the proteins of the 75 protein set was also performed. Table A.2 lists the number of tertiary contacts for each protein, the number of residues between each tertiary contact, and the pair averaged RSA value of the tertiary contact. Upon tabulating the various tertiary contact parameters presented for the 75 protein set, there are a total of 527 tertiary contacts, 177 of which do not have 10 or more residues between them while 350 of them do. There are 248 tertiary contacts that do not have a pair averaged RSA value less than or equal to 20.0 and 279 that do. Of the 350 tertiary contacts that has 10 or more residues in between them, 191 (55%) of them have an RSA value that is less than or equal to 20.0. This set of 350 tertiary contacts is used to create the tertiary contact threshold value which is hoped to be used as a binary protein prediction filter of buried or surface amino acid residues.

4.2 Sequence Entropy versus Inverse Packing Density

With respect to sequence entropy, packing density, and RSA values there are a total four learning sets: non-averaged values, all-pair averaged values, pair averaged

tertiary contact values, and 10-separated pair averaged tertiary contact values. The 10-separated pair averaged tertiary contact values is the tertiary contacts that have 10 or more amino acid residues in between them. For each learning set of sequence entropy, packing density, and RSA values sequence entropy versus inverse packing density was plotted.

For the learning set of non-averaged values, which are Perl outputted, there were a total of 19158 query residue values. The inverse packing density values were averaged at each inverse packing density position with subsequent averaging of sequence entropy values. The correlation plot shown in Figure 4.2 shows two major regions. In one region, sequence entropy increased linearly with increasing inverse packing density, values between 0 and 0.10. And in the other region sequence entropy stayed approximately the same with increasing inverse packing density values from 0.10 to 0.25.

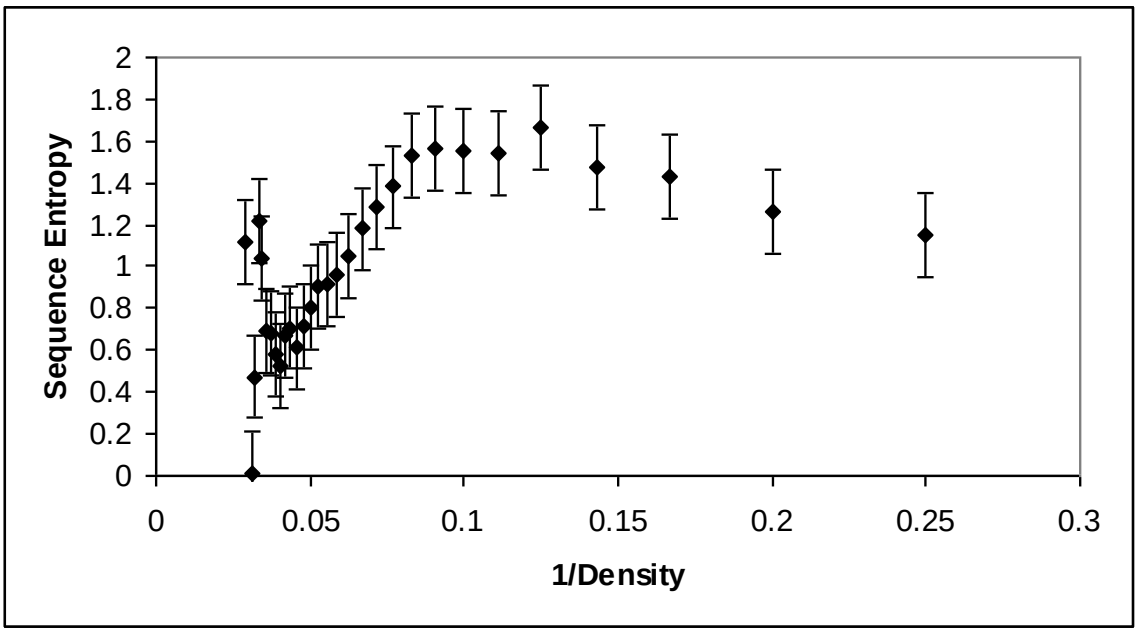

Figure 4.2: Correlation plot for 75 protein set of non-averaged sequence entropy and inverse packing density values. The aggregate sequence entropy values correspond to 19158 query residues of the 75 protein set with known tertiary contacts and are calculated by averaging the sequence entropy values at each inverse packing density position.

For the learning set of all-pair averaged values, averaged for every possible pair of amino acids, there were a total of 3556690 pair averaged query residue values. The inverse packing density values were double averaged at each inverse packing density position with subsequent averaging of sequence entropy values. The correlation plot shown in Figure 4.3 shows two major regions. For the general two region morphology, in Region I (Liao et al., 2005) sequence entropy increased linearly with increasing inverse packing density, values between 0 and 0.091, and in Region II sequence entropy stayed approximately the same with increasing inverse packing density values greater than 0.091. The sequence entropy values were generally lower than the non-averaged sequence entropy values.

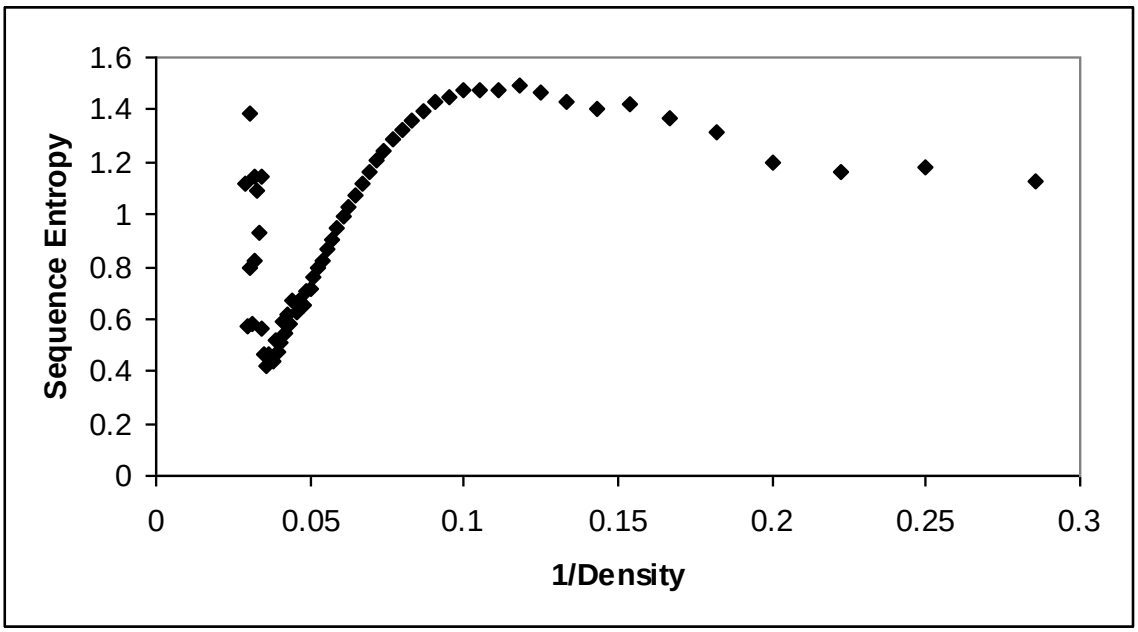

Figure 4.3: Correlation plot for 75 protein set of all-pair averaged sequence entropy and inverse packing density values. The sequence entropy values correspond to 3556690 pair averaged query residues of the 75 protein set with known tertiary contacts and are calculated by double averaging the pair averaged sequence entropy values at each inverse packing density position. Standard deviations are typically 0.2.

For the learning set of pair averaged tertiary contact values there were a total of 527 pair averaged query residues values. The inverse packing density values were averaged at each inverse packing density position with subsequent averaging of sequence entropy values. The correlation plot is shown in Figure 4.4. Sequence entropy values generally increase with increasing inverse packing density values between 0.04 and 0.10, major Region I. After the trend of linear increasing values, there are some points that obtrude from the linear trend where this set of values does not show a discernable trend itself. This set of values with no change range from inverse packing density values greater than 0.10, major Region II. Sequence entropy and range of inverse packing

density values (0.04 to 0.16) are generally lower than non-averaged and all-pair averaged sequence entropy and range of inverse packing density values.

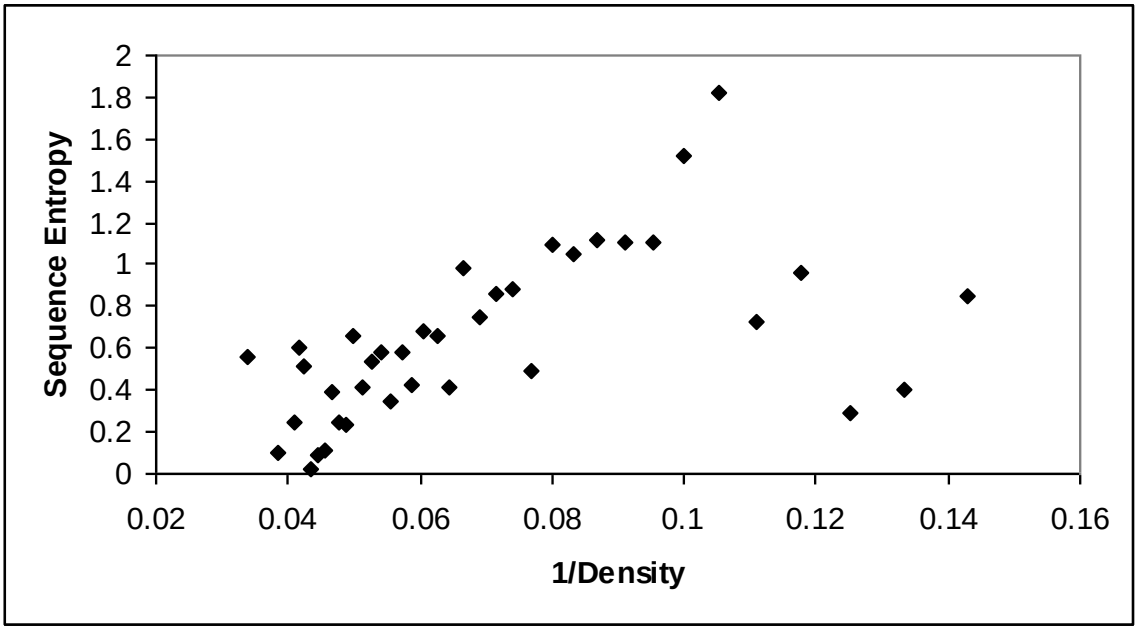

Figure 4.4: Correlation plot for 75 protein set of pair averaged tertiary contact sequence entropy and inverse packing density values. The sequence entropy values correspond to 527 pair averaged query residues of the 75 protein set with known tertiary contacts and are calculated by averaging the pair averaged sequence entropy values at each inverse packing density position. Standard deviations are typically 0.2.

For the learning set of the 10-separated pair averaged tertiary contact values there were a total of 350 pair average query residue values. The inverse packing density values were averaged at each inverse packing density position with subsequent averaging of sequence entropy values. The correlation plot is shown in Figure 4.5. Sequence entropy values generally increase with increasing inverse packing density values from 0.04 to 0.10, major Region I. After the trend of linear increasing values, there are some points that obtrude from the linear trend where this set of values does not show a discernable

trend itself. Sequence entropy and range of inverse packing density values (0.04 to 0.16 in major Region I) are generally lower than non-averaged, all-pair averaged, and pair averaged tertiary contact sequence entropy and inverse packing density values.

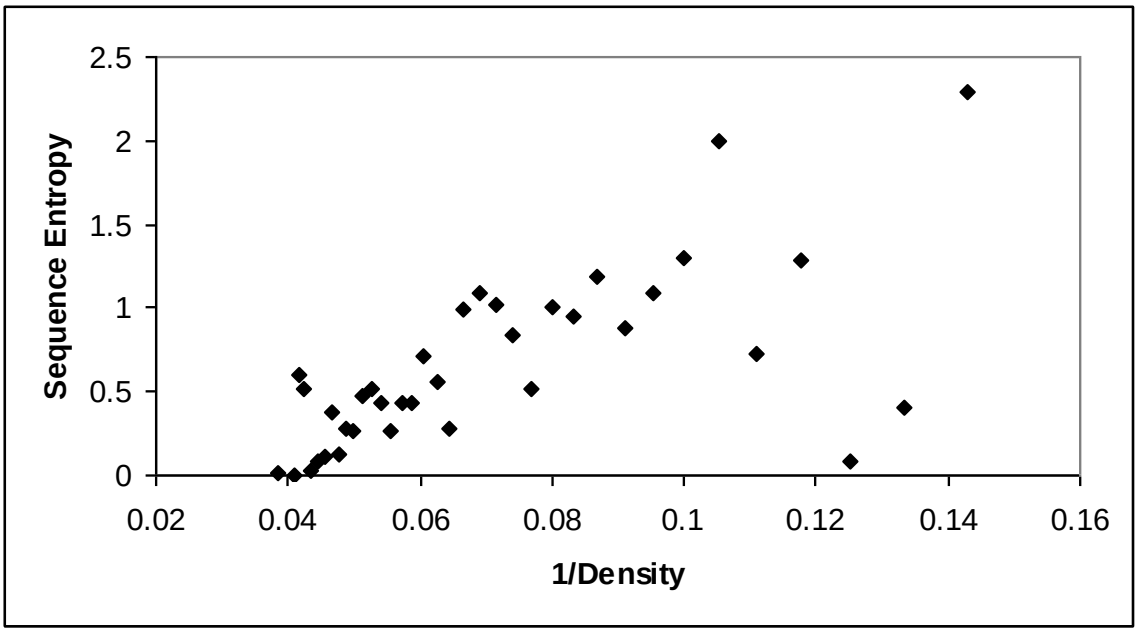

Figure 4.5: Correlation plot for 75 protein set of 10-separated pair averaged tertiary contact sequence entropy and inverse packing density values. The sequence entropy values correspond to 350 pair averaged query residues of the 75 protein set with known tertiary contacts and are calculated by averaging the pair averaged sequence entropy values at each inverse packing density position. For this smaller set standard deviations are typically 0.5.

An aggregate plot of all four learning sets of the correlation plots are shown in Figure 4.6. From this plot it can be seen that the non-averaged learning set plot generally has the highest sequence entropy values, the all-pair averaged learning set has the second highest, and both the pair averaged tertiary contact and 10-separated pair averaged tertiary contact learning set have the lowest sequence entropy values. Also, the inverse

packing density of both tertiary contact learning set values range on the lower end of Region I, whereas non-averaged and all-pair averaged learning set values range over a full range from 0 – 0.30. There is a general two-region morphology for all learning sets but the morphology is more distinct in the non-averaged and all-pair averaged learning set then in the tertiary contact learning sets.

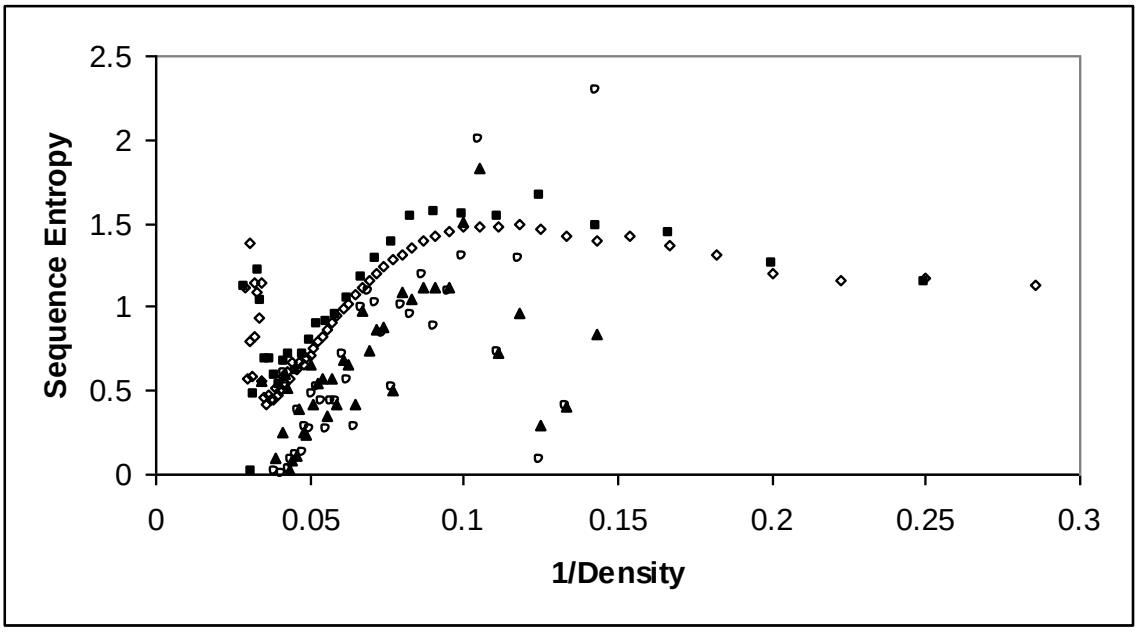

Figure 4.6: Comparison for 75 protein set of different classes of correlation data of sequence entropy and inverse packing density values. Note: non-averaged (square), all-pair averaged (diamond), pair averaged tertiary contacts (triangle), and 10-separated pair averaged tertiary contacts (circle).

4.3 Sequence Entropy versus RSA

Sequence entropy versus RSA was also plotted for all the learning sets of sequence entropy, packing density, and RSA values. RSA values for the non-averaged,

pair averaged tertiary contact, and 10-separated pair averaged tertiary contact learning sets were averaged at each RSA position with subsequent averaging of sequence entropy values. RSA values for the all-pair averaged learning set were double averaged at each RSA position with subsequent averaging of sequence entropy values. The RSA values for each protein for the all-pair averaged learning set were averaged and then combined together and averaged again.

The plots for each learning set showed that as RSA increased so did sequence entropy from an RSA range of 0 – 100. RSA values higher then 100 are not typical, but occasionally do occur. An aggregate plot (Figure 4.11) shows that the non-averaged learning set generally has the highest sequence entropy values, all-pair averaged learning set the second highest, and the tertiary contact learning sets have the lowest sequence entropy values.

The correlation plots of sequence entropy for the non-averaged, all-pair averaged, pair averaged tertiary contacts, and 10-separated pair averaged tertiary contacts are shown in Figures 4.7-4.10, respectively. The aggregate plot of all four learning sets of sequence entropy versus RSA is shown in Figure 4.11.

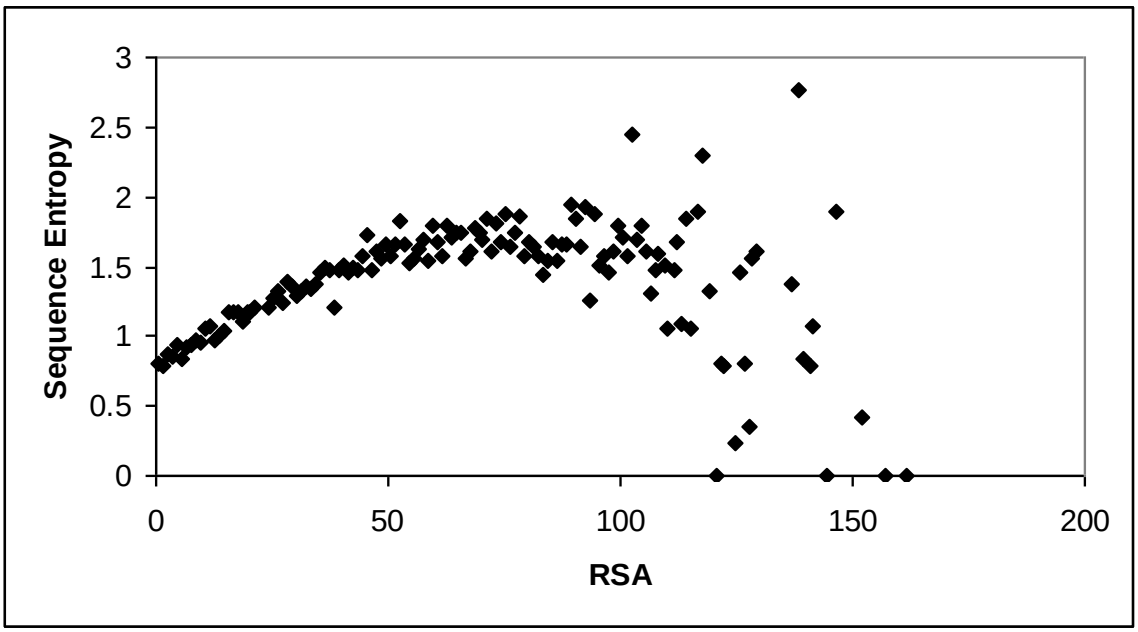

Figure 4.7: Correlation plot for 75 protein set of non-averaged sequence entropy and RSA values. The sequence entropy values correspond to 19158 query residues of the 75 protein set with known tertiary contacts and are calculated by first averaging the sequence entropy values at each RSA position and then averaging RSA value within bins of increment 1 and subsequent averaging of sequence entropy values. Standard deviations are typically 0.5.

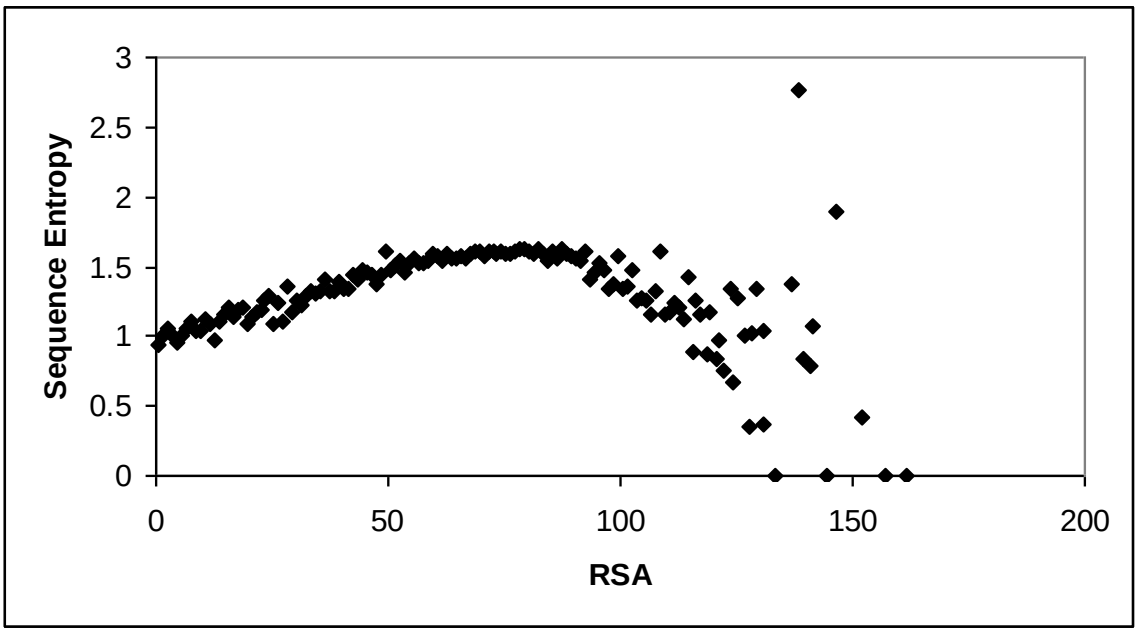

Figure 4.8: Correlation plot for 75 protein set of all-pair averaged sequence entropy and RSA values. The sequence entropy values correspond to 3556690 pair averaged query residues of the 75 protein set with known tertiary contacts and are calculated by first double averaging the sequence entropy values at each RSA position and then averaging RSA value within bins of increment 1 and subsequent averaging of sequence entropy values. Standard deviations are typically 0.5.

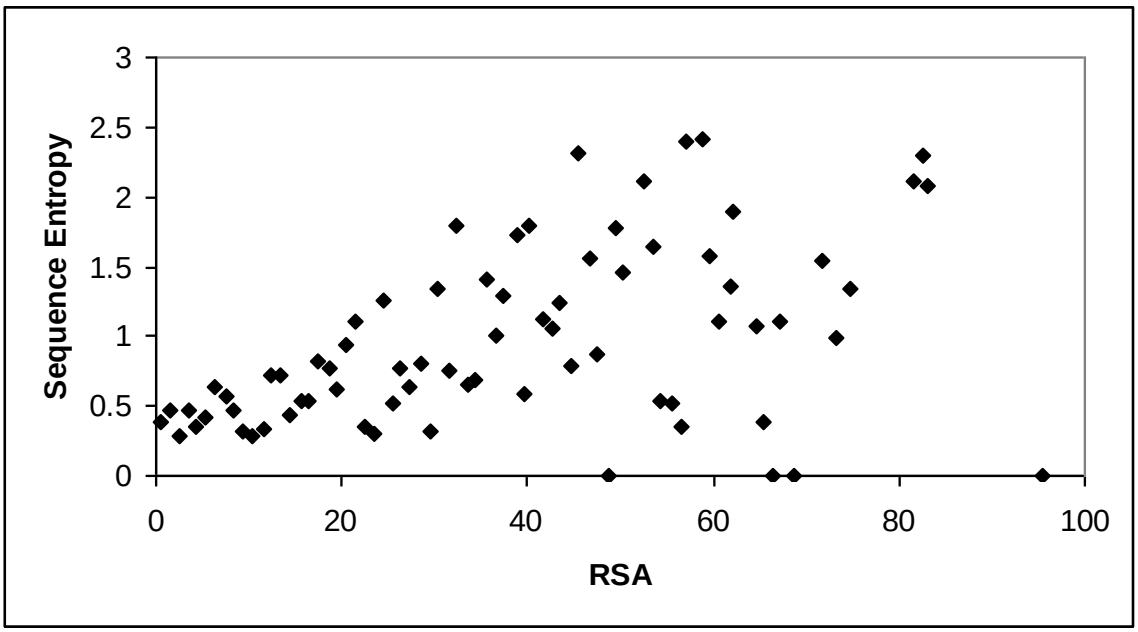

Figure 4.9: Correlation plot for 75 protein set of pair averaged tertiary contact sequence entropy and RSA values. The sequence entropy values correspond to 527 pair averaged query residues of the 75 protein set with known tertiary contacts and are calculated by first averaging the sequence entropy values at each RSA position and then averaging RSA value within bins of increment 1 and subsequent averaging of sequence entropy values. Standard deviations are typically 0.5.

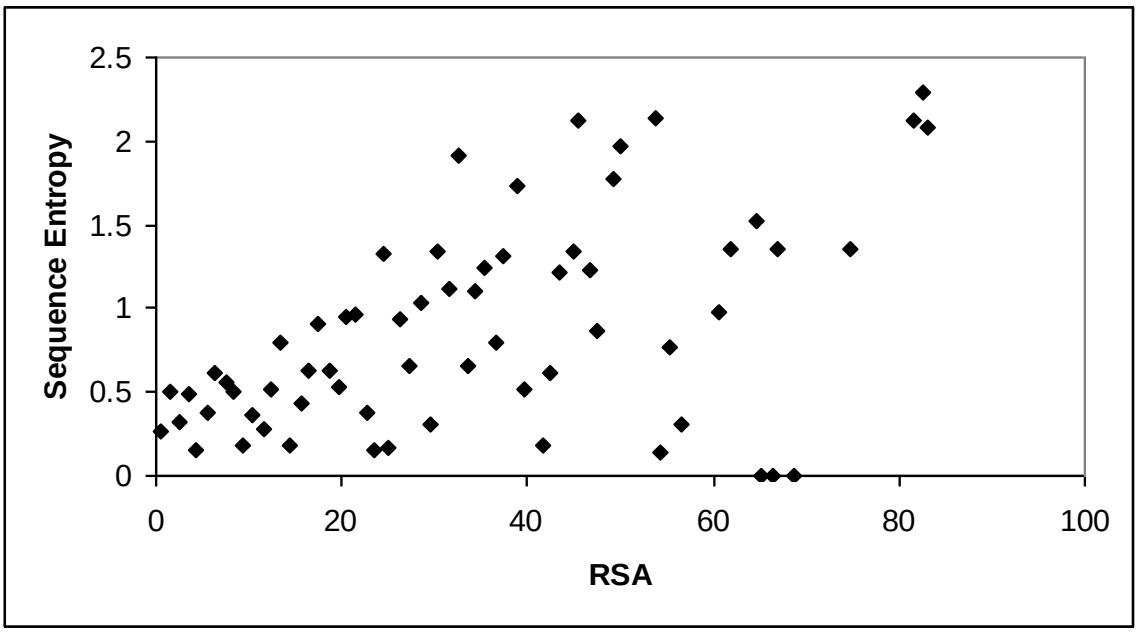

Figure 4.10: Correlation plot for 75 protein set of 10-separated pair averaged tertiary contact sequence entropy and RSA values. The sequence entropy values correspond to 350 pair averaged query residues of the 75 protein set with known tertiary contacts and are calculated by first averaging the sequence entropy values at each RSA position and then averaging RSA value within bins of increment 1 and subsequent averaging of sequence entropy values. Standard deviations are typically 0.5.

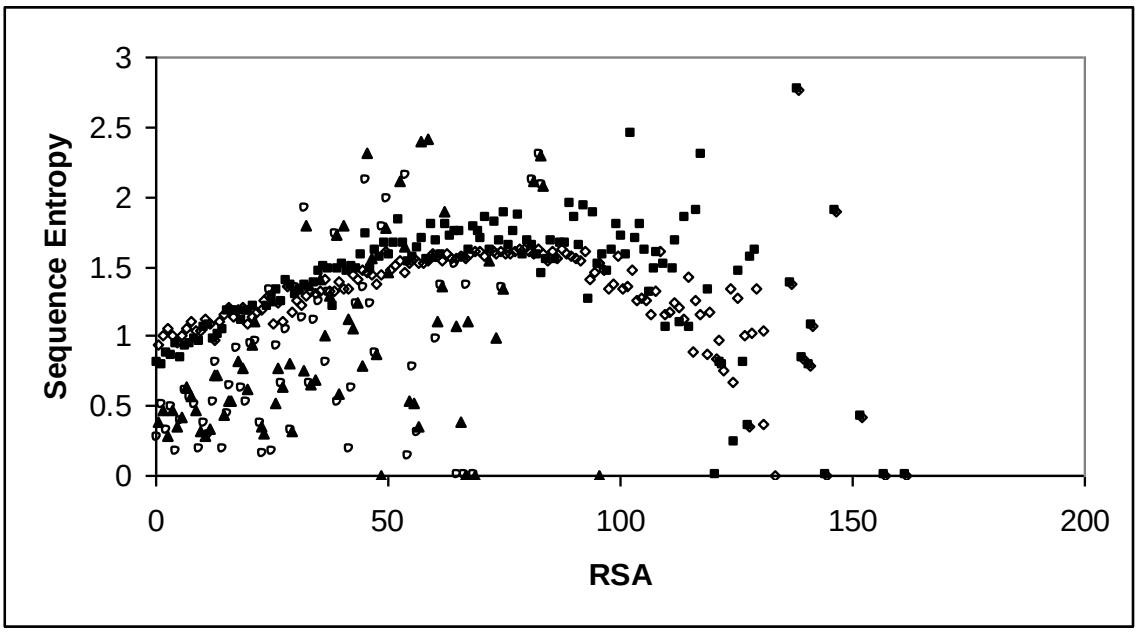

Figure 4.11: Comparison for 75 protein set of different classes of correlation data of sequence entropy and RSA values. Note: non-averaged (square), all-pair averaged (diamond), pair averaged tertiary contact (triangle), and 10-separated pair averaged tertiary contact (circle).

4.4 Tertiary Contact Analysis

From the data set of 75 proteins there were a total of 527 literature-derived tertiary contacts. The major types of tertiary contacts that were found are hydrogen bonds, ionic interactions, polar interactions, salt bridges, and disulfide bonds. Proteins that only had tertiary contacts that were hydrophobic interactions were not included in the final protein set. For this research, there was no further analysis on what these different types of tertiary contacts could imply to protein structure prediction; further analysis can be performed for future research. This set of 527 tertiary contacts is referred to as the "pair averaged tertiary contacts" learning set. Along with the non-averaged and all-pair

averaged learning sets, these are to be tested against the computationally derived 10-separated pair averaged tertiary contact learning set.

The 10-separated pair averaged tertiary contacts learning set was formed by excluding tertiary contacts that were less than 10 amino acids apart from the 527 pair averaged tertiary contact learning set. There are a total of 350 10-separated pair averaged tertiary contact sequence entropy values from which a tertiary contact threshold value can be designated. A threshold value was created by deeming 95% of the 350 sequence entropy values (from the 10-separated pair average tertiary contacts learning set) to be correctly characterized as tertiary contacts. Of those lowest sequence entropy values, the highest value was chosen to be the tertiary contact threshold value, which is 2.2285.

4.5 Frequency Distributions of Learning Sets

The frequency distribution of sequence entropy, packing density, and RSA values was performed for all learning sets. The tertiary contact threshold value of 2.2285, which is a sequence entropy value, was applied to the all-pair averaged and pair averaged tertiary contact learning sets. After applying the tertiary contact threshold, the homology-based values were separated by values that were greater than the threshold and values that were less than or equal to the threshold. Those iterations of values were subsequently applied to an RSA threshold of 20.0. Frequency distribution plots were performed for the learning sets where no threshold was applied, where tertiary contact threshold was applied, and where RSA threshold was applied subsequent the tertiary contact threshold.

4.6 Frequency Distributions of Non-Averaged Values

Frequency distribution plots were made for sequence entropy, packing density, and RSA values for the non-averaged learning set. There are a total of 19158 homology-based values. The packing density values range from 0 – 35, and the distribution follows a Gaussian-like distribution with an apparent maximum at packing density value 15 (Figure 4.12A). The sequence entropy values range from 0 – 3.863, and the distribution is right-skewed with an apparent maximum at sequence entropy value 0 (Figure 4.12B). The RSA values range from 0 – 161.4, and the distribution is right-skewed with an apparent maximum at RSA value 5 (Figure 4.12C). The frequency distributions for the non-averaged learning set are comparable to the frequency distributions for the complete 268 protein set (Mishra, 2010).

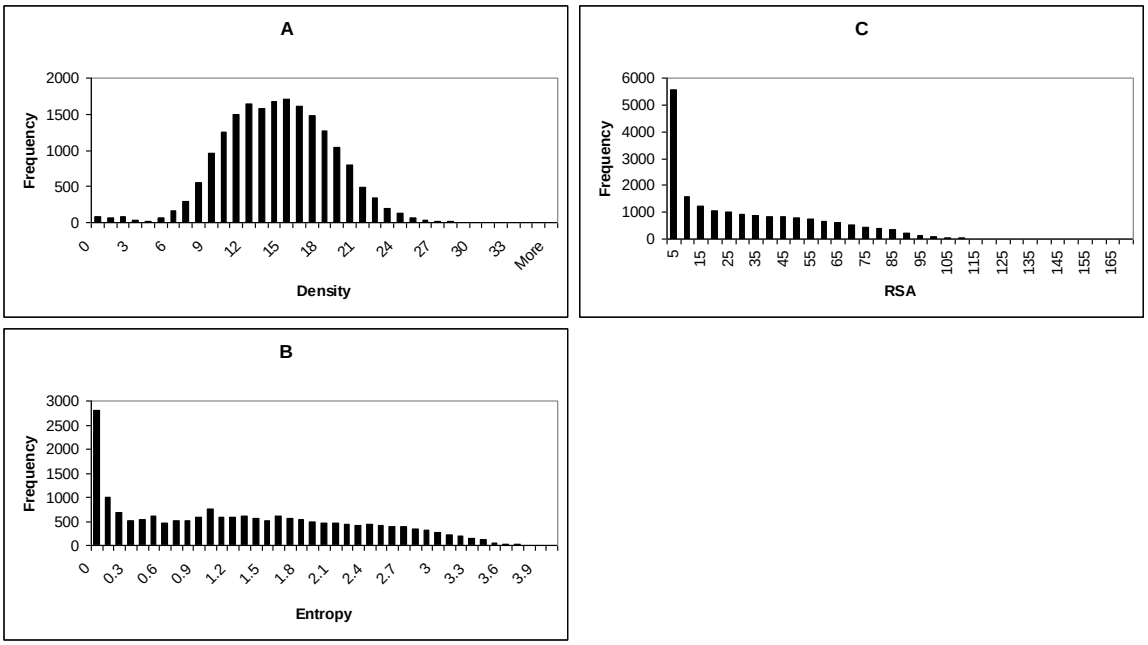

Figure 4.12: Frequency distribution plots for non-averaged packing density, sequence entropy, and RSA values. There are a total of 19158 query residues from the 75 protein set with known tertiary contacts. A. Frequency distribution plot for 19158 query residues with respect to packing density. B. Frequency distribution plot for 19158 query residues with respect to sequence entropy. C. Frequency distribution plot for 19158 query residues with respect to RSA.

4.7 Frequency Distributions of 10-Separated Pair Averaged Tertiary Contact Values

For the 10-separated pair averaged tertiary contacts learning set, there are a total of 350 tertiary contact pairs. The packing density values range from 7 – 26, and the distribution follows a Gaussian-like distribution with an apparent maximum at packing density value of 14 (Figure 4.13A). Sequence entropy values range from 0 – 3.267, and the distribution is right-skewed with apparent maximum at sequence entropy value of 0.1

(Figure 4.13B). RSA values range from 0 – 83.1 and the RSA values seems to decrease linearly from at RSA value of 5 (Figure 4.13C).

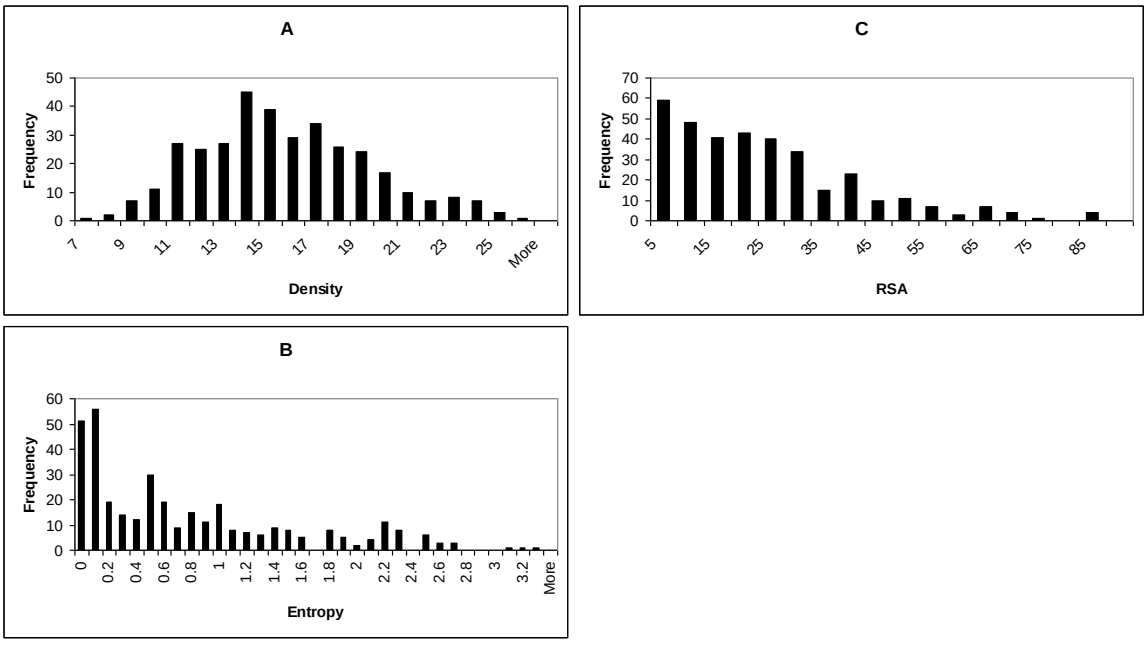

Figure 4.13: Frequency distribution plots for 10-separated pair averaged tertiary contact packing density, sequence entropy, and RSA values. There are a total of 350 pair averaged query residues from the 75 protein set with known tertiary contacts. A. Frequency distribution plot for 350 pair averaged query residues with respect to packing density. B. Frequency distribution plot for 350 pair averaged query residues with respect to sequence entropy. C. Frequency distribution plot for 350 pair averaged query residues with respect to RSA.

4.8 Frequency Distributions of Pair Averaged Tertiary Contact Values

There are a total of 527 tertiary contact pairs as defined in Table A.2. The packing density values range from 7 – 30, and the distribution follows a Gaussian-like distribution with apparent maxima at packing density values of 14 and 15 (Figure 4.14A). The sequence entropy values range from 0 – 3.267, and the distribution is right-skewed

with an apparent maximum at sequence entropy value 0.1 (Figure 4.14B). The RSA values range from 0 – 95.4 and seem to decrease linearly from an apparent maximum of an RSA value of 5 (Figure 4.14C).

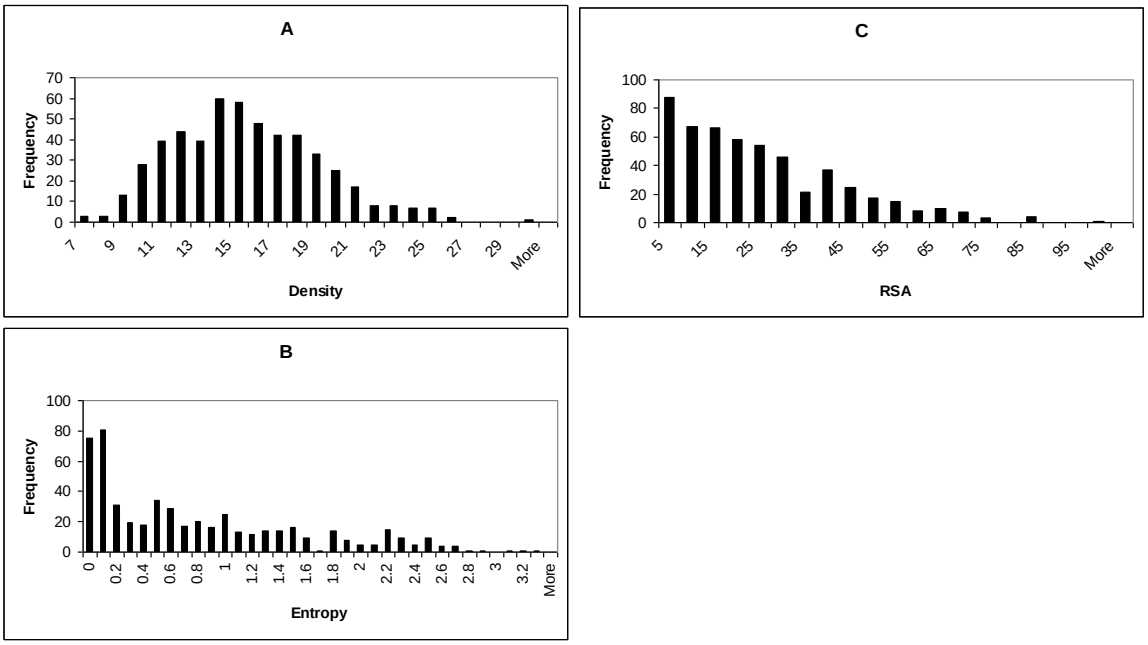

Figure 4.14: Frequency distribution plots for pair averaged tertiary contact packing density, sequence entropy, and RSA values. There are a total of 527 pair averaged query residues from the 75 protein set with known tertiary contacts. A. Frequency distribution plot for 527 pair averaged query residues with respect to packing density. B. Frequency distribution plot for 527 pair averaged query residues with respect to sequence entropy. C. Frequency distribution plot for 527 pair averaged query residues with respect to RSA.

Figures 4.15 and 4.16 show frequency distributions of both the packing density values and the RSA values with the tertiary contact threshold applied, respectively. Note that both of these frequency distributions give the same binary results of 496 and 31 residue pairs, respectively, when the tertiary contact threshold is applied. Also, note that there is a decoupling of the values when the tertiary contact threshold is applied for both

packing density and RSA. For the aggregate frequency distribution plot of packing density values (Figure 4.15C), when the tertiary contact threshold is applied significant decoupling of component distributions is shown. Note the RSA component distributions (Figure 4.16A and Figure 4.16B) are reasonably decoupled as noted in the overlay in Figure 4.16C.

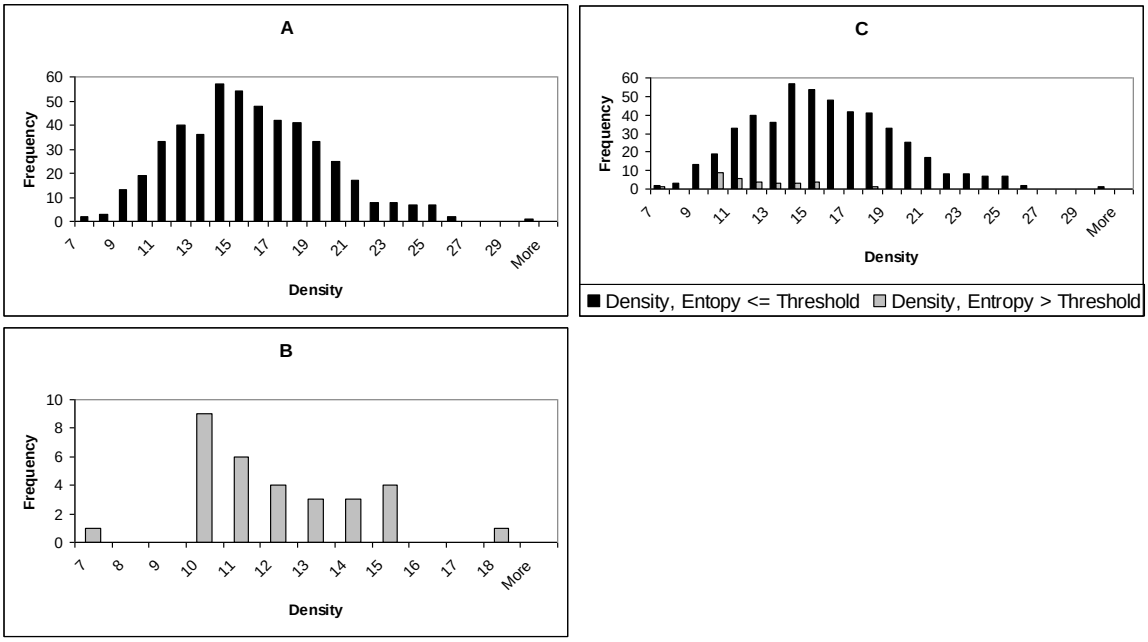

Figure 4.15: Frequency distribution plots for pair averaged tertiary contact packing density values with tertiary contact threshold value of 2.2285 applied. There are a total of 527 pair averaged query residues from the 75 protein set with known tertiary contacts. A. Frequency distribution plot for 496 pair averaged query residues with respect to packing density where corresponding sequence entropy values are less than or equal to tertiary contact threshold. B. Frequency distribution plot for 31 pair averaged query residues with respect to packing density where corresponding sequence entropy values are greater than tertiary contact threshold. C. Aggregate frequency distribution plot for known tertiary contact packing density values with tertiary contact threshold applied.

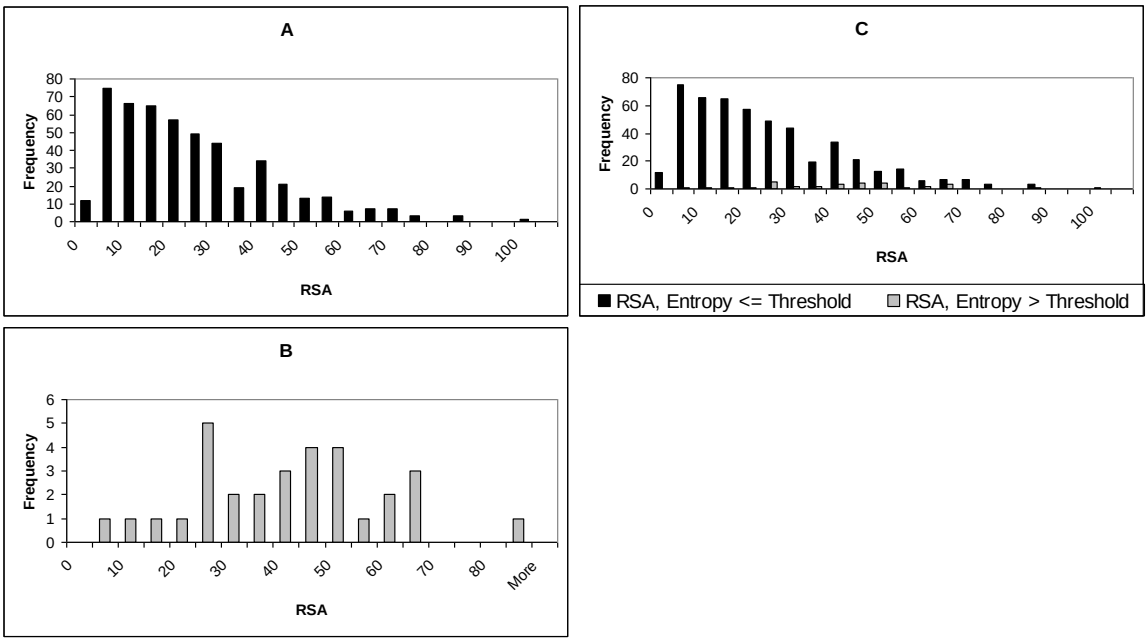

Figure 4.16: Frequency distribution plots for pair averaged tertiary contact RSA values with tertiary contact threshold value of 2.2285 applied. There are a total of 527 pair averaged query residues from the 75 protein set with known tertiary contacts. A. Frequency distribution plot for 496 pair averaged query residues with respect to RSA where corresponding sequence entropy values are less than or equal to tertiary contact threshold. B. Frequency distribution plot for 31 pair averaged query residues with respect to RSA where corresponding sequence entropy values are greater than tertiary contact threshold. C. Aggregate frequency distribution plot for known tertiary contact RSA values with tertiary contact threshold applied.

Next the RSA threshold value of 20.0 was applied to the 496 residue pairs that had corresponding sequence entropy values less than or equal to the tertiary contact threshold and resulted in 275 (55%) of their RSA values to be less than or equal to the RSA threshold and 221 (45%) of the RSA values to be greater than the RSA threshold. The entropy values that have corresponding RSA values less than or equal to the RSA threshold are right-skewed with an apparent maximum at entropy value of 0.1 (Figure 4.17A). The entropy values, with corresponding RSA values greater than the RSA

threshold one, is broadly distributed (Figure 4.17B). Figure 4.17C shows the aggregate plot of these frequency distributions. Lastly, there does not seem to be a decoupling of the two component distributions.

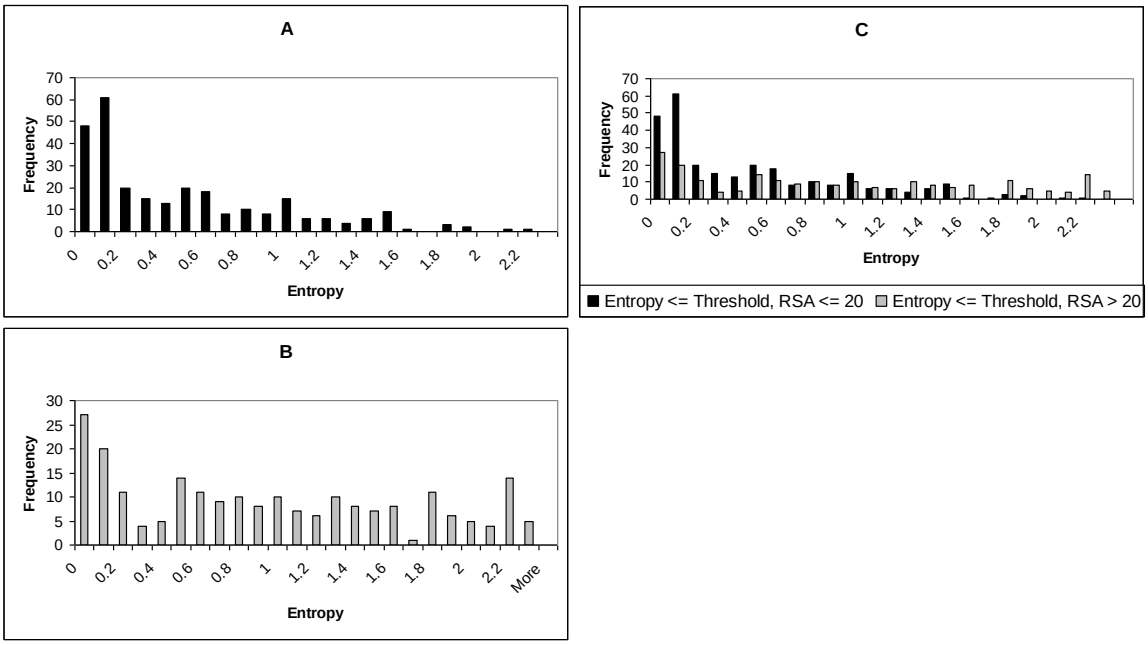

Figure 4.17: Frequency distribution plots for pair averaged tertiary contact sequence entropy values that are less than or equal to tertiary contact threshold value of 2.2285 with RSA threshold value of 20.0 applied. There are a total of 496 pair averaged query residues from the 75 protein set with known tertiary contacts. A. Frequency distribution plot for 275 pair averaged query residues with respect to sequence entropy values that are less than or equal to tertiary contact threshold where corresponding RSA values are less than or equal to RSA threshold. B. Frequency distribution plot for 221 pair averaged query residues with respect to sequence entropy values that are less than or equal to tertiary contact threshold value where corresponding RSA values are greater than RSA threshold value. C. Aggregate frequency distribution plot for pair averaged tertiary contact sequence entropy values that are less than or equal to tertiary contact threshold with RSA threshold applied.

Next the RSA threshold value of 20.0 was applied to the 31 residue pairs that had corresponding sequence entropy values greater than the tertiary contact threshold and

resulted in 5 (16%) of their RSA values to be less than or equal to the RSA threshold and 26 (84%) of the corresponding RSA values to be greater than the RSA threshold. Figure 4.18 shows the frequency distribution of the 31 entropy values whose corresponding RSA values are less than or equal to and greater than the RSA threshold value of 20.0. Figure 4.18C shows the aggregate plot of these frequency distributions.

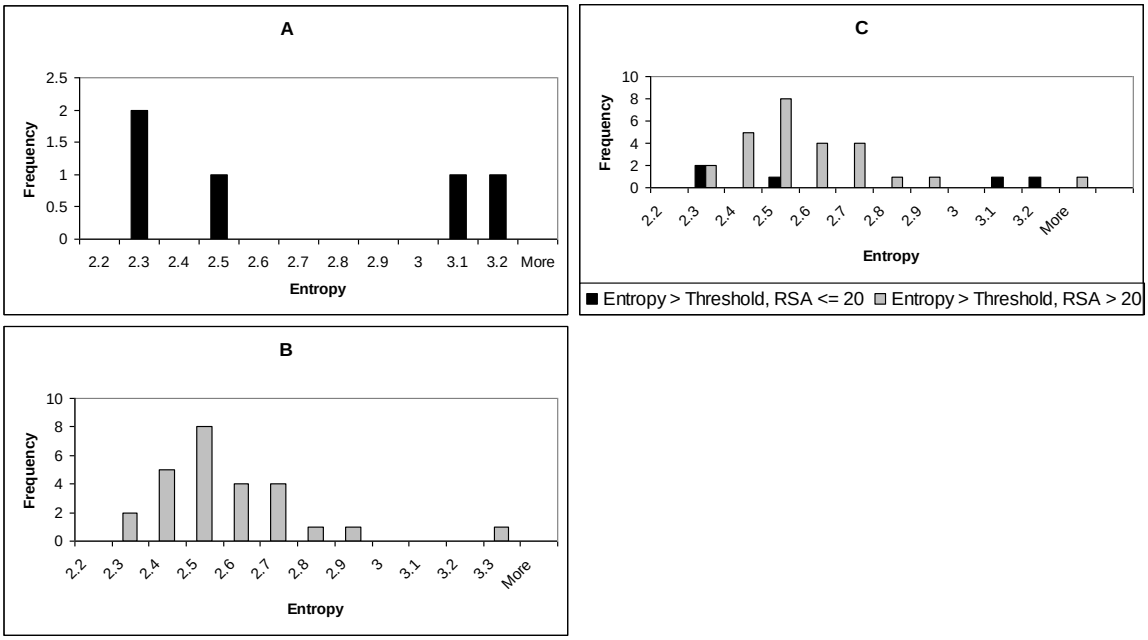

Figure 4.18: Frequency distribution plots for pair averaged tertiary contact sequence entropy values that are greater than tertiary contact threshold value of 2.2285 with RSA threshold value of 20.0 applied. There are a total of 31 pair averaged query residues from the 75 protein set with known tertiary contacts. A. Frequency distribution plot for 5 pair averaged query residues with respect to sequence entropy values that are greater than tertiary contact threshold where corresponding RSA values are less than or equal to RSA threshold. B. Frequency distribution plot for 26 pair averaged query residues with respect to sequence entropy values that are greater than tertiary contact threshold where corresponding RSA values are greater than RSA threshold. C. Aggregate frequency distribution plot for known tertiary contact sequence entropy values that are greater than tertiary contact threshold with RSA threshold applied.

4.9 Frequency Distributions of All-Pair Averaged Values

Frequency distribution plots were made for sequence entropy, packing density, and RSA values for all-pair averaged learning set. There are a total of 3556690 homology-based values. The packing density values range from 2 – 35 and the distribution follows a Gaussian-like distribution with an apparent maximum packing density value of 15 (Figure 4.19A). The sequence entropy values are right-skewed from 0 to approximately 4.4 with an apparent maximum sequence entropy value of 0 (Figure 4.19B). The RSA values range from 0 to approximately 165 with an apparent maximum RSA value of 5 (Figure 4.19C).

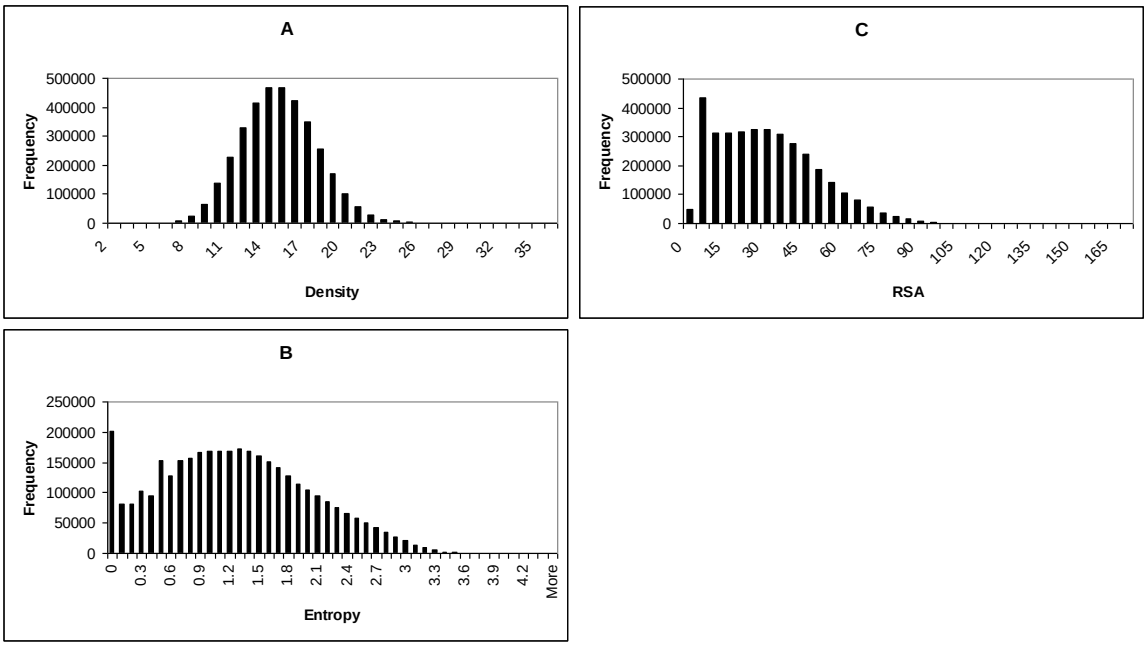

Figure 4.19: Frequency distribution plots for all-pair averaged packing density, sequence entropy, and RSA values. There are a total of 3556690 pair averaged query residues from the 75 protein set with known tertiary contacts. A. Frequency distribution plot for 3556690 pair averaged query residues with respect to packing density. B. Frequency distribution plot for 3556690 pair averaged query residues with respect to sequence entropy. C. Frequency distribution plot for 3556690 pair averaged query residues with respect to RSA.

The tertiary contact threshold value of 2.2285 was applied to the all-pair averaged sequence entropy values and resulted in 3170067 (89%) of the sequence entropy values to be less than or equal to the tertiary contact threshold and 386623 (11%) of the sequence entropy values to be greater than the tertiary contact threshold (Figure 4.20 and Figure 4.21, respectively). Both of these iterations of sequence entropy values were separated along with the corresponding packing density and RSA values. For the sequence entropy values that were separated along with the corresponding packing density values both distributions showed a Gaussian-like distribution. For less than or

equal to tertiary contact threshold the packing density values range from 2 – 35 with an apparent maximum at packing density value 15 (Figure 4.20A). For greater than tertiary contact threshold the packing density values range from 4 – 29 with an apparent maximum packing density value at 13 (Figure 4.20B). Some decoupling is noted for the overlay of the two component distributions (Figure 4.20C).

Figure 4.21 shows frequency distributions of the RSA values with the tertiary contact threshold applied, respectively. The component distributions involving RSA values (Figure 4.21A and Figure 4.21B) are right-skewed and Gaussian-like, respectively. The aggregate plot indicates some decoupling (Figure 4.21C).

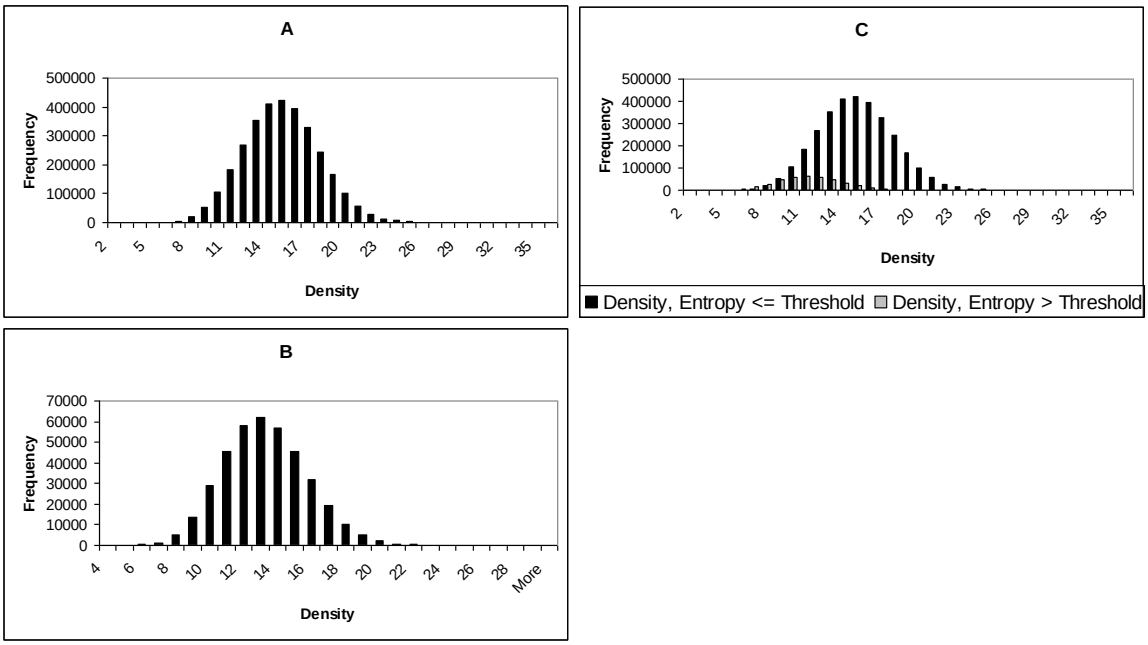

Figure 4.20: Frequency distribution plots for all-pair averaged packing density values with tertiary contact threshold value of 2.2285 applied. There are a total of 3556690 pair averaged query residues from the 75 protein set with known tertiary contacts. A. Frequency distribution plot for 3170067 pair averaged query residues with respect to packing density where corresponding sequence entropy values are less than or equal to tertiary contact threshold. B. Frequency distribution plot for 386623 pair averaged query residues with respect to packing density where corresponding sequence entropy values are greater than tertiary contact threshold. C. Aggregate frequency distribution plot for all-pair averaged packing density values with tertiary contact threshold applied.

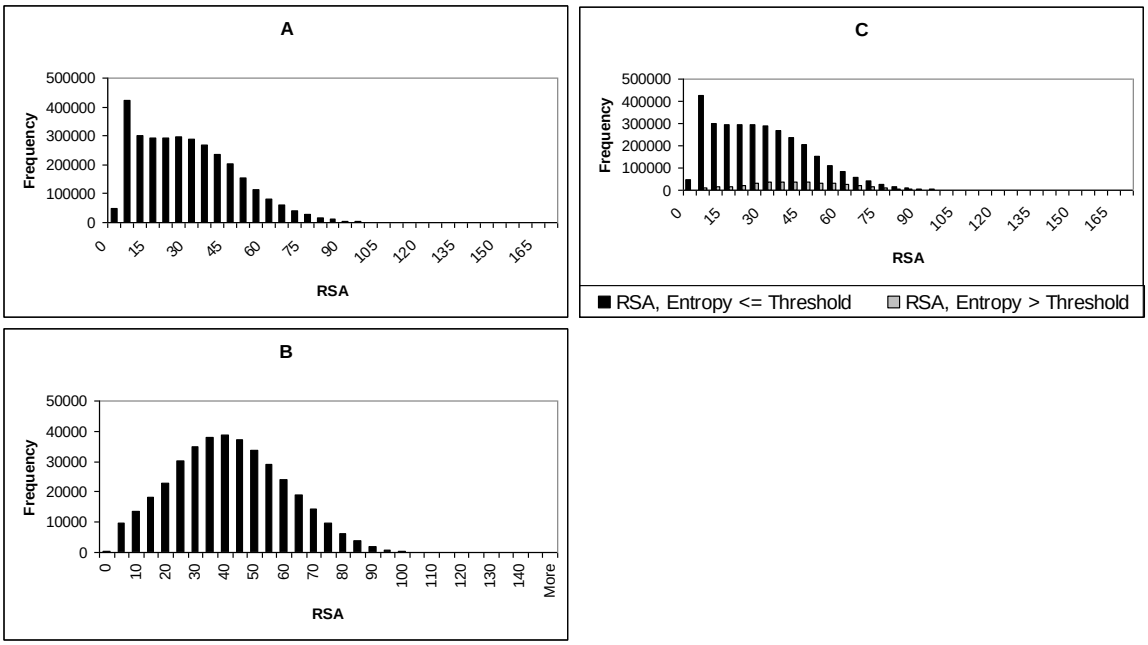

Figure 4.21: Frequency distribution plots for all-pair averaged RSA values with tertiary contact threshold value of 2.2285 applied. There are a total of 3556690 pair averaged query residues from the 75 protein set with known tertiary contacts. A. Frequency distribution plot for 3170067 pair averaged query residues with respect to RSA where corresponding sequence entropy values are less than or equal to tertiary contact threshold. B. Frequency distribution plot for 386623 pair averaged query residues with respect to RSA where corresponding sequence entropy values are greater than tertiary contact threshold. C. Aggregate frequency distribution plot for all-pair averaged RSA values with tertiary contact threshold applied.

Next the RSA threshold value of 20.0 was applied to the 3170067 residue pairs that had corresponding sequence entropy values less than or equal to the tertiary contact threshold value of 2.2285 (Figure 4.22) and resulted in 1357909 (43%) of the corresponding RSA values to be less than or equal to the RSA threshold (Figure 4.22A) and 1812158 (57%) of the corresponding RSA values to be greater than the RSA threshold (Figure 4.22B). Note the component distributions appear somewhat decoupled (Figure 4.22C).

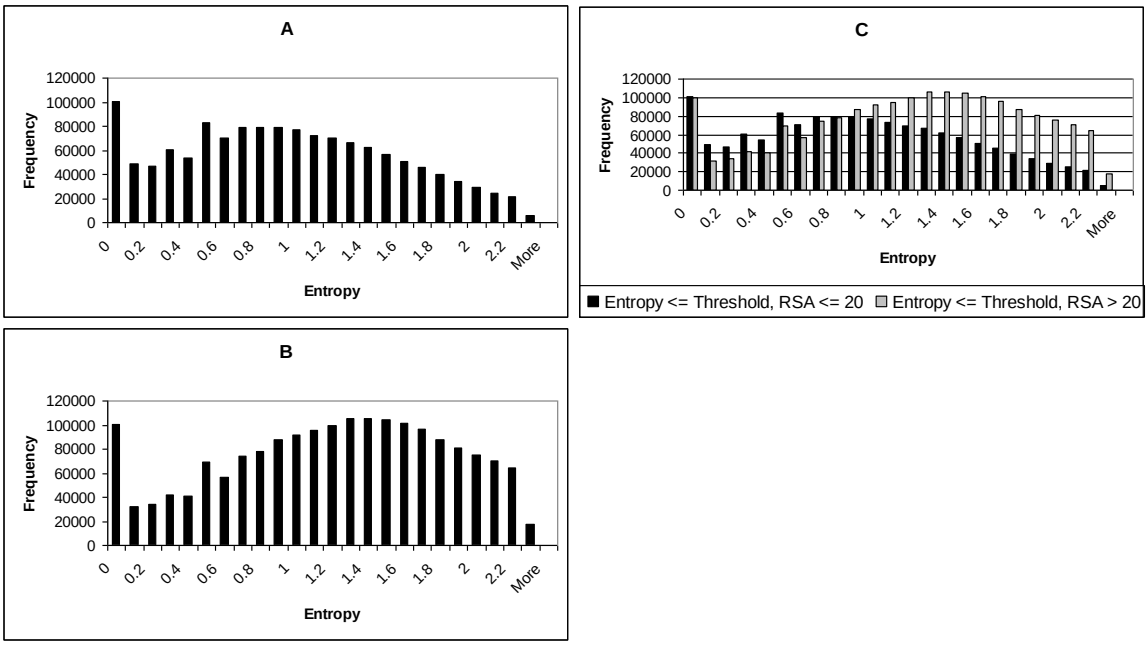

Figure 4.22: Frequency distribution plots for all-pair averaged sequence entropy values that are less than or equal to tertiary contact threshold value of 2.2285 with RSA threshold value of 20.0 applied. There are a total of 3170067 pair averaged query residues from the 75 protein set with known tertiary contacts. A. Frequency distribution plot for 1357909 pair averaged query residues with respect to sequence entropy values that are less than or equal to tertiary contact threshold where corresponding RSA values are less than or equal to RSA threshold. B. Frequency distribution plot for 1812158 pair averaged query residues with respect to sequence entropy values that are less than or equal to tertiary contact threshold where corresponding RSA values are greater than to RSA threshold. C. Aggregate frequency distribution plot for all-pair averaged sequence entropy values that are less than or equal to tertiary contact threshold with RSA threshold applied.

Next the RSA threshold value of 20.0 was applied to the 386623 residue pairs that had corresponding sequence entropy values greater than the tertiary contact threshold and resulted in 64480 (17%) of the corresponding RSA values to be less than or equal to the RSA threshold (Figure 4.23A) and 322143 (83%) of the corresponding RSA values to be greater than the RSA threshold (Figure 4.23B). Both component distributions are right-skewed and have similar apparent maximum values (Figure 4.23C).

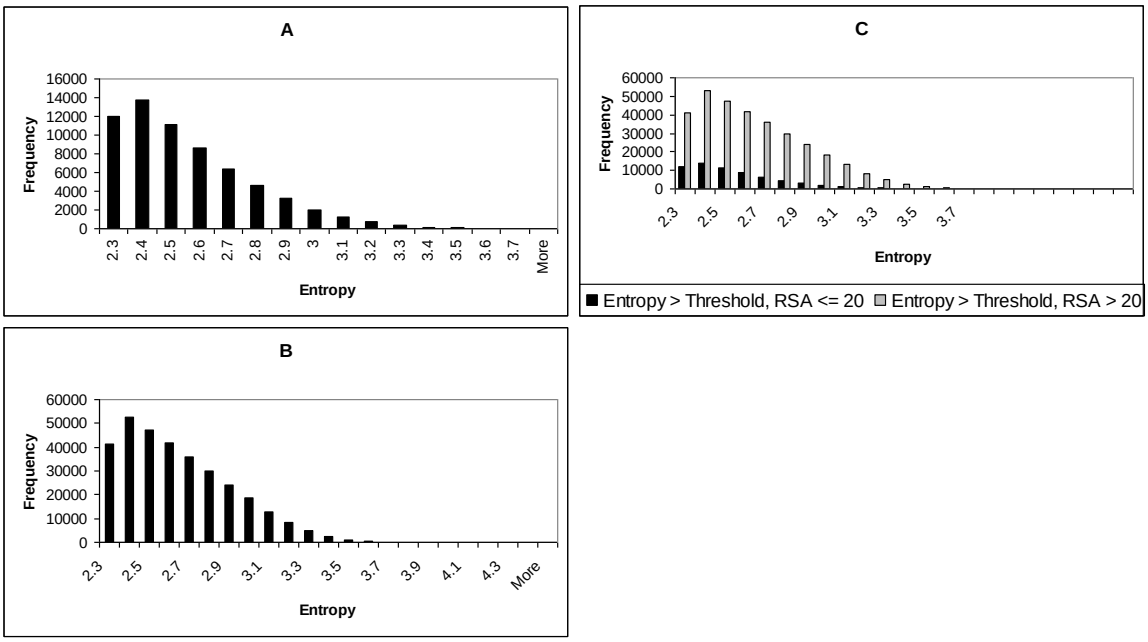

Figure 4.23: Frequency distribution plots for all-pair averaged sequence entropy values that are greater than tertiary contact threshold value of 2.2285 with RSA threshold value of 20.0 applied. There are a total of 386623 pair averaged query residues from the 75 protein set with known tertiary contacts. A. Frequency distribution plot for 64480 pair averaged query residues with respect to sequence entropy values that are greater than tertiary contact threshold where corresponding RSA values are less than or equal to RSA threshold. B. Frequency distribution plot for 322143 pair averaged query residues with respect to sequence entropy values that are greater than tertiary contact threshold where corresponding RSA values are greater than RSA threshold. C. Aggregate frequency distribution plot for all-pair averaged sequence entropy values that are greater than tertiary contact threshold with RSA threshold applied.

4.10 Frequency Distributions with Packing Density Threshold Applied

When applying the packing density threshold to the pair averaged tertiary contact packing density values that had corresponding sequence entropy values less than the tertiary contact threshold, the results showed 54 (11%) of the packing density values to less than the packing density threshold and 442 (89%) of the packing density values to be

greater than or equal to the packing density threshold. When applying the packing density threshold to the packing density values that had corresponding sequence entropy values greater than the tertiary contact threshold, the results showed 14 (45%) of the packing density values to be less than the packing density threshold and 17 (55%) of the packing density values to be greater than or equal to the packing density threshold. For these iterations of packing density values, frequency distribution analysis was performed for the corresponding RSA values and is shown in Figure 4.24A and Figure 4.24B.

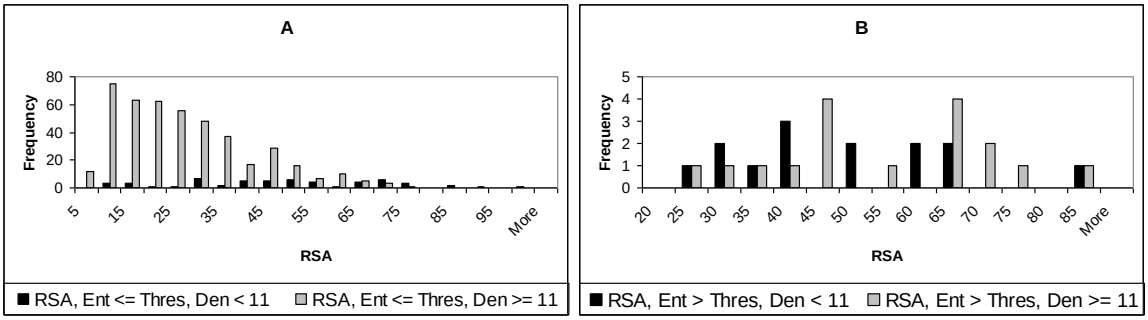

Figure 4.24: Frequency distribution plots of pair averaged RSA values where tertiary contact threshold value of 2.2285 was applied to corresponding sequence entropy values with subsequent application of packing density threshold value of 11. There are a total of 527 pair averaged query residues from the 75 protein set with known tertiary contacts. A. Aggregate frequency distribution plot of pair averaged RSA values where corresponding sequence entropy values are less than or equal to tertiary contact threshold with packing density threshold applied. B. Aggregate frequency distribution plot of pair averaged RSA values where corresponding sequence entropy values are greater than tertiary contact threshold with packing density threshold applied.

When applying the packing density threshold to the non-averaged packing density values that had corresponding sequence entropy values less than the tertiary contact threshold (Figure 4.25), the results showed 2544 (16%) of the packing density values to be less than the packing density threshold and 12979 (84%) of the packing

density values to be greater than or equal to the packing density threshold (Figure 4.25A). When applying the packing density threshold to the packing density values that had corresponding sequence entropy values greater than the tertiary contact threshold, the results showed 1023 (28%) of the packing density values to be less than the packing density threshold and 2612 (72%) of the packing density values to be greater than or equal to the packing density threshold (Figure 4.25B).

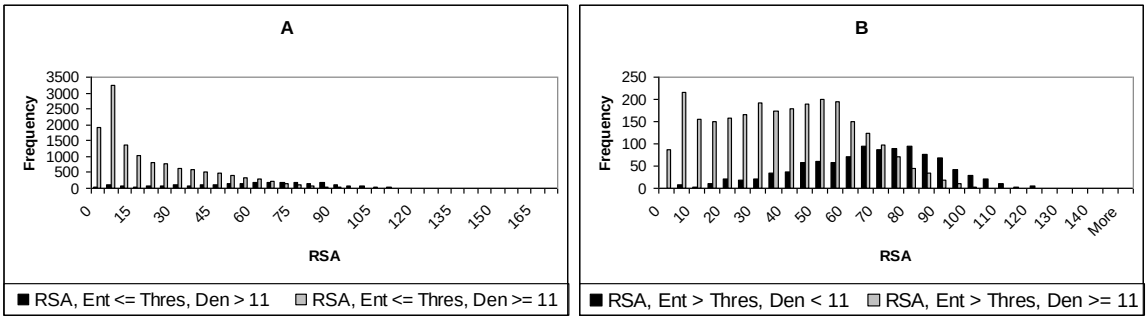

Figure 4.25: Frequency distribution plots of non-averaged RSA values where tertiary contact threshold value of 2.2285 was applied to corresponding sequence entropy values with subsequent application of packing density threshold value of 11. There are a total of 19158 query residues from the 75 protein set with known tertiary contacts. A. Aggregate frequency distribution plot of non-averaged RSA values where corresponding sequence entropy values are less than or equal to tertiary contact threshold with packing density threshold applied. B. Aggregate frequency distribution plot of non-averaged RSA values where corresponding sequence entropy values are greater than tertiary contact threshold with packing density threshold applied.

When applying the packing density threshold to the all-pair averaged packing density values that had corresponding sequence entropy values less than the tertiary contact threshold, the results showed 266990 (8%) of the packing density values to be less than the packing density threshold and 2902953 (92%) of the packing density values to be greater than or equal to the packing density threshold (Figure 4.26A). When

applying the packing density threshold to the packing density values that had corresponding sequence entropy values greater than the tertiary contact threshold, the results showed 69416 (18%) of the packing density values to be less than the packing density threshold and 317331 (82%) of the packing density values to be greater than or equal to the packing density threshold (Figure 4.26B). For these iterations of packing density values, frequency distribution analysis was performed for the corresponding RSA values and is shown in Figure 4.26.

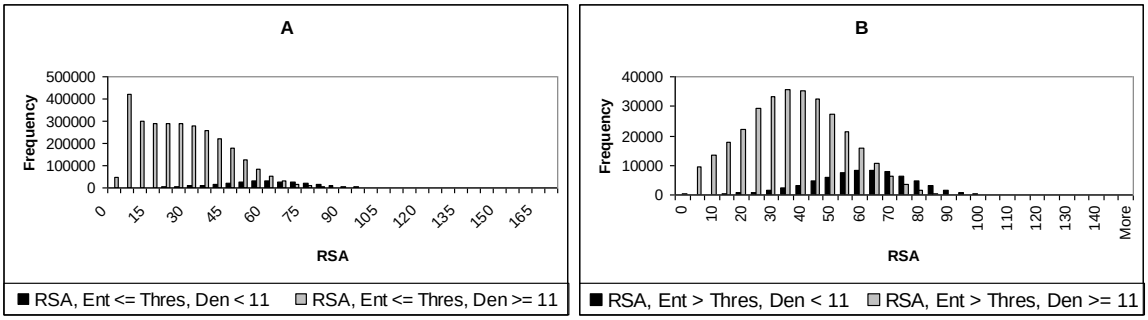

Figure 4.26: Frequency distribution plots of all-pair averaged RSA values where tertiary contact threshold value of 2.2285 was applied to corresponding sequence entropy values with subsequent application of density threshold value of 11. There are a total of 3556690 query residues from the 75 protein set with known tertiary contacts. A. Aggregate frequency distribution plot of all-pair averaged RSA values where corresponding sequence entropy values are less than or equal to tertiary contact threshold with density threshold applied. B. Aggregate frequency distribution plot of all-pair averaged RSA values where corresponding sequence entropy values are greater than tertiary contact threshold with packing density threshold applied.

Chapter 5

Discussion

5.1 Tertiary Contacts: More Conserved, Densely Packed, and Low Surface Accessibility

From the aggregate correlation plots of sequence entropy and inverse packing density it can clearly be seen that the tertiary contact values, both 10-separated pair averaged tertiary contacts and pair averaged tertiary contacts, generally have lower sequence entropy values. This can mean that tertiary contacts are more conserved within the protein sequence. From the aggregate plot it can also be seen the tertiary contact values are found to be on the lower range of the inverse packing density values which can mean that they are more densely packed and can be found within the core of a protein (Do, S.; Lustig, B. San Jose State University. Unpublished work, 2010).

From the aggregate correlation plot of sequence entropy and RSA it can be seen that the tertiary contact values also have lower sequence entropy values when compared to the non-averaged and all-pair averaged homology-based values. More importantly the tertiary contact values (both 10-separated pair averaged and pair averaged) are found on the lower portion of the range of RSA values. This can mean that tertiary contacts are less accessible to the surface of proteins, or solvents interacting with the surface of proteins, and are found buried within the core of proteins. Along with the instances that

tertiary contacts are more conserved in protein sequences and more densely packed in the core of a protein, this is consistent with the notion that tertiary contacts play an important part in maintaining protein structure, folding, and stability (Daggett and Fersht, 2003).

It can also be seen from both aggregate correlation plots of sequence entropy and inverse packing density and sequence entropy and RSA that the sequence entropy values from the all-pair averaged learning set is also lower then the non-averaged learning set. This affirms that the pair averaging of sequence entropy, packing density, and RSA of tertiary contact pairs (and ultimately all possible pairs of amino acids) appears to be a legitimate method of analysis. Moreover, the averaged values of these parameters may be used as a constraint to simulate the protein folding process and can indicate that the burial of each amino acid residue and its contacts to the surrounding residues is optimized during folding (Bahadur and Chakrabarti, 2009).

5.2 Frequency Distribution Analysis

When the tertiary contact threshold value of 2.2285 (which is a pair averaged sequence entropy tertiary contact value) was applied to the learning sets, significant separation was shown in the filtering of buried versus surface residues. The tertiary contact threshold value was determined from the 10-separated pair averaged tertiary contacts learning set, where frequency distribution analysis was performed for sequence entropy, packing density, and RSA. The frequency distribution of the packing density values shows a Gaussian-like distribution and validates that the learning set is consistent

with larger learning sets previously determined to be a diverse representation of proteins (Mishra, 2010; Rose et al., 2011). This also suggests an equally weighted binary distribution of amino acid residues that are on the surface of a protein (typically less dense) and directly in the center of a protein (typically most dense) and an average distribution of residues that are buried but not directly in the center or core (average density). The normal distribution of packing density values is logical because the majorities of amino acid residues are buried, and on the tail ends of the distribution are the surface and core residues. This methodology of average packing density does not present ambiguities because it does not depend on optimization methods (Moret et al., 2006). The frequency distribution of the sequence entropy and RSA values are right-skewed, which can mean that the majority of amino acid residues are buried and/or more conserved within the protein (Liao et al., 2005). Also, the frequency distribution of the sequence entropy and RSA for the pair averaged tertiary contacts, all-pair averaged, and non-averaged values are also right-skewed, showing that most residues are conserved and have low surface accessibility.

When the tertiary contact threshold value of 2.2285 was compared to the sequence entropy values of the learning sets (the two sets of pair averaged tertiary contacts and all-pair averaged) it was shown that a significant portion of the sequence entropy values are less than or equal to the tertiary contact threshold value. Since tertiary contacts are generally more conserved within the protein sequence it is consistent that the majority of the sequence entropy values are below the tertiary contact threshold due to protein amino acid residues mainly being conserved.

When applying the tertiary contact threshold value to the sequence entropy values of the learning sets, the corresponding packing density and RSA values were also separated along with the sequence entropy values. With the learning sets separated by the tertiary contact sequence entropy threshold, the next step was to apply the RSA threshold of 20.0. When applying the RSA threshold for the values that were less than or equal to the tertiary contact sequence entropy threshold, no trend was noted among all learning sets. For example, the pair averaged tertiary contact values showed little differentiation via RSA threshold (55% were less than or equal vs. 45% that were greater), while the all-pair averaged learning set showed just incrementally more values that were greater than the RSA threshold (57% were greater vs. 43% that were less than or equal). Still, no preference is indicated for buried tertiary contacts with respect to their exposure to solvent. It is conceivable that higher levels of packing arrangements (i.e. tertiary and quaternary) can exist despite an absence of preferred binary interactions, or tertiary contacts (Behe et al., 1991).

Lastly, when the packing density threshold value of 11 was applied to the corresponding RSA values, what was shown was somewhat counter-intuitive. The corresponding RSA values that were less than the packing density threshold were often on the lower spectrum of the range of RSA values. This is counter-intuitive because lower RSA values, especially values under 20.0, are designated as buried residues, while low packing density values are inferred to be on the surface of a protein. This discrepancy is also seen for the corresponding RSA values that were greater than or equal to the packing density threshold where both homology-based values were on the higher

range of values. It has been proposed that it is necessary that certain densely packed protein regions must also be considered as inaccessible to the surface (Moreira et al., 2007). However, conservation, or sequence entropy, at each amino acid position can be treated as an independent random value even with respect to packing density values that depend solely on solvent accessibility at a position (Mirny and Shaknovich, 1999), thus validating this set of sequence entropy values. Still, this phenomenon is quite intriguing and is a definite area of future research.

5.3 Tertiary Contacts as a Protein Structure Prediction Filter

The RSA threshold of 20.0 was also applied to the sequence entropy and corresponding RSA values of the learning sets that were greater than the tertiary contact threshold, and the results showed quite robust trends. When the RSA threshold was applied to the learning sets (pair averaged tertiary contacts, all-pair averaged, and non-averaged), there was an average where approximately 20% of the values were less than or equal to RSA threshold and 80% of the values were greater than the RSA threshold. An RSA value that is greater than 20.0 (Carugo, 2000) means that an amino acid residue is considered to be on the surface of a protein. Also, amino acid residues with sequence entropy values that are greater than the tertiary contact threshold means that these residues may not be conserved throughout the protein sequence, and it can be inferred that these residues are also on the surface of a protein. What is astounding is that for the sequence entropy values that are greater than the tertiary contact threshold, an

overwhelming percentage of approximately 80%, as stated earlier, of the corresponding RSA values are also greater than the RSA threshold. A regression-based first stage method of the Lustig group was an ideal platform for a first-stage RSA predictor for a standard 215 protein test set (Rose et al., 2011). Calculations using the tertiary contact threshold value of 2.2285 as a second stage filter to predict surface residues showed limited improvement in prediction accuracy (Nepal, R.; Lustig, B. San Jose State University. Unpublished work, 2012). However, a more intensive approach is being formulated.

One of the common problems in protein structure prediction is the lack of information on long range interactions, in this case tertiary contacts, but with preliminary knowledge of long range interactions or at least some limited information on them, the number of possible three-dimensional protein folds can greatly be reduced and thus predicted more accurately (Dosztanyi et al., 1997). Here, the tertiary contact threshold value has been utilized to determine, with high accuracy, amino acid residues that are on the surface of a protein. These results demonstrate that the tertiary contact threshold can be used as a valid prediction filter for whether amino acid residues are buried or on the surface of a protein.

Chapter 6

Future Studies

Since tertiary contacts are a relatively new phenomenon in protein structure prediction, there are many avenues that can be taken to further understand the nature of tertiary contacts. Further research of tertiary contacts can include:

- The search for tertiary contacts throughout the literature should be performed more extensively and comprehensively to increase the protein learning set.

- The tertiary contact threshold filter should be applied to other protein structure prediction methods, including methods involving secondary and quaternary structure.

- Tertiary contacts can be further classified in relation to the types of amino acids.

- Explore the peculiar phenomenon where the corresponding RSA and packing density values indicate that an amino acid residue is both buried and on the surface of a protein.

References

Adamczak, R.; Porollo, A.; Meller, J. *Proteins: Struct., Funct., Bioinf.* **2004**, *56*, 753-767.

Adamczak, R.; Porollo, A.; Meller, J. *Proteins: Struct., Funct., Bioinf.* **2005**, *59*, 467-475.

Ahmad, S.; Gromiha, M. M.; Sarai, A. *Proteins: Struct., Funct., Bioinf.* **2003**, *50*, 629-635.

Bahadur, R. P.; Chakrabarti, P. *BMC Struct. Biol.* **2009**, *9*, 1472-6807.

Behe, M. J.; Lattman, E. E.; Rose, G. D. *Proc. Natl. Acad. Sci. U. S. A.* **1991**, *88*, 4195-4199.

Bonneau, R.; Baker, D. *Annu. Rev. Biophys. Biomol. Struct.* **2001**, *30*, 173-189.

Bramucci, E.; Paiardini, A.; Bossa, F.; Pascarella, S. *BMC Bioinformatics* **2012**, *13*, 1-6.

Carugo, O. *Protein Eng., Des. Sel.* **2000**, *13*, 607-609.

Chen, H.; Zhou, H.; Hu, X.; Yoo, I. *Conferences in Research and Practice in Information Technology: 2nd Asia-Pacific Bioinformatics Conference (APBC2004)*, Dunedin, New Zealand, **2004**.

Daggett, V.; Fersht, A. R. *Trends Biochem. Sci.* **2003**, *28*, 18-25.

Dale, G. E.; Oefner, C.; D'Arcy, A. *J. Struct. Biol.* **2003**, *142*, 88-97.

DeSantis, T. Z.; Hugenholtz, P.; Keller, K.; Brodie, E. L.; Larsen, N.; Piceno, Y. M.; Phan, R.; Andersen, G. L. *Nucleic Acids Res.* **2006**, *34*, 394-399.

Dosztanyi, Z.; Fiser, A.; Simon, I. *J. Mol. Biol.* **1997**, *272*, 597-612.

Elcock, A. H.; McCammon, J. A. *Proc. Natl. Acad. Sci. U. S. A.* **2001**, *98*, 2990-2994.

Frishman, D.; Argos, P. *Proteins: Struct., Funct., Bioinf.* **1997**, *27*, 329-335.

Gerstein, M.; Altman, R. B. *J. Mol. Biol.* **1995**, *251*, 161-175.

Hubbard, S. J.; Thornton, J. M. NACCESS, Computer Program, Department of Biochemistry and Molecular Biology, University College London., http://www.bioinf.manchester.ac.uk/naccess/, 1993.

Kallblad, P.; Dean, P. M. *Proteins: Struct., Funct., Bioinf.* **2004**, *56*, 693-703.

Kim, H.; Park, H. *Proteins: Struct., Funct., Bioinf.* **2004**, *54*, 557-562.

Liang, J.; Dill, K. A. *Biophys. J.* **2001**, *81*, 751-766.

Liao, H.; Yeh, W.; Chiang, D.; Jernigan, R. L.; Lustig, B. *Protein Eng., Des. Sel.* **2005**, *18*, 59-64.

Mirny, L. A.; Shakhnovich, E. I. *J. Mol. Biol.* **1999**, *00*, 1-19.

Mishra, R. P. M. S. Thesis, San Jose State University, San Jose, CA, 2010.

Montelione, G. T.; Anderson, S. *Nature Struct. Biol.* **1999**, *6*, 11-12.

Moreira, I. S.; Fernandes, P. A.; Ramos, M. J. *J. Phys. Chem. B* **2007**, *111*, 2697-2706.

Moret, M. A.; Santana, M. C.; Nogueira, E. Jr.; Zebende, G. F. *Physica A* **2006**, *361*, 250-254.

Moret, M. A.; Zebende, G. F. *Phys. Rev. E: Stat., Nonlinear, Soft Matter Phys.* **2007**, *75*, 011920.

Naderi-Manesh, H.; Sadeghi, M.; Arab, S.; Movahedi, A. A. M. *Proteins: Struct., Funct., Bioinf.* **2001**, *42*, 452-459.

National Center for Biotechnology Information (NCBI), Protein Blast (BLASTP), http://www.ncbi.nlm.nih.gov/. Accessed 2012.

Needleman, S. B.; Wunsch, C. D. *J. Mol. Biol.* **1970**, *48*, 443-453.

Nicholas Jr., H. B.; Deerfield II, D. W.; Ropelewski, A. J. *BioTechniques* **2000**, *28*, 1174-1191.

Richardson, C. J.; Barlow, D. J. *Protein Eng., Des. Sel.* **1999**, *12*, 1051-1054.

Rodionov, M. A.; Blundell, T. L. *Proteins: Struct., Funct., Bioinf.* **1998**, *33*, 358-366.

Romero, P.; Obradovic, Z.; Li, X.; Garner, E. C.; Brown, C. J.; Dunker, A. K. *Proteins: Struct., Funct., Bioinf.* **2001**, *42*, 38-48.

Rose, D. A.; Nepal, R.; Mishra, R.; Lau, R.; Gholizadeh, S.; Lustig, B. *Proceedings of the 22nd International Workshop on Database and Expert Systems Application* **2011**, 70-74.

Rose, G. D. *Annu. Rev. Biophys.* **1993**, *22*, 381-415.

Rost, B.; Sander, C. *Proteins: Struct., Funct., Bioinf.* **1994**, *20*, 216-226.

Rutgers, the State University of New Jersey and San Diego Supercomputer Center (SDSC) and Skaggs School of Pharmacy and Pharmaceutical Sciences. RCSB PDB Protein Data Bank. http://www.rcsb.org/pdb/home/home.do (06/01/12).

Sander, C.; Schneider, R. *Proteins: Struct., Funct., Bioinf.* **1991**, *9*, 56-58.

Shenkin, P. S.; Erman, B.; Mastrandrea, L. D. *Proteins: Struct., Funct., Bioinf.* **1991**, *11*, 297-313.

Smith, T. F.; Waterman, M. S. *J. Mol. Biol.* **1981**, *147*, 195-197.

Ting, K. H.; Jernigan, R. L. *J. Mol. Evol.* **2002**, *54*, 425-436.

Tisdal, J. Beginning Perl for Bioinformatics; O'Reilly Media: Sebastopol, CA, 2001.

Valdar, W. S. J. *Proteins: Struct., Funct., Bioinf.* **2002**, *48*, 227-241.

Wagner, M.; Adamczak, R.; Porollo, A.; Meller, J. *J. Comp. Biol.* **2005**, *12*, 355-369.

Wang, G.; Dunbrack Jr., R. L. *Bioinformatics* **2003**, *19*, 1589-1591.

Wang, J.; Lee, H.; Ahmad, S. *Proteins: Struct., Funct., Bioinf.* **2007**, *68*, 82-91.

Appendices

Appendix A Additional Tables

Table A.1: 102 protein set with known tertiary contacts. Listed for each protein is the protein chain identifier, number of query residues, and number of alignments. Note PDB ID and chains of proteins that are not included in the final 75 protein set (due to containing only hydrophobic interaction tertiary contacts) are *italicized*.

PDB ID	Chain	# Query Residues	# Alignments	PDB ID	Chain	# Query Residues	# Alignments
1A2K	A	127	401	1DIN	A	236	986
1A32	A	88	982	1E5M	A	416	1000
1A48	A	306	1001	1EEH	A	437	1008
1A4I	A	301	1001	*1ES6*	*A*	n/a	n/a
1A6Q	A	382	1064	*1IHP*	*A*	n/a	n/a
1AA7	*A*	158	1000	*1LS9*	*A*	n/a	n/a
1ADE	A	431	1000	*1M6J*	*A*	n/a	n/a
1AF3	A	196	513	*1MSO*	*A*	n/a	n/a
1AFW	*A*	393	1001	*1OKI*	*A*	n/a	n/a
1AG9	A	175	747	1RBP	A	182	289
1AK0	*A*	270	213	*1UGM*	*A*	n/a	n/a
1AK4	C	145	1000	*1WQ5*	*A*	n/a	n/a
1AL8	*A*	359	1008	2ACY	A	98	796
1AMK	A	251	1000	*2ARC*	*A*	164	130
1AMP	A	291	1001	*2ATJ*	*A*	308	1002
1AMU	A	563	2309	*2BLS*	*A*	358	1000
1AOB	A	265	1007	2G3P	A	225	219
1AQ6	A	253	1000	2HDH	A	293	1195
1ATL	A	202	1005	*2IHL*	*A*	129	916
1AUO	A	218	915	2ILK	A	160	240
1AVW	*B*	177	342	2JEL	P	85	1017
1AW5	*A*	340	1000	*2LIG*	*A*	164	222
1AW7	A	194	86	2LIV	A	344	1004
1AW9	A	216	1000	*2MBR*	*A*	340	970
1AYX	*A*	492	161	2OHX	A	374	1005
1B3A	A	67	607	2RN2	A	155	1000
1B5E	A	246	563	2SCP	A	174	447
1B67	A	68	312	2SHP	A	525	1577
1B8A	A	438	1122	2SIC	I	107	55
1BAM	A	213	15	2SQC	A	631	970

1BBH	A	131	162	2TCT	A	207	1002
1BD0	A	388	1000	2TGI	A	112	1000
1BEA	A	127	340	*2UGI*	*A*	84	3
1BF2	A	750	1029	*3CLA*	*A*	213	303
1BIA	A	321	1000	3GRS	A	478	1001
1BIQ	A	375	1001	3PFK	A	319	1222
1BIS	*A*	166	1000	3PMG	A	561	1004
1BJW	A	327	1002	3RN3	A	124	626
1BMD	A	110	106	3SDH	A	146	633
1BN6	*A*	294	1002	3SGB	E	185	349
1BRS	A	110	106	3SGB	I	56	572
1BRW	A	433	849	4DRF	A	159	1000
1BT3	A	345	1046	4HTC	I	65	35
1BUL	*A*	265	1000	5CPA	A	307	1075
1BXQ	A	323	1004	5CSM	A	256	87
1CB0	A	283	989	6LDH	A	330	1001
1CEX	A	214	272	8ATC	A	310	1000
1CJX	A	357	615	*8PRK*	*A*	287	1001
1CRC	A	105	1012	8PTI	A	58	1543
1DCS	A	311	636	9PAP	A	212	1001
1DHT	A	327	1001	9WGA	A	171	1556

Table A.2: Characterization of the tertiary contacts for the 75 protein set. Listed for each protein is the protein chain identifier, tertiary contact amino acid and position, and pair averaged RSA value for each tertiary contact. Also listed is whether the tertiary contact is separated by at least 10 residues and whether they are less than or equal to an RSA value of 20.0.

PBD ID (PDB file reference(s))	Chain	# Tertiary contacts, AA and position	Pair averaged RSA value	# Residues between tertiary contact $\geq$ 10	Pair averaged RSA value $\leq$ 20
1A2K (primary)	A	T42-H124 E42-R76 K71-D92 K71-D94	6.35 9.5 27.35 27.3	Yes Yes Yes Yes	Yes Yes No No
1A32 (primary)	A	D48-S51 D73-R76 R76-E25	11.55 36.55 29.7	No No Yes	Yes No No
1A48 (primary)	A	L231-R73 Y258-F243 E47-V278 D239-K53	18.65 4.8 36.05 17.9	Yes Yes Yes Yes	Yes Yes No Yes

		D239-K260 D259-K260	23.65 7.45	Yes No	No Yes
1A4I (primary)	A	R137-D183	6.1	Yes	Yes
1A6Q (primary)	A	R33-H62 R33-R186	30.45 42.25	Yes Yes	No No
1ADE (primary, 1)	A	Q A171-Y A176 Y B167-Y A176 H A232-T B250 R A257-S B323 R A317-D B203 R A147-D B231 E A101-R A147 K A140-D B231	13.75 14.75 19.85 18.8 16.45 5.2 7.75 12.5	No No Yes Yes Yes Yes Yes Yes	Yes Yes Yes Yes Yes Yes Yes Yes
1AF3 (primary)	A	Y22-D156 R165-P116 D11-R91 S2-N175 N5-N175 N5-E179 E7-K87 D11-R91 D11-K87 Y15-D95 K16-D95 K16-E98 Q19-D95 Y22-D156 Q26-S164 E98-S145 Q111-E129 P116-R165 E124-H177 N128-Y173	45.6 46.95 20.75 55.55 15.65 7.05 37.3 20.75 18.45 28.25 16.4 15.45 39.5 45.6 30.85 15.75 42.4 46.95 31.95 21.45	Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes	No No No No Yes Yes No No Yes No Yes Yes No No No Yes No No No No
1AG9 (primary)	A	W56-Y57 E60-A61	25 19.15	No No	No Yes
1AK4 (primary, 1)	C	A88-I91 R55-P90 P1-D51	23.5 28.5 47.55	No Yes Yes	No No No
1AMK (primary, 1)	A	S96-A64 A42-W12	6.1 0.8	Yes Yes	Yes Yes
1AMP (primary)	A	D117-D118 C223-C227	2.65 11.95	No No	Yes Yes

1AMU (primary, 1)	A	A322-N321 D413-Y425 Y323-D413	5.35 16 10.1	No Yes Yes	Yes Yes Yes
1AOB (primary)	A	Y209-A144 H207-Y94 D169-H147	26.45 6.45 14.4	Yes Yes Yes	No Yes Yes
1AQ6 (primary, 1)	A	F175-F146 F175-Y10 R39-N173	1.35 3.1 8.15	Yes Yes Yes	Yes Yes Yes
1ATL (primary, 1)	A	L108-G109	11.85	No	Yes
1AUO (primary)	A	H199-D69	16.5	Yes	Yes
1AW7 (primary, 1, 2)	A	T128-N65 H135-Y13 G16-S15 L113-K114 Q139-T138 Q139-R145	7.45 19.7 34.75 48.65 16.05 18.9	Yes Yes No No No No	Yes Yes No No Yes Yes
1AW9 (primary)	A	R72-E96 Q53-H105 S103-H104	47.6 60.65 39.2	Yes Yes No	No No No
1B3A (primary)	A	S5-C50	27.05	Yes	No
1B5E (primary)	A	H216-Y218 D4-D4 E9-E9 E10-E10 E21-E21 D23-D23 D44-D44 E45-E45 D121-D121 D158-D158	8.2 65.9 50.3 41.8 71.8 37.7 56.9 27.5 36 95.4	No No No No No No No No No No	Yes No No No No No No No No No
1B67 (primary)	A	R19-K53 R19-I55 S21-I55	54.2 31.85 13.6	Yes Yes Yes	No No Yes
1B8A (primary)	A	R412-A227 E361-I362 R214-R368 R214-K195 R412-H223	16.2 14.15 19.6 19.65 22.85	Yes No Yes Yes Yes	Yes Yes Yes Yes No
1BAM	A	F-159-L162	1.1	No	Yes

(primary)		E163-Y165	44.75	No	No
		F166-T169	1	No	Yes
		M1-E182	21.75	Yes	No
		K5-E160	33.55	Yes	No
		K61-D94	22.65	Yes	No
		R76-E62	29.85	Yes	No
		K106-D19	12.15	Yes	Yes
		K106-E101	14.65	No	Yes
		R107-E98	31.35	No	No
		K126-E111	22.25	Yes	No
		K146-E161	54.1	Yes	No
		K207-D70	29.75	Yes	No
		K213-E211	55.8	No	No
		K132-E170	65.15	Yes	No
		K132-E167	68.65	Yes	No
		H133-E167	66.5	Yes	No
1BBH	A	T62-R12	1.1	Yes	Yes
(primary)		K25-S52	6.05	Yes	Yes
		E17-E10	40.25	No	No
		K122-E126	65.15	No	No
1BD0	A	M312-R136	9.15	Yes	Yes
(primary)		Y265-K39	23.45	Yes	No
1BEA	A	C6-C55	13	Yes	Yes
(primary)		L80-R79	16.6	No	Yes
		I62-L63	3.35	No	Yes
1BF2	A	Y250-D292	7.45	Yes	Yes
(primary)		D292-R373	0.5	Yes	Yes
		C520-C590	9.5	Yes	Yes
		C712-C740	12.95	Yes	Yes
		E124-R260	4.4	Yes	Yes
		D178-R633	17.55	Yes	Yes
		D716-R559	21.65	Yes	No
		D716-R559	21.65	Yes	No
		D716-R563	16.05	Yes	Yes
1BIA	A	C107-T90	1.75	Yes	Yes
(primary)					
1BIQ	A	Y208-D84	10.7	Yes	Yes
(primary)		D84-E204	9.2	Yes	Yes
1BJW	A	T36-N21	24.75	Yes	No
(primary)		K13-S15	53.25	No	No
		V18-N21	53.1	No	No
		N21-T36	24.75	Yes	No
1BMD	A	M154-V128	0	Yes	Yes

(primary)		A89-N130	17.6	Yes	Yes
		H186-N130	3.7	Yes	Yes
		D158-H186	2.6	Yes	Yes
		D158-R161	3.5	No	Yes
		R149-E275	28.4	Yes	No
		Y17-A243	12.2	Yes	Yes
		E27-K31	10.45	No	Yes
		E57-K168	0	Yes	Yes
		E27-K31	10.45	No	Yes
		E57-K168	0	Yes	Yes
		E275-R149	28.4	Yes	No
		E27-R22	6.15	No	Yes
		E27-K31	10.45	No	Yes
		E27-K31	10.45	No	Yes
		E27-K31	10.45	No	Yes
		E27-K31	10.45	No	Yes
		R149-E275	28.4	Yes	No
1BRS (primary)	A	R59-E76	10.7	Yes	Yes
		H102-D35	3.3	Yes	Yes
		N33-H102	27.05	Yes	No
		H102-D39	27.05	Yes	No
		H102-G31	30.15	Yes	No
		Y29-N84	36.55	Yes	No
		G31-R83	32.6	Yes	No
		N33-H102	27.05	Yes	No
		L34-H102	23.3	Yes	No
		D35-E60	22.45	Yes	No
		D39-R59	37.65	Yes	No
		D39-R83	29.7	Yes	No
		T42-R83	3.4	Yes	Yes
		G43-R87	10.45	Yes	No
		E76-H102	0.3	Yes	Yes
1BRW (primary)	A	H82-S83	15.7	No	Yes
		H116-G205	14.35	Yes	Yes
		E255-T90	3.45	Yes	Yes
1BT3 (primary)	A	C11-C28	18.4	Yes	Yes
		C27-C89	0.35	Yes	Yes
1BXQ (primary, 1, 2)	A	C249-C283	23.8	Yes	No
1CB0 (primary)	A	D220-D222	5.65	No	Yes
		T118-H137	74.85	Yes	No
		T118-H65	67	Yes	No
1CEX	A	S42-N84	14.9	Yes	Yes

(primary)		S42-Q121	10.15	Yes	Yes
		S120-N84	8.2	Yes	Yes
		C31-C109	18.9	Yes	Yes
		C171-C178	16.9	No	Yes
1CJX (primary)	A	D160-R326	10.15	Yes	Yes
1CRC (primary)	A	R91-E69	21.6	Yes	No
		H18-P30	13.95	Yes	Yes
		M80-Y67	19.6	Yes	Yes
		K27-A15	35.05	Yes	No
		K27-Q16	49.15	Yes	No
		R91-K86	37.1	No	No
		R91-M65	13.4	Yes	Yes
1DCS (primary)	A	M180-V262	18.7	Yes	Yes
		R258-S260	9.7	No	Yes
1DHT (primary)	A	K159-Y155	14.5	No	Yes
1DIN (primary)	A	Y144-T224	11.25	Yes	Yes
		E101-Y137	13.1	Yes	Yes
		S10-R66	18.65	Yes	Yes
		Q35-E36	3.65	No	Yes
		T224-H166	10.45	Yes	Yes
		Y197-H166	29.2	Yes	No
		W196-T183	34.35	Yes	No
		S49-R45	42.9	No	No
		E36-R206	36.8	Yes	No
		R206-S208	41.9	No	No
		S203-H202	5.1	No	Yes
		C123-H202	8.7	Yes	Yes
		C123-L124	3.6	No	Yes
		H202-D171	54.65	Yes	No
		H202-C123	8.7	Yes	Yes
		E36-C123	7.25	Yes	Yes
		D99-R66	16.4	Yes	Yes
1E5M (primary)	A	H307-K339	0.9	Yes	Yes
		C167-G170	0.25	No	Yes
		L346-G350	1.7	No	Yes
		G343-L345	0.35	No	Yes
		H307-R401	7.55	Yes	Yes
		K339-H307	0.9	Yes	Yes
		K339-H307	0.9	Yes	Yes
		K339-E353	0.35	Yes	Yes
		E195-G111	11.1	Yes	Yes
		E195-I112	17.25	Yes	Yes

		E195-G347	0	Yes	Yes
1EEH (primary)	A	R186-P72	18.3	Yes	Yes
1RBP (primary)	A	C4-C160	36.65	Yes	No
		C70-C174	27.45	Yes	No
		C120-C129	15.9	No	Yes
		T109-Y114	3.45	No	Yes
		T109-Y111	5.45	No	Yes
		D110-T113	32.45	No	No
		D110-Y114	22.6	No	No
		Y111-D16	25	Yes	No
2ACY (primary)	A	F14-R77	39.75	Yes	No
		Q18-T46	23.2	Yes	No
		R21-Y25	62.05	No	No
2G3P (primary)	A	C7-C36	0	Yes	Yes
		C46-C53	0	No	Yes
2HDH (primary)	A	R209-E117	50.05	Yes	No
2ILK (primary)	A	Q38-T100	56.45	Yes	No
		K57-E142	60.35	Yes	No
2JEL (primary)	P	S41-H76	32.5	Yes	No
		H15-N12	60.5	No	No
2LIV (primary)	A	C53-C78	20.5	Yes	No
2OHX (primary)	A	I368-T347	21	Yes	No
		R369-L345	2	Yes	Yes
		C132-K135	25.95	No	No
		A278-S265	0.4	Yes	Yes
		L307-R312	15.3	No	Yes
		C46-R47	22.75	No	No
		S48-H51	16.2	No	Yes
		D49-H67	0.85	Yes	Yes
		C46-D49	2.3	No	Yes
		C46-S48	11.7	No	Yes
		G181-K185	19.6	No	Yes
		G181-V186	5.85	No	Yes
		R47-D50	24.8	No	No
2RN2 (primary)	A	R138-D134	44.1	No	No
		Q4-K117	45.55	Yes	No
		E6-R27	24.05	Yes	No
		E6-R27	24.05	Yes	No
		T9-T69	2.45	Yes	Yes
		T9-T69	2.45	Yes	Yes

		D10-G11	13.05	No	Yes
		D10-G11	13.05	No	Yes
		C13-G15	27.35	No	No
		C13-G18	12.55	No	Yes
		C13-T42	7.8	Yes	Yes
		C13-N44	28.45	Yes	No
		R27-E32	36.1	No	No
		R27-E129	21.55	Yes	No
		R27-E129	21.55	Yes	No
		Y28-K60	46.2	Yes	No
		Y28-E61	26.05	Yes	No
		Y28-E61	26.05	Yes	No
		E32-R132	42.5	Yes	No
		T42-D148	4.5	Yes	Yes
		R46-N100	7.15	Yes	Yes
		R46-D102	20.25	Yes	No
		R46-D102	20.25	Yes	No
		R46-D148	5.45	Yes	Yes
		R46-D148	5.45	Yes	Yes
		R46-D148	5.45	Yes	Yes
		E48-S71	9.15	Yes	Yes
		E57-R106	43.45	Yes	No
		E57-R106	43.45	Yes	No
		E61-C63	27.65	No	No
		E61-H114	28.95	Yes	No
		H62-Q113	64.85	Yes	No
		S68-E119	17.4	Yes	Yes
		S68-N130	5.4	Yes	Yes
		T69-D70	21.1	No	No
		T69-S71	4.3	No	Yes
		S71-V74	4.05	No	Yes
		T73-N100	11.75	Yes	Yes
		T73-W104	5.55	Yes	Yes
		R75-T79	54.1	No	No
		Q76-Q80	46.6	No	No
		T79-Q80	58.8	No	No
		Q80-N84	57.15	No	No
		K86-D108	38.8	Yes	No
		T92-D94	52.5	No	No
		T92-K96	41.65	No	No
		T92-K96	41.56	No	No
		A110-H114	12.5	No	Yes
		E119-H127	35.85	No	No
		H127-N130	23.85	No	No

		D134-R138	44.1	No	No
		D148-G150	14.8	No	Yes
		I78-H83	42.8	No	No
		S36-L136	33.55	Yes	No
		E57-R106	43.45	Yes	No
		D10-D70	27.65	Yes	No
		R41-D148	30	Yes	No
		R41-D148	30	Yes	No
		Y73-K99	33.4	Yes	No
		T43-N100	14.65	Yes	Yes
		T9-E48	7.3	Yes	Yes
		Q80-W81	43.5	No	No
		R46-N100	7.15	Yes	Yes
		R46-D102	20.25	Yes	No
		R46-D148	5.45	Yes	Yes
		R27-E6	24.05	Yes	No
		R27-E32	36.1	No	No
		R27-E129	21.55	Yes	No
		E32-R132	42.5	Yes	No
		S68-N130	5.4	Yes	Yes
		E119-H127	35.85	No	No
		W104-Y73	5.55	Yes	Yes
		H62-Q113	64.85	Yes	No
		H127-E119	35.85	No	No
		H114-A110	12.5	No	Yes
		H114-E61	28.95	Yes	No
2SCP (primary)	A	S52-N59	4.7	No	Yes
		D58-R101	16.05	Yes	Yes
		D58-R101	16.05	Yes	Yes
		N59-S52	4.7	No	Yes
		N85-E148	35.7	Yes	No
		S90-E93	1.55	No	Yes
		R101-D58	16.05	Yes	Yes
		E148-N85	35.7	Yes	No
		T24-F12	9.8	Yes	Yes
		D71-Y11	23.05	Yes	No
		D61-F12	24.55	Yes	No
2SHP (primary)	A	D61-D459	10.35	Yes	Yes
2SIC (primary)	I	H64-D32	20.4	Yes	No
2SQC (primary)	A	W406-W351	1.25	Yes	Yes
		W558-F616	5.15	Yes	Yes
		D374-D377	4.35	No	Yes

		E454-R623	14.1	Yes	Yes
2TCT (primary)	A	E114-L169	36.6	Yes	No
		R128-Q184	47.6	Yes	No
		Y132-Q184	26.2	Yes	No
		G102-E147	26.15	Yes	No
		G102-H151	44.9	Yes	No
		N18-I22	73.15	No	No
		R94-D95	5.05	No	Yes
		Q116-S67	31.55	Yes	No
		N82-Y66	10.15	Yes	Yes
		T103-H100	67.15	No	No
		H139-G182	55.15	Yes	No
		Y132-A185	1.8	Yes	Yes
		D178-R104	61.75	Yes	No
		D11-R62	16.75	Yes	Yes
2TGI (primary)	A	C7-C16	1.4	No	Yes
		C15-C78	3.25	Yes	Yes
		C44-C109	8.15	Yes	Yes
		C48-C111	2.9	Yes	Yes
3GRS (primary, 1)	A	C58-C63	26.05	No	No
		S373-P375	2.5	No	Yes
		T469-E472	28.4	No	No
3PFK (primary)	A	R252-H160	39.45	Yes	No
3PMG (primary)	A	H384-D389	0.1	No	Yes
		E376-S377	17.6	No	Yes
		S116-R22	36.85	Yes	No
		S116-R22	36.85	Yes	No
		H117-S116	31.65	No	No
		H117-R292	12.45	Yes	Yes
3RN3 (primary)	A	C26-C84	0	Yes	Yes
		C40-C95	13.6	Yes	Yes
		C58-C110	3.4	Yes	Yes
		C64-C72	2.55	No	Yes
		D121-H119	28.8	No	No
		N44-H12	6.9	Yes	Yes
		R10-E2	39.4	No	No
3SDH (primary)	A	Y75-D82	25.9	No	No
		K30-D89	18.65	Yes	Yes
		Y75-N79	2.45	No	Yes
		D89-K30	18.65	Yes	Yes
3SGB (primary)	E	C16-C24	41.3	No	No

3SGB (primary)	I	N36-N33 N33-T17	15.45 8.4	No Yes	Yes Yes
4DFR (primary)	A	R57-P55 N59-L54 N59-R57	34.5 7.35 8.25	No No No	No Yes Yes
4HTC (primary)	I	C39-P48 K47-T4 D5-N12 K47-T4 K47-D5 D6-D14 C16-C28 C16-C28 C22-C39	50.2 19.2 27.1 19.2 35.95 5.25 4.1 4.1 14.45	No Yes No Yes Yes No Yes Yes Yes	No Yes No Yes No Yes Yes Yes Yes
5CPA (primary)	A	C138-C161 E292-R272 Y238-E270 N188-K190 Y90-D101 N112-K128 N146-S172 D104-R59 Q76-E72 S41-E175 S70-N112 S254-S194 S258-S266 S266-S258 T75-E72 T129-S131 T129-V141 T119-W73 W147-A143 Y238-E270 H69-D142 D104-R59	43.3 5.7 12.95 35.35 39.85 0.85 9.45 13.15 2.15 2.65 0 1.2 0 0 1.8 7.4 1.9 7.9 0.45 12.95 2.25 13.15	Yes Yes Yes No Yes Yes Yes Yes No Yes Yes Yes No No No No Yes Yes No Yes Yes Yes	No Yes Yes No No Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes
5CSM (primary)	A	I239-I192 R16-E198 R157-E198 S226-E228 R75-R76 D24-K208 D24-R204	10.65 39.2 16.45 53.55 28.25 18.9 26.75	Yes Yes Yes No No Yes Yes	Yes No Yes No No Yes No
6LDH	A	D47-N22	6.4	Yes	Yes

(primary)		D53-V29	60.6	Yes	No
		D44-L41	36.55	No	No
		Q67-I78	46.9	Yes	No
		S88-Y85	43.4	No	No
		S139-A98	19.55	Yes	Yes
		R173-C187	11.9	Yes	Yes
		S169-V191	10.95	Yes	Yes
		N207-G205	53.5	No	No
		N207-S210	59.65	No	No
		N219-D221	50	No	No
		R267-D256	20.65	Yes	No
		N264-D295	53.8	Yes	No
		D5-K304	82.7	Yes	No
		N22-N22	5.2	No	Yes
		N22-N22	5.2	No	Yes
		D44-K263	29.25	Yes	No
		S247-D65	39.8	Yes	No
		D44-H74	6.45	Yes	Yes
		K58-E62	9.45	No	Yes
		D84-S86	23.05	No	No
		N140-H195	41.75	Yes	No
		Y147-K151	9.4	No	Yes
		D168-H195	64.6	Yes	No
		E194-S198	12.85	No	Yes
		S204-K308	28.65	Yes	No
		K222-D224	41.2	No	No
		D224-K227	12.1	No	Yes
		K243-S247	64.3	No	No
		E259-K263	26	No	No
		N264-D295	15	Yes	Yes
		R267-T260	53.8	No	No
		R267-T260	21.15	No	No
		K276-D277	49.9	No	No
		D295-N264	53.8	Yes	No
		S318-E194	36.55	Yes	No
8ATC (primary)	A	Q73-N105	14.55	Yes	Yes
		E101-V127	0	Yes	Yes
		R234-R167	11.1	Yes	Yes
		H134-S171	1.75	Yes	Yes
		E108-R113	21.8	No	No
		E239-K164	22.95	Yes	No
		E204-R130	34.05	Yes	No
		R128-E204	34.4	Yes	No
		R130-D200	10	Yes	Yes

		E117-K139	22.55	Yes	No
		E239-K164	22.95	Yes	No
8PTI (primary)	A	A58-Y23	24.85	Yes	No
		A58-C55	24.2	No	No
		I18-G35	37.5	Yes	No
		G35-I18	37.5	Yes	No
		G35-G12	25.15	Yes	No
		C14-G12	22.15	No	No
		I18-A16	49.5	No	No
		R39-G36	18.55	No	Yes
		A58-Y23	24.85	Yes	No
		A58-C55	24.2	No	No
9PAP (primary)	A	C22-C63	16.6	Yes	Yes
		D55-C95	13.2	Yes	Yes
		C153-C200	15.5	Yes	Yes
		V113-Q114	35.9	No	No
		E35-K174	3.45	Yes	Yes
		E50-K17	7.35	Yes	Yes
		E35-E50	2	Yes	Yes
		P2-Y166	29.1	Yes	No
		W7-I125	5.6	Yes	Yes
		N212-D108	81.5	Yes	No
		S29-V161	2	Yes	Yes
		Y61-Y67	45.05	No	No
		D108-N212	81.5	Yes	No
		N175-H159	4.8	Yes	Yes
		S97-E52	3.2	Yes	Yes
		K17-T14	8.75	No	Yes
		K17-P15	41.6	No	No
		K17-Q47	24.5	Yes	No
		R191-G167	9.45	Yes	Yes
		R191-Q118	19.8	Yes	Yes
		Q118-Y203	12	Yes	Yes
		T204-G201	12.25	No	Yes
		N212-R41	83.1	Yes	No
		R96-E89	32.05	No	No
		R191-D140	31.25	Yes	No
9WGA (primary)	A	C3-C18	0.6	Yes	Yes
		C12-C24	25.15	Yes	No
		C17-C31	0.1	Yes	Yes
		C35-C40	10.2	No	Yes
		C46-C61	1.25	Yes	Yes
		C55-C67	5.2	Yes	Yes
		C60-C74	3.4	Yes	Yes

		C78-C83 C89-C104 C98-C110 C103-C117 C121-C126 C132-C147 C141-C153 C146-C160 C164-C169	16.15 0 10.8 4 17.95 0.15 15.7 3.9 25.65	No Yes Yes Yes No Yes Yes Yes No	Yes Yes Yes Yes Yes Yes Yes Yes No

Table A.3: Characterization of the tertiary contacts that were not included for the 75 protein set. Listed for each protein is the protein chain identifier, tertiary contact amino acid and position, and pair averaged RSA value for each tertiary contact. Also listed is whether the tertiary contact is separated by at least 10 residues and whether they are less than or equal to an RSA value of 20.0.

PBD ID (PDB file reference(s))	Chain	# Tertiary contacts, AA and position	Pair averaged RSA value	# Residues between tertiary contact $\geq$ 10	Pair averaged RSA value $\leq$ 20
1AA7 (primary)	A	P A90-P B90 M A93- M B93 V A97- V B97	12.5 1.3 1.1	No No No	Yes Yes Yes
1AFW (primary)	A	T101-Q124 H375-T380	17.7 2.55	Yes No	Yes Yes
1AK0 (primary, 1)	A	F61-V132 Y144-Y155	46.5 58.45	Yes Yes	No No
1AL8 (primary, 1, 2)	A	F172-L161	77.75	Yes	No
1AVW (primary)	B	F64-S62 R65-Y151 K135-D95 K135-T98 R61-R174 R61-T175 D71-Y99	0.8 25 25.75 59.4 7.3 7.3 20.4	No Yes Yes Yes Yes Yes Yes	Yes No No No Yes Yes No
1AW5 (primary)	A	F46-F89	13.4	Yes	Yes
1AYX (primary)	A	Y63-E48 W67-E52 W139-E120 L208-E177 W209-E178	32.8 3.05 26.65 7.9 28.15	Yes Yes Yes Yes Yes	No Yes No Yes No

		Y351-E311	11.6	Yes	Yes
		W362-E317	4	Yes	Yes
		L471-E415	41.4	Yes	No
		W473-E417	5.65	Yes	Yes
1BIS	A	D64-D116	n/a	Yes	n/a
(primary)		D64-D116	n/a	Yes	n/a
1BN6	A	H289-D260	22.55	Yes	No
(primary)		H289-D260	22.55	Yes	No
		E141-V256	19.3	Yes	Yes
		E141-I258	23.4	Yes	No
		E141-H283	1.55	Yes	Yes
1BUL	A	E166-N170	33.55	No	No
(primary)		C69-C238	0.8	Yes	Yes
1ES6	A	Q184-D253	n/a	Yes	n/a
		Q184-K256	n/a	Yes	n/a
1IHP	A	C8-C17	n/a	No	n/a
		C48-C391	n/a	Yes	n/a
		C241-C259	n/a	Yes	n/a
		C413-C421	n/a	No	n/a
1LS9	A	G9-A77	n/a	Yes	n/a
		K53-E76	n/a	Yes	n/a
1M6J	A	H86-C14	n/a	Yes	n/a
1MSO	A	F1-E17	n/a	Yes	n/a
		F1-E17	n/a	Yes	n/a
		N3-S9	n/a	No	n/a
		Q4-L17	n/a	Yes	n/a
		C6-C11	n/a	No	n/a
		C7-C7	n/a	No	n/a
		C20-C19	n/a	No	n/a
1OKI	A	E116-P171	n/a	Yes	n/a
		K117-W174	n/a	Yes	n/a
		G118-W174	n/a	Yes	n/a
		Y130-N124	n/a	No	n/a
		P171-E116	n/a	Yes	n/a
		Y174-K117	n/a	Yes	n/a
		V175-K117	n/a	Yes	n/a
1UGM	A	I23-V20	n/a	No	n/a
		G85-H86	n/a	No	n/a
		R11-R16	n/a	No	n/a
		R10-T50	n/a	Yes	n/a
		H86-E102	n/a	Yes	n/a
		R16-D106	n/a	Yes	n/a
		R11-D19	n/a	No	n/a

1WQ5	A	D56-K167 D56-K167 Q65-S161 V133-Q19 E134-Q19 E135-Y8 E135-Y8 E135-M15 N157-I20 N157-Y181	n/a n/a n/a n/a n/a n/a n/a n/a n/a n/a	Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes	n/a n/a n/a n/a n/a n/a n/a n/a n/a n/a
2ARC (primary)	A	N154-Q158 E157-N154 Y31-Y82 Y82-W95 M42-I36	8.05 1.65 39.75 17.4 6.95	No No Yes Yes No	Yes Yes No Yes Yes
2ATJ (primary)	A	C18-C18	26.95	No	No
2BLS (primary)	A	S64-S318 S70-Q237 N346-T316	0.85 10.65 6.2	Yes Yes Yes	Yes Yes Yes
2IHL (primary)	A	Y94-G117 S31-N103 W33-N106 N101-Y23 Y102-Y23 D55-R112 D99-R112	54.25 34.2 30.3 30.7 36.05 25.85 24.35	Yes Yes Yes Yes Yes Yes Yes	No No No No No No No
2LIG (primary)	A	Y149-R64	50.7	Yes	No
2MBR (primary)	A	E325-R159	6.4	Yes	Yes
2UGI (primary)	A	I22-M56 L16-I18 V55-L57 P67-L70	13.6 16.5 0.05 2	Yes No No No	Yes Yes Yes Yes
3CLA (primary)	A	K10-E82 R18-D199 R18-D199 R26-D167 R26-D167 K38-D156 K45-D49 K45-D49	52.4 44.5 44.5 14.4 14.4 45.2 75.45 75.45	Yes Yes Yes Yes Yes Yes No No	No No No Yes Yes No No No

		D71-R74	51.55	No	No
		K72-R205	24	Yes	No
		K78-E142	12.25	Yes	Yes
		K78-E142	12.25	Yes	Yes
		R209-E212	33.1	No	No
		E101-R205	15.65	Yes	Yes
		N68-V89	6.35	Yes	Yes
		N68-V89	6.35	Yes	Yes
		Q137-S107	36.65	Yes	No
		Q137-S107	36.65	Yes	No
		Q192-L160	42.5	Yes	No
		Q192-L160	42.5	Yes	No
		Q211-D40	45.55	Yes	No
		Q211-D40	45.55	Yes	No
		R18-F195	59.8	Yes	No
		R18-H196	33.6	Yes	No
		N159-N161	32.35	No	No
8PRK (primary)	A	D71-R78	14.8	No	Yes

Appendix B Perl Program Listings

ftp-script-1.pl

```
# !/usr/bin/perl -w

use Net::FTP;

sub doFTP {

        my ($line1) = @_ ;
        print "arg recieved, $line1 \n";
        chomp ($line1);
        $line1 = lc ($line1);
        $subdir = substr ($line1, 1, 2);
        $destDir = "/pub/pdb/data/structures/divided/mmCIF/".$subdir;
        print "$destDir\n";
        $ftp->cwd ($destDir);
        $ftp->binary();
        $filetoftp = $line1."\.cif\.Z";
        print "$filetoftp\n";
        $ftp->get ($filetoftp, $filetoftp);
```

```
}

open (FHNDL, "SeqIDlearningset286.txt");
$line = <FHNDL>;
$ftp = Net::FTP->new ("ftp.rcsb.org");
$ftp->login ("anonymous", '-anonymous@');

# $ftp->cwd ("/pub/pdb/data/structures/divided/mmCIF");

while (!eof (FHNDL)) {

        doFTP ($line);
        $line = <FHNDL>;

}

doFTP ($line);
$ftp->quit;
close (FHNDL);
```

cif2den.pl

```
# Author: Radhika Pallavi Mishra
# Date: August 20, 2008
# Purpose: Takes mmcif file in ASCII text format and calculates density. Also identifies
# missing residues in ATOM section
# Adapted from pdb2denMOD-2 written by William Yeh
# User Specified Variables to Control Analysis and Output
# For each Value, calc's #dist <= Value

@TabValues = (6, 7, 8, 9, 10, 11, 12);

# Must be increasing in value.
# Defines which @TabValues printed

@TabPrint  = (0, 1, 2, 3, 4, 5, 6);

# 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'

);

if($#ARGV < 1) {

        print "USAGE : perl cif2den.pl <Chain Name i.e. A,B etc..> <Name of .cif file>
\n";
        exit;

}

$p_directory = "CIF_files";

# Initialize all variables
# PDB name (extracted from HEADER line), in lower case

$Name = "";
$Line = "";

# Temp var for current ATOM line as list

@AtomLine = ();

# 3-letter form from SEQRES, for checking

@AASeqRes = ();

# 3-letter form from ATOM statements

@AA = ();

# 1-letter form (translated)

@AA1 = ();

# Unused for now

@Tag = ();

# Calc'd distances
```

```
@Distance = ();
$AAref = '';

# $AArefI = 'A';

$Indexref = 0;
$Tagref = '';

# Number of aa's

$AACount = 0;
$Xref = 0;
$Yref = 0;
$Zref = 0;
$i = 0;
$j = 0;
$k = 0;
@X = ();
@Y = ();
@Z = ();
@XVect = ();
@YVect = ();

# Temp var for storing vector of aa being compared

@ZVect = ();

# TabCount tracks no. < each @TabValues

@TabCount = ();

# Calc density from TabCount & TabValues

@TabDensity = ();

# Temp var holds line for printing

@PrintLine = ();
$Count = 0;

open ( IN  ,"$p_directory/$ARGV[1]") or die "Cannot open input files for read"."\n";

# Indices for AA's
```

```
$i = 0;

while( <IN> ) {

# Save current line

        $Line = $_ ;
        chomp($Line);

# =====  Extract Amino Acid seq from SEQRES statements  =====

        if ($Line =~ /_pdbx_poly_seq_scheme\.pdb_ins_code/) {

                while((!eof(IN))&&($Line !~ /loop_/)) {

                        $Line = <IN> ;
                        chomp($Line);
                        @LineArray = split(/ +/,$Line);

                        if($LineArray[9] eq $ARGV[0]) {

                                push(@AASeqRes, $AADictionary{$LineArray[3]});
                                push(@FASTAPOS, $LineArray[4]);
                                push(@PDBPOSSEQRES, $LineArray[6]);

                        }

                }

        }

# =====  Extract alpha-C (x,y,z) from ATOM statements  =====

        if ($Line =~ /_atom_site\.pdbx_PDB_model_num/) {

                while((!eof(IN))&&($Line !~ /\#/)) {

                        $Line = <IN> ;
                        chomp($Line);

# Find alpha-Carbon ATOM lines

                        if($Line =~ /^ATOM +[\d]+ +[A-Z]+ +CA +/ ) {
```

```
                                @AtomLine = split(/ +/, $Line);
                                ($AAref, $Xref, $Yref, $Zref) = @AtomLine[5,10,11,12];
                                $NextLine = <IN>;
                                @NextLineArray = split(/ +/,$NextLine);

# Radhika 09/10/08

                                @NewAtomline = (@AtomLine,@NextLineArray);

# Radhika 09/10/08

                                $AAREFI = @NewAtomline[22];

# $q = 0;
# while($NextLineArray[$q] =~/\?/) {
#         $q = $q + 1;
# }

                                $pdbpos = $NewAtomline[20];

                                if ($AADictionary{$AAref} ne '' && ($AAREFI
eq$ARGV[0])) {

# if ( ($AADictionary{$AAref} ne '') && ($AAREFI eq "A"))
# ONLY EXTRACT CHAIN A...

                                        push( @AA , $AAref);
                                        push( @AA1, $AADictionary{$AAref});
                                        push( @X  , $Xref);
                                        push( @Y  , $Yref);
                                        push( @Z  , $Zref);
                                        push( @PDBPosArray, $pdbpos);
                                        print " AAREF is $AAref, AAREFI is $AAREFI,
pdbpos is $pdbpos and x,y,z are $Xref, $Yref and $Zref \n";

                                }

# ENd of if

                        }

# End of while

                }
```

```
	}

# ===== Extract PDB name from HEADER line

	if ($Line =~ /^data_/) {

		chomp($Line);
		@AtomLine = split( /_/, $Line);
		$Name = $AtomLine[1];
		$Name = lc($Name);
		print "Name is $Name \n";

	}

}

# Make sure it's ascending

$AACount = 1 + $#AA;
@TabValues = sort { $a <=> $b } @TabValues;

# Output Filename...

open ( OUT ,'>'."$Name".$ARGV[0].".den") or die "Cannot open out_pdbden.txt for
write.\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

# Compute density

                * ($TabValues[$j]**3));

        }

# ===== Store density values in a hash corresponding to their PDB Position =====

        $valueForHash = "";

# Output count C()

        foreach $i (@TabPrint) {
                $valueForHash = $valueForHash."_".$TabCount[$i];
        }

        $posDenHash{$PDBPosArray[$i]} = $valueForHash ;

}

$numSeqResEntries = $#AASeqRes + 1;
print "Number of residues in Sequence = $numSeqResEntries \n" ;
print OUT "Number of residues in Sequence = $numSeqResEntries \n" ;

for($k=0; $k < $numSeqResEntries; $k++) {

        if($PDBPOSSEQRES[$k] eq "\?") {

                $PrintLine = "D "
                .$Name
                ."_".sprintf("%03d", $FASTAPOS[$k])
```

```
                ."_".sprintf("%1s " , $AASeqRes[$k]);

# Output count C() = NA

                foreach $l (@TabPrint) {

                        $PrintLine = $PrintLine
                        ."C(". $TabValues[$l] .")= "
                        ."NA" ." " ;

                }

                printf OUT  $PrintLine ." \?\n";

        }

        else {

                $PrintLine = "D "
                .$Name
                ."_".sprintf("%03d", $FASTAPOS[$k])
                ."_".sprintf("%1s " , $AASeqRes[$k]);
                $density = $posDenHash{$PDBPOSSEQRES[$k]};
                print "density is $density \n";
                @densityArray = split(/_/, $density);
                $m = 1;

# Output count C()

                foreach $n (@TabPrint) {

                        $PrintLine = $PrintLine
                        ."C(". $TabValues[$n] .")= "
                        .sprintf("% 3d", $densityArray[$m]) ." " ;
                        $m++;

                }

                printf OUT  $PrintLine ." $PDBPOSSEQRES[$k]\n";

        }

}
```

```
close(IN);
printf OUT "\n\n";
close(OUT);
```

Chainselectivecif2den.pl

```
#!/usr/bin/perl -w

open (FHNDL,"SeqIDlearningset268.txt") or die "Cannot open filename ";
$lines = <FHNDL>;

while (!eof(FHNDL)) {

        chomp ($lines);
        $identifier = substr ($lines,4,1);
        $filename = substr ($lines,0,4);
        $filename = $filename."\.cif";

# print " filename is $filename\n";
# print "identifier is $identifier and line is $lines\n";

        system ("perl cif2den.pl $identifier $filename" );
        $lines = <FHNDL>;

}

close (FHNDL);
end;
```

bst2entMOD2.pl

```
# ! /usr/perl
# File: bst2ent.pl   v4 9/12/2002 D.Chiang
# "Blast to Entropy Calculation"

# Usage: 'perl bst2ent.pl <PDB-name> <infile> <outfile> (<logfile>)'
# Eg 'perl bst2ent.pl 1agm 1agmibst.txt 1agmoent.txt 1agmobst2entlog.txt'
# generates '1agmoent' from processing '1agmibst.txt'.
# The PDB name is used to tag all residues, which is then used to match the
# PDB post-processing done by Perl script 'pdb2den.pl'. Therefore, the
# protein-name MUST BE IDENTICAL to that in the corresponding PDB file.
# Also generates logfile (default 'bst2entlog.txt') = a superset
# of outfile for data verification purposes.
# BLAST output file <infile> must be saved in Text format.
```

```
# Action: Takes BLAST output file in ASCII test format

# 1. Extracts all Query and Subject sequence pairs
# 2. Compacts the Query/Subject sequences back to length(Query)
#       by omitting all insertions in the Subject sequence.
#       (Deletions in the Subject seq are kept.)
# 3. Puts the Query and all Subject seqs in output matrix.
# 4. Calculates entropy values for "qualified" database sequences (ie such
#       as those with Identity% scores higher than $IdenPercentMin) and
#       "qualified" positions (ie there are a sufficient number of homologs
#       with non-deletions in that position, as specified in $HomologsMin)
# 5. Qualified sequences are specified in upper case residue codes, while
#       unqualified sequences use lower case codes.
#       Unqualified positions are flagged with entropy value output of "-1".

# Detail Notes:

# 1. Relies on first line chars being " Score =" to flag beginning
#       section of Query and Subject sequences. ScoreBits, Expect, IdenPercent,
#       and PositivePercent values are extracted. These are used to flag
#       whether sequence is counted in entropy calculations.
# 2. Concatenates seq sections until 2 blank lines encountered.
# 3. Format of *.out file:
#       1st line is original Query (extracted from 1st match pair)
#       Subsequent lines are Subject sequences modified by deleting
#       insertions and filling out both prefix and suffix to have
#       same length as original Query.
# 4. Note that BLAST can substitute 'X' (proteins) or 'N' (nucleotides) into
#       the Query sequence to filter out "low complexity" regions. These
#       residues are kept as X or N in the entropy calculation. However, they
#       can be post-processed when correlated with the PDB information using
#       the residue position number. They are converted to lower case
#       in the output (trick to help merging with pdb2den.pl output,
#       since lowercase sorts after all upper case).

# Revision History

# v1.0 6/27/02  Initial version.
# v1.1 6/28/02  Minor addition of ';' to output file.
# v2.0 6/28/02  Change output format to transposed form.
# v3   7/18/02  Change name to bst2ent.pl (from bl2seq.pl).
#       - Use " Score =" (not "Query") to flag sequence sections.
#       - Add entropy calculations.
```

```
#         - Clean up misc code
# v4   9/12/02  Extracted all '-' from Query sequence reported from 1st
#         match in BLAST, to take care of case when the 1st match
#         includes insertions (ie the query itself is not found).
#         Also reduced Identity% and #HomologMin parameters

# 8/5/08 Modified by Radhika Pallavi Mishra to include the chain name and bitcutoff set
to 0.

#-----------------------------------------------------------------------------------------

# Specify User Parameters
# Min value of Identity% score for qualified seq

$IdenPercentMin = 0.10;

# (IdenPercentMin is another approach of BLAST results cut-off. Not used for this study)
# Min value of non-deleted homologs for entropy calc

$HomologMin = 1;

# (HomologMin is needed for entropy calc, so an error won't occur when dividing by 0)
# Min value of match score

$ScoreMin = 100;
$BitCutOff = 40;
$p_directory = "nblast_all";
opendir (DIRECTORY, $p_directory);

while (defined($p_filename = readdir(DIRECTORY))) {

        if ($p_filename != "." || $p_filename != "..") {

# Initialize all variables
# Passed param; used to tag residues in output only

                $PDBName = "";
                $Line = "";
                @LineSplit = ();
                $ScoreBits = 0;
                $Expect = 0;
                $IdenPercent = 0;
                $PositivePercent = 0;
```

```
# Original Query Seq (from 1st pair)

                $QuerySeq = "";

# Query Seq in pair for manipulation

                $QuerySeqTemp = "";

# Subject Seq in pair for manipulation

                $SubjectSeqTemp = "";

# Initial seq offset (always 1 for 1st seq)

                $QueryOffset = 0;

# Temp vars ...

                $LengthDiff = 0;
                $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";

# Get PDB protein name as 1st parameter

                $PDBName = substr($p_filename,0,5);
                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> ) {
```

```
# Save current line

                        $Line = $_ ;
                        chomp($Line);

# ----------Extracting BitScore, Expected Value, Identity, Positives----------
# Find next set of Query/Sbjct

                        while ( $Line =~ /^ Score =/ ) {

# Reset values

                                $QueryOffset = 0 ;
                                $QuerySeqTemp = "";
                                $SubjectSeqTemp = "";
                                @LineSplit = split(/ +/, $Line);
                                $ScoreBits = $LineSplit[3];

# Extract Expect value

                                $Expect = $LineSplit[8];

# If format "e-xxx" add '1' prefix

                                if ($Expect =~ /^e/) { $Expect = "1".$Expect };

# Save next line

                                $Line = <IN>;
                                chomp($Line);
                                @LineSplit = split(/ +/, $Line);

# Extract Identity%, stripping

                                $IdenPercent = $LineSplit[4];

# strip out of "(xxx%)," format

                                $IdenPercent =~ s/[(),%]//g;
                                $IdenPercent = $IdenPercent / 100.0;

# Same with Positive%

                                $PositPercent = $LineSplit[8];
```

```
                    $PositPercent =~ s/[(),%]//g;
                    $PositPercent = $PositPercent / 100.0;

# Skip next line (should be blank)

                    $Line = <IN>;
                    $Line = <IN>;

# ----------Extracting Query Sequences, Subject Sequences, Query Offsets----------
# Find 1st Query line in set

                    while ( $Line =~ /^Query  / ) {

# Separate into fields

                        ($Dummy, $QueryOffDummy, $Line, $Dummy2)
= split(/ +/, $Line);

# Keep 1st $QueryOffset

                        if ($QueryOffset == 0) {
                            $QueryOffset = $QueryOffDummy
                        }

# Combine running seq

                        $QuerySeqTemp = $QuerySeqTemp.$Line;

# Throw away 2nd line

                        $Line = <IN>;

# Save 3rd = Sbjct line

                        $Line = <IN>;
                        chomp($Line);

# Strip seq prefix

                        $Line =~ s/^Sbjct  \d+ +//;

# Strip suffix

                        $Line =~ s/ *\d+.*$//;
```

```
# Combine running seq

                                        $SubjectSeqTemp = $SubjectSeqTemp.$Line;

# Throw away 2nd line

                                        $Line = <IN>;

# Next line (another Query?)

                                        $Line = <IN>;
                                        chomp($Line);

                                    }

# Very 1st Query is saved

                                    if ( $QuerySeq eq "") {

                                        $QuerySeq = $QuerySeqTemp;

# However, convert special X,N chars to lower case

                                        $QuerySeq =~ s/X/x/g;

# (Should be removed, N is used for nucleotides only)
# $QuerySeq =~ s/N/n/g;
# Also, extract insertations ('-')

                                        $QuerySeq =~ s/-//g;
                                        $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";

# Write out complete seq

                                printf OUTD $QuerySeqTemp . "\n";
                                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";

# Write out complete seq

                                printf OUTD $QuerySeqTemp . "\n";
                                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  =====
# if ($IdenPercent < $IdenPercentMin) {
# If not-qualified, flag as lower case
# If not-qualified, flag as lower case

                                if ($ScoreBits < $ScoreMin) {
                                        $SubjectSeqTemp = lc($SubjectSeqTemp);
                                }

# =====  WRITE OUT SEQUENCE  =====

                                printf OUTD "# === Matched pair with insertions
omitted\n";

# Write out complete seq

                                printf OUTD $QuerySeqTemp . "\n";
                                printf OUTD $SubjectSeqTemp . "\n\n";

# ===== REMOVE UNQUALIFIED SEQS FROM THE FINAL ENTROPY FILE
OUTPUT =====

                                if ($ScoreBits >= $ScoreMin) {
                                        push(@SeqList, $SubjectSeqTemp."\n");
                                }

                        }

                }

                close(IN);

# Why is the -2 value there?
```

```
                for ($i = 0; $i <= -2+length( @SeqList[0] ); $i += 1) {

                        $LineOut = "";

                        for ($j = 1; $j <= $#SeqList; $j += 1) {

# Cycling thru seq j, at pos i

                                $LineOut = $LineOut . substr( @SeqList[$j], $i, 1);

                        }

# =====  COMPUTE ENTROPY  =====

                        $Line = $LineOut;

# Delete anything not capital (IMPORTANT!!!!)

                        $Line =~ s/[^A-Z]//g;
                        @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 "@"
                                ." Entropy= "        .$Entropy
                                ." EntropyCount= "   .join(" ",%EntropyCount)
                                ." EntropyCountTot= ".$EntropyCountTot
                                ." AllEntValues= "   .join(" ",@AllEntValues)
                                ."\n";

                        }

                        else {

# Flag as error -- too few homologs

                                $Entropy = -1;

                        }

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

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

                        $LineHeader = "D "
                        .$PDBName."_"
                        .sprintf("%03d",1+$i)
                        ."_". substr(@SeqList[0], $i, 1);
                        printf OUT $LineHeader ." ". $LineOut . "\n";

                }

                close(OUTD);
                close(OUT);

        }

}
```

Radhika-6pointentropy.pl

```
# ! /usr/perl
# File: bst2ent.pl   v4 9/12/2002 D.Chiang
# "Blast to Entropy Calculation"

# Usage: 'perl bst2ent.pl <PDB-name> <infile> <outfile> (<logfile>)'
# Eg 'perl bst2ent.pl 1agm 1agmibst.txt 1agmoent.txt 1agmobst2entlog.txt'
# generates '1agmoent' from processing '1agmibst.txt'.
# The PDB name is used to tag all residues, which is then used to match the
# PDB post-processing done by Perl script 'pdb2den.pl'. Therefore, the
# protein-name MUST BE IDENTICAL to that in the corresponding PDB file.
# Also generates logfile (default 'bst2entlog.txt') = a superset
# of outfile for data verification purposes.
# BLAST output file <infile> must be saved in Text format.

# Action: Takes BLAST output file in ASCII test format

# 1. Extracts all Query and Subject sequence pairs
# 2. Compacts the Query/Subject sequences back to length(Query)
#       by omitting all insertions in the Subject sequence.
#       (Deletions in the Subject seq are kept.)
# 3. Puts the Query and all Subject seqs in output matrix.
# 4. Calculates entropy values for "qualified" database sequences (ie such
#       as those with Identity% scores higher than $IdenPercentMin) and
#       "qualified" positions (ie there are a sufficient number of homologs
#       with non-deletions in that position, as specified in $HomologsMin)
# 5. Qualified sequences are specified in upper case residue codes, while
#       unqualified sequences use lower case codes.
#       Unqualified positions are flagged with entropy value output of "-1".

# Detail Notes:

# 1. Relies on first line chars being " Score =" to flag beginning
#       section of Query and Subject sequences. ScoreBits, Expect, IdenPercent,
#       and PositivePercent values are extracted. These are used to flag
#       whether sequence is counted in entropy calculations.
# 2. Concatenates seq sections until 2 blank lines encountered.
# 3. Format of *.out file:
#       1st line is original Query (extracted from 1st match pair)
#       Subsequent lines are Subject sequences modified by deleting
#       insertions and filling out both prefix and suffix to have
#       same length as original Query.
# 4. Note that BLAST can substitute 'X' (proteins) or 'N' (nucleotides) into
```

```
#         the Query sequence to filter out "low complexity" regions. These
#         residues are kept as X or N in the entropy calculation. However, they
#         can be post-processed when correlated with the PDB information using
#         the residue position number. They are converted to lower case
#         in the output (trick to help merging with pdb2den.pl output,
#         since lowercase sorts after all upper case).

# Revision History

# v1.0 6/27/02  Initial version.
# v1.1 6/28/02  Minor addition of ';' to output file.
# v2.0 6/28/02  Change output format to transposed form.
# v3   7/18/02  Change name to bst2ent.pl (from bl2seq.pl).
#         - Use " Score =" (not "Query") to flag sequence sections.
#         - Add entropy calculations.
#         - Clean up misc code
# v4   9/12/02  Extracted all '-' from Query sequence reported from 1st
#         match in BLAST, to take care of case when the 1st match
#         includes insertions (ie the query itself is not found).
#         Also reduced Identity% and #HomologMin parameters

# ------------------------------------------------------------------------------------------

# Specify User Parameters
# Min value of Identity% score for qualified seq

$IdenPercentMin = 0.10;

# (IdenPercentMin is another approach of BLAST results cut-off. Not used for this study)
# Min value of non-deleted homologs for entropy calc

$HomologMin = 1;

# (HomologMin is needed for entropy calc, so an error won't occur when dividing by 0)
# Min value of match score

$ScoreMin = 100;
$BitCutOff = 40;
$p_directory = "nblast_all";
opendir (DIRECTORY, $p_directory);

while (defined($p_filename = readdir(DIRECTORY))) {

        if ($p_filename != "." || $p_filename != "..") {
```

```
# Initialize all variables
# Passed param; used to tag residues in output only

                $PDBName = "";
                $Line = "";
                @LineSplit = ();
                $ScoreBits = 0;
                $Expect = 0;
                $IdenPercent = 0;
                $PositivePercent = 0;

# Original Query Seq (from 1st pair)

                $QuerySeq = "";

# Query Seq in pair for manipulation

                $QuerySeqTemp = "";

# Subject Seq in pair for manipulation

                $SubjectSeqTemp = "";

# Initial seq offset (always 1 for 1st seq)

                $QueryOffset = 0;

# Temp vars ...

                $LengthDiff = 0;
                $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";

# Get PDB protein name as 1st parameter

		$PDBName = substr($p_filename,0,4);
		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> ) {

# Save current line

			$Line = $_ ; chomp($Line);

# ----------Extracting BitScore, Expected Value, Identity, Positives----------
# Find next set of Query/Sbjct

			while ( $Line =~ /^ Score =/ ) {

# Reset values

				$QueryOffset = 0 ;
				$QuerySeqTemp = "";
				$SubjectSeqTemp = "";
				@LineSplit = split(/ +/, $Line);
				$ScoreBits = $LineSplit[3];

# Extract Expect value

				$Expect = $LineSplit[8];

# If format "e-xxx" add '1' prefix

				if ($Expect =~ /^e/) { $Expect = "1".$Expect };

# Save next line

				$Line = <IN>; chomp($Line);
				@LineSplit = split(/ +/, $Line);

# Extract Identity%, stripping
```

```
                                $IdenPercent = $LineSplit[4];

# strip out of "(xxx%)," format

                                $IdenPercent =~ s/[(),%]//g;
                                $IdenPercent = $IdenPercent / 100.0;

# Same with Positive%

                                $PositPercent= $LineSplit[8];
                                $PositPercent =~ s/[(),%]//g;
                                $PositPercent = $PositPercent / 100.0;

# Skip next line (should be blank)

                                $Line = <IN>;
                                $Line = <IN>;

# ----------Extracting Query Sequences, Subject Sequences, Query Offsets----------
# Find 1st Query line in set

                                while ( $Line =~ /^Query  / ) {

# Separate into fields

                                        ($Dummy, $QueryOffDummy, $Line, $Dummy2)
= split(/ +/, $Line);

# Keep 1st $QueryOffset

                                        if ($QueryOffset == 0) {
                                                $QueryOffset = $QueryOffDummy
                                        }

# Combine running seq

                                        $QuerySeqTemp = $QuerySeqTemp.$Line;

# Throw away 2nd line

                                        $Line = <IN>;

# Save 3rd = Sbjct line
```

```
                                        $Line = <IN>; chomp($Line);
# Strip seq prefix
                                        $Line =~ s/^Sbjct  \d+ +//;
# Strip suffix
                                        $Line =~ s/ *\d+.*$//;
# Combine running seq
                                        $SubjectSeqTemp = $SubjectSeqTemp.$Line;
# Throw away 2nd line
                                        $Line = <IN>;
# Next line (another Query?)
                                        $Line = <IN>; chomp($Line);
                                }
# Very 1st Query is saved
                                if ( $QuerySeq eq "") {
                                        $QuerySeq = $QuerySeqTemp;
# However, convert special X,N chars to lower case
                                        $QuerySeq =~ s/X/x/g;
# (Should be removed, N is used for nucleotides only)
# $QuerySeq =~ s/N/n/g;
# Also, extract insertations ('-')
                                        $QuerySeq =~ s/-//g;
                                        $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";

# Write out complete seq

                        printf OUTD $QuerySeqTemp . "\n";
                        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";

# Write out complete seq

                        printf OUTD $QuerySeqTemp . "\n";
                        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  =====
# If not-qualified, flag as lower case
# if ($IdenPercent < $IdenPercentMin) {
# If not-qualified, flag as lower case

                    if ($ScoreBits < $ScoreMin) {
                        $SubjectSeqTemp = lc($SubjectSeqTemp);
                    }

# =====  WRITE OUT SEQUENCE  =====

                    printf OUTD "# === Matched pair with insertions
omitted\n";

# Write out complete seq

                    printf OUTD $QuerySeqTemp . "\n";
                    printf OUTD $SubjectSeqTemp . "\n\n";

# ===== REMOVE UNQUALIFIED SEQS FROM THE FINAL ENTROPY FILE
OUTPUT =====

                    if ($ScoreBits >= $ScoreMin) {
                        push(@SeqList, $SubjectSeqTemp."\n");
                    }
```

```
                }

        }

        close(IN);

# Why is the -2 value there?

        for ($i = 0; $i <= -2+length( @SeqList[0] ); $i += 1) {

                $LineOut = "";

                for ($j = 1; $j <= $#SeqList; $j += 1) {

# Cycling thru seq j, at pos i

                        $LineOut = $LineOut . substr( @SeqList[$j], $i, 1);

                }

# =====  COMPUTE ENTROPY  =====

                $Line = $LineOut;

# Delete anything not capital (IMPORTANT!!!!)

                $Line =~ s/[^A-Z]//g;
                @LineSplit = split("", $Line);
                %EntropyCount = ();
                $EntropyCountTot = 0;

                foreach $i (@LineSplit) {

                        if(($i eq "A") || ($i eq "V") || ($i eq "L") || ($i eq "I") || ($i
eq "M") ||($i eq "C")) {

                                $category = "aliphatic";
                                $EntropyCount{$category}++ ;

# print "i is $i and count in aliphatic is $EntropyCount{$category} \n";

                        }
```

```
                elsif( ($i eq "F") || ($i eq "W") || ($i eq "Y") ||($i eq "H")) {

                        $category = "aromatic";
                        $EntropyCount{$category} =
$EntropyCount{$category} + 1 ;

                }

                elsif( ($i eq "S") || ($i eq "T") || ($i eq "N") ||($i eq "Q")) {

                        $category = "polar";
                        $EntropyCount{$category} =
$EntropyCount{$category} + 1 ;

                }

                elsif( ($i eq "K") || ($i eq "R") ) {

                        $category = "positive";
                        $EntropyCount{$category} =
$EntropyCount{$category} + 1 ;

                }

                elsif( ($i eq "D") || ($i eq "E") ) {

                        $category = "negative";
                        $EntropyCount{$category} =
$EntropyCount{$category} + 1 ;

                }

                elsif( ($i eq "G") || ($i eq "P") ) {

                        $category = "special";
                        $EntropyCount{$category} =
$EntropyCount{$category} + 1 ;

                }

                $EntropyCountTot++  ;

# print "total count is $EntropyCountTot \n ";
```

```
                        }

                        @AllEntValues = sort(values(%EntropyCount));

                        if ($EntropyCountTot >= $HomologMin) {

                                $Entropy = 0;

                                for ($j = 0; $j <= $#AllEntValues; $j++) {

# print "entropy value is $AllEntValues[$j] \n";
# print "Total count is $EntropyCountTot \n";

                                        $Prob = $AllEntValues[$j] / $EntropyCountTot ;
                                        $Entropy = $Entropy - ($Prob * ( log($Prob)/log(2)
));

                                }

# Debug Code

                                printf OUTD "@"
                                ." Entropy = " .$Entropy
                                ." EntropyCount = " .join(" ",%EntropyCount)
                                ." EntropyCountTot = " .$EntropyCountTot
                                ." AllEntValues = " .join(" ",@AllEntValues)
                                ."\n";

                        }

                        else {

# Flag as error -- too few homologs

                                $Entropy = -1;

                        }

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

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

```
                $LineHeader = "D "
                .$PDBName."_"
                .sprintf("%03d",1+$i)
                ."_". substr(@SeqList[0], $i, 1);
                printf OUT $LineHeader ." ". $LineOut . "\n";

        }

        close(OUTD);
        close(OUT);

    }

}
```

extract_fractanalysis_entropy_aggr.pl

```
# Program to extract entropy values from entropy "ENT Files" and
# compute fractional analysis print in one file for aggregate plot

if($#ARGV < 0) {

    print "Usage: perl extract_fractanalysis_entropy_aggr.pl <directory with .ent
files>\n";
    exit;

}

# Open a directory and read a file

$p_directory = $ARGV[0];
opendir (DIRECTORY, $p_directory) or die "cannot open";

while (defined ($p_filename = readdir (DIRECTORY))) {

    if ($p_filename =~ /\.ent/) {

        $filetoopen = $p_directory."/".$p_filename;
        print "file to open is $filetoopen\n";
        open (FHNDL, $filetoopen) or die "Cannot open file $p_filename";

# print" opened file $p_filename from $p_directory\n";
# to split the opened file and store in an array
# D 1AMK_001_M E=  0.000 A= abcdefg---abe
```

```
		$output_filename = substr ($p_filename,0,5)."\.fract";
		open (OUTFHNDL, ">$output_filename");
		$i=0;

# print "$lines";

		do {

			$lines = <FHNDL>;
			@filearray = split(/ +/,$lines); # @LIST = split(/PATTERN/,
STRING);
			$entropy_val = $filearray[3];

# print "$entropy_val \n";

			chomp ($entropy_val);

# split the alignments

			$alignment= $filearray[5];
			chomp($alignment);
			@align= split(//,$alignment);
			$totalLength = $#align + 1;

# calculate gapfraction -

			$numgaps = 0;
			$gapfraction = 0;

# calculate fraction small residues (Alanines A and Glycines G)

			$small_residues = 0;
			$small_residues_fraction = 0;

# calculate fraction strongly hydrophobic (V, I, L, F, Y, M, W)

			$strongly_hydrophobic = 0;
			$strongly_hydrophobic_fraction = 0;

# calculate fraction strongly hydrophobic with gaps= fraction str. hydrophobic-
fractiongaps
# fraction of small residues with gaps =
# Sequence entropy with gaps = average sequence entropy- fraction gaps
```

```
                        foreach $amino_acid (@align) {

                                if($amino_acid eq "-") {
                                        $numgaps = $numgaps +1 ;
                                }

                                if(($amino_acid eq "A") || ($amino_acid eq "G")) {
                                        $small_residues = $small_residues +1 ;
                                }

                                if(($amino_acid eq "V") || ($amino_acid eq "I")||
($amino_acid eq "L")

                                || ($amino_acid eq "F")|| ($amino_acid eq "Y") ||
($amino_acid eq "M")|| ($amino_acid eq "W") ) {
                                        $strongly_hydrophobic = $strongly_hydrophobic
+1 ;
                                }

                        }

                        $num_non_gap_amino_acids = $totalLength - $numgaps;

                        if ($num_non_gap_amino_acids > 0) {

                                $gapfraction = $numgaps/$num_non_gap_amino_acids ;
                                $small_residues_fraction =
$small_residues/$num_non_gap_amino_acids;
                                $strongly_hydrophobic_fraction =
$strongly_hydrophobic/$num_non_gap_amino_acids;
                                $non_strongly_hydrophobic_fraction = 1-
$strongly_hydrophobic_fraction;

                        }

                        else {

                                $gapfraction = $num_non_gap_amino_acids;
                                $small_residues_fraction = $num_non_gap_amino_acids;
                                $strongly_hydrophobic_fraction =
$num_non_gap_amino_acids;
                                $non_strongly_hydrophobic_fraction =
$num_non_gap_amino_acids;
```

```
                        }

                        print OUTFHNDL
"E=$entropy_val,FG=$gapfraction,FSR=$small_residues_fraction,FSHP=$strongly_hydr
ophobic_fraction,FNSHP=$non_strongly_hydrophobic_fraction\n";

                }

                while(!eof(FHNDL));
                print OUTFHNDL "\n\n";
                close(FHNDL);
                close(OUTFHNDL);

        }

}

end;
```

extract_individualfractentropy_density_aggr.pl

```
# Program to extract entropy values from entropy "ENT Files" and print in one file for
aggregate plot

if($#ARGV < 1) {
        print "Usage : perl extract_entropy_aggr.pl <directory with .fract files> <directory
with density files> \n";
}

# Open a directory and read a file

$p_directory = $ARGV[0];
$p1_directory = $ARGV[1];

# $outputfile = $ARGV[2];
# Open files for reading and writing
# open(OUTFHNDL, ">$outputfile");
# open file from density directory

opendir (DIRECTORY1, $p1_directory) or die "cannot open directory $p1_directory \n";

while (defined($p1_filename = readdir(DIRECTORY1))) {
```

```
        if ($p1_filename != "." || $p1_filename != "..") {

                print "filename is $p1_filename and directory is $p1_directory \n";
                open ( FHNDL1  ,"$p1_directory/$p1_filename") or die "Cannot open
input files for read"."\n";

# Get PDB protein name as 1st parameter

                $PDBName = substr($p1_filename,0,5);
                $PDBName = uc($PDBName);
                open(OUTFHNDL, ">$PDBName\.txt");

# open  corresponding fract file

                $filetoopen = $p_directory."/".$PDBName."\.fract";
                print "file to open is $filetoopen\n";
                open(FHNDL, $filetoopen) or die "Cannot open file $p_filename";
                $lines1 = <FHNDL1>;
                $lines = <FHNDL>;

# print "$lines1";

                do {

                        if ($lines1 =~ /C\(9\)/) {

                                @filearray1 = split(/ +/,$lines1); # @LIST =
split(/PATTERN/, STRING);

# @filearray = split(/ /,$lines);

                                $density_val = $filearray1[9];
                                $entropy_val = $lines;
                                chomp($density_val);
                                chomp($entropy_val);
                                print OUTFHNDL "Den=$density_val,$entropy_val\n";

# print "$density_val $entropy_val\n";

                                $lines= <FHNDL>;

                        }

                        $lines1 = <FHNDL1>;
```

```
                }

                while(!eof(FHNDL1) && !eof(FHNDL));
                print OUTFHNDL "\n\n";
                close(FNDHL1);
                close(FHNDL);
                close (OUTFHNDL);

        }

}

end;
```

calculate_aggr_per_protein.pl

```
# Program to extract entropy values from entropy "ENT Files" and print in one file for
aggregate plot

if ($#ARGV < 1) {

        print "Usage: perl calculate_aggr_per_protein.pl <Input directory with .txt files
with individual agg protein density and rsa calculations> <output directory name> \n";
        exit;

}

# Open a directory and read a file

$p_directory = $ARGV[0];
$p_directory1 = $ARGV[1];

# $outputfile = $ARGV[2];
# Open files for reading and writing
# open (OUTFHNDL, ">$outputfile");
# open file from density directory

opendir (DIRECTORY, $p_directory) or die "cannot open directory $p1_directory \n";

while (defined ($p_filename = readdir (DIRECTORY))) {

        if ($p_filename =~ /txt/) {
```

```
		print "filename is $p1_filename and directory is $p1_directory \n";
		open (FHNDL, "$p_directory/$p_filename") or die "Cannot open input
files for read"."\n";

# Get PDB protein name as 1st parameter

		$PDBName = substr ($p_filename, 0, 4);
		$outputfilename = $PDBName."Aggr"."\.txt";
		$outputfilename = $p_directory1."/".$outputfilename;
		open (OUTFHNDL, ">$outputfilename");

# Following Arrays will store the average value of ent, fg, fsr fshp and fnshp for den = 0
to 40 in their index 0 to 40

		my @aggr_rsa_array = ();

# Following Array will store the average number of occurences of den=i, at index i

		my @num_density_occurences = ();

		for($i=0; $i<=40; $i++) {

			$aggr_rsa_array[$i] = 0;
			$num_density_occurences[$i] = 0;

		}

		do {

			$lines = <FHNDL>;

			if ($lines =~ /^PDB/) {

# @LIST = split (/PATTERN/, STRING);

				@filearray = split (/, /, $lines);
				print "$lines ";
				@temp = split (/\=/, $filearray[2]);
				$den = $temp[1];
				print "Density is $den \n";

				if ($den ne "NA") {

					for ($i=0; $i<=40; $i++) {
```

```
                                        if ($den == $i) {

                                                @temp = split (/=/, $filearray[3]);
                                                $rsa = $temp[1];
                                                print "$rsa\n";
                                                $aggr_rsa_array[$i] =
$aggr_rsa_array[$i] + $rsa;
                                                $num_density_occurences[$i] =
$num_density_occurences[$i] + 1;

                                        }

                                }

                        }

                }

                while (!eof (FHNDL));

                for ($i=0; $i<=40; $i++) {

                        if ($num_density_occurences[$i] > 0) {

                                $aggr_rsa = $aggr_rsa_array[$i] /
$num_density_occurences[$i];
                                print OUTFHNDL "Den=$i, RSA=$aggr_rsa,
N=$num_density_occurences[$i]\n";

                        }

                        else {
                                print OUTFHNDL "Den=$i,RSA=NA,N=0\n";
                        }

                }

                print OUTFHNDL "\n\n";
                close (FHNDL);
                close (OUTFHNDL);
```

```
        }

}

end;
```

double_agg_forPlot.pl

```
# Program to extract individual entropy,etc aggregate values from aggregate entropy,etc
".txt" files
# and print in one file for aggregate plot

if ($#ARGV < 1) {
        print "Usage: perl extract_entropy_aggr.pl <directory with .txt files> <name of
output file> \n";
}

# Open a directory and read a file

$p_directory = $ARGV[0];
$outputfile = $ARGV[1];

# Open files for reading and writing

open (OUTFHNDL, ">$outputfile");

# open file from density directory

opendir (DIRECTORY, $p_directory) or die "cannot open directory $p_directory \n";

while (defined ($p_filename = readdir (DIRECTORY))) {

        if ($p_filename != "." || $p_filename != "..") {

                print "filename is $p_filename and directory is $p_directory \n";
                open (FHNDL, "$p_directory/$p_filename") or die "Cannot open input
files for read"."\n";
                $lines = <FHNDL>;

# print "$lines";

                do {

                        print OUTFHNDL "$lines";
```

```
# print "$lines\n";

                $lines= <FHNDL>;

        }

        while (!eof (FHNDL));

    }

}

close (FHNDL);
close (OUTFHNDL);
end;
```

listNoAlignments.pl

```
# Program to list the number of alignments in a blast file saved as .txt

if ($#ARGV < 2) {
    print "Usage: perl listNoAlignments.pl <directory with .txt files> <name of output
file> \n";
}

# Open a directory and read a file

$p_directory = $ARGV[0];
$output = $ARGV[1];
open (OUTFHNDL, ">$output");
opendir (DIRECTORY, $p_directory) or die "cannot open";

while (defined ($p_filename = readdir (DIRECTORY))) {

    if ($p_filename =~ /\.txt/) {

        $filetoopen = $p_directory."/".$p_filename;

# print "file to open is $filetoopen\n";

        open (IN, $filetoopen) or die "Cannot open file $p_filename";

# print" opened file $p_filename from $p_directory\n";
```

```
                $line = <IN>;
                $n=0;

                while (!eof (IN)) {

                        if ($line =~ /^ Score =/) {
                                $n++;
                        }

                        $line = <IN>;

                }

                print "$n\n";
                print OUTFHNDL "Total Number of alignments = $n\n";
                close (IN);

        }

}

close (OUTFHNDL);
end;
```

No_of_res_count.pl

```
# Program to list the number of residues in a density file

if ($#ARGV < 2) {
        print "Usage: perl extract_entropy_aggr.pl <directory with .den files> <name of
output file> \n";
}

# Open a directory and read a file

$p_directory = $ARGV[0];
$outputfile = $ARGV[1];

# Open files for reading and writing

open (OUTFHNDL, ">$outputfile");

# open file from density directory
```

```
opendir (DIRECTORY1, $p_directory) or die "cannot open directory $p_directory \n";

while (defined ($p_filename = readdir (DIRECTORY1))) {

        if ($p_filename != "." || $p_filename != "..") {

                print "filename is $p_filename and directory is $p_directory \n";
                $filetoopen = $p_directory."/".$p_filename;
                print "file to open is $filetoopen\n";
                open (FHNDL, $filetoopen) or die "Cannot open file $p_filename";
                $lines = <FHNDL>;

                while (!eof (FHNDL)) {

                        if ($lines =~ /NO_RESIDUES=/) {

                                print "$lines \n";
                                @filearray = split (/ +/,$lines);
                                $no_res = $filearray[2];
                                chomp ($no_res);
                                print OUTFHNDL "$p_filename $no_res\n";

                        }

                        $lines= <FHNDL>;

                }

                close (FHNDL);

        }

}

close (OUTFHNDL);
end;
```

Bitscorelistno_ofsubject.pl

```
# Program to list the scores in a blast file saved as .txt, and the corresponding bit score

if($#ARGV < 2) {
```

```
	print "Usage: perl Bitscorelistno_ofsubject.pl <directory with .txt files> <name of
output file> \n";
}

# Open a directory and read a file

$p_directory = $ARGV[0];
$output = $ARGV[1];
open(OUTFHNDL, ">$output");
opendir (DIRECTORY, $p_directory) or die "cannot open";

while (defined($p_filename = readdir(DIRECTORY))) {

	if ($p_filename =~ /\.txt/) {

		$filetoopen = $p_directory."/".$p_filename;

# print "file to open is $filetoopen\n";

		open(IN, $filetoopen) or die "Cannot open file $p_filename";

# print" opened file $p_filename from $p_directory\n";

		$line = <IN>;

		while(!eof(IN)) {

			if ( $line =~ /^ Score =/) {

				print "$line \n";
				@filearray = split(/ +/,$line);
				$bitscore = $filearray[3];
				chomp $bitscore;
				print "$bitscore\n";
				print OUTFHNDL "$bitscore\n";

			}

			$line = <IN>;

		}

# print "$n\n";
# print OUTFHNDL "Total Number of alignments = $n\n";
```

```
                close(IN);

        }

}

close(OUTFHNDL);
end;
```

Appendix C Additional Files

SeqIDlearningset268.txt

IDs	length	Exptl.	resolution	R-factor	FreeRvalue
2JELP	85	XRAY	2.5	0.21	0.28
3CLAA	213	XRAY	1.75	0.16	1
1TOAA	313	XRAY	1.8	0.18	0.2
8ATCA	310	XRAY	2.5	0.17	1
3SGBE	185	XRAY	1.8	0.12	1
1EFNB	152	XRAY	2.5	0.21	0.28
1QPAA	345	XRAY	1.8	0.16	1
1EFUA	385	XRAY	2.5	0.17	0.28
1QCIA	262	XRAY	2	0.23	1
1A32A	88	XRAY	2.1	0.21	0.32
1AMUA	563	XRAY	1.9	0.21	0.25
1HWGA	191	XRAY	2.5	0.2	0.29
1AQ0A	306	XRAY	2	0.17	0.21
1AFWA	393	XRAY	1.8	0.19	0.24
1GARA	212	XRAY	1.96	0.17	0.29
1CRMA	260	XRAY	2	0.18	1
1XGSA	295	XRAY	1.75	0.19	0.23
1DANL	152	XRAY	2	0.19	0.22
1AL8A	359	XRAY	2.2	0.19	0.25
1QMEA	702	XRAY	2.4	0.2	0.23
1AF3A	196	XRAY	2.5	0.23	0.27
1QR2A	230	XRAY	2.1	0.22	0.28
1NO3A	857	XRAY	2.15	0.19	0.23
1AKOA	268	XRAY	1.7	0.17	0.2
1BRWA	433	XRAY	2.1	0.23	0.28
1BMDA	327	XRAY	1.9	0.15	1
1DHKA	496	XRAY	1.85	0.18	0.22

2NACA	393	XRAY 1.8	0.15	1
1EEHA	437	XRAY 1.9	0.23	0.27
1B8AA	438	XRAY 1.9	0.17	0.2
1RHSA	296	XRAY 1.36	0.17	0.23
1XSOA	150	XRAY 1.49	0.1	0.17
1BUOA	121	XRAY 1.9	0.21	0.25
1DHSA	361	XRAY 2.2	0.15	0.24
1FLEI	57	XRAY 1.9	0.2	1
1CB0A	283	XRAY 1.7	0.18	0.2
3SDHA	146	XRAY 1.4	0.16	1
1QHAA	917	XRAY 2.25	0.21	0.28
2LIVA	344	XRAY 2.4	0.18	1
1QFHA	212	XRAY 2.2	0.22	0.27
1BIQA	375	XRAY 2.05	0.19	0.26
1GOTA	350	XRAY 2	0.21	0.29
1UBYA	367	XRAY 2.4	0.2	1
1AK0A	270	XRAY 1.8	0.21	0.23
1THTA	305	XRAY 2.1	0.23	1
1ILR1	152	XRAY 2.1	0.2	1
1AOBA	265	XRAY 2.1	0.19	0.24
1GOTG	73	XRAY 2	0.21	0.29
1CMBA	104	XRAY 1.8	0.19	1
1AZIA	153	XRAY 2	0.17	1
1HWGB	237	XRAY 2.5	0.2	0.29
1QAZA	351	XRAY 1.78	0.18	0.23
2OHXA	374	XRAY 1.8	0.17	1
1BULA	265	XRAY 1.89	0.21	0.26
2HDHA	293	XRAY 2.2	0.2	0.25
2TGIA	112	XRAY 1.8	0.17	1
1CHMA	401	XRAY 1.9	0.18	1
1YCSA	199	XRAY 2.2	0.2	0.29
1HXPA	348	XRAY 1.8	0.19	1
1BFDA	528	XRAY 1.6	0.15	0.19
1FIPA	98	XRAY 1.9	0.2	1
8PRKA	287	XRAY 1.85	0.19	0.23
5CSMA	256	XRAY 2	0.19	0.24
2SHPA	525	XRAY 2	0.2	0.27
1NP4A	184	XRAY 1.5	0.2	0.26
1BD0A	388	XRAY 1.6	0.24	0.27
830CA	168	XRAY 1.6	0.21	0.27
2RN2A	155	XRAY 1.48	0.2	1
6XIAA	387	XRAY 1.65	0.14	1
2PCCA	296	XRAY 2.3	0.17	1
1BO6A	297	XRAY 2.1	0.21	0.25

1BXQA	323	XRAY 1.41	0.14	0.18
1DFJI	456	XRAY 2.5	0.19	1
5CPVA	108	XRAY 1.6	0.19	1
1JSGA	114	XRAY 2.5	0.19	0.26
1PBGA	468	XRAY 2.3	0.16	0.24
5RUBA	490	XRAY 1.7	0.18	1
256BA	106	XRAY 1.4	0.16	1
1CJXA	357	XRAY 2.4	0.22	0.28
2LIGA	164	XRAY 2	0.18	1
1F13A	731	XRAY 2.1	0.18	0.24
1DAAA	282	XRAY 1.94	0.18	1
1EFUB	282	XRAY 2.5	0.17	0.28
1PGTA	210	XRAY 1.8	0.18	1
1SMTA	122	XRAY 2.2	0.22	0.25
3PMGA	561	XRAY 2.4	0.16	0.19
2MBRA	340	XRAY 1.8	0.2	0.26
1DANT	80	XRAY 2	0.19	0.22
6LDHA	329	XRAY 2	0.2	1
8ATCB	153	XRAY 2.5	0.17	1
1TC1A	220	XRAY 1.41	0.19	0.23
1URPA	271	XRAY 2.3	0.23	0.27
1AG9A	175	XRAY 1.8	0.2	0.25
1BA3A	550	XRAY 2.2	0.2	0.24
1GVPA	87	XRAY 1.6	0.21	0.29
5CPAA	307	XRAY 1.54	0.19	1
1BRSA	110	XRAY 2	0.17	1
1CQXA	403	XRAY 1.75	0.18	0.21
1PDAA	313	XRAY 1.76	0.19	1
1NAWA	419	XRAY 2	0.2	0.27
2ILKA	160	XRAY 1.6	0.16	1
7CATA	506	XRAY 2.5	0.21	1
1CSHA	435	XRAY 1.65	0.16	1
9WGAA	171	XRAY 1.8	0.17	1
3PFKA	319	XRAY 2.4	0.17	1
1AH7A	245	XRAY 1.5	0.2	0.23
1RPOA	65	XRAY 1.4	0.19	1
1BRSD	89	XRAY 2	0.17	1
1AUOA	218	XRAY 1.8	0.21	0.27
1AW7A	194	XRAY 1.95	0.18	1
1DCSA	311	XRAY 1.3	0.13	0.15
1E98A	215	XRAY 1.9	0.19	0.24
1C02A	166	XRAY 1.8	0.2	0.25
1ISAA	192	XRAY 1.8	0.19	1
1NSYA	271	XRAY 2	0.17	0.23

2TCTA	207	XRAY 2.1	0.18	1
1IMBA	277	XRAY 2.2	0.17	1
1EWFA	456	XRAY 1.7	0.2	0.25
2TRCP	217	XRAY 2.4	0.19	0.28
1STFI	98	XRAY 2.37	0.19	1
1KPTA	105	XRAY 1.75	0.17	0.22
4DFRA	159	XRAY 1.7	0.15	1
1FEHA	574	XRAY 1.8	0.18	0.23
1E5MA	416	XRAY 1.54	0.17	0.2
256LA	164	XRAY 1.8	0.16	1
1DHKB	223	XRAY 1.85	0.18	0.22
1GOTB	340	XRAY 2	0.21	0.29
1A48A	305	XRAY 1.9	0.15	1
1BJWA	382	XRAY 1.8	0.21	0.27
1DORA	311	XRAY 2	0.17	0.21
1DMRA	823	XRAY 1.82	0.15	0.18
2BLSA	358	XRAY 2	0.22	1
1AYXA	492	XRAY 1.7	0.15	0.18
1B3AA	67	XRAY 1.6	0.17	0.24
1BINA	143	XRAY 2.2	0.2	0.3
1MPGA	282	XRAY 1.8	0.19	0.25
1SESA	421	XRAY 2.5	0.18	1
1A4IA	301	XRAY 1.5	0.2	0.23
13PKA	415	XRAY 2.5	0.22	0.29
1CKIA	317	XRAY 2.3	0.19	0.28
1SOXA	466	XRAY 1.9	0.17	0.22
1ADEA	431	XRAY 2	0.2	1
1DINA	236	XRAY 1.8	0.15	1
3GBPA	307	XRAY 2.4	0.16	1
1VLBA	907	XRAY 1.28	0.15	0.19
1AW9A	216	XRAY 2.2	0.2	1
1VBTA	165	XRAY 2.3	0.2	0.25
1B5EA	246	XRAY 1.6	0.19	0.21
1QJPA	171	XRAY 1.65	0.15	0.2
1OSPO	257	XRAY 1.95	0.23	0.29
1BIAA	321	XRAY 2.3	0.19	1
1HJRA	158	XRAY 2.5	0.16	1
2UGIA	84	XRAY 2.2	0.23	0.28
2BC2A	227	XRAY 1.7	0.2	0.25
1BYOA	99	XRAY 2	0.19	0.23
1YQVL	211	XRAY 1.7	0.2	0.23
1FGKA	310	XRAY 2	0.21	0.26
1CSEE	274	XRAY 1.2	0.18	1
1ICWA	72	XRAY 2.01	0.19	0.27

1YCSB	239	XRAY 2.2	0.2	0.29
1BISA	166	XRAY 1.95	0.2	0.26
3GRSA	478	XRAY 1.54	0.19	1
1IVYA	452	XRAY 2.2	0.21	0.27
1ADDA	349	XRAY 2.4	0.18	1
1DYSA	348	XRAY 1.6	0.18	0.24
2SQCA	631	XRAY 2	0.15	0.19
1SHKA	173	XRAY 1.9	0.17	0.22
1YDRE	350	XRAY 2.2	0.19	1
1VOKA	200	XRAY 2.1	0.2	1
1CDCA	99	XRAY 2	0.19	1
1REGX	122	XRAY 1.9	0.18	0.21
2ARCA	164	XRAY 1.5	0.18	0.23
3SGBI	56	XRAY 1.8	0.12	1
1AORA	605	XRAY 2.3	0.15	1
1CZJA	111	XRAY 2.16	0.2	0.26
1RNEA	340	XRAY 2.4	0.18	1
1MCTA	223	XRAY 1.6	0.17	1
1NOXA	205	XRAY 1.59	0.19	0.2
1CSEI	70	XRAY 1.2	0.18	1
1A1IA	90	XRAY 1.6	0.19	0.22
1DQSA	393	XRAY 1.8	0.17	0.22
1DANU	121	XRAY 2	0.19	0.22
1CNZA	363	XRAY 1.76	0.2	0.26
12ASA	330	XRAY 2.2	0.16	0.29
1A6QA	382	XRAY 2	0.21	1
1HIAI	48	XRAY 2.4	0.2	0.31
1B67A	68	XRAY 1.48	0.19	0.27
1A4UA	254	XRAY 1.92	0.2	0.24
1BG0A	356	XRAY 1.86	0.2	0.22
1AA7A	158	XRAY 2.08	0.21	0.28
1BAMA	213	XRAY 1.95	0.19	1
1YQVH	215	XRAY 1.7	0.2	0.23
1MORA	368	XRAY 1.9	0.19	1
1GUAB	81	XRAY 2	0.22	1
2TPSA	227	XRAY 1.25	0.18	0.22
1AN9A	340	XRAY 2.5	0.2	0.26
2ACYA	98	XRAY 1.8	0.17	0.23
1NSEA	444	XRAY 1.9	0.21	0.28
1CTTA	294	XRAY 2.2	0.19	1
1AW5A	340	XRAY 2.3	0.2	0.27
1FINB	260	XRAY 2.3	0.21	1
1SMNA	245	XRAY 2.04	0.17	1
1NMBN	470	XRAY 2.2	0.21	1

1AYLA	541	XRAY 1.8	0.2	0.23
2SPCA	107	XRAY 1.8	0.2	1
1BN6A	294	XRAY 1.5	0.17	0.17
1KBAA	66	XRAY 2.3	0.2	1
2G3PA	225	XRAY 1.9	0.26	0.3
1AK4C	145	XRAY 2.36	0.24	0.31
1QHIA	366	XRAY 1.9	0.23	0.29
1AJSA	412	XRAY 1.6	0.17	1
1B8JA	449	XRAY 1.9	0.18	0.2
1AMPA	291	XRAY 1.8	0.16	1
1BF2A	750	XRAY 2	0.16	0.21
1FROA	183	XRAY 2.2	0.21	0.23
1BXGA	356	XRAY 2.3	0.17	1
1DHTA	327	XRAY 2.24	0.19	0.28
1TOXA	535	XRAY 2.3	0.23	0.31
1OPYA	131	XRAY 1.9	0.2	0.27
1BT3A	345	XRAY 2.5	0.17	0.25
1E1KA	460	XRAY 1.95	0.18	0.23
1AVWB	177	XRAY 1.75	0.19	0.21
1G2AA	168	XRAY 1.75	0.19	0.25
1AQ6A	253	XRAY 1.95	0.19	0.25
1FMTA	314	XRAY 2	0.21	0.26
1UTGA	70	XRAY 1.34	0.23	1
1DPGA	485	XRAY 2	0.21	0.26
1BXKA	355	XRAY 1.9	0.2	1
1RBPA	182	XRAY 2	0.18	1
1SLTA	134	XRAY 1.9	0.17	1
1FJMA	330	XRAY 2.1	0.18	1
1ATLA	202	XRAY 1.8	0.16	1
1BBHA	131	XRAY 1.8	0.18	1
1JHGA	101	XRAY 1.3	0.13	0.17
1CEXA	214	XRAY 1	0.09	0.12
2IHLA	129	XRAY 1.4	0.17	1
1M6PA	152	XRAY 1.8	0.22	0.28
3RN3A	124	XRAY 1.45	0.22	1
1CG2A	393	XRAY 2.5	0.2	0.22
1KWAA	88	XRAY 1.93	0.25	0.3
4HTCI	65	XRAY 2.3	0.17	1
2ATJA	308	XRAY 2	0.18	0.2
1OACA	727	XRAY 2	0.16	1
1CRZA	403	XRAY 1.95	0.19	0.24
1EHYA	294	XRAY 2.1	0.19	0.23
1MKBA	171	XRAY 2	0.18	0.24
8PTIA	58	XRAY 1.8	0.16	1

1AMKA	251	XRAY 1.83	0.11	1
2SICI	107	XRAY 1.8	0.18	1
1TRKA	680	XRAY 2	0.16	1
1EBHA	436	XRAY 1.9	0.19	1
1TX4A	198	XRAY 1.65	0.17	0.21
1BSLA	324	XRAY 1.95	0.19	1
1TX4B	177	XRAY 1.65	0.17	0.21
1QTQA	553	XRAY 2.25	0.24	0.25
1KPFA	125	XRAY 1.5	0.21	0.24
1BLZA	331	XRAY 1.45	0.2	0.22
1FKDA	107	XRAY 1.72	0.18	1
1BEAA	127	XRAY 1.95	0.2	0.29
2SCPA	174	XRAY 2	0.18	1
1HF8A	289	XRAY 2	0.19	0.22
1GJMA	414	XRAY 2.2	0.18	0.22
1A2KA	127	XRAY 2.5	0.21	0.27
1COZA	129	XRAY 2	0.2	0.26
5ACNA	754	XRAY 2.1	0.21	1
9PAPA	212	XRAY 1.65	0.16	1
1CRCA	104	XRAY 2.08	0.18	1
3DAPA	320	XRAY 2.2	0.17	0.23

Appendix D Additional Notes for Flowchart for Perl Scripts

Additional Notes:

1. *ftp-script-1.pl* downloads mmcif files from the RCSB PDB (Protein Data Bank) website (http://www.rcsb.org) and saves the files as .cif (ex. 12AS.cif).
    a) ftp-script-1.pl outputs .cif files, which is the input for *cif2den.pl* and *Chainselectivecif2den.pl*.

2. *cif2den.pl* and *Chainselectivecif2den.pl* takes .cif files and outputs .den files.
    a) *SeqIDlearningset268.txt* must be in same folder as scripts.

3. *bst2entMOD2.pl* or *Radhika-6pointentropy.pl* takes BLASTp .txt input files and outputs .ent and .dbg files.
    a) BLASTp .txt files were downloaded after running protein sequence with BLASTp algorithm, from NCBI website (http://www.ncbi.nlm.nih.gov).
    b) To run *bst2entMOD2.pl* or *Radhika-6pointentropy.pl*, BLASTp .txt files must be in folder named “nblast_all”.

4. *ftp-scriptHSSP.pl* used to download .ent and .dbg files.

5. *extract_fractanalysis_entropy_aggr.pl* uses .ent files as input and outputs .fract files.
    a) .ent and .dbg files must be in a user defined folder and then executed through Perl script.

6. *extract_individualfractentropy_density_aggr.pl* takes .den and .fract files, aligns the residue positions, and outputs a .txt file of the aligned residues.
    a) .den and .fract files must be in a user defined folder and then executed through Perl script.

7. *calculate_aggr_per_protein.pl* takes the .txt file of the aligned residues and outputs a .txt file of an aggregate analysis.
    a) .txt files must be in a user defined folder and then executed through Perl script.
    b) Output directory is also user defined and executed through Perl script.

8. *double_agg_forPlot.pl* takes .txt file of aggregate analysis and outputs a user defined file of single aggregate values.
    a) .txt files must be in a user defined folder and then executed through Perl script.

9. *listNoAlignments.pl* takes BLASTp .txt input files and outputs .txt files of frequency of query proteins vs. number of alignments.
    a) BLASTp .txt files were downloaded after running protein sequence with BLASTp algorithm, from NCBI website (http://www.ncbi.nlm.nih.gov).
    b) .txt files must be in user defined directory and executed through Perl script.

10. *No_of_res_count.pl* takes .den files and outputs .txt files with the frequency of query proteins to the length of query proteins.
    a) .den files must be in user defined directory and executed through Perl script.

11. *Bitscorelistno_ofsubject.pl* takes BLASTp .txt input files and outputs .txt files of the frequency of subject proteins at BLAST bit score.
    a) BLASTp .txt files were downloaded after running protein sequence with BLASTp algorithm, from NCBI website (http://www.ncbi.nlm.nih.gov).
    b) .txt files must be in user defined directory and executed through Perl script.