Chapter 23: Mining Data in Bioinformatics

Mohammed J. Zaki [*], Rensselaer Polytechnic Institute

Running Title: Mining Data in Bioinformatics
Keywords:
Amino Acids
Association Mining
Bioinformatics
CASP
Classification
Closed Itemsets
Contact Map
DNA
Frequent Itemsets
Genes
Hidden Markov Model
Protein Data Bank
Protein Folding Pathways
Proteins
Protein Structure Prediction
RNA
Secondary Structure Motifs

1 Introduction

Bioinformatics is the science of storing, extracting, organizing, analyzing, interpreting, and utilizing information from biological sequences and molecules. It has been mainly fueled by advances in DNA sequencing and genome mapping techniques. The Human Genome Project has resulted in rapidly growing databases of genetic sequences, while the Structural Genomics Initiative is doing the same for the protein structure database. New techniques are needed to analyze, manage and discover sequence, structure and functional patterns or models from these large sequence and

[*]This work was supported in part by NSF CAREER Award IIS-0092978, and NSF Next Generation Software Program grant EIA-0103708.

structural databases. High performance data analysis algorithms are also becoming central to this task.

Bioinformatics provides opportunities for developing novel data analysis methods. Some of the grand challenges in bioinformatics include protein structure prediction, homology search, multiple alignment and phylogeny construction, genomic sequence analysis and gene finding, as well as applications in gene expression data analysis, drug discovery in pharmaceutical industry, etc. For example, in protein structure prediction, one is interested in determining the secondary, tertiary and quaternary structure of proteins, given their amino acid sequence. Homology search aims at detecting increasingly distant homologues, i.e., proteins related by evolution from a common ancestor. Multiple alignment and phylogenetic tree construction are interrelated problems. Multiple alignment aims at aligning a whole set of sequences to determine which subsequences are conserved. This works best when a phylogenetic tree of related proteins is available. Finally, gene finding aims at locating the genes in a DNA sequence.

In this chapter we will focus on how data mining methods can be utilized for protein structure prediction, where one is interested in determining 3D structure of proteins given their amino acid sequence. We will begin with a brief introduction to concepts of molecular biology, and then we will focus on the protein folding problem and various mining techniques applied in this domain.

2 Background

2.1 Basic Molecular Biology

There are three main building blocks of biological systems formed from nucleic or amino acids. *DNA* (DeoxyriboNucleic Acid) acts as the information carrier/encoder, *RNA* (RiboNucleic Acid) acts as the bridge from DNA to proteins, and *Proteins* (composed of Amino Acids) act as the action molecules.

DNA is an unbranched double helical polymer composed of nucleotides, which have four kinds of bases (the nucleic acids): adenine (A), thymine (T), cytosine (C) and guanine (G). A always pairs with T and G pairs with C to form complimentary base pairing in the double helical DNA strand. The primary structure of DNA can be represented simply as a string of bases, e.g.,

$ATGAATCGTAA\cdots$.

Unlike DNA, RNA is a single stranded molecule also made up of four nucleic acids, but instead of T RNA has uracil (U). U also binds with A just like T. There are different kinds of RNA in the cell; messenger (mRNA) and transfer RNAs (tRNA) are of most relevance for protein folding.

Proteins are unbranched polymers with a peptide backbone formed from a chain of amino acids. An amino acid has a central carbon atom, called the alpha carbon (C_α). Attached to C_α atom is a hydrogen atom, an amino group, a carboyl group, and a side chain. There are twenty different side chains, which differentiate one amino acid from another, giving 20 different amino acids. The side chains differ in size (ranging from a single hydrogen atom to having carbon rings), polarity (polar/non-polar), attractivity to water (hydrophilic/hydrophobic), etc.

The primary structure of a protein can be represented as a sequence formed from a 20 letter alphabet (one letter per amino acid), e.g., $MNRRGLNAGNTMTSQANID\cdots$. Since proteins are 3D molecules they have higher order structures. The secondary structure of a protein refers to the structures formed by short subsequences (e.g., alpha helixes and beta sheets are the main secondary structures). The global fold of the proteins is called its tertiary structure (i.e., the arrangement of secondary structures in 3D). Multiple protein domains also arrange themselves into complexes, refered to as the quaternary structure. The shape of the protein is crucial to the function; its irregular surface allows different kinds of molecules to bind to the protein and this facilitates various activities (regulatory, enzymatic, etc.) in the cell. This is why the protein folding problem is so important.

So how do proteins form? The *genes*, which are contiguous stretches of DNA, encode the information to manufacture proteins. Often there is a very complex regulatory network involving several genes that controls the production of proteins. A DNA strand thus consists of the genes at different positions interspersed with the intergenic regions, which do not code for any protein. A gene itself can be composed of exons (expressed segments) and the introns (unexpressed segments).

Protein formation happens via two main steps: *transcription* and *translation*. First a copy of the the gene is made; subsequently the unexpressed introns are spliced out to yield a mRNA molecule that only consists of the exons. This process is called transcription. A subsequence of three mRNA bases, called a codon, codes for a single amino acid. Transfer RNA (tRNA) molecules

are the ones that implement the translation between the codon and the amino acids. The ribosomal assembly unit slides along the mRNA molecule and for successive codons the tRNA molecule attracts different amino acids, with the result that as the ribosomal unit slides there is a growing polypeptide chain. This process is called translation. The protein synthesis process stops when a stop codon is encountered. As mentioned above, the proteins take on unique three dimensional shapes, which govern their function.

2.2 Mining Methods in Protein Structure Prediction

It is well known that proteins fold spontaneously and reproducibly to a unique 3D structure in aqueous solution. Despite significant advances in recent years, the goal of predicting the three dimensional structure of a protein from its one-dimensional sequence of amino acids, without the aid of evolutionary information, remains one of greatest and most elusive challenges in bioinformatics. The current state of the art in structure prediction provides insights that guide further experimentation, but falls far short of replacing those experiments.

Today we are witnessing a paradigm shift in predicting protein structure from its known amino acid sequence $(a_1, a_2, \cdots, a_n)$. The traditional or Ab initio folding method employed first principles to derive the 3D structure of proteins. However, even though considerable progress has been made in understanding the chemistry and biology of folding, the success of ab initio folding has been quite limited.

Instead of simulation studies, an alternative approach is to employ learning from examples using a database of known protein structures. For example, the Protein Data Bank (PDB) records the 3D coordinates of the atoms of thousands of protein structures. Most of these proteins cluster into around 700 fold-families based on their similarity. It is conjectured that there will be on the order of 1000 fold-families for the natural proteins (Wolf, Grishin, & Koonin 2000). The PDB thus offers a new paradigm to protein structure prediction by employing data mining methods like clustering, classification, association rules, hidden Markov models, etc.

The ability to predict protein structure from the amino acid sequence will do no less than revolutionize molecular biology. All genes will be interpretable as three-dimensional, not one-dimensional, objects. The task of assigning a predicted function to each of these objects (arguably

a simpler problem than protein folding) would then be underway. In the end, combined with proteomics data (i.e. expression arrays), we would have a flexible model for the whole cell, potentially capable of predicting emergent properties of molecular systems, such as signal transduction pathways, cell differentiation, and the immune response.

Protein Folding Pathways Proteins are chains of amino acids having lengths ranging from 50 to 1000 or more residues. The early work of Levinthal (Levinthal 1968) and Anfinsen (Anfinsen & Scheraga 1975) established that a protein chain folds spontaneously and reproducibly to a unique three dimensional structure when placed in aqueous solution. The sequence of amino acids making up the polypeptide chain contains, encoded within it, the complete building instructions. Levinthal also proved that the folding process cannot occur by random conformational search for the lowest energy state, since such a search would take millions of years, while proteins fold in milliseconds. As a result, Anfinsen proposed that proteins must form structure in a time-ordered sequence of events, now called a "pathway". The nature of these events, whether they are restricted to "native contacts" (defined as contacts that are retained in the final structure) or whether they might include non-specific interactions, such as a general collapse in size at the very beginning, were left unanswered. Over time, the two main theories for how proteins fold became known as the "molten globule" or "hydrophobic collapse" (invoking non-specific interactions) and the "framework" or "nucleation/condensation" model (restricting pathways to native contacts only).

Over the years, the theoretical models for folding have converged somewhat, in part due to a better understanding of the structure of the so-called "unfolded state" and due to a more detailed description of kinetic folding intermediates. The "folding funnel" model (Nolting *et al.* 1997) has reconciled hydrophobic collapse with the nucleation-condensation model by envisioning a distorted, funicular energy landscape and a "minimally frustrated" pathway. The view remains of a gradual, counter-entropic search for the hole in the funnel as the predominant barrier to folding.

The 3D conformation of a protein may be compactly represented in a symmetrical, square, boolean matrix of pairwise, inter-residue contacts, or "contact map." The contact map provides a host of useful information about the protein's structure. For example, clusters of contacts represent certain secondary structures, and it also captures non-local interactions giving clues to the tertiary structure.

Below we describe how data mining can be used to extract valuable information from contact maps. More specifically we focus on two main tasks: 1) Given a database of protein sequences and their 3D structure in the form of contact maps, build a model to predict if pairs of amino acids are likely to be in contact or not. 2) Discover common (non-local) contact patterns or "features" that characterize physical "protein-like" contact maps. We show via experiments that our techniques are very effective in predicting and characterizing contacts. We further highlight promising directions of future work. For example, how mining can help in generating heuristic rules of contacts, and how one can generate plausible folding pathways in contact map conformational space.

The protein folding problem will be solved gradually, by many investigators who share their results at the bi-annual CASP (Critical Assessment of protein Structure Prediction) meeting (Moult *et al.* 1995), which offers a world-wide blind prediction challenge. Here, we will investigate how mining can uncover interesting knowledge from contact maps.

3 Mining Protein Contact Maps

[INSERT FIGURE 1 ABOUT HERE]

The contact map of a protein (see Figure 1) is a particularly useful representation of protein structure. Two amino acids in a protein that come into contact with each other form a non-covalent interaction (hydrogen-bonds, hydrophobic effect, etc.). More formally, we say that two residues (or amino acids) a_i and a_j in a protein are in *contact* if the 3D distance $\delta(a_i, a_j)$ is at most some threshold value t (a common value is $t = 7\mathring{A}$; $\mathring{A}$ stands for angstrom, $1\mathring{A} = 10^{-10}m$), where $\delta(a_i, a_j) = |\mathbf{r_i} - \mathbf{r_j}|$, and $\mathbf{r_i}$ and $\mathbf{r_j}$ are the coordinates of the α-Carbon atoms of amino acids a_i and a_j (an alternative convention uses beta-carbons for all but the glycines). We define *sequence separation* as the distance between two amino acids a_i and a_j in the amino acid sequence, given as $|i - j|$. A contact map for a protein with N residues is an $N \times N$ binary matrix C whose element $C(i, j) = 1$ if residues i and j are in contact, and $C(i, j) = 0$ otherwise.

The contact map provides a host of useful information. For example, clusters of contacts represent certain secondary structures: α-Helices appear as bands along the main diagonal since they involve contacts between one amino acid and its four successors; β-Sheets are thick bands parallel or anti-parallel to the main diagonal (see Figure 1). Tertiary structure may also be obtained

by reverse projecting into 3D space using the MAP algorithm (Vendruscolo, Kussell, & Domany 1997) or other distance geometry methods. Vendruscolo et al (Vendruscolo, Kussell, & Domany 1997) have also shown that it is possible to recover the 3D structure from even corrupted contact maps. For predicting the elusive global fold of a protein we are usually interested in only those contacts that are far from the main diagonal. In this chapter we thus ignore any pair of residues whose sequence separation $|i - j| < 4$.

Contact Map Mining Tasks In this chapter we focus on two main contact map mining tasks: 1) Given a database of protein sequences and their 3D structure in the form of contact maps, build a model to predict if pairs of amino acids are likely to be in contact or not. 2) Discover common (non-local) contact patterns or "features" that characterize physical "protein-like" contact maps. We also highlight other complementary mining tasks, namely, how to mine heuristic rules of contacts in real proteins, and how to generate folding pathways in contact map conformational space.

3.1 Classifying Contacts versus Non-Contacts

For training and testing of our classification model we used a non-redundant database of proteins of known structure, PDBselect:December 1998 (Hobohm & Sander 1994) containing 691 proteins and their sequence families. The proteins in the set have $< 25\%$ sequence similarity. Disordered or missing coordinates in the middle of a protein sequence were addressed by dividing the sequence at that point. This produces a set of 794 files, most of them containing an entire protein sequence, but some of these correspond to proteins that were split.

Given a PDB protein file, we have to transform the data into a format that can be easily mined, i.e., we need to prepare the data in tabular format where we have multiple attributes (columns) for each example (rows) or record. Since we are interested in predicting the contact between a pair of amino acids, we use each pair as an example in the training set, associated with a special *class* attribute indicating whether it is a contact (C) or non-contact (NC); amino acids a_i and a_j are in contact if $\delta(a_i, a_j) < 7\mathring{A}$, i.e., the distance between α-carbons of amino acids a_i and a_j is less then $7\mathring{A}$. Our new database has an entry showing the two amino acids and their class for each pair of amino acids for each protein. In order to avoid predicting purely local contacts we ignore all pairs

whose sequence separation $|i - j| < 4$. Note also that the number of contacts N_C is a lot smaller than the number of non-contacts N_{NC} for any protein.

We found that the percentage of contacts (or number of database entries with class 1) over all pairs is less than 1.7%. Across the 794 files, the longest sequence had length 907, while the smallest had length 35. There were 17,618,115 pairs over all proteins, while only 292,126 pairs were in contact. This database thus corresponds to a highly biased binary classification problem. That is, we have to build a mining model that can discriminate between contacts and non-contacts between amino acids pairs, where the examples are overwhelmingly biased toward the non-contacts.

3.2 Mining Methodology

Given these databases our goal is to find high support and high confidence rules of the form $A \Rightarrow C$ and $A \Rightarrow NC$, that discriminate between the contact pairs and the non-contact pairs, respectively. *Support* is the joint probability of rule antecedent and consequent, while *confidence* is the conditional probability of the consequent given the antecedent. Below we describe the mining/training and testing phases, where we learn from examples using the frequent closed itemsets (Zaki & Hsiao 2002), and then classify unseen examples as being contacts or non-contacts, respectively.

3.2.1 Mining on Known Examples

The goal of the mining phase is to learn from known contact and non-contact examples and build a model or rule set that discriminates between the two classes. We selected a random 90% of the files for training, out of a total of 794 files. The remaining 10% of the files were kept aside for testing the mined rule set.

Looking at all pairwise amino acids one can obtain two databases. One data set consists of all pairs that are in contact (from all the proteins), denoted as $\mathcal{D}_C$. The other database consists of all pairs not in contact, denoted as $\mathcal{D}_{NC}$. Since we are primarily interested in predicting the contacts rather than the non-contacts, we mine only on the contacts database $\mathcal{D}_C$. However, we do use the non-contacts database $\mathcal{D}_{NC}$ to prune out those patterns that are frequent in both sets. Building a discriminative rule set consists of the following steps, in order:

1. *Mining*: We use CHARM (Zaki & Hsiao 2002) to mine all the frequent closed itemsets in

$\mathcal{D}_C$. An *itemset* is a set of attribute values; it is frequent if it occurs at least *min_freq* times in the database, and it is *closed* if there is no proper superset with the same frequency. The frequency of an itemset is also called *support*, denoted $\sigma(X, \mathcal{D})$, where X is an itemset and $\mathcal{D}$ a database. Let's denote the set of mined frequent closed itemsets as $\mathcal{F}$.

2. *Counting*: We next compute the support of all itemsets in $\mathcal{F}$ in the non-contacts database $\mathcal{D}_{NC}$.

3. *Pruning*: We compute the probability of occurrence of each itemset in $\mathcal{F}$ in both the contact and non-contact databases. The probability of occurrence is simply the support of the itemset divided by the number of examples in the given dataset. For example, if itemset $X \in \mathcal{F}$, then the probability of its occurrence in $\mathcal{D}_C$ is given as $P(X, \mathcal{D}_C) = \sigma(X, \mathcal{D}_C)/|\mathcal{D}_C|$.

   As a first step in pruning we can remove all itemsets $X \in \mathcal{F}$ which have a greater probability of occurrence in the non-contact database than in the contact database, i.e., if $P(X, \mathcal{D}_{NC}) > P(X, \mathcal{D}_C)$. Actually, we compute the ratio of the contact probability versus the non-contact probability for X, and prune it if this ratio is less than some suitably chosen threshold ρ, i.e., we prune X if $P(X, \mathcal{D}_C)/P(X, \mathcal{D}_{NC}) < \rho$ (in our experiments we tried various values of ρ ranging from 2 to 10). In other words we want to retain only those itemsets that have a much greater chance of predicting a contact rather than a non-contact.

3.2.2 Testing on Unknown Examples

The goal of the testing phase is to find how accurately the mined set of rules predict the contacts versus the non-contacts in new examples not used for training. We used a random 10% of the files in the database for testing. The test set had a total of 2,336,548 pairs, out of which 35,987 or 1.54% were contacts. Since we do know the true class of each example, it is easy for us to find out how good our rules are for prediction.

For testing we generate a combined database $\mathcal{D}_t$ containing all pairs of amino acids in contact or otherwise. For each example we know the true class. We assign each example a predicted class using the following steps:

1. *Evidence Calculation:* For each example E in the test dataset $\mathcal{D}_t$, we compute which itemsets

in the set of mined and pruned closed frequent itemsets $\mathcal{F}$ are subsets of E. Let's denote the set of these itemsets as S. We next calculate the cumulative contact and non-contact support for example E, i.e., the sum of the supports of all itemsets in S in the contact and non-contact database. Finally, we compute the evidence for E being a contact, i.e., we take the ratio of the cumulative contact support over cumulative non-contact support, denoted as ρ_E, and defined as

$$\rho_E = \frac{\sum_{X \in S} \sigma(X, \mathcal{D}_C)}{\sum_{X \in S} \sigma(X, \mathcal{D}_{NC})}$$

Any example E with zero contact support is taken to be a non-contact and discarded, and only the examples or test pairs with positive contact support are retained for the next step.

2. *Prediction:* To make the final prediction if a test pair of residues is in contact or not, we sort all test examples E (with positive cumulative contact support) in decreasing order of contact evidence ρ_E. Finally, the top γ fraction of examples in terms of ρ_E are predicted to be contacts and the remaining $1 - \gamma$ fraction of examples as non-contacts. How γ is chosen will be explained below.

3.2.3 Model Accuracy and Coverage

In predicting contacts versus non-contacts for the test examples, we have to evaluate the mined model based on two metrics: *Accuracy* and *Coverage*. Furthermore, we are interested only in the prediction of contacts; thus accuracy and coverage is considered only for contacts. Accuracy is the ratio of correct contacts to the predicted contacts, while coverage is the percentage of all contacts correctly predicted. Thus, accuracy tells us how good the model is, while coverage tells us the number of contacts predicted.

More formally, let N_{tc} denote the number of true contacts in the test examples, N_{pc} the number of predicted contacts, N_{tpc} the number of true predicted contacts, and let N_a denote the number of all possible contacts, i.e., $N_a = (N-3) \times (N-2)/2$ (where N is the protein length), since the contact map is symmetric and pairs with sequence separation less than 4 are ignored. The accuracy of the model is given as:

$$A = N_{tpc}/N_{pc}$$

The coverage of the model is given as:

$$C = N_{tpc}/N_{tc}$$

The number of contacts predicted N_{pc} of course depends on how we chose γ, since the top γ fraction of test examples based on evidence is predicted as contacts. Since a protein is characterized by N_{tc} true contacts, we set $\gamma = N^*_{tc}/N^*_a$ and then predict the top γ fraction of examples as contacts. Note that N^*_{tc} and N^*_a denote the actual contacts and all pairs, respectively, that have positive contact support, since we discard examples with zero contact support. By adopting the above method, the number of predicted contacts is limited to those actually present in the protein. Further, this method has been used by previous approaches to contact map prediction (Fariselli & Casadio 1999; Olmea & Valencia 1997), and we retain it to facilitate comparison with previous results. Finally, we also compare our model against a random predictor. The accuracy of random prediction of contacts is defined as:

$$A_r = N_{tc}/N_a$$

3.3 How much information is there in Amino Acids Alone?

[INSERT FIGURE 2 ABOUT HERE]

Our first goal is to test how much information is contained in the amino acids only, i.e., for both training and testing, each example consisted of only the two amino acids a_i and a_j, along with their class (contact or non-contact). Figure 2 shows the accuracy, coverage, and improvement of the mined model over the random predictor for the test set. The accuracy and coverage is the mean value over all proteins. The figure plots the accuracy and coverage as percentages. It also plots the improvement of the model over the random predictor in ratios. The x-axis shows the *prediction factor*, which is related to the γ value (used to predict the top fraction of pairs as contacts). The prediction factor is in multiples of N^*_{tc}, the number of true contacts in the protein with positive contact evidence. For example, a value of 10 means that the top $(10 \times N^*_{tc})/N^*_a$ fraction of the examples are predicted as contacts. As one increases the number of contacts predicted (increasing γ ratios) the coverage increases and the accuracy decreases, exhibiting the classic accuracy versus coverage tradeoff, i.e., as one makes more and more predictions one clearly increases the chances

of finding more true contacts, but at the same time the fraction of true predictions decreases. Looking at the ratio line, we find that the information obtained from frequent pairs of amino acids leads to a model that is around two times better than a random predictor (i.e., randomly tagging a test pair as contact or non-contact). Thus we observe that while amino acids alone have some information that can be used to predict contacts versus non-contacts, this information is not too good.

The left-most graph in Figure 2 shows the accuracy and coverage of the predictor over test proteins of all lengths. The other two figures on the right show how accuracy and coverage change with protein length. We have divided the test proteins into four bins: $1 \leq N < 100$, $100 \leq N < 170$, $170 \leq N < 300$, and $300 \leq N$.

We find that over all proteins the amino acids by themselves can be used to give an 8.5% accuracy, 1.5% coverage, and an improvement over a random predictor by a factor of 2.4. Note also the interesting trend in the graph. As the prediction factor increases we get better and better coverage, but the accuracy trails off. Which value to choose for the prediction factor depends on what is more important. It has been reported in (Vendruscolo, Kussell, & Domany 1997) that the 3D structure of proteins can be recovered quite robustly, even from corrupted contacts maps. This implies that coverage should have an higher weight than accuracy. In any case, if we had to choose a value representing the best trade-off, we can pick the point where the accuracy and coverage curves intersect. This happens for a prediction factor of 7, where we have roughly 7% accuracy and coverage, and which is 2 times better than random. For 6.3% accuracy we can increase coverage to 14%.

When we consider the results for proteins of different lengths, we find the same trade-off between accuracy and coverage. Looking at the crossover point, we get around 13% accuracy and coverage for short proteins with $N < 100$, 6% for $100 \leq N < 170$, 4.5% for $170 \leq N < 300$, and around 2% of longer proteins.

3.4 Using Local Structures for Contact Prediction

Previous work (Bystroff & Baker 1998) showed that some small, fast-folding regions of a protein may be identified by their sequence alone. A library of 262 short sequence patterns that fold fast was compiled by cluster analysis of the database of known protein structures. Evidence from NMR

and molecular dynamics simulations found that these database-derived sequence motifs are folding initiation sites or I-sites.

Subsequent work developed HMMSTR (Bystroff, Thorsson, & Baker 2000), a hidden Markov model (HMM) for generalized protein sequence. Generalized HMMs are directed, cyclic graphs where each node is a single symbol emitter. HMMSTR is a "parallel HMM" which emits, from one Markov state, a single amino acid and a symbol for the backbone phi and psi angles. HMMSTR was designed as condensed representation of all known local structure motifs, as defined in the I-sites database. Therefore, collectively, the allowed paths through the directed graph represent all known local structure motifs, including alpha-helices, beta strands, helix caps, and a variety of loops and turns. They are represented in the model in proportion to the frequency at which they are found in the database of protein structures. The HMMSTR model from the merged I-sites motifs, results in 282 Markov states, as shown in Figure 3. Each HMMSTR state can produce, or "emit", amino acids and structure symbols according to a probability distribution specific to that state.

[INSERT FIGURE 3 ABOUT HERE]

3.4.1 Using HMMSTR Output

After HMMSTR is built we again took the 691 proteins from PDBselect and computed for each protein the optimal HMMSTR states that agree with the observed amino acids in the protein. The output probability distribution of all the states thus chosen for a protein sequence is used as input for the frequent itemset mining algorithm. In fact, rather than a single state associated with a given residue, we have available the probability that the residue at the given position is associated with all the states of HMMSTR, i.e., we have available $P(q_i|a_j)$ for all the 282 HMMSTR states ($1 \leq i \leq 282$) for all the residues in a given protein ($1 \leq j \leq N$), where q_i is a Markov state, a_j is a given residue, and N is the length of the protein.

For each residue we also know the amino acid at that position. Additional outputs describe the probability of observing a particular amino acid, secondary structure, backbone angle region, or structural context descriptor. We also have the spatial coordinates of the α-Carbon atom $\langle x, y, z \rangle$; a distance vector of length n giving the distance of this residue from all other residues in the protein; and the 20 amino acid profiles for that position.

A protein data file looks like this:

```
PDB Name: 153l_
Sequence Length: 185

Position: 1
Residue: R
Coordinates: 0.0  -73.2  177.6
Profile: 0.0 ... 1.0 ... 0.0 #20 Values
HMMSTR State Probabilities:
        0.0 ... 0.7 .... 0.3 ... 0.0 #282 Values
Distance Vector: 0 3 5 ... 18 15 13 #185 Values, i.e., Seq Length

Position: 2
Residue: T
Coordinates: -124.4  0.2  -177.1
Profile: 0.0 ... 1.0 ... 0.0 #20 Values
HMMSTR State Probabilities:
        0.0 ... 0.9 ... 0.1 ... 0.0 #282 Values
Distance Vector: 3 0 3 ... 15 13 10 #185 Values

...

Position: 185
Residue: Y
Coordinates: -88.7  0.0  0.0
Profile: 0.0 ... 0.4 ... 0.6 ... 0.0 #20 Values
HMMSTR State Probabilities:
        0.0 ... 0.2 ... 0.5 ... 0.3 ... 0.0 #282 Values
Distance Vector: 15 13 10 ... 5 3 0 #185 Values
```

We have a file like the one shown above for all of the 691 non-redundant set of proteins from PDBSelect. Disordered or missing coordinates in the middle of a protein sequence were addressed by dividing the sequence at that point. This produces a set of 794 files, most of them containing an entire protein sequence, but some of these correspond to proteins that were split. Using all the available information from HMMSTR we create a tabular dataset with a row per pair of amino acids, and with multiple features (columns) comprised of the amino acid profiles, HMM state probabilities, other structural context information, and a class (contact or non-contact).

Since association rules only work for categorical attributes, we need to convert the continuous state probabilities into discrete values. To do this we take the ratio of each of the 282 HMMSTR state probabilities for a_i against the background or prior probability of an amino acid being in that state; if the ratio is more than some threshold we include the state in the context of a_i, else we ignore it. We repeat the same process for a_j. Using a similar thresholding method one can incorporate the amino acid profiles for positions i and j. With all this context information for both a_i and a_j we obtain a new database to be used to find the frequent closed itemsets characterizing the contacts and non-contacts. In summary the database has the following columns for pairs of amino acids over all proteins:

```
Protein and Position Information: ProteinID  PairID i j |i-j|
Amino Acids and Context: ai aj di dj ri rj ci cj
Profile: pi1 pi2 ... pj1 pj2 ...
HMMSTR: qi1 qi2 ... qj1 qj2 ...
Class: C or NC
```

Note that the number of columns can be variable for different pairs depending on the profile and HMMSTR state probabilities. p_{i_1}, p_{i_2}, etc. show the other amino acids that can appear in position i (provided the probability is more than some threshold), and finally q_{i_1}, q_{i_2}, etc. show HMMSTR states with probabilities more than some factor of the prior probability of those states. For additional details see (Zaki, Jin, & Bystroff 2000).

3.4.2 Mining Predictive Rules

[INSERT FIGURE 4 ABOUT HERE]

To mine the rules predictive of contacts, we consider the augmented database of examples obtained by adding in the amino acid profile, structural contextual information, and the Markov states for each residue. We then mine the frequent patterns that distinguish contacts from non-contacts as described in Section 3.2.

Figure 4 shows the prediction results using this augmented database. If we look at the cross-over point we get almost 19% accuracy and coverage, while the model is 5.2 times better than random. For 18% accuracy we can get coverage of 25% (still 5.1 times better than random).

If we look at proteins of various lengths in Figure 4, we find that for $N < 100$, we get 26% accuracy and 63% coverage at the extreme point (4 times over random). For $100 \leq N < 170$ we get 21.5% accuracy and 10% coverage towards the end (6 times over random), for $170 \leq N < 300$ we get 13% accuracy and around 7.5% coverage (6.5 times over random), and for longer proteins we get 9.7% accuracy and 7.5% coverage (7.8 times over random).

[INSERT FIGURE 5 ABOUT HERE]

Figure 5 shows the results in a slightly different format. It plots the improvement in coverage/accuracy over a random prediction. For example, at around 1% coverage we have a model that is 7.25 better than the random predictor, while at 25% coverage the model is about 5.1 better than random prediction. For the accuracy plot, at 18% accuracy the model is 5.1 better than random, while at 25% accurayc it is 7.25 better.

The above results are comparable to or better than the results recently reported in (Zhao & Kim 2000), where they examined pairwise amino acid interactions in the context of secondary structural environment (helix, strand, and coil), and used the environment dependent contact energies for contact prediction experiments. For about 25% coverage our model does more than 5 times better than the random predictor, as compared to the 4 times improvement reported in (Zhao & Kim 2000).

We believe these results are the best, or at least comparable to those reported so far in the literature on contact map prediction (Fariselli & Casadio 1999; Olmea & Valencia 1997). For example, Fariselli and Casadio (Fariselli & Casadio 1999), used a Neural Network based approach

over pairs database, with other contextual information like sequence context windows, amino acid profiles, and hydrophobicity values. They reported an 14.4% accuracy over all proteins, with an 5.4 times improvement over random. They also got 18% accuracy for short proteins with an 3.1 times improvement over random. Olmea and Valencia (Olmea & Valencia 1997) on the other hand used correlated mutations in multiple sequence alignments for contact map prediction. They added other information like sequence conservation, alignment stability, contact occupancy, etc. to improve the accuracy. They reported 26% accuracy for short proteins, but they did not report the result for all proteins. While we believe that our hybrid approach does better, we should say that direct comparison is not possible, since previous works used a different (and smaller) PDB_select database for training and testing. One draw back of these previous approaches is that they do not report any coverage values, so it is not clear what percentage of contacts are correctly predicted. Another approach to contact map prediction was presented in (Thomas, Casari, & Sander 1996), which was based on correlated mutations. They obtained an accuracy of 13% or 5 times better than random.

3.4.3 Predicted Contact Map

[INSERT FIGURE 6 ABOUT HERE]

Figure 6 shows the predicted contact map for the protein $2igd$ from Figure 1. We got 35% accuracy and 37% coverage for this protein. The figure shows the true contacts, the contacts correctly predicted, and all the contacts predicted (correctly or incorrectly). Our prediction was able to capture true contacts representing portions of all the major interactions. For example, true contacts were found for the alpha helix, the two anti-parallel beta sheets and the parallel beta sheet. However, some spurious clusters of contacts were also discovered, such as the triangle in the lower left corner. In the next section, we will show how one can eliminate such false contacts by recognizing the fact that they never occur in real proteins, since such a cluster is physically impossible.

4 Characterizing Physical, Protein-like Contact Maps

Proteins are self-avoiding, globular chains. A contact map, if it truly represents a self-avoiding and compact chain, can be readily translated back to the three-dimensional structure from which it came. But, in general, only a small subset of all symmetric matrices of ones and zeros have this property. The task is to output a contact map that both satisfies the geometrical constraints and is likely to represent a low-energy structure. Interactions between different subsequences of a protein are constrained by a variety of factors. The interactions may be initiated at several short peptides (initiation sites) and propagate into higher-order intra or inter-molecular interactions. The properties of such interactions depend on (1) the amino acid sequence corresponding to the interactions, (2) the physical geometry of all interacting groups in three dimensions, and (3) the immediate contexts (linear, and secondary components for tertiary structural motifs) within which such interactions occur.

We describe below in detail the method that we use for mining frequent dense patterns or structural motifs in contact maps. These motifs represent the typical non-local structures that appear in physical protein-like contact maps, and which can be used to improve the quality of contact map prediction by eliminating impossible contacts (those that never occur in real proteins). Briefly, there are three major stages for the approach: (1) Data generation, which involves creating a large set of protein-like contact maps, (2) Mining, which involves computation of all the frequent dense patterns, and (3) Pruning mined frequent patterns and integration of these patterns with biological data.

4.1 Generating a Database of Protein-like Structures

12,524 protein-like structures were generated using the following procedure. We first generate 60-residue random amino acid sequences using HMMSTR (Bystroff, Thorsson, & Baker 2000) model based on the I-sites library of sequence-structure motifs. Here it is used as a stochastic emitter of Markov states. A Markov state sequence, length 60, is generated by starting in a random state and selecting the next state stochastically, using the state-state transition probabilities. The resulting state sequence is converted to an amino acid sequence by selecting each amino acid stochastically from the Markov state "profile" (amino acid probability distribution for the given

state). The resulting random amino acid sequences are (locally) protein-like. We then use the ROSETTA server (Simons *et al.* 1997) to generate the structures for those random sequences. The algorithm underlying the ROSETTA server is the so called Monte Carlo fragment insertion. It picks fragments from the I-sites library based on sequence homology, calculates the energy of the new structure, then either discards or keeps the fragments based on Monte Carlo. This is repeated until convergence.

To describe the procedure in greater detail, first a multiple sequence alignment is generated by PSI-BLAST(Altschul *et al.* 1997). The alignment is converted into a profile, which is used to generate the move-set (from which the inserted fragment is chosen from). We use a sliding window of size 3 or 9 to move along the sequence and match each of 3 or 9 residue fragments from the target sequence with the motifs of the I-sites library. The 25 highest-scoring matches to the motifs are selected and put into the move-set. For a single move, a fragment is chosen randomly from the move-set and inserted as a part of the structure. The energy of the current structure is evaluated with an energy function. If the new energy passes the Metropolis criterion (i.e., an exponentially distributed random variable), then the change is kept, otherwise the inserted fragment is discarded and the original structure is restored. 12,000 cycles of simulated annealing are applied to broaden the search for the global energy.

The five structures with the lowest energies are selected as the candidates of the resulting structure. In order to exclude the structures that are not as compact as natural proteins, we examine the radius of gyration (rg) of each candidate: $rg = \frac{\sum_{i=1}^{N}\sum_{j=i+1}^{N} dist(i,j)}{N^2-N}$, where $dist(i,j)$ is the distance between i and j. The higher the rg, the less compact the structure. Of the five, low-energy candidates, we select only those having $rg < 11.0$ as our results. The final 12,524 structures were converted into contact maps for further data mining analysis, i.e., for each structure we noted the pairwise residue distances and created the contact map with $7\mathring{A}$ cutoff threshold.

4.2 Mining Dense Patterns in Contact Maps

To enumerate all the frequent 2D dense patterns we scanned the database of 12524 contact maps with a 2D sliding window of a user specified size. For all structures, any sub-matrix under the window that had a minimum "density" (the number of '1's or contacts) was captured. For a $N \times N$

contact map, using a 2D $W \times W$ window, there are $(N - W) \times (N - W)/2$ possible submatrices. We have to tabulate those which are dense, using different window sizes. We chose window sizes from 5 to 10, to capture denser contacts close to the diagonal (i.e., short-range interactions), as well as the sparser contacts far from the diagonal (i.e., long-range interactions).

Due to the intrinsic constraints in protein secondary and tertiary structures, the density of the contacts naturally decreases with chain separation distance (in the contact map, the distance from the main diagonal). In order to also capture these less dense but possibly significant patterns, we scaled the minimum density cutoff as a function of the chain separation distance. The density weighing function we used is as follows:

$$min\_d = minDensity * (1 - (|i - j|)/N)$$

where *minDensity* is the user specified density threshold, i and j are the starting indices of a window in the 2D contact map (here it represents the top left position of a sub-matrix).

Counting Dense Patterns As we slide the $W \times W$ window, the sub-matrix under the window will be added to a dense pattern list if its density exceeds the *min_d* threshold. However, we are interested in those dense patterns that are frequent, i.e., when adding a new pattern to the list of dense patterns we need to check if it already exists in the list. If yes, we increase the frequency of the pattern by one, and if not, we add it to the list initialized with a count of one.

The main complexity of the method stems from the fact that there can be a huge number of candidate windows. For instance with a window size of $W = 5$, and for $N = 60$, we have 1485 windows per contact map. This translates to 18,598,140 possible windows for the 12,524 contact maps. At the other extreme, for $W = 10$, there are 15,341,900 windows. Of these windows only relatively few will be dense, since the number of contacts is a lot less then the number of non-contacts. Still we need an efficient way of testing if two submatrices are identical or not. Assume that P is the number of current dense patterns of size $W \times W$. The naive method to add a new pattern is to check for equality against all P patterns, where each check takes $O(W^2)$ time, giving a total time of $O(W^2P)$ per equality check. A better approach is to use a hash table of dense patterns instead of a list. This can cut down the time to $O(W^2)$ per equality check if a suitable hash function is found. We will describe below how we can further improve the time to just $O(W)$

per check.

Counting Dense Patterns via Hashing For fast hashing and equality checking, we encoded each sub-matrix in the following way: Each row of the $\{0, 1\}$ sequence will be converted into a number corresponding to the binary value represented by the sequence, and all the numbers computed this way will be concatenated into a string. For example the 5×5 submatrix below is encoded as the string: $0.12.8.8.0$.

```
      submatrix       binary value of row
        00000         0
        01100         12
        01000         8
        01000         8
        00000         0
  stringId (concatenate row values) = 0.12.8.8.0
```

According to our submatrix encoding scheme, each dense $W \times W$ window M is encoded as the string $stringId(M) = v_1.v_2.\cdots.v_W$, where v_i is the value of the row treated as a binary string. For fast counting we employed a 2-level hashing scheme. For the first level we use the sum of all the row values as the hash function:

$$h_1(M) = \sum_{i=1}^{W} v_i$$

The second level hashing uses the *stringId* as the hash key and therefore is an exact hashing, i.e., $h_2(M) = stringId(M)$. The use of this 2-level hashing scheme allows us to avoid many unnecessary checks. The first level hashing (h_1) narrows the potential matching submatrices to a very small number. Then the second level hashing (h_2) computes the exact matches. Computing h_1 and h_2 both take $O(W)$ time; thus equality check of a submatrix takes $O(W)$ time.

After all dense areas are hashed into the second level slot, the support counts for each unique *stringId* of the dense patterns are collected, and those patterns that have support counts more than a user specified *minSupport* will be considered frequent dense patterns and will be output for analysis.

4.3 Pruning and integration

After obtaining mined patterns that are frequent and are relatively dense, we pruned them using a number of heuristics in order to extract biologically meaningful structural motifs. All the potential structure motifs fall into two categories: that of secondary structures which primarily consist of alpha helices or beta sheets (parallel or anti-parallel), and that of tertiary structures which involve interactions between secondary structure components. For example, alpha helices, beta sheets and beta turn regions can have multiple contacts between them, such that components that are farther away in linear sequence could be brought together to form functional groups. These tertiary structures are particularly important for biological processes such as high-specificity binding of ligands and receptors.

Due to the intrinsic characteristics and constraints of the secondary structures, alpha helices form contact patterns that line along the main diagonal of the contact matrix, whereas beta-sheets form contact patterns that are either perpendicular (anti-parallel beta sheets) or parallel to the main diagonal. The positions at which these patterns could occur are also constrained. In contrast, the contact patterns that belong to a tertiary structure (interactions between two secondary structural components) are more likely to be less dense and distant from the main diagonal. Furthermore, they do not have definitive contact shapes compared to the well defined secondary structure groups. Thus it is difficult to extract these and isolate them from other patterns. We took several approaches to attack the difficulty: first, as described in the previous section, we weighed the minimum density according to the distance of each sub-matrix to the main diagonal, such that distal regions have smaller density threshold than proximal regions; second, by varying window-size until an appropriate size is reached, we can differentiate the tertiary interactions from the rest by measuring the density.

Once dense patterns are found, the final step is to incorporate the sequence information with them. That is, for each dense pattern and its occurrences in the different contact maps (i.e., in different 60 residue protein segments in PDB format), we note the protein id, the start positions of the window (given as (X,Y) coordinates), the amino acid sub-sequences associated with the X and Y dimensions of the window, and the type of interaction. This information is then used to visualize the mined patterns. An example dense pattern with associated information is shown below (only a

few occurrences are shown, even though the pattern occurs 170 times):

```
StringId:0.12.8.8.0, Support = 170
00000
01100
01000
01000
00000

Occurrences:
pdb-name        (X,Y)    X_sequence   Y_sequence      Interaction
1070.0          52,30      ILLKN         TFVRI        alpha::beta
1145.0          51,13      VFALH         GFHIA        alpha::strand
1251.2          42,6       EVCLR         GSKFG        alpha::strand
1312.0          54,11      HGYDE         ATFAK        alpha::beta
1732.0          49,6       HRFAK         KELAG        alpha::beta
2895.0          49,7       SRCLD         DTIYY        alpha::beta
...
```

By applying above methods, two major groups of patterns were isolated, as described in the next section: one group being major secondary structure components such as alpha and beta sheets, another group being tertiary structural motifs involving two secondary structural components.

4.4 Experimental Results

[INSERT TABLE 1 ABOUT HERE]

We discovered frequent submatrix patterns whose supports reach 1 to 2% when using a sliding window of size 8 and a minimum density of 0.125. These patterns turned out to correspond to beta sheets secondary structures (parallel and antiparallel ’1’s in the binary contact map). Examples of the most frequent dense patterns are given in Table 1.

The second class of patterns involve interactions between different secondary structures. The kinds of the interactions revealed with the frequent dense patterns differ in terms of the involved

secondary structure components, multiplicity of interacting atoms, and the contexts (linear and secondary) surrounding such interactions. Roughly, based on the type of secondary components that are involved, we observed that the mined tertiary structure motifs can be categorized into the following classes:

1. Alpha helix a_1 :: Alpha helix a_2
2. Alpha helix a :: Beta sheet b
3. Alpha helix a :: Beta turn bt
4. Beta sheet b :: Beta turn bt

[INSERT FIGURE 7 ABOUT HERE]

Figure 7 shows an example of each of these four types of interactions. The above classes can be further divided into sub-classes according to the number of contacts involved in each component, multiplicity of interacting atoms (one to one, one to many, or many to many), sequence specificities, and the linear/secondary structural contexts of the interaction. We are currently creating a library of all possible non-local interactions in "real" contact maps.

5 Future Directions for Contact Map Mining

Besides predicting and characterizing contact potentials between residues, one can formulate additional important mining tasks to extract interesting knowledge from contact maps. Here we highlight two such tasks: how to mine heuristic rules of contacts and how to mine folding pathways.

5.1 Heuristic Rules for "Physicality"

Simple geometric considerations may be encoded into heuristics that recognize physically possible and protein-like patterns within contact maps, C. For example, we may consider the following to be rules that are never broken in true protein structures:

If $C(i,j) = 1$ and $C(i+2, j+2) = 1$, then $C(i, j+2) = 0$, and $C(i+2, j) = 0$.

If $C(i+2, j) = 1$ and $C(i, j+2) = 1$), then $C(i,j) = 0$, and $C(i+2, j+2) = 0$.

These rules encode the observation that a beta sheet (contacts in a diagonal row) is either

parallel or anti-parallel, but not both. Another example may be drawn from contacts with alpha helices.

If $C(i, i+4) = 1$ and $C(i, j) = 1$ and $C(i+4, j) = 1$, then $C(i+2, j) = 0$

This follows from the fact that $i + 2$ lies on the opposite side of the helix from i and $i + 4$, and therefore cannot share contacts with non-local residue j. Local structure may be used in the definition of the heuristics. For example, if an unbroken set of $C(i, i + 4) = 1$ exists, the local structure is a helix, and therefore, for all $|j - i| > 4$ in that segment, $C(i, j) = 0$. The question is whether one can mine these rules automatically.

5.1.1 Finding Heuristics by Data Mining

Consider the contact map for the parallel beta sheet shown in Table 1. We can discover "positional" rules, i.e., the heuristic geometric rules by considering an appropriate neighborhood around each contact $C(i, j)$ and noting down the relative coordinates of the other contacts and non-contacts in the neighborhood, conditional on the local structure type(s). Consider a lower 1-layer (denoted LL1) neighborhood for a given point, $C(i, j)$. This includes all the coordinates within $i + 1$ and $j + 1$, i.e. each point has 3 other points in its LL1 neighborhood, namely $C(i, j + 1)$, $C(i + 1, j)$ and $C(i + 1, j + 1)$. If we repeat this process for each point we obtain a database which can be mined for frequent combinations.

For instance looking at the parallel beta sheet submatrix in Table 1 we would get the set $C(i, j) = 1$, $C(i + 1, j) = 0$, $C(i, j + 1) = 0$, $C(i + 1, j + 1) = 1$ for each contact. If one were to do this for many other proteins one would find the pattern "If $(Ci, j) = 1$ and $C(i + 1, j + 1) = 1$, then $C(i, j + 1) = 0$ and $C(i + 1, j) = 0$," among several others. This is the same rule deduced by hand above.

Other patterns can be found by defining an appropriate neighborhood, which can be t layers thick (where t is the maximum coordinate difference between points), and can encompass all points within t or some subset of that region. From each of these we can construct examples which can be mined for frequent patterns to obtain heuristic contact rules. We can also incorporate sequence information to mine more complicated patterns. We are currently developing techniques to mine such heuristic rules of contact automatically.

5.2 Rules for Pathways in Contact Map Space

Currently, there is no strong evidence that specific non-native contacts (i.e., those that are not present in the final 3D structure) are required for the folding of any protein. Many simplified models for folding, such as lattice simulations, tacitly assume that non-native contacts are "off pathway" and are not essential to the folding process. Therefore, we choose to encode the assumption of a "native pathway" into our algorithmic approaches. This simplifying assumption allows us to define potential folding pathways based on a known three-dimensional structure. We may further assume that native contacts are formed only once in any given pathway.

[INSERT FIGURE 8 ABOUT HERE]

The formation of a contact between two amino acids in the chain, an elemental part of a folding pathway, forms a cycle and incurs the loss of configuration entropy in that piece of chain. Loss of configuration entropy (i.e. the ordering of the backbone) is the main opposing force to protein folding. Entropy loss must be balanced by favorable energetic interactions, such as hydrophobic contacts or hydrogen bonds. We may impose a limit onto our pathway model by assuming that any new contact must occur within S_{max} residues of a contact that is already formed. In other words we assume that $U(i,j) \leq S_{max}$, where $U(i,j)$ is the number of "unfolded" residues between i and j. Intervening residues are "folded" when a contact forms. This will be called the "condensation rule." In computational experiments using lattice models of folding, the pathways that follow this rule are dominant.

A pathway in contact map space consists of a time-ordered series of contacts. The pathway is initiated by high-confidence I-sites, and thereafter it follows a tree-search format (Figure 8). Each level of the tree is the addition of a contact that satisfies the condensation rule. A maximum of M branches can be selected based on the energy. In addition, contacts that are not physically possible can be rejected, using heuristics or mined rules for physicality. Identical branches (same set of contacts, different order) can of course be merged.

Note, that the energy criterion includes an entropic penalty proportional to $U(i,j)$, the number of unfolded residues between i and j. If $U(i,j) = 0$, there is no entropic penalty, since it implies that a previous contact has already cyclized that segment. We believe that the rules for folding in contact map space are consistent with the accepted biophysical theories of folding, while confining

the search to a greatly simplified and reduced space. We are currently developing methods to discover the folding pathways in the contact map space. It is worth observing that while the structure prediction problem has attracted a lot of attention, the pathway prediction problem has received almost no attention. However, the solution of either task would greatly enhance the solution of the other, hence it is natural to try to solve both of these problems within a unifying framework. Our current work is a step toward this unified approach.

6 Summary

In this chapter we illustrated two main applications of data mining in protein structure prediction, namely, classifying contacts versus non-contacts, and recognizing common 2D patterns in protein-like contact maps. For the former we described data mining techniques to predict 3D contact potentials among protein residues using a hybrid approach. We first used Hidden Markov Models to extract folding initiation sites, and then applied frequent closed itemset mining to discover contact potentials. The new hybrid approach achieved accuracy better than those reported previously.

For the latter problem, we described a novel string encoding and hashing technique to extract all the dense submatrices by sliding a 2D window across the contact map. We discovered common non-local patterns using a dynamic density threshold and several pruning techniques. Using our approach we were able to extract some typical interactions that occur in "physical" protein-like contact maps. These patterns can be used to refine the contact map predictions to remove non-physical predictions. Currently we are in the process of compiling a library of such non-local interactions between different secondary structures. This library would be analogous to the I-sites library, but while the I-sites library records the common motifs for short contiguous segments (3-19 residues), the new library will record interactions between non-contiguous segments.

We would like to close by saying that in general data mining approaches seem ideally suited for Bioinformatics, since it is data-rich, but lacks a comprehensive theory of life's organization at the molecular level. The extensive databases of biological information create both challenges and opportunities for developing novel KDD methods. We hope that this chapter has been successful in highlighting some aspects of how mining can be applied in the protein prediction domain.

References

Altschul, S.; Madden, T.; Schaffer, A.; Zhang, J.; Zhang, Z.; Miller, W.; and Lipman, D. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Research, 25(17), 3389-402.

Anfinsen, C., and Scheraga, H. (1975). Experimental and theoretical aspects of protein folding. Adv. Protein Chemistry, 29, 205-300.

Bystroff, C., and Baker, D. (1998). Prediction of local structure in proteins using a library of sequence- structure motifs. Journal of Molecular Biology, 281(3), 565-77.

Bystroff, C.; Thorsson, V.; and Baker, D. (2000). HMMSTR: A hidden markov model for local sequence-structure correlations in proteins. Journal of Molecular Biology, 301, 173-190.

Fariselli, P., and Casadio, R. (1999). A neural network based predictor of residue contacts in proteins. Protein Engineering, 12(1), 15-21.

Hobohm, U., and Sander, C. (1994). Enlarged representative set of protein structures. Protein Science, 3(3), 522-524.

Levinthal, C. (1968). Are there pathways for protein folding? J. Chem. Phys., 65, 44-45.

Moult, J.; Pedersen, J.; Judson, R.; and Fidelis, K. (1995). A large-scale experiment to assess protein structure prediction methods. Proteins, 23(3), ii-v.

Nolting, B.; Golbik, R.; Neira, J.; Soler-Gonzalez, A.; Schreiber, G.; and Fersht, A. (1997). The folding pathway of a protein at high resolution from microseconds to seconds. Proc. Natl. Acad. Sci. USA, 94(3), 826-30.

Olmea, O., and Valencia, A. (1997). Improving contact predictions by the combination of correlated mutations and other sources of sequence information. Folding & Design, 2, S25-S32.

Simons, K.; Kooperberg, C.; Huang, E.; and Baker, D. (1997). Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and bayesian scoring functions. Journal of Molecular Biology, 268(1), 209-25.

Thomas, D.; Casari, G.; and Sander, C. (1996). The prediction of protein contacts from multiple sequence aligments. Protein Engineering, 9(11), 941-48.

Vendruscolo, M.; Kussell, E.; and Domany, E. (1997). Recovery of protein structure from contact maps. Folding & Design, 2(5), 295-306.

Wolf, Y. I.; Grishin, N. V.; and Koonin, E. V. (2000). Estimating the number of protein folds and families from complete genome data. Journal of Molecular Biology, 299(4), 897-905.

Zaki, M. J., and Hsiao, C.-J. (2002). CHARM: An efficient algorithm for closed itemset mining. In 2nd SIAM International Conference on Data Mining.

Zaki, M. J.; Jin, S.; and Bystroff, C. (2000). Mining residue contacts in proteins using local structure predictions. In IEEE International Symposium on Bioinformatics and Biomedical Engineering.

Zhao, C., and Kim, S.-H. (2000). Environment-dependent residue contact energies for proteins. Proc. Natl. Acad. Sci. USA, 97(6), 2550-5.

List of Figure Captions

Figure 1. A Contact Map: 3D structure for protein G (PDB file 2igd, $N = 61$) and its contact map showing parallel (top left cluster) and anti parallel sheets (bottom left and top right cluster), and helix features (thin cluster close to main diagonal).

Figure 2. Predicting Contacts Using Only Amino Acids

Figure 3. HMMSTR Hidden Markov Model

Figure 4. Predicting Contacts Using HMMSTR States and Amino Acids

Figure 5. Improvement over Random Prediction

Figure 6. Predicted Contact Map (PDB file 2igd)

Figure 7. Frequent Patterns between Secondary Structures: 1) Alpha Helix - Alpha Helix 2) Alpha Helix - Beta Sheet, 3) Alpha Helix - Beta Turn, 4) Beta Sheet - Beta Turn

Figure 8. Folding Pathways in Contact Map: Large triangles represent the contact map, initially empty. Each branch defines a region of the contact map by setting pairs of amino acids to be in contact (dots) or not (space). Each node in the tree corresponds uniquely to a three dimensional structure (left), where the dotted lines represent segments of the chain that have undefined structure

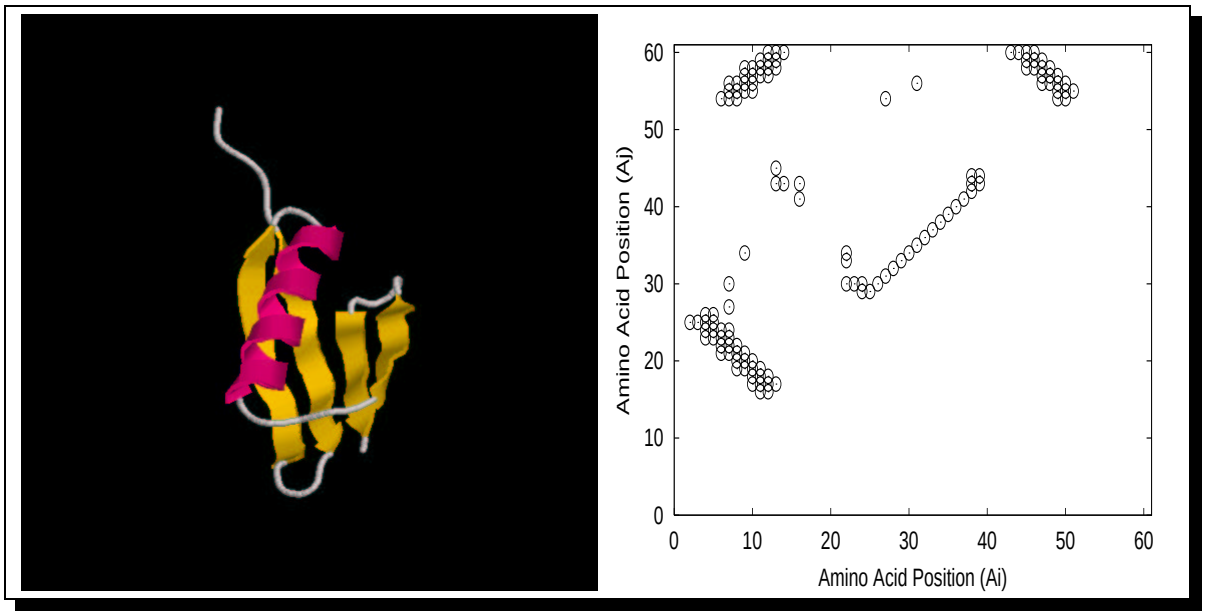

Figure 1: A Contact Map: 3D structure for protein G (PDB file 2igd, $N = 61$) and its contact map showing parallel (top left cluster) and anti parallel sheets (bottom left and top right cluster), and helix features (thin cluster close to main diagonal).

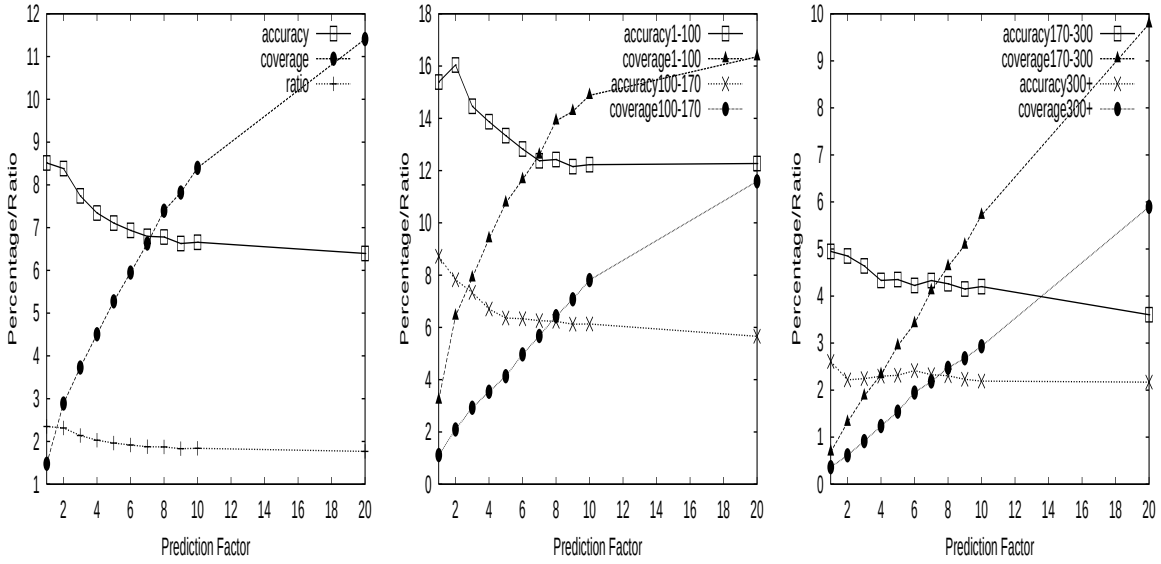

Figure 2: Predicting Contacts Using Only Amino Acids

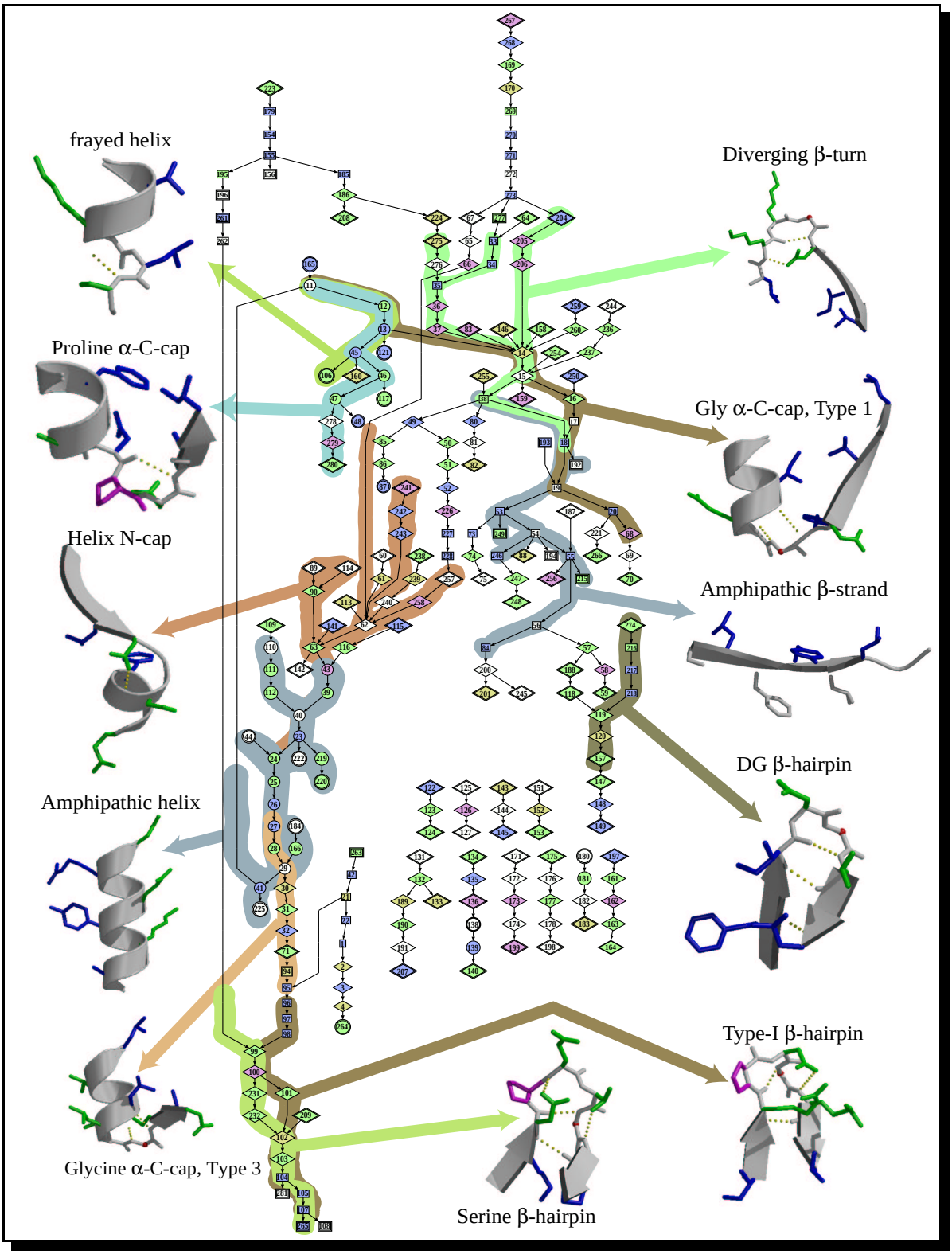

Figure 3: HMMSTR Hidden Markov Model

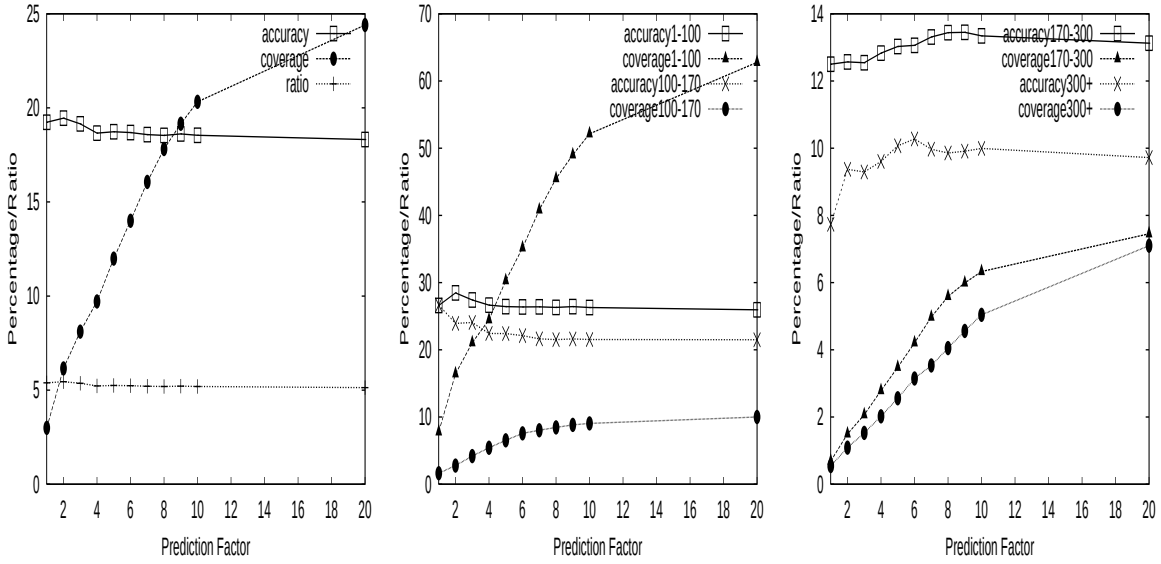

Figure 4: Predicting Contacts Using HMMSTR States and Amino Acids

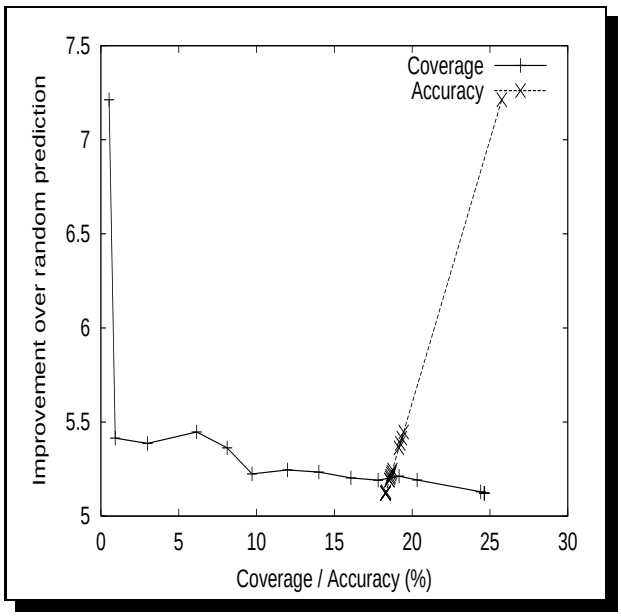

Figure 5: Improvement over Random Prediction

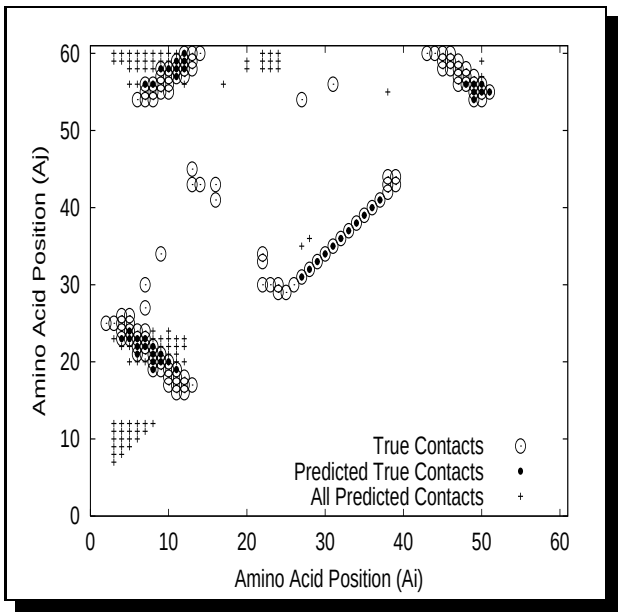

Figure 6: Predicted Contact Map (PDB file 2igd)

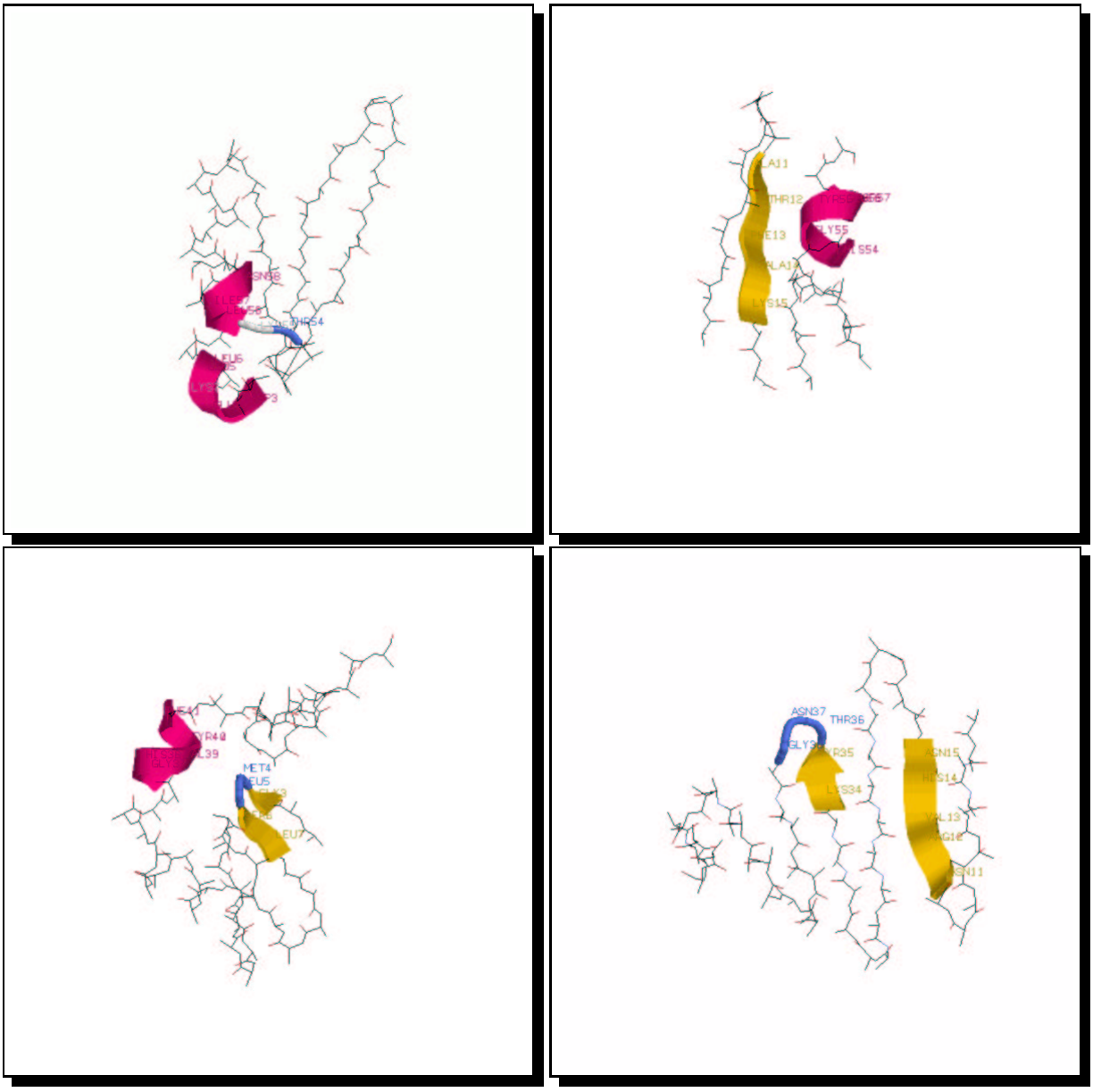

Figure 7: Frequent Patterns between Secondary Structures: 1) Alpha Helix - Alpha Helix 2) Alpha Helix - Beta Sheet, 3) Alpha Helix - Beta Turn, 4) Beta Sheet - Beta Turn

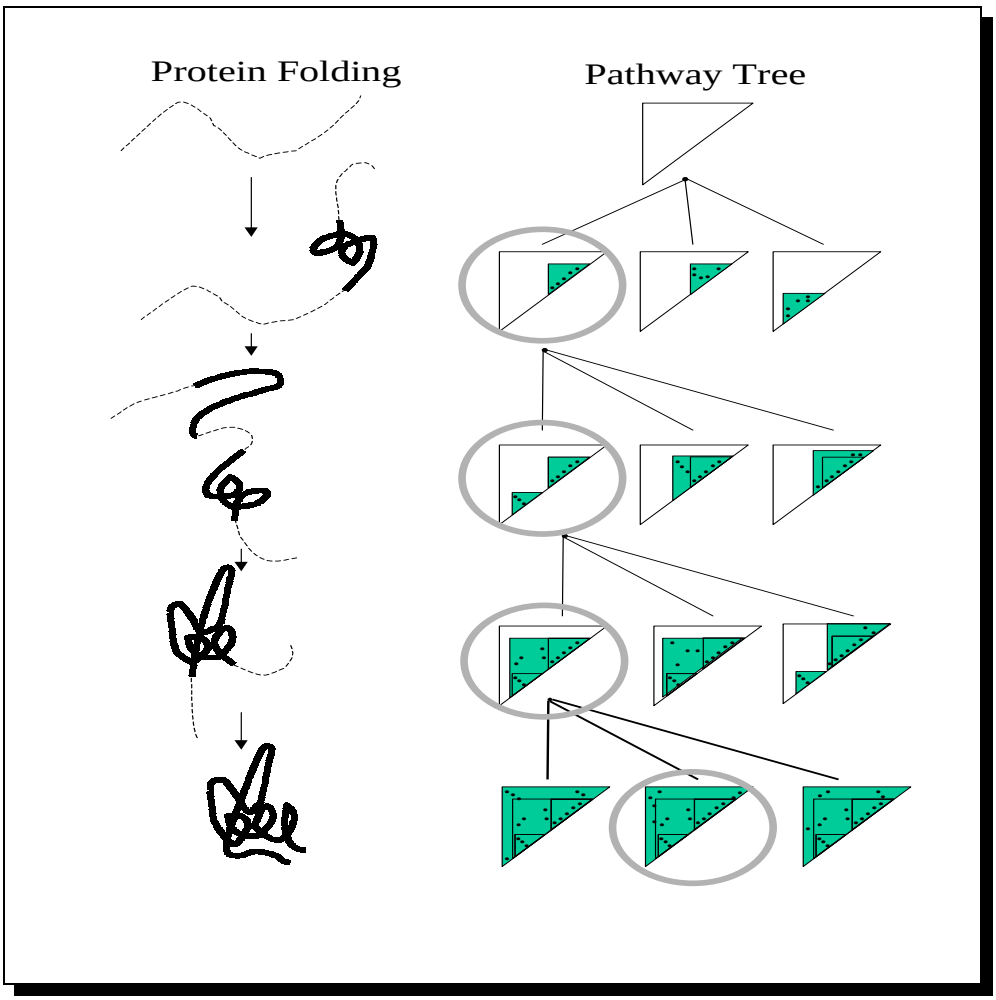

Figure 8: Folding Pathways in Contact Map: Large triangles represent the contact map, initially empty. Each branch defines a region of the contact map by setting pairs of amino acids to be in contact (dots) or not (space). Each node in the tree corresponds uniquely to a three dimensional structure (left), where the dotted lines represent segments of the chain that have undefined structure

Table 1: Frequent Dense Submatrices

Submatrix	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0	0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1
Frequency	2.0%	2.2%	1.9%
Physical Phenomena	anti-parallel beta	anti-parallel beta	parallel beta