Methods for Mining Protein Contact Maps

Mohammed J. Zaki[†][*], Jingjing Hu[†], and Chris Bystroff[‡]
[†]Computer Science Department
[‡]Biology Department

Rensselaer Polytechnic Institute
110 8th Street, Troy, NY 12180-3590
Email: huj5,zaki@cs.rpi.edu, bystrc@rpi.edu

Abstract

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. In this paper we describe how data mining can be used to extract valuable information from contact maps. For example, clusters of contacts represent certain secondary structures, and also capture non-local interactions, giving clues to the tertiary structure. We show that using contact maps and a hybrid mining approach, we can construct “contact rules” to predict the structure of an unknown protein. Furthermore, we mine non-local frequent dense contact patterns that discriminate physical from non-physical maps. We cluster these patterns based on their similarities and evaluate the clustering quality, and show that our techniques are effective in characterizing contact patterns across different proteins.

1.1 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. Genome sequencing projects have 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 structural

[*] This work was supported in part by NSF CAREER Award IIS-0092978, DOE Early Career Award DE-FG02-02ER25538, NSF grant EIA–0103708, and RPI Exploratory Seed Grant

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

Bioinformatics is an emerging field, undergoing rapid and exciting growth. Knowledge discovery and data mining (KDD) techniques will play an increasingly important role in the analysis and discovery of sequence, structure and functional patterns or models from large sequence databases. One of the grand challenges of bioinformatics still remains, namely the protein folding problem.

Proteins fold spontaneously and reproducibly (on a time scale of milliseconds) into complex three-dimensional globules when placed in an aqueous solution, and, the sequence of amino acids making up a protein appears to completely determine its three dimensional structure (Branden & Tooze 1991). Given a protein amino acid sequence (*linear structure*), determining its three dimensional folded shape, (*tertiary structure*), is referred to as the *Structure Prediction Problem*; it is widely acknowledged as an open problem, and a lot of research in the past has focused on it.

Traditional approaches to protein structure prediction have focused on detection of evolutionary homology (Altschul *et al.* 1997), fold recognition (Bryant 1996; Sippl 1996), and where those fail, ab initio simulations (Skolnick, Kolinski, & Ortiz 2000) that generally perform a conformational search for the lowest energy state (Simons *et al.* 1997). However, the conformational search space is huge, and, if nature approached the problem using a complete search, a protein would take longer to fold than the age of the universe, while proteins are observed to fold in milliseconds. Thus, a structured folding pathway (time ordered sequence of folding events) must play an important role in this conformational search. Strong experimental evidence for pathway-based models of protein folding has emerged over the years, for example, experiments revealing the structure of the "unfolded" state in water (Mok *et al.* 1999), burst-phase folding intermediates (Colon & Roder 1996), and the kinetic effects of point mutations ("phi-values" (Nolting *et al.* 1997)). These pathway models indicate that certain events always occur early in the folding process and certain others always occur later.

It appears that the traditional approaches, while having provided considerable insight into the chemistry and biology of folding, are still struggling when it comes to structure prediction (Bonneau & Baker 2001; Honig 1999), hence, a novel approach is called for, for example, using data mining. Mining from examples is a data driven approach that is generally useful when physical models are intractable or unknown, however, data representing the process is available. Thus, our problem appears to be ideally suited to the application of mining – physical models of folding are either intractable or not well understood, and data in the form of a protein data base exists.

1.2 Modeling Protein Folding

It is well known that proteins fold spontaneously and reproducibly to a unique 3D structure in aqueous solution (Branden & Tooze 1991). 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,

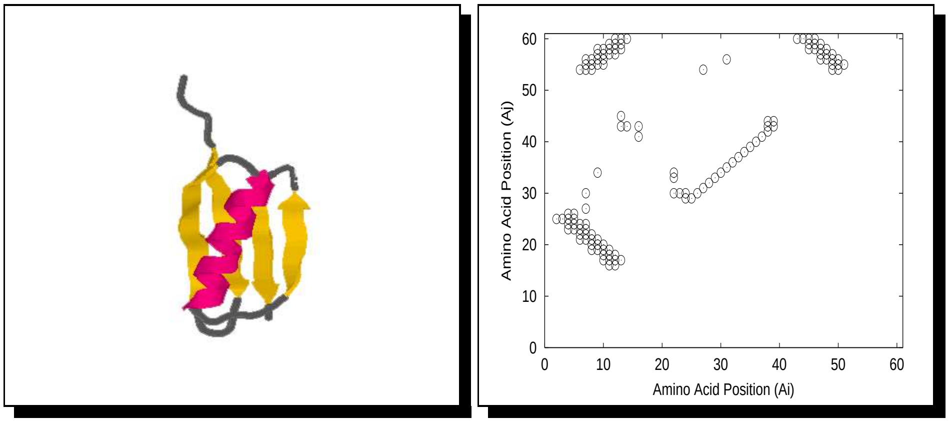

Figure 1.1: **Left:** 3D structure for protein G (PDB file 2igd, Sequence Length 61). **Right:** Contact Map for protein G – Circles indicate residue contacts, while non-contacts are represented by empty space. Only the upper triangle of the matrix is shown, since the contact map is symmetric. Clusters of circles indicate certain secondary structures, for example, the cluster along the main diagonal is an α-helix, and the clusters parallel and anti-parallel to the diagonal are parallel and anti-parallel β-sheets, respectively.

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. See (Hardin, Pogorelov, & Luthey-Schulten 2002; Bonneau & Baker 2001) for recent surveys on ab initio methods.

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. See (Rost 2001; Moult 1999; Schonbrun, Wedemeyer, & Baker 2002) for a review of existing structure prediction methods.

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 Contact Maps A *contact map* is a particularly useful two dimensional representation of a protein's tertiary structure. An example is shown in Figure 1.1. Two residues (or amino acids) a_i and a_j in a protein are in *contact* if the 3D distance is less than some threshold value t (we used $t = 7\AA$). Using this definition, every pair of amino acids is either in contact or not. Thus, for a protein with N residues, this information can be stored in an NxN binary symmetric matrix C, called the contact map. Each element, C_{ij}, of the contact map is called a contact, and is 1 if residues a_i and a_j are in contact, and 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 appear as thick bands parallel or anti-parallel to the main diagonal. Tertiary structure may also be obtained by reverse projecting into 3D space using the MAP algorithm (Vendruscolo, Kussell, & Domany 1997).

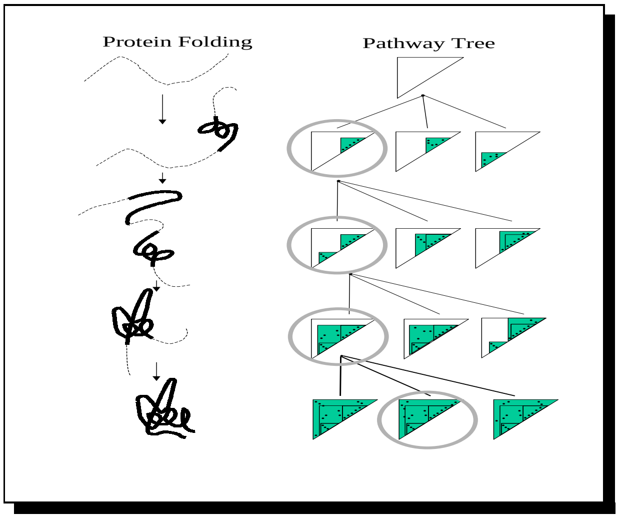

Figure 1.2: Folding Pathway and Tree Search: Large triangles represent the contact map, initially empty. Each branch defines a region of the contact map (green) by setting pairs of amino acids to be in contact (dots) or not (space). Each node in the tree corresponds uniquely to a three dimensional structure (left), where the dotted lines represent segments of the chain that have undefined structure.

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.

It is natural to define a **pathway** as a time-series of possible folding events, i.e., a sequence of contacts or sets of contacts (which we call blocks). A **folding pathway** will then be a pathway along with an assignment of 1 or 0 to each contact. Starting from an unfolded protein, the set of possible folding pathways can be represented on a tree, as is illustrated in Figure 1.2. A path from the root to a leaf is a particular folding pathway. Each branch defines a region of the contact map by setting pairs of amino acids to be in contact or not. Each set of contacts is physically realizable. The left column in the figure shows how each (circled) node in the tree corresponds uniquely to a three dimensional structure, where the dotted lines represent segments of the chain that have undefined structure. The final nodes of the tree represent complete, physical contact maps, which may be projected back into three dimensions using the MAP algorithm (Vendruscolo, Kussell, & Domany 1997).

Sources of Data Since we are using a data driven learning approach, we take a moment to discuss our data sources. The Protein Data Bank (PDB) records the 3D coordinates of the atoms of thousands of protein structures. The set of all known, globular proteins cluster into around 700 families based on their sequence similarity (PDBselect (Hobohm & Sander 1994)). It is conjectured that there will be on the order of 1000-2000 fold-families for the natural proteins. Thus from the PDB we can extract a set of proteins along with their known contact maps. These contact maps form the “rule learning data”, which will be used to mine for association rules. Part of this data set will be used for learning meaningful patterns, and a part will be set aside for validation.

Contributions In this paper we describe how data mining can be used to extract valuable information from contact maps. More specifically we focus on the following 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) Use contact maps to discover an extensive set of non-local dense patterns and compile a library of such non-local interactions. 3) Cluster these patterns based on their similarities and evaluate the clustering quality. 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.

1.3 Mining Contacts using Local Structure

We used a generalized hidden Markov model (HMM) called HMMSTR based on the I-sites library (Bystroff, Thorsson, & Baker 2000) to model statistical interactions between adjacent motifs on the chain, and were thus able to model the local propagation of structure. I-sites (or Initiation-Sites) are local sequence motifs that tend to fold the same way across protein families independent of the context. A rule-based method for predicting tertiary contacts in proteins, using HMMSTR as a preprocessor has already been developed (Zaki, Jin, & Bystroff 2000) and can be extended to sequentially output probabilities for subsets of contacts.

Super-local Contact Potentials Sequences from the database of 700 non-redundant proteins with less than 25% sequence similarity (the PDBselect dataset (Hobohm & Sander 1994)) were pre-processed using HMMSTR. The associated structures were converted to contact maps. The whole dataset was then mined to find common association rules for tertiary contacts. The rules were tested on a subset not used in the data mining.

The database of known proteins was divided into a training set and a test set. Each protein sequence was submitted to Psi-Blast (Altschul *et al.* 1997) to generate a sequence family, from which a sequence profile was summed, and backbone angles were discretized (Bystroff, Thorsson, & Baker 2000). For the training set, the amino acid profile and the backbone angle regions were the input data for the forward/backward calculation (Rabiner 1989), which produced a “gamma” matrix of position-dependent HMMSTR Markov state probabilities. For the test set, only the amino acid profile was used to generate the gamma matrix. A contact map was calculated for each member of the training set using a alpha-carbon distance cutoff of 7Å. From these data, a database of “item sets” was constructed. One entry corresponds to an ij residue pair, where $|i-j| > 4$, and consists of the amino acid pair and two sets of Markov state identifiers. Markov state identifiers were included as “items” associated with the ij contact if the position-dependent probability of the state was greater than twice the *a priori* probability of that state. Each entry had a label “1” meaning i and j were in contact in the structure, or “0” if they were not. All items are discrete symbols.

Association rule data mining (Agrawal *et al.* 1996; Zaki 2000) was applied to the database of item sets to extract rules which were predictive of contacts. A rule

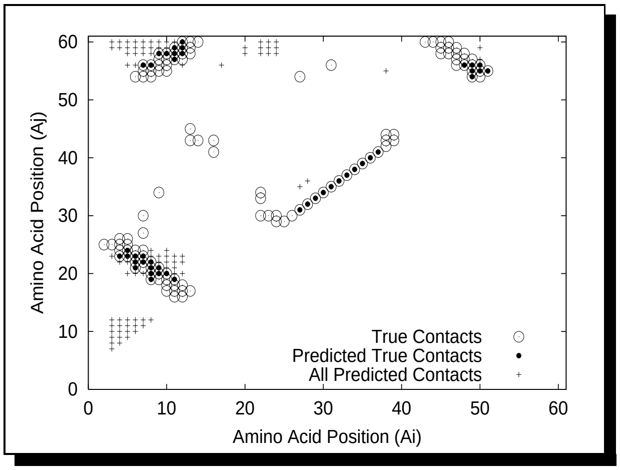

Figure 1.3: Predicted Contact Map (PDB protein 2igd). We were able to predict parts of the major structures.

has the form: "item1" $+$ "item2" $+$ ... $\Rightarrow C$, where C is 1 or 0. Items may be amino acids, predicted or observed secondary structure symbols, predicted or observed Markov state identifiers, or sequence separation ranges. Prediction of contacts in the test set was carried out by comparing the rule support for contact prediction with the rule support for non-contact prediction. The ratio of these values for all ij pairs in a given protein was sorted, and the top N pairs were predicted to be in contact, where N is the expected number of contacts, which depends on the sequence length. We shall call our method HMMassoc for later reference. Figure 1.3 shows an example contact map prediction. Previous work on contact prediction has employed Neural Networks (Fariselli & Casadio 1999), and statistical techniques based on correlated mutations (Olmea & Valencia 1997; Thomas, Casari, & Sander 1996). Recent work by Vendruscolo (Vendruscolo, Kussell, & Domany 1997) has also shown that it is possible to recover the 3D structure from even corrupted contact maps. Our recent results (Zaki, Jin, & Bystroff 2000) show that our model obtains around 20% accuracy and coverage over the set of all proteins; the model is also 5.2 times better than a random predictor. We can significantly enhance coverage to over 40% if we sacrifice accuracy (13%). For short proteins (length< 100) we get 30% accuracy and coverage (4.5 times better than random); if we lower accuracy to 26% we can get coverage up to 63%. While these results are better than (or equal to) those reported previously, we have still a long way to go before the goal of protein structure prediction is fully realized. Generating three-dimensional structure from the predicted contact maps is now a subject of our investigations.

1.4 Other Methods for Contact Map Prediction

In this section we review some previous work on contact map prediction. It has been shown that neural networks score higher than statistical approaches for contact map prediction (Fariselli & Casadio 1999). We will thus briefly review the statistical approaches and then focus more on the neural network based methods.

1.4.1 Statistical Approaches

A number of earlier methods for contact predictions have relied on Correlated Mutation Analysis (Gobel *et al.* 1994; Shindyalov, Kolchanov, & Sander 1994; Thomas, Casari, & Sander 1996; Olmea & Valencia 1997), which compares multiple members of a protein family and detects residues that remain constant or mutate together. The basic idea is to note the distance between residues that appear at a given position in a multiple sequence alignment. Next the correlation coefficient between pairs of positions in an alignment of n proteins is computed as follows:

$$r_{ij} = \frac{1}{n^2} \sum_{k=1}^{n} \sum_{l=1}^{n} \frac{(s_{ikl} - \mu_i)(s_{jkl} - \mu_j)}{\sigma_i \sigma_j}$$

where s_{ikl} is the similarity between residues k and l, μ_i is the mean similarity, and σ_i the standard deviation of the similarity at position i in the alignment. Completely conserved positions or those with gaps over 10% are ignored.

(Olmea & Valencia 1997) used correlated mutations and other information like sequence conservation, alignment stability, contact occupancy, etc. to improve the accuracy. The prediction results from their approach are shown in Table 1.1; they did not report the result for all proteins. Another correlated mutation based approach to contact map prediction was presented in (Thomas, Casari, & Sander 1996), they obtained an accuracy of 13%.

Recent work by (Singer, Vriend, & Bywater 2002) has also suggested use of a contact likelihood matrix derived from PDB to improve correlated mutation analysis to predict contacts. Work by (Zhao & Kim 2000) examined pairwise amino acid interactions in the context of secondary structural environments; they reported a set of residue contact energies for a 20 × 3 ($\alpha, \beta,$ coil) = 60 residue alphabet. They also tested the use of these environment-dependent contact energies (ERCE) for contact predictions. The predictive accuracy results from the ERCE method are shown in Table 1.1.

1.4.2 Neural Networks

(Fariselli & Casadio 1999) were the first to apply a neural network (NN) to predict pairwise residue contacts. They also trained on the PDBselect dataset (Hobohm & Sander 1994) (an older version with 608 proteins). They used a classical feed-forward NN with a standard back-propagation algorithm (Rumelhart, Hinton, & Williams 1986). The network architecture consisted of one output neuron for contact propensity between an amino-acid pair, one hidden layer with two neurons, and an input layer with variable number of neurons depending on the input encoding. They implemented five different networks with increasing input complexity.

Given residues at position i and j, every network had 2 fixed input neurons: one input neuron for the normalized sequence separation $i-j$, and another for the normalized sequence length. The different networks tried included:

Method	$N < 100$	$100 \leq N < 170$	$170 \leq N < 300$	$N \geq 300$	All
AA	13	6	4.5	2	8.5
CM	13	13	11	3	-
CC	19	12	9	4	-
ERCE	19.6	12.1	7.2	-	13.5
Net1	18	16.1	13.3	12.7	14.4
Net2	18	16.4	13.6	13.1	14.7
Net3	19	16.9	13.8	13.2	15
Net4	18	16.7	14.1	13.7	15.1
Net5	19	18	14.9	**14.4**	16
NETCSS	**26**	21	**15**	11	**21**
HMMassoc	**26**	**21.5**	13	9.7	19

Table 1.1: Comparative Prediction Accuracy for Different Methods

- Net1: it uses only the amino acids for prediction. Each input neuron encodes a particular ordered pair of residue types, contributing $\binom{20}{2} = 190$ input neurons for distinct pairs (e.g., A-G) and 20 input neurons for pairs of same residue type (e.g., A-A). The input neuron for the pair (a_i, a_j) is set to 1 while the remaining 209 input neurons are set to 0. With the 2 fixed neurons, there are 212 input neurons.

- Net2: it incorporates, in addition to the amino acids, the hydrophobicity scale for each reside (averaged over a window of 7 consecutive residues); it thus has 214 input neurons.

- Net3: it encodes (in addition to information from Net1 and Net2) evolutionary information obtained from the multiple sequence alignment (MSA) of the protein to other similar proteins taken from the HSSP database (Sander & Schneider 1991). For the positions i and j, they count the different residue types that occur in those positions in the MSA. The normalized pair counts are used as input to the NN. Two additional neurons record how well positions i and j are conserved in the MSA. The total input neurons is thus 216.

- Net4: it tries to capture the sequence context for each residue. They center a window of length 3 at positions i and j and consider the 5 possible pairings of residues (parallel and antiparallel) ($210 \times 5 = 1050$ neurons). With hydrophobic values for each position, it has 1054 input neurons.

- Net5: it is Net4 augmented with evolutionary information as in Net3.

The accuracy of prediction for proteins of various lengths and on the entire testing set are shown in Table 1.1. It appears that all the additional information is helpful in the prediction. Net5 performs the best overall and it incorporates the hydrophobicity values, evolutionary information, and sequence context.

The recent Neural Networks proposed in (Fariselli *et al.* 2001) improve upon the work by (Fariselli & Casadio 1999). One factor is that they have a more recent and larger version of the PDBselect dataset with 822 non-redundant proteins. Another factor is that the new NNs incorporate more information. The basic architecture consists of one output, 8 hidden, and variable input layer neurons. The different NNs trained were:

- NET: same as Net4 above.
- NETC: same as NET with three additional inputs: two for residue conservation values for i and j and third for the correlated mutation between them.
- NETCW: same as NETC but with 6 residue conservation values and 9 correlated mutation values for the length 3 window around i and j.
- NETCSEP: same as NETC, but with an additional neuron coding for sequence separation.
- NETCSS: adds secondary structure information (3 states: α-helix, β-sheet, and coil) to NETC. For each of the 3 positions around i and j (6 total), add an input assigning the residues to one of 3 states, i.e., 18 extra inputs.

They found NETCSS to perform the best. Generally the more information used by the NN the better it performed. The accuracy values for NETCSS are shown in Table 1.1.

1.4.3 Comparative Performance

A summary of known results on contact map prediction accuracy is shown in Table 1.1. AA (using amino acids only) and HMMassoc (HMMSTR with association mining) values are taken from (Zaki, Jin, & Bystroff 2000), CM (correlated mutation) and CC (correlation + conservation) from (Olmea & Valencia 1997), ERCE (environment dependent contact energy) is taken from (Zhao & Kim 2000), Net1-Net5 from (Fariselli & Casadio 1999), NETCSS from (Fariselli *et al.* 2001). We add as a caveat that direct comparison is not possible, since previous works used a different (and smaller) PDBselect databases 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. Having said that it is still possible to draw some overall conclusions.

The best results in each column are highlighted in bold. The proteins have been separated into bins based on sequence length N. It can be seen that the NN-based methods and HMMassoc outperform other statistically-based methods. Among the NN methods, NETCSS is the best. NETCSS and HMMassoc perform the same on proteins up to 170 residues long, with NETCSS having an advantage of about 2% for longer proteins. It should be noted that while NETCSS explicitly incorporates the evolutionary information, sequence context, hydrophobicity, etc., HMMassoc does so implicitly in the HMM state output in the form of the “gamma” matrix. We plan to add more explicit information to the association mining approach to test its efficacy.

Also, it has been reported that post-processing the predicted contact maps by filtering out overpredicted contacts can help improve the accuracy (Olmea & Valencia 1997; Fariselli *et al.* 2001). The results shown above are without such post filtering.

1.4.4 Other Advances

There has been some recent work by (Pollastri & Baldi 2002) on using recurrent neural network architectures for contact prediction. Their PDBselect dataset is even larger, consisting of 1484 proteins, and the inputs consist of an orthogonal encoding of pairs of amino acids (pairs of binary vectors of length 20), evolutionary information in terms of profiles, correlated mutations, specific secondary structural features (α, β, coil), and relative solvent accessibility (buried, exposed). They report an overall accuracy of 27% for all proteins, which would make it the best current method. They concluded that the use of secondary structure and solvent accessibility is more useful than profiles and correlated profiles.

Support Vector Machines (SVM) have also been used by (Zhao & Karypis 2003) for contact map mining. They trained 15 different SVMs using a different number of features. The possible features included sequence conservation values for i and j, sequence separation, correlated mutation values, predicted secondary structure values, and sequence profiles (for evolutionary information). The best SVM method used all of the available features to give an overall accuracy of 22.4%. However, it should be noted that they used a different set of only 177 proteins, and not the non-redundant set from PDBselect.

1.5 Characterizing 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. Previous work (Zaki, Jin, & Bystroff 2000) has generated a method 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 the method that we use for mining frequent dense patterns or structural motifs in contact maps. All protein sequences used are from Protein Data Bank (PDB). Briefly, there are four major stages in our approach: (1) Mining dense patterns,(2) Pruning mined patterns, (3) Clustering the dense patterns, and (4) Integration of these patterns with biological data.

1.5.1 Mining Dense Patterns in Contact Maps

To enumerate all the frequent 2D dense patterns we scan the database of contact maps with a 2D sliding window of a user specified size. Across all proteins in the database, any sub-matrix under the window that has a minimum “density” (the number of ’1’s or contacts) is captured. For a $N \times N$ contact map (N is the length of the protein), using a 2D $W \times W$ window, there are $(N - W) \times (N - W)/2$ possible sub-matrices. We have to tabulate those which are dense, using different window sizes. We choose 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).

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 roughly 28 million possible windows for a database with 18,455 contact maps (equal to the number of proteins stored in the PDB database). Of these windows only relatively few will be dense, since the number of contacts is a lot less than the number of non-contacts. Still we need an efficient way of testing if two sub-matrices are identical or not. We 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 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 will encode each sub-matrix in the following way: each row of the contact map, i.e., 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
Hashing of a Dense Pattern
```

According to our sub-matrix 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 will employ 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 sub-matrices 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 the equality check of a sub-matrix 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 further analysis.

After a pruning step to remove redundant patterns (Hu *et al.* 2002), we generated the possible dense patterns with *minSupport* 1, i.e., the exhaustive set of dense patterns that appear in our database. We also varied the amino acid contact threshold while creating the contact map (recall that two amino acids are in contact if they are at most t distance apart in 3D; we used $t =$ 5,6 and 7 $\mathring{A}$ in our experiments). When using sliding window size less than 5, the dense patterns generated are trivial and didn’t show enough structural meaning. With window size 6 and above, we generated only slightly more dense patterns than with window size 5. We consider 5 an important window size to generate existing dense patterns. In the following study, only data with sliding window size 5 will be listed. An example dense pattern with associated information is shown below (its support count is 5 and its volume, the number of 1’s, is 10):

```
Sup:5 Str:0.28.12.15.1. Vol:10
00000
11100
01100
01111
00001
a dense pattern example
```

The numbers of non-redundant dense patterns extracted using different contact thresholds is shown in the second column of Table 1.2 (it also shows other clustering information which will be explained in the next section).

1.5.2 Clustering Dense Patterns

In the mining step, a large number of possible dense patterns are generated even after pruning. Instead of analyzing these non-local patterns directly it is beneficial to group

Contact Threshold	# Patterns	# Clusters	Cluster Quality
5 $\mathring{A}$	2508	83	0.8931
6 $\mathring{A}$	9929	99	0.8633
7 $\mathring{A}$	21231	367	0.8367

Table 1.2: Clustering of Dense Patterns

them into groups of similar interactions. To characterize all the dense patterns that we have mined, clustering provides an effective way to obtain a gross view.

There are two main approaches to clustering. 1) Partition-based clustering tries to divide the data of N objects into k partitions or groups using heuristic search or iterative methods (e.g., k-means clustering). 2) Hierarchical clustering comes in two flavors. a) Agglomerative clustering technique starts with each object in its own cluster. At each step pairs of clusters with minimum distance between them are successively merged. b) Divisive clustering takes the opposite approach, it starts with all the records in one cluster, and then successively splits clusters into small pieces.

In this paper, we used agglomerative clustering to group the mined dense patterns to find the dominant non-local interactions, using the methodology described below.

Calculating distance

First, the distance between every pair of patterns is calculated using the formula:

$$Distance(M_i, M_j) = \sum_{k=1}^{W^2} |M_i[k] - M_j[k]| \tag{1.1}$$

where M_i and M_j are dense patterns, and k is the position in the $W \times W$ matrix taken as a linear array (top left corner is position 0 and bottom right is $W \times W$). Thus $M_i[k]$ is either 0 or 1, indicating a non-contact and contact, respectively. The smaller the distance between two patterns, the more likely the two patterns are similar to each other.

We also need to define the distance between two clusters, say c_1 and c_2. Let the size of c_1 be n and the size of c_2 be m patterns. Then the distance between the pair of clusters is given as: $\sum_{i=1}^{n} \sum_{j=1}^{m} Distance(M_i, M_j)$ (with pattern $M_i \in c_1$ and $M_j \in c_2$), i.e., the sum of all pair-wise distances between patterns in a cluster.

Clustering

Before we start the clustering, we need to determine a threshold distance for a cluster, namely, the maximum average distance among the patterns in one cluster. Once this is done, the procedure is as follows: 1) Compare all pairs of clusters and mark the pair that is closest. 2) The distance between this closest pair of clusters is compared to the threshold value. If the distance is less than the threshold distance, these clusters become linked and are merged into a single cluster. Return to Step 1 to continue the clustering. If the distance between the closest pair is greater than the threshold, the clustering stops. If the threshold value is too small, there will still be many groups present at the end, and many of them will be singletons. Conversely, if the threshold is too large, objects that are not very similar may end up with the same cluster. We used distance 4 as the threshold for clustering.

```
Cluster No.1, Count = 59
Contact Probabilities:                     Representative:
0:0.05  1:0.05  2:0.68  3:0.85  4:0.71    |    00111
5:0.03  6:0.02  7:0.14  8:0.07  9:0.09    |    00000
10:0.05 11:0.05 12:0.12 13:0.09 14:0.03   |    00000
15:0.03 16:0.05 17:0.15 18:0.27 19:0.85   |    00001
20:0.25 21:0.10 22:0.59 23:0.92 24:0.83   |    00011
```

After the agglomerative clustering step, for each cluster, we need a way to compactly describe the dominant interactions represented by all members of the cluster. For this we calculated the contact probability at each of the $W \times W$ positions in the submatrix. Assume that there are n patterns grouped in cluster c. Contact probability at position k is defined as the ratio of the number of contacts at that position divided by the cluster cardinality, and is given as $p_k^c = (1/n) \times \sum_{i=1}^{n} M_i[k]$. Based on these probability values, a representative contact pattern is generated for each cluster. In a representative contact pattern, we record a ’1’ at position k whenever p_k^c is greater than some probability threshold r and a ’0’ otherwise. An example cluster is shown below with associated information. Count is the number of patterns in the cluster and the notation 0:0.05 means that the probability of contact at position 0 is 0.05. The representative contact pattern for the cluster with a probability threshold $r = 0.65$ is also shown.

The number of clusters generated using different amino acid contact threshold are listed in Table 1.2, Column 2 (with clustering threshold of 4 and window size 5). For instance at $6\mathring{A}$ contact threshold we obtained 99 clusters from the 9929 mined patterns.

Evaluating Clustering Quality

After clustering is finished, we need a method to evaluate how effective it is. One way is to define an objective notion of clustering quality. While this may be hard in general, the contact probabilities for a cluster gives a good indication about how good the cluster is. For example, a cluster with very high values at some positions and very low values at some positions is a good cluster, while a cluster that has contact probabilities close to 0.5 is not very good. In other words, if a majority of the cluster members agree on a position (mostly 0’s or mostly 1’s), that indicates a good clustering.

We use the formula below to generate the sum of contact probabilities at each position in the window within a cluster c:

$$S_c^1 = \sum_{k=1}^{W^2} p_k^c, \text{ if } p_k^c > 0.5 \tag{1.2}$$

$$S_c^0 = \sum_{k=1}^{W^2} (1 - p_k^c), \text{ if } p_k^c \leq 0.5 \tag{1.3}$$

The quality of a cluster c is then given by the sum $Q_c = S_c^1 + S_c^0$. A high Q_c value close to 1 indicates a good cluster, while a value close to 0.5 indicates a poor cluster.

The final clustering quality across all the clusters is given as the weighted sum of individual cluster quality values, as shown in the formula below:

$$Q = \frac{\sum_{i=1}^{|C|} |c_i| \times Q_{c_i}}{N}, (0.5 \leq Q \leq 1) \tag{1.4}$$

where C is the set of clusters, $|C|$ is the number of clusters, $c_i \in C$ is one cluster in the set, $|c_i|$ is the numbers of patterns in the cluster, and N is the total number of patterns. Note how the clustering quality Q varies from 0.5 to 1, with a higher value suggests better clustering quality because it clusters patterns which share similar occurrence positions for '1's and '0's. A cluster with the same contact pattern has a $Q = 1$. The clustering quality corresponding to clusters generated in our experiments is listed in Table 1.2, Column 3 (with clustering threshold of 4 and window size 5). For example, given window size of 5, contact distance threshold of 6 $\mathring{A}$ and clustering threshold of 4, the clustering quality of the 99 clusters is 0.865753.

1.5.3 Integration and Visualization

After dense patterns are found and clustered, the final step is to incorporate the protein sequence/structure information with them. That is, for each dense pattern and its occurrences in the different contact maps (that is in different protein segments in PDB), we note the protein id, the start positions of the window (given as (X, Y) coordinates of the top left corner), and the type of interaction. This information is then used to visualize the mined patterns or interactions. An example of a dense pattern with associated information is shown below. This pattern with 11 contacts, occurs only once in PDB file with id $1vjs\_.1$, at position $(134, 109)$, i.e., it represents a non-local interaction between protein segment at positions 134-138 (the X axis) and the protein segment at positions 109-113 (the Y axis), in this case an interaction between two beta-strands.

```
Sup:1 Str:1.5.31.24.16 Vol:11
00001
00101
11111
11000
10000
pdb- x_start y_start interaction
1vjs_.1  134   109   beta strand-beta strand
```

1.5.4 Experimental Results

We used a non-redundant set of 2702 proteins from the PDB for our experiments. Preliminary distance maps for protein were produced based on the 3D coordinates of the α-Carbon atoms of each amino acid Based on these distance maps, binary contact maps were generated using several contact thresholds As described above, we discovered 9929 dense patterns when using a sliding window of size 5, maximum amino acid contact threshold of $6\AA$ and a minimum density of 0.125. When agglomerative clustering is applied, 99 clusters are generated using a clustering threshold of 4. The clustering quality is 0.8633. Two example clusters with their four associated patterns and corresponding interactions are given below:

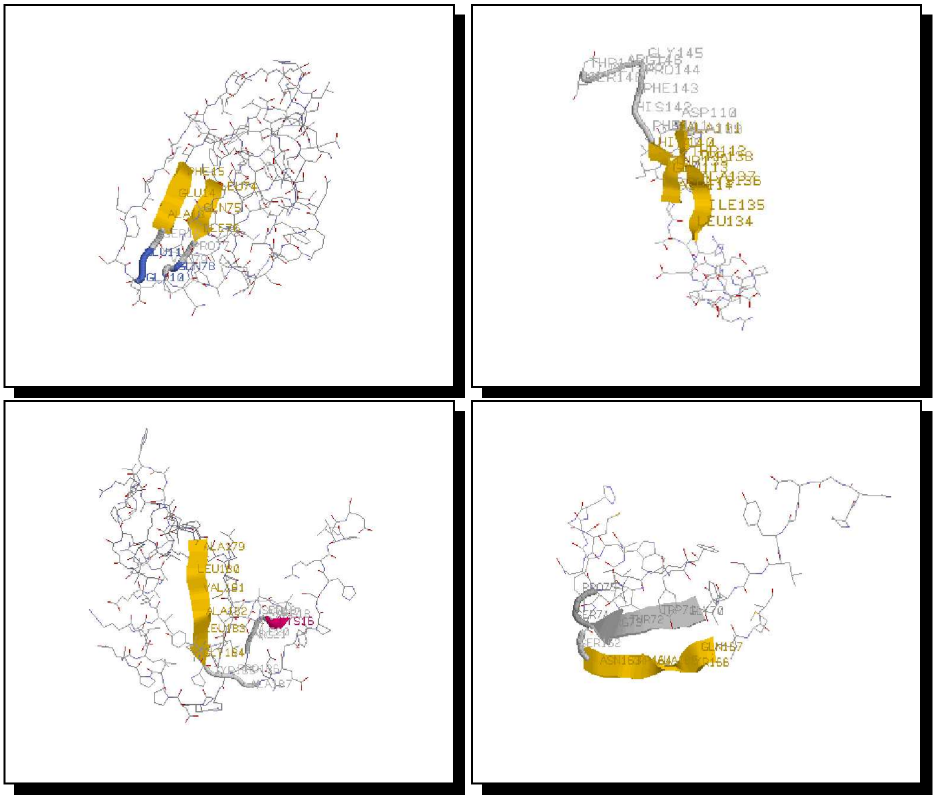

Figure 1.4: Secondary Structures of four different patterns from one cluster–Beta Strand vs. Beta Strand: Upper left: pattern 1355, Upper right: pattern 3496, Lower left: pattern 6282, Lower right: pattern 7980

Figure 1.4 shows an example of the structures of four different patterns from one of the mined clusters. Beta strand interacting with beta strand is the dominant non-local motif in this cluster. In other clusters, different dominant interactions were discovered. These interactions 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.

These experiments shows that we efficiently clustered patterns according to their similarities both in sub-matrix level and structure level. With our clustering method, we can compile a library of possible dense patterns for further application in extracting valuable information to improve the accuracy of protein structure and pathway prediction. For instance the exhaustive collection of mined patterns can be used in a post-processing step to filter out over predicted contacts, similar to the approach in (Olmea & Valencia 1997; Fariselli *et al.* 2001).

1.6 Future Directions

1.6.1 Improving Prediction of Contact Maps

As mentioned above we applied a hybrid method based on hidden Markov Models and association rule mining to predict the contact map for a given protein sequence (see (Zaki, Jin, & Bystroff 2000) for details). Fig. 1.3 shows the predicted contact map for the protein $2igd$ from Fig. 1.1. 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 or the block of contacts in top left and middle regions of the contact map. Using the extensive library of non-local motifs, one can eliminate such false contacts by recognizing the fact that they never occur in real proteins, and thus these blocks of contacts are physically impossible. In future work we will describe the effectiveness of this post-processing approach (by filtering out physically impossible blocks) in improving the prediction of contact maps.

1.6.2 Mining 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: a) If $C(i,j) = 1$ and $C(i+2, j+2) = 1$, then $C(i, j+2) = 0$, and $C(i+2, j) = 0$. b) 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 (respectively), 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 to $i+4$, and therefore cannot share contacts with non-local residue j. Local structures 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.

One approach is to 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). For instance, consider a lower 1-layer (denoted LL1) neighborhood for a given point, $C(i,j)$. LL1 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 $C(i+1,j+1)$. From the LL1 region around each point we obtain a database which can be mined for frequent combinations. Other rules can be found by defining an appropriate neighborhood and by incorporating sequence information. We are currently developing techniques to mine such heuristic rules of contact automatically.

1.6.3 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.

A pathway in contact map space consists of a time-ordered series of contacts. The pathway is initiated by high-confidence Initiation-sites (Bystroff, Thorsson, & Baker 2000), and thereafter it follows a tree-search format (see Figure 1.2). We may impose a "condensation rule" 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. Each level of the tree is the addition of a contact that satisfies the condensation rule. A maximum of k branches can be selected based on the energy. In addition, contacts that are not physically possible can be rejected, using the mined clusters of dense patterns or using the heuristics rules for physicality. Identical branches (same set of contacts, different order) can of course be merged.

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.

1.7 Conclusions

In this paper we described how data mining can be used to extract valuable information from contact maps. More specifically we focused on the following 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) Use contact maps to discover an extensive set of non-local dense patterns and compile a library of such non-local interactions. We reviewed previous methods for contact map predictoins and we highlighted some 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.

References

Agrawal, R.; Mannila, H.; Srikant, R.; Toivonen, H.; and Verkamo, A. I. 1996. Fast discovery of association rules. In Fayyad, U., and et al., eds., *Advances in Knowledge Discovery and Data Mining*, 307–328. AAAI Press, Menlo Park, CA.

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.

Bonneau, R., and Baker, D. 2001. Ab initio protein structure prediction: Progress and prospects. *Annual Review of Biophysics and Biomolecular Structure* 30:173–189.

Branden, C., and Tooze, J. 1991. *Introduction to Protein Structure*. Garland Publishing, NY.

Bryant, S. 1996. Evaluation of threading specificity and accuracy. *Proteins* 26(2), 172-85.

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

Colon, W., and Roder, H. 1996. Kinetic intermediates in the formation of the cytochrome c molten globule. *Nature Structural Biology* 3(12), 1019-25.

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

Fariselli, P.; Olmea, .; Valencia, A.; and Casadio, R. 2001. Prediction of contact maps with neural networks and correlated mutations. *Protein Engineering* 14(11), 835-843.

Gobel, U.; Sander, C.; Schneider, R.; and Valencia, A. 1994. Correlated mutations and residue contacts in proteins. *Proteins* 18:309–317.

Hardin, C.; Pogorelov, T.; and Luthey-Schulten, Z. 2002. Ab initio protein structure prediction. *Curr Opin Struct Biol* 12(2):176–181.

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

Honig, B. 1999. Protein folding: from the levinthal paradox to structure prediction. *Journal of Molecular Biology* 293(2), 283-93.

Hu, J.; Shen, X.; Shao, Y.; Bystroff, C.; and Zaki, M. 2002. Mining protein contact maps. In *2nd BIOKDD Workshop on Data Mining in Bioinformatics*.

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

Mok, Y.; Kay, C.; Kay, L.; and Forman-Kay, J. 1999. Noe data demonstrating a compact unfolded state for an sh3 domain under non-denaturing conditions. *J. Mol. Biology* 289(3), 619-38.

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.

Moult, J. 1999. Predicting protein three-dimensional structure. *Curr. Opin. Biotechnol.* 10(6):583–8.

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.

Pollastri, G., and Baldi, P. 2002. Prediction of contact maps by recurrent neural network architectures and hidden context propagation from all four cardinal corners. *Bioinformatics* 18 Suppl 1:S62–70.

Rabiner, L. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77(2):257-86.

Rost, B. 2001. Review: protein secondary structure prediction continues to rise. *J Struct Biol* 134(2-3):204–218.

Rumelhart, D.; Hinton, G.; and Williams, R. 1986. Learning representations by back-propagating errors. *Nature* 323:533–536.

Sander, C., and Schneider, R. 1991. Database of homology derived protein structures and the structural meaning of sequence alignment. *Proteins* 9:56–68.

Schonbrun, J.; Wedemeyer, W. J.; and Baker, D. 2002. Protein structure prediction in 2002. *Current Opinion in Structural Biology* 12(3):348–354.

Shindyalov, I.; Kolchanov, N.; and Sander, C. 1994. Can three-dimensional contacts in protein structures be predicted by analysis of correlated mutations? *Protein Engineering* 7:349–358.

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.

Singer, M.; Vriend, G.; and Bywater, R. 2002. Prediction of protein residue contacts with a pdb-derived likelihood matrix. *Protein Engineering* 15(9):721–725.

Sippl, M. 1996. Helmholtz free energy of peptide hydrogen bonds in proteins. *J. Mol. Biology* 260(5), 644-8.

Skolnick, J.; Kolinski, A.; and Ortiz, A. 2000. Derivation of protein-specific pair potentials based on weak sequence fragment similarity. *Proteins* 38(1), 3-16.

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

Zaki, M. J. 2000. Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering* 12(3):372-390.

Zhao, Y., and Karypis, G. 2003. Prediction of contact maps using support vector machines. In *IEEE Intl. Conferene 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.