Protein folding and chart parsing

Julia Hockenmaier **Aravind K. Joshi**
Institute for Research in Cognitive Science
University of Pennsylvania
Philadelphia, PA 19104, USA
{juliahr, joshi}@cis.upenn.edu

Ken A. Dill
Dept. of Pharmaceutical Chemistry
University of California, San Francisco
San Francisco, CA 94143, USA
dill@maxwell.compbio.ucsf.edu

Abstract

How can proteins fold so quickly into their unique native structures? We show here that there is a natural analogy between parsing and the protein folding problem, and demonstrate that CKY can find the native structures of a simplified lattice model of proteins with high accuracy.

1 Introduction

In statistical parsing, the task is to find the most likely syntactic structure for an input string of words, given a grammar and a probability model over the analyses defined by that grammar. Proteins are sequences of amino acids (polypeptide chains) that form unique, sequence-specific three-dimensional structures. The structure into which a particular protein folds has a lower energy than all other possible structures. In protein structure prediction, the task is thus to find the lowest-energy physical structure for an input sequence of amino acids, given a representation of possible structures and a function that assigns an energy score to these structures. There is therefore a natural analogy between these two seemingly unrelated computational problems. Based on this analogy, we propose an adaptation of the CKY chart parsing algorithm to protein structure prediction, using a well-known simplified model of proteins as proof of concept.

Models of protein folding additionally aim to explain the process by which this structure formation takes place, and their validity depends not only on the accuracy of the predicted structures, but also on their physical plausibility. One common proposal in the biophysical literature is that the folding process is hierarchical, and that folding routes are tree-shaped. CKY provides an explicit computational recipe to efficiently search (and return) all possible folding routes. This sets it apart from existing folding algorithms, which are typically based on Monte Carlo simulations, and can only sample one possible trajectory.

Since we believe that there is much scope for future work in applying statistical parsing techniques to more detailed models of proteins, a secondary aim of this paper is to provide an introduction to the research questions that arise in protein folding to the NLP community.

Proteins are essential components of the cells of any living organism, and their biological function (eg. as enzymes that catalyze certain reactions) depends on their three-dimensional structure. However, genes only specify the linear, sequence of the amino acids, and the ribosome (the cell's "protein factory") uses this information to assemble the polypeptide chain. Under "natural" conditions, these polypeptide chains then fold rapidly and spontaneously into their unique final structures, or native states. Therefore, protein folding is often referred to as the second half of the genetic code, and the ability to predict the native state for a primary sequence is great practical importance, eg. in drug design, or in our understanding of the genome.

Levinthal (1968), who was the first to frame the folding process as a search problem, showed that folding cannot be guided by a random, exhaustive search: he argued that a chain of 150 amino acids has on the order of 10^{300} possible structures, but since folding takes only a few seconds, not more 10^{8} of these structures can be searched. Under the assumption that a better understanding of the physical folding process will ultimately be required to design accurate structure prediction techniques, this observation has lead researchers to try to identify sequence-specific pathways along which folding may proceed or a general mechanism that makes this process so fast and reliable.

Our aim of understanding the folding process is different from a number of approaches which have used formal grammars to represent the structure of biological molecules such as RNAs or proteins (Searls, 2002; Durbin et al., 1998; Chiang, 2004). These studies have typically focused on a specific classes of protein folds, and are not generally applicable yet. Our folding algorithm restricts the possible order of folding events, but places no explicit restrictions on the structures it can account for (other than those imposed by the spatial model used to represent them, and those that are implied by the hierarchical nature of the folding process).

2 A brief introduction to protein folding

2.1 Protein structure

The *primary structure* describes the linear sequence of amino acids that are linked via peptide bonds (and form the backbone of the polypeptide chain). Each amino acid has one side chain which branches off the backbone. Proteins contain twenty different kinds of amino acids, which differ only in the size and chemical properties of their side-chains. One important distinction is that between hydrophobic (water-repelling) and hydrophillic (polar) amino acids.

The *secondary structure* refers to patterns of local structures such as α-helices or β-sheets, which occur in many different folded structures. These secondary structure elements often assemble into larger *domains*. The *tertiary structure* represents the fully folded three-dimensional conformation of a single-chain protein, and typically consists of multiple domains. Since proteins in the cell are surrounded by water, hydrophobic side-chains are typically inside this structure and in close contact to each other, forming a *hydrophobic core*, whereas polar side-chains are more likely to be on the surface of this structure. This *hydrophobic effect* is known to be the main driving force for the folding process.

Computational models of protein folding often use a very simplified representation of these structures. Ultimately, models which explicitly capture all atoms and their physical interactions are required to study the folding of real proteins. However, since such models often require huge computational resources such as supercomputers or distributed systems, novel search strategies and other general properties of the folding problem are usually first studied with coarse-grained, simplified representations, such as the HP model (Lau and Dill, 1989; Dill et al., 1995) used here.

2.2 Folding and thermodynamics

As first shown by Anfinsen (1973), protein folding is a reversible process: under “denaturing” conditions, proteins typically unfold into a random state (which still preserves the chain connectivity of the primary amino acid sequence), and refold again into their unique native state if the natural folding conditions are restored. Thus, all the information that is necessary to determine the folded structure has to be encoded in the primary sequence. This is analogous to natural language, where the meaning of sentences such as *I drink coffee with milk* vs. *I drink coffee with friends* is also determined by their words.

Since folding occurs spontaneously, the native state has to be the thermodynamically optimal structure (under folding conditions), ie. the structure that results in the lowest *free energy*. The free energy $G = H - TS$ of a system depends on its energy H, its entropy S (the amount of disorder in the system), and the temperature T. A computational model therefore requires an *energy function* $\phi : R^n \rightarrow R$, which maps n-dimensional vectors that describe the structure of a polypeptide chain (eg. in terms of the coordinates of its atoms) to the free energies of the corresponding structures. The native state is assumed to be the global minimum of this function. This is again analogous to statistical parsing, where the correct analysis is assumed to be the structure with the highest probability. In the case of proteins, we can use the laws of physics to determine the energy function, whereas in language, the “energies” have to be estimated from corpora.[1]

The energy H of a single protein structure depends essentially on the interactions (*contacts*) between side-chains and on the bond angles along the backbone, whereas the entropy S also depends on the surrounding solvent (water). It is this impact on S which creates the hydrophobic effect. For simplicity’s sake most computational models use an *implicit solvent* energy function, which captures the hydrophobic effect by assuming that the contact energies between hydrophobic side-chains are particularly favorable. Since bond angles alone cannot capture the hydrophobic effect (Dill, 1999), simplified models typically ignore their impact and represent the energy of a conformation only

[1] We note, however, that so-called “knowledge-based” or “statistical potentials”, whose parameters are also estimated from known structures, are often used as well.

in terms of the side chain contacts. One particularly well-known example is the Miyazawa-Jernigan (1996) energy function, a 20x20 matrix of contact potentials whose parameters are estimated from the Protein Data Bank, a database of experimentally verified protein structures. These simplified energy functions are therefore very similar to the bi-lexical dependency models that are commonly used in statistical parsing.

It is this similarity between inter-residue contacts and word-word dependencies that grammar-based approaches (Searls, 2002) exploit. The set of contacts for a given structure can be represented as a *polymer graph*, although often only the edges of this graph are given in the form of a *contact map* (a triangular matrix whose entry C_{ij} corresponds to the contact between the ith and jth residue). The edges in this graph are inherently undirected. In α-helices and parallel β-sheets, the edges are crossing. Although grammars that capture the "dependencies" in specific kinds of protein structures have been written (Chiang, 2004), it is at present unclear whether such an approach can be generalized. The difficulty for all approximations to structural representations (grammar-based or otherwise) lies in accounting for *excluded volume* or *steric clashes* (the fact that no two amino acids can occupy the same point in space).

The so-called "New View" of protein folding (Dill and Chan, 1997) assumes that the speed of the folding process can be explained by the shape of the *energy landscape* (ie. the surface of the energy function for all possible structures of a given chain). Folding is fastest if the landscape is funnel-shaped (ie. has no local minima, and there is a direct downward path from all points to the native state). If the energy landscape is rugged (ie. has many local minima) or golf-course shaped (ie. all structures except for the native state have the same, high, energy), folding is slow. In the first case, energetic barriers slow down the folding process: the chain gets stuck in local minima, or kinetic traps. Such traps correspond to structures that contain "incorrect" (non-native) contacts which have to be broken (thus increasing the energy) before the native state can be reached. In the case of a plateau in the landscape, the search for the native state is slowed down by entropic barriers, i.e. a situation where a large number of equivalent structures with the same energy are accessible. Implicit in the landscape perspective is the assumption that folding is a greedy search – that local moves in the landscape can successfully identify the global minimum. Not all amino acid sequences have such landscapes, and in fact, most random amino acid sequences are unlikely to fold into a unique structure. This is again similar to language, where random sequences of words are also unlikely to form a grammatical sentence.

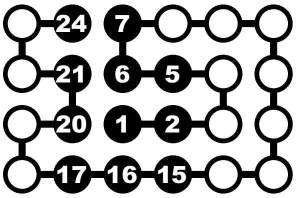

Figure 1: A conformation in the HP model with a "Greek key" β-sheet (1-17) and α-helix (17-24)

Computational simulations of the folding process are typically based on Monte Carlo or related techniques. These approaches require an energy function as well as a "move set" (a set of rules which describe how one conformation can be transformed into another). However, since each individual simulation can only capture the folding trajectory of a single chain, many runs are typically required to sample the entire landscape to a sufficient degree.

2.3 The HP model

The HP model (Lau and Dill, 1989; Dill et al., 1995) is one of the most simplified protein models. Here, proteins are short chains that are placed onto a 2-dimensional square lattice (Figure 1). Each HP sequence consists of two kinds of monomers, hydrophobic (H) and polar (P), and each monomer is represented as a single bead on a lattice site. The chain is placed onto the lattice such that each lattice site is occupied by at most one bead, and beads that are adjacent in the sequence are on adjacent lattice sites, so that it forms a self-avoiding walk (SAW) on the lattice. Such lattice models are commonly used in polymer physics, since they capture excluded volume effects, and the properties of such SAWs on different types of lattices are a well-studied problem in combinatorics.

Each distinct SAW corresponds to one "conformation", or possible structure. The energy of a conformation is determined by the contacts between two H monomers i and j that are not adjacent in the sequence. Contacts arise if the chain is in a configuration such that monomers i and j

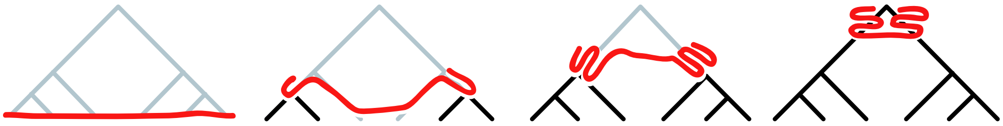

Figure 2: Trees describe folding routes. Tree cuts describe the state of the chain at any point in time.

$(i < j)$ are located on adjacent lattice sites. Each HH-contact contributes -1 to the energy. The energy $E(c)$ of a conformation c with n HH contacts is therefore $-n$. We consider only sequences that have a single lowest-energy conformation (native state), since these are the most protein-like. All unique-folding sequences up to a length of 25 monomers and their natives states are known (Irbäck and Troein, 2002). In our experiments, we will concentrate on the set of all unique-folding HP sequences of length 20, of which there are 24,900. These 20-residue chains have 41,889,578 viable conformations on the 2D lattice.

Despite its simplicity, the HP model is commonly used to test protein folding algorithms, since it captures essential physical properties of proteins such as chain connectivity and the hydrophobic effect, and since finding the lowest energy conformation is an NP-complete problem (Crescenzi et al., 1998; Berger and Leighton, 1998), as in real proteins.

3 Folding as hierarchical search

3.1 Evidence for hierarchical folding

There is substantial evidence in the experimental literature (starting with Crippen (1978) and Rose (1979); but see also Baldwin and Rose (1999a; 1999b)) that the folding process is guided by a hierarchical search strategy, whereby folding begins simultaneously and independently in different parts of the chain, leading initially to the formation of local structures which either grow larger, or assemble with other local structure. Folded protein structures can typically be recursively decomposed, and in many proteins, small, contiguous parts of the chain form near-native structures during early stages of the folding process. On the theoretical side, Dill et al. (1993) demonstrate that local contacts are easiest to form when the chain is unfolded, and facilitate the subsequent formation of less local contacts, leading to a "zipping" effect, where small, local structures grow larger before being assembled.

3.2 Folding routes as trees

Folding routes describe how individual chains move from the unfolded to the native state. If protein folding is a recursive, parallel process, as assumed here, folding routes are trees whose leaf nodes represent substrings of the primary sequence, and whose root represents the folded structure of the entire chain (Figure 2). The nodes in between the leaves and root correspond to chain segments whose length lies between that of the shortest initial segments and the final complete chain. Folding begins independently and simultaneously at each of the leaves, and moves toward the root. Each internal node of a folding route tree represents a set of partially folded conformations of the corresponding chain segment that is found by combining conformations of smaller pieces formed in previous steps.

Figure 2 also shows that the state of the entire chain at different stages during the folding process is given by a horizontal treecut, a set of nodes whose segments span the entire chain, but do not overlap.

Because we assume that folding routes are trees, contacts between two adjacent segments A and B can only be formed when A and B are combined to form their parent C. Our assumption also implies that in a sequence uvw, contacts between v and w or between v and u have to be formed before or at the same time as contacts between u and w.

Trees provide a unified representation of the growth and assembly process assumed by hierarchical folding theories: A growth step corresponds to a local tree in which a non-terminal node and a leaf node are combined, whereas an assembly step corresponds to a local tree in which two non-terminal nodes are combined.

Folding route trees thus play a very different role from the traditional phrase structure trees in natural language, since they represent merely the process by which the desired structure was formed, and not the structure itself. This is more akin to the role of syntactic derivations in formalisms such as CCG (Steedman, 2000): in CCG, syntactic derivation trees do not constitute an autonomous level of representation, but only specify how the semantic interpretation of a sentence is constructed. We will see below that proteins, like sentences in CCG, have a “flexible” constituent structure, with multiple folding routes leading to the native state.

4 Protein folding as chart parsing

Here, we show how the CKY algorithm (Kasami, 1965; Younger, 1967) can be adapted to protein folding in the HP model. Although we use a simplified lattice model, our technique is sufficiently general to be applicable to other representations. As in standard CKY, structures for substrings $i..j$ are formed from pairs of previously identified structures for substrings $i...k$ and $k+1..j$, and, as in standard probabilistic CKY, we use a pruning strategy akin to Viterbi search, and only retain the lowest energy structures in each cell.

The complexity of standard CKY is $O(n^3|G|)$, where n is the length of the input string and $|G|$ the “size” of the grammar. Since we do not have a grammar with a fixed set of nonterminals, which would allow us to compactly represent all possible structures for a given substring, the constant factor $|G|$ is replaced by an exponential factor n^c, representing the number of possible conformations of a chain of length n. Our pruning strategy captures the physical assumption that only locally optimal structures are stable enough not to unfold before further contacts can be made. With a larger set of amino acids and a corresponding energy function, a beam search strategy (with threshold pruning) may be more appropriate. Pruning is an essential part of our algorithm – without it, it would amount to exhaustive enumeration, repeated $O(n^3)$ times.

The chart Since only HH contacts contribute to the energy of a conformation, the dimensions of the chart are determined by the number of H monomers in the sequence. We segment every HP sequence into h substrings that contain one H each (splitting long substrings of Ps in the middle). For efficiency reasons, non-empty prefixes or suffixes of P monomers (eg. in sequences of the form $PPPH.....HP$) may also be split off as additional substrings (and are then only combined with the rest of the chain once the substring from the first to the last H monomer has been analyzed). These substrings correspond to the leaf nodes in the folding trees. Other regimes are also conceivable. Since no adjacent H monomers can form a contact, up to three consecutive Hs may be kept in the same substring. While this typically leads to an increase in efficiency, it comes at a slight cost in accuracy with our current pruning strategy. Long substrings of Ps could also be treated as separate substrings in a manner similar to P pre- and suffixes.

Chart items The items in our chart represent the lowest-energy conformations that are found for the corresponding substring. Unlike in standard CKY, each cell contains the full set of structures for its substring (which leads to the exponential worst-case behavior observed above). Therefore, the chart does not need to be unpacked to obtain the desired output structure. Backpointers from items in $chart[i][j]$ to pairs of items in $chart[i][k]$ and $chart[k+1][j]$ represent the folding route trees, and thus record the history of the folding process. Each item can only have at most $j-i$ pairs of backpointers, since it can only be constructed from one pair of conformations in each pair of cells.

Initializing the chart The chart is initialized by filling the cells $chart[i][i]$ which correspond to the ith substring. Since each initial substring has at most one H, all its conformations are equivalent (and the size of $chart[i][i]$ is thus exponential in the length of its substring). This exhaustive enumeration can be performed off-line.

Filling the chart As in standard CKY, the internal cells $chart[i][j]$ are filled by combining the entries of cells $chart[i][k]$ and $chart[k+1][j]$ for $i \leq k < j$. Two conformations $l \in chart[i][k]$ and $r \in chart[i][k]$ are combined like two pieces of a jigsaw puzzle where the only constraint is that two pieces may not overlap. That is, we append all (rotational and translational) variants of r to any free site adjacent to the site of l’s last monomer, and add all resulting viable conformations c (ie. those where no two monomers occupy the same lattice site) into $chart[i][j]$.

With our current pruning strategy, only the lowest-energy conformations in each cell are kept.

CKY terminates when the top cell, $chart[1][n]$, is filled. It has succeeded if the top cell contains an item with only one conformation, the native state.

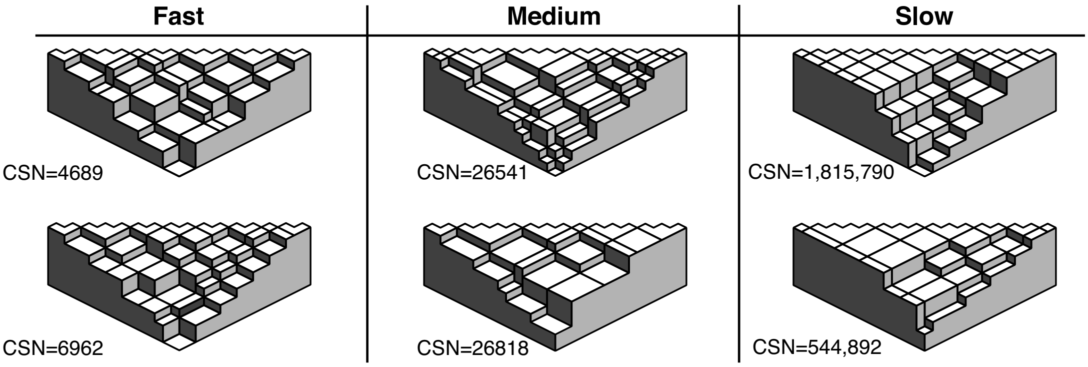

Figure 3: The amount of search depends on the shape of the ”chart energy landscapes”

Contact maps as node labels We have also developed a variant of this algorithm where the entries in a cell correspond to contact maps (sets of HH-contacts), and where each entry corresponds in turn to the set of conformations that corresponds to this contact map. Conformations that have the same contact map are assumed to be physically equivalent. While the number of possible contact maps is also exponential in the length of the substring (Vendruscolo et al., 1999), it is obviously much smaller than the number of actual conformations. In our current implementation, the amount of search required is identical in both variants; but in extending this approach beyond the lattice, it may be possible to use a more efficient sampling approach to speed up the combination of conformations in two cells.

5 Results

5.1 Folding accuracy

With our current pruning strategy, CKY finds the native state of 96.7% of all 24,900 unique-folding 20mers, confirming our hypothesis that the hierarchical greedy search that is implemented in CKY is a viable strategy. With exhaustive search, the “conformational search number” (CSN), ie. total number of conformations searched per sequence (summed over all cells), corresponds on average to 2.5% of all possible conformations for a sequence of length 20. We have also explored restrictions where an initial contact is only allowed between H monomers whose distance along the backbone is smaller than or equal to a given threshold Δ. For $\Delta = 7$, accuracy drops slightly to 95.2%, but the number of searched conformations corresponds to only 1% of the search space.

5.2 The chart landscape

Since we employ a beam search strategy, all conformations that remain in a cell after pruning have the same energy level. Therefore, CKY identifies the substring or *chart energy landscape* of each sequence, a function $f(i,j)$ which maps substrings (i,j) to their lowest accessible energy level. Since the energy of a conformation in the HP model is determined by the number of HH contacts, $f(i,j) \leq f(i',j')$ for all $i' \leq i, j \leq j'$. That is, unlike standard energy functions, f has no local minima. As shown in figure 3 (where the size of the cells is adjusted to reflect the length of the corresponding substrings), the “slope” of f determines the amount of search required to fold a sequence. Sequence that require little search have a steep funnel, whereas sequence that require a lot of search have a flat, golf-course like landscape. HH contacts impoose constraints on the number of conformations, therefore a cell with lower energy will also have fewer entries than a cell with higher energy that spans a string of the same length. This is analogous to standard energy landscapes (Dill and Chan, 1997), where a plateau corrresponds to an *entropic* barrier, which requires a lot of search.

5.3 The “constituent structure” of proteins

We can extract the set of all folding routes (all trees which lead to the native state) from the chart, visualize the ensemble-averaged “constituent structure” of a chain by coloring each cell in the (adjusted) chart by the posterior probability that native routes go through it (here black:p=1 and white:p=0). A probability of one corresponds to a structure that has to be formed by all routes, whereas a probability of zero represents a set of misfolded structures. Misfolding arises if the lowest energy structures contain non-native (incor-

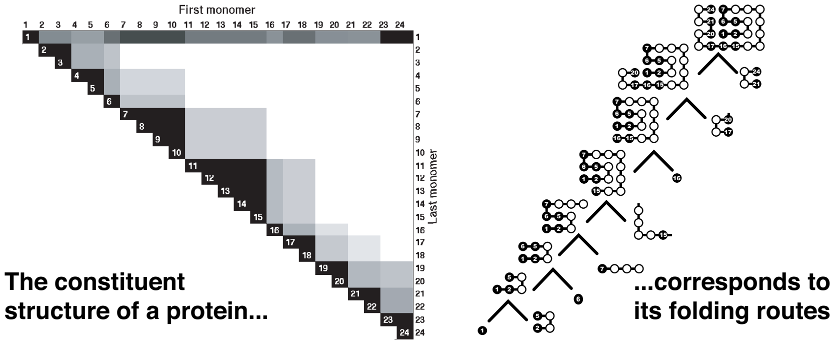

Figure 4: CKY identifies the “constituent structures” of proteins, which correspond to their folding routes

rect) contacts. Since these contacts have to be broken before the native state can be reached, requiring an uphill step in energy, they correspond to *energetic* barriers.

Figure 4 shows the “constituent structure” of the conformation shown in Figure 1, and one of its corresponding folding routes. Many sequences show very specific patterns of folding routes, as in the example given here, where the β-strands 7-10 and 11-16 and the α-helix from 17-24 “grow” onto the hairpin from 1-5.

A number of proteins are known to form so-called “foldons” (Maity et al., 2005). These are substrings of the chain which can be found in their near-native conformation before the entire chain is completely folded. In our parsing perspective on protein folding, these foldons correspond to nodes that are shared by sufficiently many native routes that they can be detected experimentally.

6 Conclusions and future work

This paper has demonstrated that an adaptation of the CKY chart parsing algorithm can be succcessfully applied to protein folding in the 2D HP model, a commonly used simplified lattice model which captures essential physical and computational properties of the real folding process. Both syntactic parsing and protein folding algorithms search for the globally optimal structure for a given input string. And any given sentence has a large number of possible interpretations, just as any amino acid sequence has an astronomical number of possible spatial conformations. Therefore it is not surprising if similar techniques can be applied to both tasks. In both cases, it seems to be possible to exploit locally available information with a greedy, hierarchical search strategy, which starts with local, independent searches for small substrings (to first determine which small phrases might make sense, or to find partially stable peptide structures) and then either: (a) ‘grows’ one substring into a larger substring, or (b) ‘assembles’ two substrings together into a larger substring. More interestingly, in the protein folding case, such recursive hierarchical search strategies, which imply tree-shaped folding routes, have been postulated independently for biological and biophysical reasons. This may indicate a deeper, natural connection between these two processes.

Given that hierarchical search strategies for protein folding have been proposed in the biological literature, our primary interest here has been the question of whether a greedy, hierarchical search as implemented in CKY is able to identify the native state of proteins in the HP model. The research presented here aims to verify these predictions with an explicit computational model. Therefore, we were less concerned with improving efficiency, and more with the properties of this algorithm, which we consider a baseline method upon which more sophisticated techniques such as best-first parsing (Caraballo and Charniak, 1998) or A* search (Klein and Manning, 2003) may well be able to improve.

We also plan to adapt this technique to other, more realistic, representations of proteins, and to longer sequences. For longer sequences, we will take advantage of the fact that CKY is easily parallelizable, since any operation which combines the entries of two cells $chart[i][k]$ and $chart[k+1][j]$ is completely independent of other parts of the chart.

If the routes by which proteins fold really are trees, a dynamic programming technique such as CKY is inherently suited to model this process, since it is the most efficient way to search all possible trees. This distinguishes it from more established techniques such as Monte Carlo, which can only follow one trajectory at a time, and require multiple runs to sample the underlying landscape to a sufficient degree. What CKY by itself does not give us is an accurate prediction of the rates that govern the folding process, including misfolding and unfolding events. However, we believe that it is possible to obtain this information from the chart by extracting all tree cuts (which corresond to the states of the chain at different stages during the folding process) and calculating folding rates between them.

Our work is only the beginning of a larger research program: eventually we would like to be able to model the folding process of real proteins. One aim of this paper was therefore to point out the fundamental similarities between statistical parsing and protein folding. We believe that this is a fertile area for future work where other natural language processing techniques may also prove to be useful.

Acknowledgements

This research is supported by NSF ITR grant 0205456. We would like to thank our colleagues at Penn and UCSF, in particular Vince Voelz, Banu Ozkan, John Chodera, David Chiang, Liang Huang, Fernando Pereira and Mitch Marcus, for many comments and conversations.

References

Christian B. Anfinsen. 1973. Principles that govern the folding of protein chains. *Science*, 181(96):223–230, July.

Robert L. Baldwin and George D. Rose. 1999a. Is protein folding hierarchic? I. local structure and peptide folding. *Trends Biochem. Sci.*, 24(1):26–33, January.

Robert L. Baldwin and George D. Rose. 1999b. Is protein folding hierarchic? II. folding intermediates and transition states. *Trends Biochem. Sci.*, 24(1):77–83, February.

Bonnie Berger and Frank Thomson Leighton. 1998. Protein folding in the hydrophobic-hydrophilic(HP) model is NP-complete. *Journal of Computational Biology 5(1): 27-40*, 5(1):27–40.

Sharon A. Caraballo and Eugene Charniak. 1998. New figures of merit for best-first probabilistic chart parsing. *Computational Linguistics*, 24(2):275–298.

David Chiang. 2004. *Evaluation of Grammar Formalisms for Applications to Natural Language Processing and Biological Sequence Analysis*. Ph.D. thesis, University of Pennsylvania.

Pierluigi Crescenzi, Deborah Goldman, Christos H. Papadimitriou, Antonio Piccolboni, and Mihalis Yannakakis. 1998. On the complexity of protein folding. *Journal of Computational Biology*, 5(3):423–466.

Gordon M. Crippen. 1978. The tree structural organization of proteins. *J. Mol. Biol.*, 126(3):315–32, December.

Ken A. Dill and Hue Sun Chan. 1997. From Levinthal to pathways to funnels. *Nature Structural Biology*, 4(1):10–19, January.

Ken A. Dill, Klaus M. Fiebig, and Hue Sun Chan. 1993. Cooperativity in protein folding kinetics. *Proc. Natl. Acad. Sci.*, 90:1942–1946, March.

Ken A. Dill, Sarina Bromberg, Kaizhi Yue, Klaus M. Fiebig, David P. Yee, Paul D. Thomas, and Hue Sun Chan. 1995. Principles of protein folding – a perspective from simple exact models. *Protein Science*, 4:561–602.

Ken A. Dill. 1999. Polymer principles and protein folding. *Protein Science*, 8:1166–1180.

Richard Durbin, Sean R. Eddy, Anders Krogh, and Graeme Mitchison. 1998. *Biological sequence analysis*. Cambridge University Press.

Anders Irbäck and Carl Troein. 2002. Enumerating designing sequences in the HP model. *Journal of Biological Physics*, 28:1–15.

T. Kasami. 1965. An efficient recognition and syntax algorithm for context-free languages. Scientific Report AFCRL-65-758, Air Force Cambridge Research Laboratory, Bedford MA.

Dan Klein and Christopher D. Manning. 2003. A* parsing: Fast exact Viterbi parse selection. In *Proceedings of HLT-NAACL '03*.

KF Lau and KA. Dill. 1989. A lattice statistical mechanics model of the conformational and sequence spaces of proteins. *Macromolecules*, 22:638–642.

Cyrus Levinthal. 1968. Are there pathways for protein folding? *J. Chim. Phys*, 65:44–45.

H. Maity, M. Maity, M. Krishna, L. Mayne, and S. W. Englander. 2005. Protein folding: The stepwise assembly of foldon units. *Proc. Natl. Acad. Sci.*, 102:4741–4746.

Sanzo Miyazawa and Robert L. Jernigan. 1996. Residue-residue potentials with a favorable contact pair term and an unfavorable high packing density term, for simulation and threading. *Journal of Molecular Biology*, pages 623–644.

George D. Rose. 1979. Hierarchic organization of domains in globular proteins. *J. Mol. Biol.*, 134:447–470.

David B. Searls. 2002. The language of genes. *Nature*, 420:211–217, November.

Mark Steedman. 2000. *The Syntactic Process*. MIT Press, Cambridge, MA.

Michele Vendruscolo, Balakrishna Subramanian, Ido Kanter, Eytan Domany, and Joel Lebowitz. 1999. Statistical properties of contact maps. *Physical Review*, 59:977–984.

D. H. Younger. 1967. Recognition and parsing of context-free languages in time $O(n^3)$. *Information and Control*, 10(2):189–208.