PROTEINS: Structure, Function, and Bioinformatics 66:1–15 (2007)

Routes Are Trees: The Parsing Perspective on Protein Folding

Julia Hockenmaier,[1]* Aravind K. Joshi,[1] and Ken A. Dill[2]
[1]*Institute for Research in Cognitive Science and Department of Computer and Information Science, University of Pennsylvania*
[2]*Department of Pharmaceutical Chemistry, University of California at San Francisco*

ABSTRACT **An important puzzle in structural biology is the question of how proteins are able to fold so quickly into their unique native structures. There is much evidence that protein folding is hierarchic. In that case, folding routes are not linear, but have a tree structure. Trees are commonly used to represent the grammatical structure of natural language sentences, and chart parsing algorithms efficiently search the space of all possible trees for a given input string. Here we show that one such method, the CKY algorithm, can be useful both for providing novel insight into the physical protein folding process, and for computational protein structure prediction. As proof of concept, we apply this algorithm to the HP lattice model of proteins. Our algorithm identifies all direct folding route trees to the native state and allows us to construct a simple model of the folding process. Despite its simplicity, our model provides an account for the fact that folding rates depend only on the topology of the native state but not on sequence composition. Proteins 2007;66:1–15. © 2006 Wiley-Liss, Inc.**

Key words: folding; HP model; contact order; dynamic programming

INTRODUCTION

Protein molecules have an astronomical number of possible three-dimensional conformations. Yet, they fold quickly into unique, sequence-specific native structures. What makes the folding process so fast, and so reliable? The search for the native state cannot be random or exhaustive; it has to be biased or guided in some way.[1,2] An understanding of this underlying bias or guiding principle might lead to more efficient algorithms for protein structure prediction, and could therefore have important practical implications. In this paper, we assume that the folding process is guided by a parallel, hierarchical search for locally optimal, that is, lowest energy, structures. We show that this hypothesis allows us to adapt efficient dynamic programming techniques for ab initio protein folding, and argue that it provides an explanation for the well-known fact that folding rates depend on the topology of the native structure.[3]

There is much evidence that folding is hierarchical: proteins have recursive domains,[4–6] some small peptide segments (“autonomous folding units”) fold into near-native structures by themselves, and for some proteins, stable intermediates that consist of secondary structure elements (“foldons”) have been observed during folding.[7] Also, theoretical models, for example the hydrophobic zipping model,[8] have shown that at the beginning of the folding process the formation of very local contacts is favorable, and that these local contacts in turn facilitate the formation of contacts that are increasingly non-local. Proposals for hierarchical folding algorithms have been made before.[9] However, no computational model is yet able to identify the native state and all hierarchical folding pathways from the primary sequence alone.

Based on the assumption that the folding process is hierarchical, we suggest that folding routes have an underlying tree structure. This leads us to propose an adaptation of the Cocke–Kasami–Younger (CKY) chart parsing algorithm[10,11] for structure prediction and protein folding. As proof of concept, we apply this algorithm to the simple HP lattice model[12,13] of proteins. CKY is an efficient dynamic programming technique that was originally developed for so-called context-free grammars, and is commonly used in compilers for programming languages and to predict the sentence structure of natural languages. CKY searches all possible binary trees over the input sequence in a parallel, greedy fashion. Like all dynamic programming approaches, it recursively decomposes the global search problem—in our case, the search for the native state of the entire sequence—into a number of smaller, local searches (for the best structures of shorter sequence fragments). The results of these local searches (the structures of the fragments) are stored in a table, called the parse chart, and re-used in the search for larger structures. In contrast to other folding algorithms, CKY is

Grant sponsor: ITR, NSF; Grant number: 0205456; Grant sponsor: NIH; Grant number: GM34993.

*Correspondence to: Julia Hockenmaier, Institute for Research in Cognitive Science, University of Pennsylvania, 3401 Walnut Street, Suite 400A, Philadelphia PA 19104-6228 USA.
E-mail: juliahr@cis.upenn.edu

Received 21 March 2006; Revised 4 July 2006; Accepted 5 July 2006

Published online 24 October 2006 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/prot.21195

therefore able to return all hierarchical folding routes that lead directly to the native state. From the output of our algorithm, we construct a very simple model of folding rates. These rates depend on the amount of entropic search that is required to find the native state directly, and on the prevalence of energetic barriers. It is well know that folding rates of real proteins are strongly correlated with the topology of the folded structure,[3] but not with the composition of the primary sequence. Despite the obvious simplifications of the physics in our model, the folding rates we predict show a similar correlation with native topology and are also independent of sequence composition.

EVIDENCE FOR HIERARCHICAL FOLDING

There has been much interest in identifying a principle or mechanism by which proteins fold; that is, a general explanation of how the search can be so efficient, despite the diversity of microscopic trajectories and folded protein structures. There is substantial evidence that the microscopic folding process may be guided by a hierarchical search strategy*: several researchers have proposed that secondary structures form earlier than tertiary structures or that secondary structures nucleate tertiary contacts or both.[5,7,8,16–20] Furthermore, Lesk and Rose[5,6] show that the domains in globular proteins are recursively decomposable, and argue that this structural property is a consequence of the folding process, which they assume follows hierarchic pathways, or trees. Apparent counterevidence against hierarchical search comes from hydrogen exchange experiments involving cytochrome c[7], which identify an intermediate state where the chain segments at the N and C terminal form a unit before they combine with intervening segments.[21] However, cytochrome c has a large heme cofactor, and folding simulations[22] indicate that the heme plays a critical role in stabilizing the contact between the terminal segments. This indicates that cytochrome c may be a special case. Also, recent work[23,24] demonstrates that hierarchical pathways can be found for all single chain proteins in the PDB, despite the prevalence of N-C terminal contacts in two-state folders, indicating that static native structures are not sufficient to determine the folding routes.

It has also been shown recently that experimental Φ values, which give information about protein folding routes, can often be predicted qualitatively by a mechanism in which local pieces of structure form independently throughout the chain, and either grow (zip) or coalesce (assemble) with other such pieces, leading to the accretion of native structure.[8,19,25–28] Such a mechanism of zipping and assembly also predicts tree-shaped folding pathways (see below). However, all of these models are based on prior knowledge of the native structure of the protein. What is needed is a method that has access only to the primary amino acid sequence, and finds all folding routes without a prohibitively expensive computational search. The algorithm presented in fourth section is a proof of principle of a method that can do this. For simplicity's sake, we focus only on direct folding routes, that is, those paths to the native state that involve no misfolding and subsequent unfolding. We believe these are the dominant routes for fast-folding two-state proteins.

*See also reviews of Baldwin and Rose[14,15] for experimental evidence.

Fig. 1. Trees describe folding routes. Horizontal treecuts describe the state of the chain at any point in time. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

FOLDING ROUTES AS TREES

Protein folding kinetics are often modeled in terms of very broad macrostates. However, the coarse macroscopic level of description that is commonly obtained from laboratory experiments is not sufficient to design computational models and folding algorithms. For this purpose, an explicit understanding of the microscopic processes by which individual chains search and reach their native structure is necessary. In order to capture the hierarchical, parallel nature of the folding process, we represent the microscopic folding routes as trees† whose leaf nodes represent short segments of covalently linked chain (substrings), and whose root represents the entire chain (Fig. 1). 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. If a folding route leads to the native state, this set of conformations will include the native substructure of the corresponding chain segment (as shown in Fig. 1), but may be much larger if the native substructure is not uniquely determined by the segment alone. Therefore, our definition of folding route trees neither requires secondary structure elements to be stable without tertiary contacts, nor does it preclude this possibility.

Because we assume that folding routes are trees, only conformations of sequence-adjacent, non-overlapping segments can be combined. This means that if node A spans the chain segment from monomer i to k, and node B spans the segment from monomer $k + 1$ to j, their corresponding structures can be combined to form a new node C (the par-

†Formally, a (rooted) tree is defined as a connected acyclic graph $G = \{\mathbf{V},\mathbf{E}\}$, consisting of a set of vertices (nodes) $\mathbf{V}$ and a set of directed edges (links) $\langle V_i, V_j\rangle$ from one node $V_i \in \mathbf{V}$ to another node $V_j \in \mathbf{V}$ ($V_j \neq V_i$). $\mathbf{V}$ contains a special root node, which has no incoming edges. With the exception of the root, every node V has exactly one incoming edge $\langle$W,V$\rangle$. W is the parent (direct ancestor) of V. Nodes that have no outgoing edges are called leaf nodes. Following standard usage in graph theory, we assume that edges are directed away from the root, although the folding process that is represented by a folding route tree moves from the leaves to the root.

ent of A and B), which spans the larger segment from monomer i to j. But A cannot be combined with a node D that spans the segment from l to m (unless $l = k + 1$ or $i = m + 1$). Our assumption that folding routes are trees also implies that contacts between the segments of A and B can only be formed when A and B are combined into C.

Furthermore, we assume that only the lowest energy conformations that can be found for a segment are stable enough not to unfold on a time scale needed to reach the next stage. Computationally, this corresponds to a greedy search, which retains only the optimal conformations at each node.

Each node represents a set of energetically equivalent structures. In the HP model, the energy of a conformation is determined by the number of non-local interactions, or contacts, between hydrophobic residues. Two conformations that have the same number of hydrophobic contacts are therefore energetically equivalent. But two energetically equivalent conformations are not physically equivalent if the transition from one to the other requires one or more hydrophobic contacts to be broken. In order to calculate route-specific folding times (fifth section), we therefore consider two conformations equivalent if they have the same contact map, that is, the same set of hydrophobic contacts, and label the nodes of folding route trees with these contact maps. There can be multiple lowest-energy contact maps for the same chain segment, corresponding to different nodes (with the same energy, but belonging to different folding route trees).

Each node in a folding route tree can be characterized as either a "growth" or an "assembly" event. A growth step combines either two leaf nodes (unfolded parts of the chain), or adds one leaf node to an internal node (partially folded structure). In contrast, an assembly step combines two internal nodes. In Figure 1, the second and third trees show growth steps, whereas the last tree shows an assembly step. The more assembly steps a route has, the flatter its tree, and the more parallel its folding process. Figure 1 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.

For technical reasons explained below, we will assume that all folding route trees are binary branching. Some folding events, such as the zipping up of a hairpin may be better described as a ternary tree. Our algorithm will instead identify two distinct binarized versions of this tree.

PROTEIN FOLDING AS PARSING

Folding route trees describe the process by which a protein may find its native state. Trees are also commonly used in linguistics, because the grammatical structures of natural language are also recursive and hierarchical.[29] In order to understand and distinguish the meaning of sentences such as *We eat sushi with tuna* and *We eat sushi with chopsticks*, it is necessary to parse them, that is to identify their correct syntactic structure. Figure 2 shows the possible parse trees for both sentences. Each node represents a "constituent" of the sentence, and is labeled by its corresponding syntactic category, for example, S (sentence), NP (noun phrase), VP (verb phrase), or PP (prepositional phrase). As this example shows, the correct parse depends on the words in the sentence, just as the primary sequence of a protein determines its native structure.

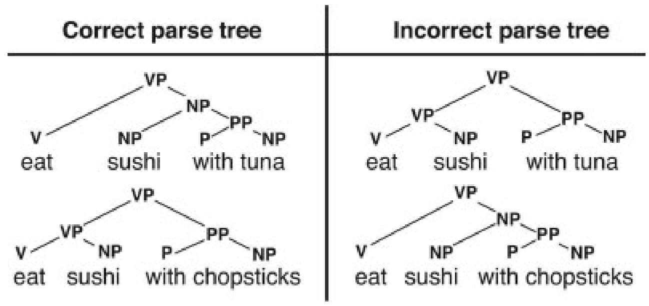

Fig. 2. Parse trees show the grammatical structures of sentences.

Parsing is the computational task of identifying the correct syntactic structure of an input string according to a grammar that defines the set of possible syntactic structures. Chart parsing algorithms such as CKY are dynamic programming techniques that exploit the independence assumptions implicit in the tree representation to search all possible trees for the given sequence of words in an efficient and systematic manner. Statistical parsers[30,31] rely on a probability model to rank competing analyses and to prune away unlikely structures early on in order to speed up the search process. In our adaptation of CKY to proteins, the folding process is also guided by a greedy search in which each step identifies those conformations whose energies are locally optimal among all the possible conformations it identifies for the same chain segment.

The HP Model

We illustrate our search method for protein folding using the HP model.[12,13] In the HP model, proteins are short chains having access to an ensemble of conformations on a 2-dimensional square lattice. 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 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 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.[32] 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.

The HP model has been commonly used to test protein conformational search protocols. It is the model of choice here because (1) the full conformational space of a protein molecule can be completely enumerated, so the native state can be known with no assumption or approximation, (2) the model has the same NP-complete search problem as in real proteins,[33,34] where the size of the search space grows exponentially with chain length, and yet, for many sequences, there is only a single native structure, and (3) despite its simplicity, it captures the physics of hydrophobic interactions, steric excluded volume, and chain conformational freedom that are key components of real protein folding. Thus the HP model allows much more extensive and unambiguous testing of search strategies than other models.

Adapting CKY to the HP Model

CKY[10,11] is a dynamic programming algorithm that was originally developed for context-free grammars.[‡] Although originally developed for the analysis of language, context-free grammars are also used in bioinformatics, for example, to model RNA hairpin structures.[35] An introduction to CKY that is geared towards bioinformaticists can be found in the textbook of Durbin et al.[36]

The main insight behind CKY is that parse trees can be constructed recursively from smaller trees, and that constituents such as the PPs (*with tuna*) in Figure 2 can be stored in a so-called parse chart, and re-used in multiple analyses of the same string. A *parse chart* is a table in which the analyses of all substrings (sentence fragments) are stored. If the input sentence contains n words, the chart is an $n \times n$ array whose cells *chart* $[i][j]$ contain data structures that represent parse trees for substrings from the ith to the jth word in the input string. Since the rules of context-free grammars only specify the labels of the parent and immediate child nodes, all nodes that span the same string and have the same label can be considered equivalent. This insight is exploited in CKY, where each item, or element in a cell, is a data structure that specifies a node label and a list of pairs of backpointers to the child (edges) from which this node can be obtained. Therefore, the cell that corresponds to *eat sushi with tuna* contains one item, with label VP and two pairs of backpointers {⟨VP, PP⟩, ⟨V, NP⟩} to the items in the corresponding cells. Although the number of possible trees is exponential in the length of the string, the number of items in any cell in a parse chart (and thus the number of operations required to combine the items in two cells) is bounded by the number of syntactic categories (node labels) defined by the grammar. CKY's complexity is therefore polynomial ($O(n^3|G|)$ for a string of n tokens (words) and a context-free grammar G with $|G|$ rules). When CKY is used with probabilistic context-free grammars,[37] where each rule is associated with a numerical weight, greedy search strategies are typically used, so that in each cell only those structures with the highest scores (Viterbi search) or with scores that are close to the highest score (beam search) remain.[§]

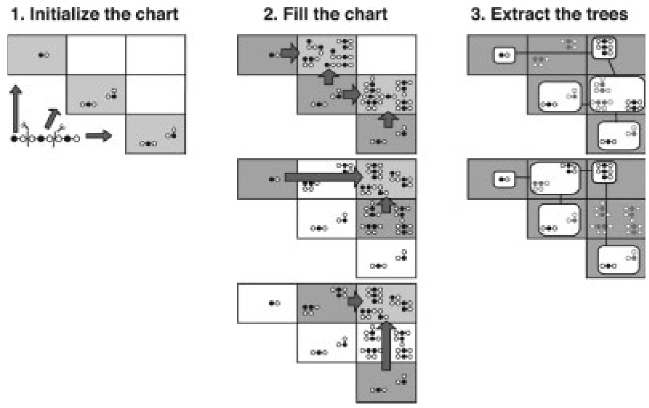

Fig. 3. The CKY algorithm. **1:** The sequence is split into n fragments. The chart is an $n \times n$ table. The diagonal cells contain the structures of the initial fragments. **2:** The chart is filled by combining pairs of structures from adjacent fragments. Only the lowest energy-structures are retained in each cell. **3:** If folding is successful, the native state is in the top right cell, and all folding route trees can be extracted by recursively following the backpointers to the smaller structures.

Here we adapt CKY to the HP model (Fig. 3). We also use a greedy search strategy akin to Viterbi search, and keep only the lowest-energy conformations for each substring.[¶] Instead of a predefined grammar with a fixed set of categories, we use contact maps (sets of hydrophobic contacts) as node labels. These contact maps describe sets of equivalent conformations for the corresponding chain segment. Like the constituent labels in a grammar, they define pathway-independent state variables, which make it possible to use an algorithm like CKY to systematically explore all folding routes.

The chart

Since only HH contacts contribute to the energy of a conformation, the full HP sequence of the whole protein is split into n substrings that contain one H each. The chart is a table of size $n \times n$. Each cell *chart*$[i][j]$ is associated with the substring that begins with the first monomer of the ith substring and ends in the last monomer of the jth substring. Cells *chart*$[i][i]$ along the main diagonal are associated with the ith initial substring and correspond to the leaf nodes of the folding routes, whereas the top cell *chart*$[1][n]$ is associated with the entire sequence (and thus the root of all native folding route trees).

‡CKY requires a grammar in so-called Chomsky Normal Form, where all trees are binary branching. But since every context-free grammar can be transformed (reversibly) into an equivalent grammar in Chomsky Normal Form, CKY can be used with any context-free grammar. For simplicity's sake, we will assume that folding routes are binary trees.

§For more information on CKY as applied to natural language, the interested reader is referred to the textbooks of Jurafsky and Martin[38] and Manning and Schütze.[39]

¶Since the number of possible conformations is exponential in the length of the chain, the worst case complexity of our algorithm is exponential. This is to be expected, since protein folding even in the simplified HP model is known to be NP hard. More important for practical purposes is the observed average case complexity, which we examine below.

Chart items

The items in our chart represent nodes in folding route trees as well as their associated conformations. An item I in cell $chart[i][j]$ is a tuple $\langle i, j, \mathbf{C}, C, Children\rangle$. It corresponds to the set of conformations C (for the substring of $chart[i][j]$) that have all and only those HH contacts specified by the contact map $\mathbf{C}$. I also represents all folding route trees (for its substring) with root $\mathbf{C}$. These trees are not enumerated, but are stored implicitly in *Children*, a set of pairs of backpointers $\langle L,R\rangle$ to items L in $chart[i][k]$ and R in $chart[k+1][j]$ (with $i \leq k < j$). If L represents a set of l trees with root $\mathbf{L}$, and R represents a set of r trees with root $\mathbf{R}$, $\langle L,R\rangle$ identifies a set of $l \times r$ trees with root node $\mathbf{C}$, left child node $\mathbf{L}$ and right child node $\mathbf{R}$. Items in the diagonal cells $chart[i][i]$ represent leaf nodes, and have no *Children*. We will refer to the set of conformations represented by all items in the same cell as the entries of that cell, and define the size of a cell as its number of entries.

Instead of grouping conformations by their contact maps, the items in each cell could also correspond to the conformations themselves. This would not affect the amount of search required to fill the chart, but would make it more difficult to calculate folding rates or other physically meaningful quantities.

Initializing the chart

The chart is initialized by filling the cells $chart[i][i]$ that correspond to the ith substring. Since each initial substring has at most one H, these cells contain only one item whose contact map is the empty set. The conformations of these initial items are enumerated. We will refer to the set of conformations represented by all items in the same cell as the entries of that cell, and define the size of a cell as the size of its set of entries. The size of the initial cell $chart[i][i]$ is thus exponential in the length of its substring.

Filling the chart

The internal cells $chart[i][j]$ are filled by combining the entries of cells $chart[i][k]$ and $chart[k+1][j]$. For $\Delta = 1...n$, and for $i = 1...n$, each cell TARGET $= chart[i][j]$ (with $j = I + \Delta$) is filled by combining the items of all pairs of cells $\langle$LEFT, RIGHT$\rangle = \langle chart[i][I+k], chart[k+1][j]\rangle$ (for $0 \leq k < I + \Delta$). Two items $L \in$ LEFT and $R \in$ RIGHT are combined by combining all their entries $l \in L$ and $r \in R$. The conformations l and r are combined by appending all (rotational and translational) variants of r to any free site adjacent to the site of l's last monomer. All resulting viable conformations c whose energy is identical to or lower than that of the current entries in TARGET are entered into TARGET. Since we use a greedy search strategy in which cells store only those items that correspond to the lowest obtainable energy for their substring, all current items in TARGET are deleted if c's energy is lower than that of the current entries in TARGET.** If there is no item in TARGET with c's contact map $\mathbf{C}$, a new item with label C (and one entry, c) is created. Otherwise, c is only added to the item C if there is not already an entry in C that is a (rotational or translational) variant of c. In order to keep track of the folding route trees, a pair of backpointers to $\langle L,R\rangle$is also added to $\mathbf{C}$'s *Children*. 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.

The representation of folding routes

Like standard CKY, our algorithm represents folding route trees in a "shared packed forest," where each item corresponds to a set of trees (or nodes). In standard CKY, the trees themselves are the structures that we are interested in, but in our algorithm, they only represent a record of the process by which chain conformations (the structures we are actually interested in) are formed. Because each node corresponds to a set of conformations, node-sharing is only possible if all nodes with the same contact map label C for the same substring correspond to the same set of conformations. Since a conformation c is only entered into an item I if it has exactly the HH contacts specified by the item's contact map $\mathbf{C}$, I corresponds to only those conformations that have the HH contacts specified by $\mathbf{C}$. If I corresponds to an initial substring, its entries are enumerated exhaustively.†† When two items L and R are combined, all possible combinations of their entries are tried. Thus, if L and R contain all conformations specified by their contact maps $\mathbf{L}$ and $\mathbf{R}$, all conformations that have at least the contacts specified by $\mathbf{L}$ and $\mathbf{R}$ will be found (and associated with their appropriate contact map).

Reducing the amount of search

As described above, the CKY method searches all possible folding route trees of the protein. This may include routes where the first contacts are between monomers that are far apart in the sequence. However, as demonstrated by Fiebig and Dill,[25] in the absence of sequence-local contacts, the formation of such non-local contacts is highly unlikely. The locality of a contact between monomers n and m is indicated by its contact order[3] (CO), or the distance between n and m along the backbone: CO $= |m - n|$. We therefore restrict CKY's search to folding routes where the first (initial) contacts have a contact order that is equal to, or lower than, a given threshold Δ: if the conformations in both $chart[i][k]$ and $chart[k+1][j]$ have no HH contact, they are only combined if one of the H monomers in the span of $chart[i][k]$ can form an HH contact having a contact order $\leq\Delta$ with a monomer to its right or if one of the H monomers in the span of $chart[k+1][j]$ can form an HH contact with CO $\leq \Delta$ with a monomer to its left. In the following, we vary Δ from 3 to 11. We find that these initial contact order (ICO) restrictions greatly reduce the amount of search required,

**Less greedy beam search strategies are equally possible, and may be more appropriate if more than two types of monomers (and a corresponding, more complex, energy function) are considered.

††If the first or last monomer is not sequence-final, there has to be at least one free lattice site adjacent to it, in order to guarantee chain connectivity.

and yet rarely prevent the process from finding the single globally optimal native conformation.

Search Efficiency and Prediction Accuracy

There is a general assumption in the literature that hierarchic assembly of locally stable structures minimizes the amount of search required. CKY, which implements precisely such an assembly process, allows us to examine this assumption. CKY's overall search strategy is greedy, because it only retains the optimal conformations in each cell. However, in order to fill a cell, an exhaustive search of all the possible combinations of conformations from prior cells is performed. While a more informed search strategy may be preferable from an engineering point of view, we are here interested in examining the efficiency of such an exhaustive, hierarchical search.

To explore the search efficiency, let s_α be the total number of conformations that CKY searches in all cells when folding a particular HP sequence α (of length n). We assume that the size of the search space S_α is equal to the number of possible conformations for sequences of length n on the 2D lattice, S_n. S_n is exponential in n. Then the relative amount of search is given by s_α/S_α, and the speedup, or search reduction factor, in comparison with exhaustive enumeration is given by $r_\alpha = S_\alpha/s_\alpha$. We determine the "success" of CKY on a set of sequences as the percentage of those sequences for which CKY finds the native state. Our simulations show that the CKY method is successful, yet efficient (see Fig. 4), and therefore demonstrate the viability of our proposed hierarchical search strategy. This means that CKY generally finds the single globally optimal conformation (known from prior exhaustive enumeration) while searching only a small fraction of the whole conformational space to reach it. The ICO restrictions, which are similar in spirit to the hydrophobic zipping model,[8,25] are particularly effective in reducing the amount of search required.

Search Entropy and Energy Landscapes

We find that the total amount of search varies greatly among sequences of the same length. What is the reason for these differences?

Since our greedy search retains only the lowest energy conformations in each cell of the parse chart, every cell corresponds to a specific energy level, and thus the parse chart itself serves as a surrogate energy landscape (the "chart landscape"), which maps substrings to their lowest kinetically accessible energy levels. This energy level is 0 for the cells along the main diagonal, which correspond to the initial chain segments, and equal to the energy of the native state in the corner cell most distant from the main diagonal.

These chart landscapes (Fig. 5) show that the fast folders (i.e., those involving the quickest conformational searches) have broad funnel-shaped landscapes, while the slow folders have landscapes that are narrower and more golf course-like.

This relationship between folding speeds and the shapes of chart landscapes is readily understood. Any decision to form a contact (even a wrong one!) reduces not only the energy, but also the entropy (i.e. the number of conformations that remain) in the corresponding cell, and therefore also reduces subsequent conformational searching. Therefore, there is a sort of "leveraging" of the search: contacts that are formed early reduce the overall amount of search far more than contacts that are formed towards the end of the folding process. When two partially folded segments are combined, the lowest energy structures among the resulting conformations typically have an even lower energy (and entropy), resulting in the observed funnel shape for the sequences that require the least amount of search. Since the bottleneck in our approach usually involves one or more conformational search steps, it is also consistent with the entropic transition state ensemble of Wallin and Chan.[40]

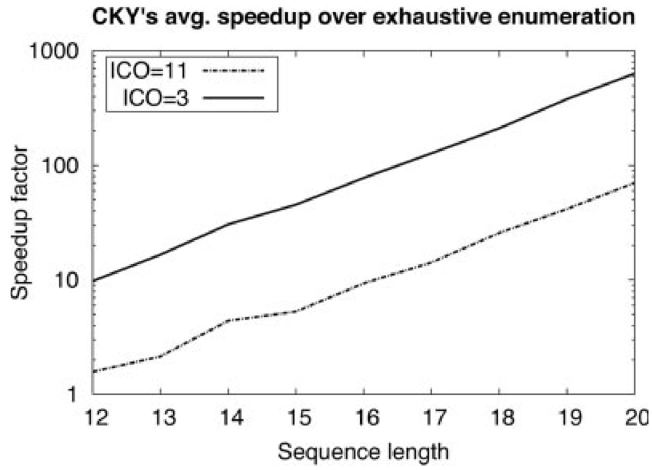

Fig. 4. Accuracy and search speedup of CKY. Left: accuracy and speedup on all unique-folding 20mers (24,900 sequences, $S = 41{,}889{,}578$ conformations). Right: average speedup for different lengths.

Even though non-native contacts can reduce the search entropy in the same way as native contacts, these chart landscapes also indicate why low contact order-proteins (e.g. α-helices) fold faster than proteins with a high native contact order (e.g. β-sheets): contacts between the ith and jth initial segment can only be found in conformations that are in $chart[i][j]$ or in cells above it or to its right. Therefore, non-local, high contact order contacts can only be formed in cells that are distant from the main diagonal and are filled relatively late during the search process, whereas local, low contact order contacts are formed early on when cells that are close to the main diagonal are being filled, and lead to a funnel-shaped chart landscape. A plateau on the chart landscape corresponds to a long segment for which no contacts could be found. Proteins whose native state has many local contacts are unlikely to have such a plateau on their chart landscape.

PHYSICAL PLAUSIBILITY AND IMPLICATIONS

Here, we show that our hierarchic search method resembles the physical search process insofar as it predicts the dependence of folding rates on the topology of the native state and not on the composition of the primary sequence.

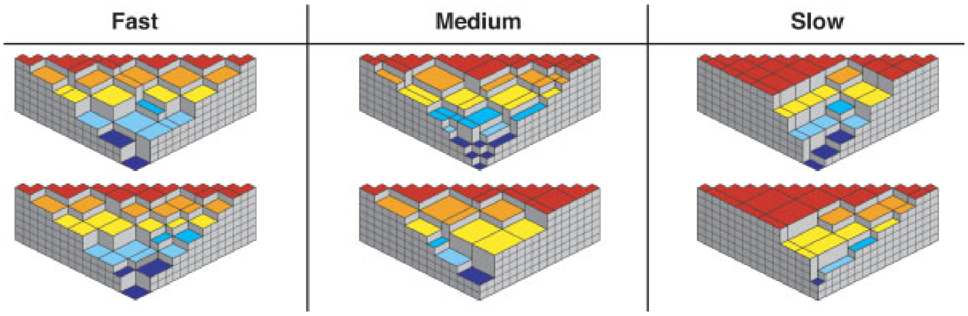

Fig. 5. The shape of the chart landscape determines the total amount of search.

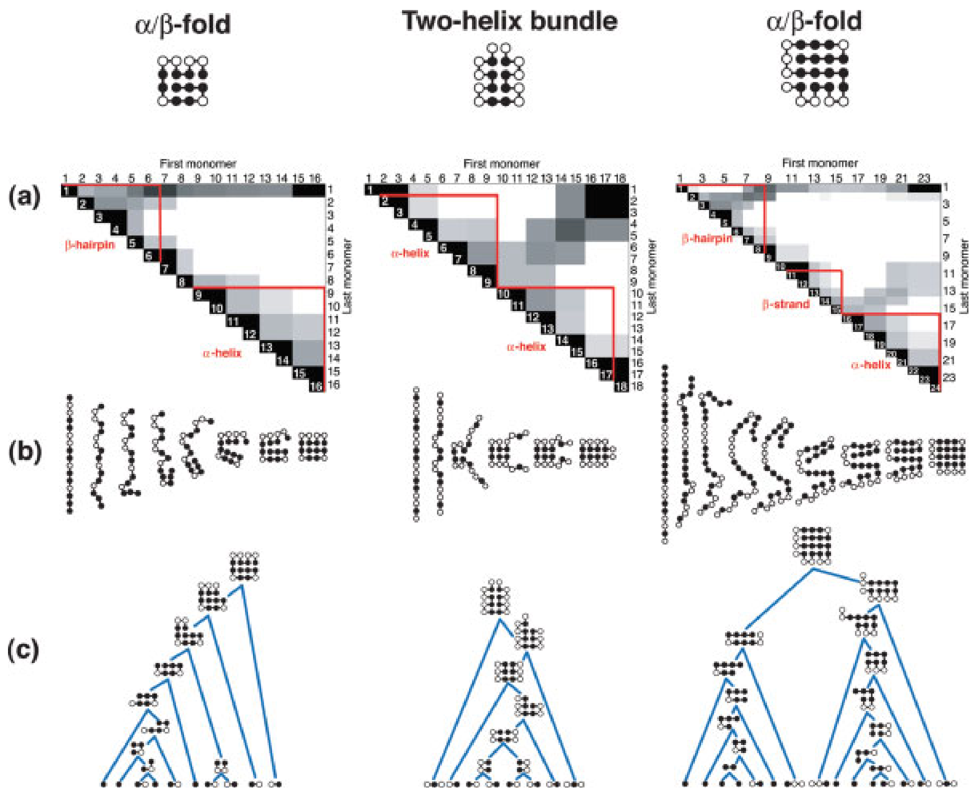

Fig. 6. Proteins have sequence-specific patterns of folding routes. The color of the cells in these parse charts (**a**) indicates their posterior probability of being on a native route (black: $P = 1$, white: $P = 0$). This corresponds to the ensemble-averaged folding routes shown in (**b**). A microscopic folding route tree is given in (**c**).

Distinct Macroscopic Pathways and Foldons Emerge

Theoretical models often predict a multiplicity of microscopic folding trajectories. But experimentalists frequently observe simple, distinct macroscopic pathways that are inconsistent with the diverse and heterogeneous folding process that theorists seem to imply.

Our algorithm may be able to reconcile these two perspectives. CKY returns the set of all microscopic folding route trees that lead directly to the native state without forming (and subsequently breaking) any non-native contacts. We find that all proteins may fold via a multitude of different routes, just like most sentences of natural language can have different syntactic analyses that achieve the same semantic interpretation.[41] However, we also find that typically only a small fraction of all possible routes leads to the native state, and that when these microscopic routes are ensemble-averaged, specific macroroutes emerge (Fig. 6). These sequence-specific patterns arise because, for many substrings, the lowest energy structures that are found by our greedy search method contain non-native contacts and cannot therefore correspond to nodes of any native folding route tree.[‡‡] Under the assumption

[‡‡]We note that we would not be able to observe this behavior in a GO model, which assumes that all potential contacts are native. A GO model approach would erroneously lead to the conclusion that all routes lead to the native state.

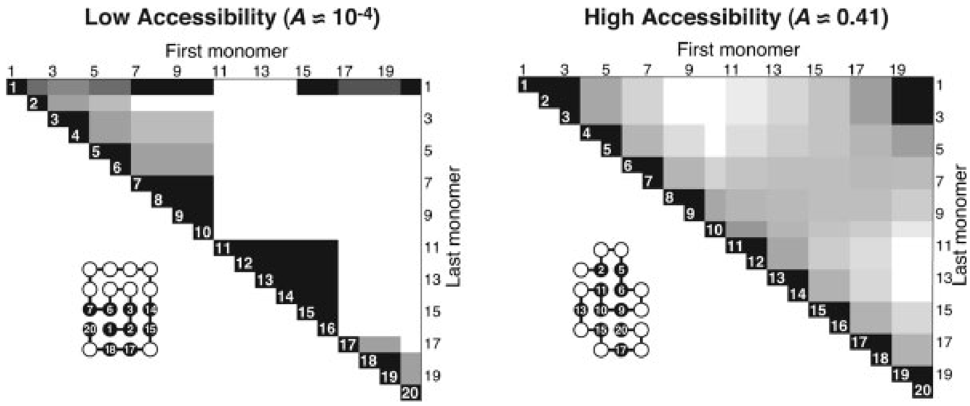

Fig. 7. Two structures with low (left) and high (right) accessibility.

that misfolded chains have to at least partially unfold before reaching the native state, the entire population of chains will eventually have to follow one of the native routes identified by our algorithm.

Many proteins are known to form partial intermediates, or "foldons,"[7] which consist of contiguous chain segments that are in near-native conformations. We believe that these foldons correspond to nodes of native folding route trees with low entropy, and are therefore manifestations of the hierarchical folding process.

Ensemble-averaged folding routes (projected onto CKY's parse chart) for three different HP sequences are shown in Figure 6. All native routes start in the cells along the main diagonal (which represent the leaf nodes of the folding route trees, or the initial segments into which the HP sequence was split), and, by definition, reach the top cell (which represents the entire chain, its possible structures, and the root node of all native folding route trees). In the first α/β structure, the hairpin can be formed in a number of ways, and the α-helix can grow onto it in a number of ways, but there is no native route where the hairpin and the entire helix assemble. In the HP model, the later pathway is excluded by our greedy search, because every substring *HPPHHPPH* folds into two turns which fold onto each other, as in the bottom of the 2-helix bundle. Therefore, the 2-helix bundle can only grow from the middle to the top, but it cannot be formed by assembly of two fully formed helices. The second α/β fold has 16,295 distinct routes, yet they are all broadly similar: in all cases, the β-hairpin zips up, and the third β-strand and helix grow onto it.

The Accessibility of the Native State

Folding rates are determined by the amount of entropic search required, and by the likelihood of the chain to get stuck in a kinetic trap, that is to misfold. We will now develop a very simple model that aims to capture those two factors. Kinetic traps arise if the chain forms any non-native contacts. Since these contacts have to be broken, such traps are separated from the native state by energetic barriers. How likely is the unfolded chain to find the native state directly, without encountering such traps or energetic barriers? We will refer to this probability as the accessibility of the native state, A. The complement of A, $1 - A$, indicates the probability of misfolding. Under the simplifying assumption that all possible routes are equiprobable, and excluding the possibility of pathways where the temporary formation of a non-native contact is a prerequisite for a native contact, we obtain an estimate of A from $N_{\tau,}$ the number of direct folding route trees to the native state returned by CKY.

In order to calculate A, we first observe that each folding route is a binary tree whose nodes are labeled with contact maps. The number of possible labeled trees is unknown. However, the number of possible unlabeled binary trees with n leaves is given by the Catalan number C:

$$C_n = (2n)!/((\mathrm{n}+1)!n!)$$

which is the number of possible binary bracketings of a string of $n + 1$ words. Therefore, $\mathrm{A} = p_{\mathrm{tree}} \times p_{\mathrm{label}}$ is the product of the probability of choosing a correct unlabeled tree and the probability of choosing the correct contact map labels for the nodes in this tree. $p_{\mathrm{tree}} = N_\tau/C_{n-1}$ because each folding route returned by CKY corresponds to a different unlabeled tree. Since all leaf nodes have the same label, p_{label} factors into the probabilities $p_{L,R \to \mathrm{P}}$ that for each pair of sister nodes L and R, the correct contact map label for the parent P is chosen. Therefore, p_{label} can be calculated efficiently by CKY. It can also be defined such that among possible parents with the same number of HH contacts, those with high entropy (and thus lower free energy) are preferred (Appendix C).

Figure 7 shows two 20mers that both have 10 H-monomers, and thus 4862 possible folding routes. The structure on the right has 2842 native routes and very high accessibility ($A = 0.41$). In this sequence, all possible (i, $i + 3$) and (i, $i + 5$) contacts are native, and only some (i, $i + 7$) contacts ((2,9) and (13,20) are not. By contrast, the structure on the left has only 27 native routes, and thus very low accessibility ($A = 0.0001$). Here, the segment 1-10 and 11-16 have to be assembled. Native routes can only differ in the way in which the segment 1-10 is formed, and how exactly the strand 17-20 grows onto 1-16.

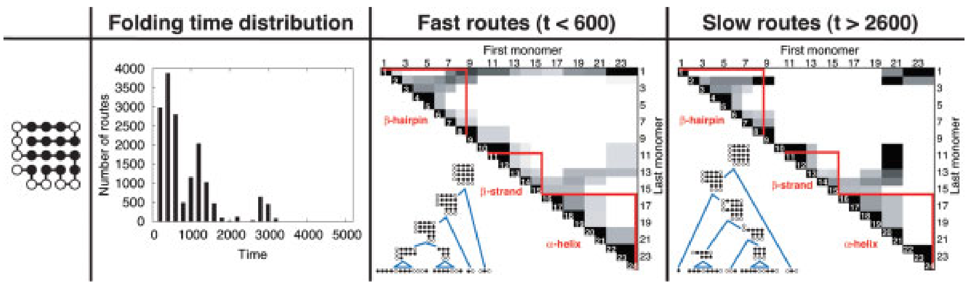

Fig. 8. The distribution of folding route times has different modes. Routes that require similar amounts of search are more similar to each other than routes that require far more or far less amounts of search.

Route-specific Search Times

In order to predict folding rates, we also have to be able to quantify the amount of entropic search required to find the native state. CKY gives us a way to calculate "computational" search times t for each individual route (Appendix B). These times are based on the amount of search required to combine pairs of sister nodes. They take into account the fact that different branches of the same tree represent simultaneous folding events. Each node N in a folding route tree corresponds to a set of $|N|$ conformations. The entropy of a node, $S_N = \log(|N|)$, is high if its chain segment is long and the number of contacts in its contact map label is small. Conversely, the node entropy is low if either the chain segment is short or if the number of contacts is high. If two individual conformations can be combined in up to K different ways, and it takes one time step to explore one such combination, it takes $\Delta t \approx K|L||R| = O(S_L + S_R)$ time steps to combine two nodes L and R,[§§] and the parent node P is formed at $t_P = \max(t_L, t_R) + \Delta t$.

For every sequence, we obtain a distribution of folding times (Fig. 8). In many cases, this distribution has multiple modes. We can also visualize folding routes with similar folding times, and observe that routes with similar folding times tend to be more similar to each other than to routes with very different folding times. In Figure 8, all the slow routes (with a folding time of 2600 time steps or more) go through the same rate-limiting step where the chain segments 2-9 (most of the initial β-hairpin) and 10-21 (the third β-strand and some of the α-helix) are assembled, whereas in most of the fast routes, the third β-strand as well as the α-helix grow onto the hairpin, and the segment 10-21 is never formed. This is because the amount of search required to combine two nodes L and R (e.g. the native nodes spanning 2-9 and 10-21) depends only on the node entropy of L and R, not on how L and R are formed, or what happens in subsequent steps.

A Simple Model of Overall Folding Rates

We can use the mean of the route-specific search time distribution, $\langle t \rangle$, and the accessibility of the native state, A, to define a simple model of overall folding rates $k = A/\langle t \rangle$. This model ignores many details, but it captures two of the main factors that determine folding speed—the amount of entropic search and the prevalence of energetic barriers. Our analysis shows that, despite its simplicity, this model predicts correctly that the folding rates depend on the topology of the folded structure, and not on the composition of the primary sequence.

The impact of sequence composition

First, we examine whether our predicted folding rates depend on the sequence composition, as measured by the number of H residues in the chain. Figure 9 shows that H-content is strongly negatively correlated with accessibility (with a correlation coefficient of $r = -0.67$) and almost equally strongly positively correlated ($r = 0.63$) with per-route search rate. That is, an increase in H-content decreases native accessibility A, and thus increases the likelihood of misfolding. At the same time, it decreases the search entropy along the native routes, resulting in higher per-route folding rates $1/\langle t \rangle$. These two effects cancel each other out, and the overall rate k is therefore independent ($r = 0.08$) of H-content. This is consistent with experimental findings, which have failed to reveal a correlation between sequence composition and folding rates in real proteins. Our model suggests that in sequences with few hydrophobic residues, slow folding may be due to increased entropic search, whereas in sequences with many hydrophobic residues, it may be caused by energetic barriers.

The impact of native topology

A remarkable observation by Plaxco et al.[3] is that the logarithm of the folding rates of two-state proteins correlates strongly ($r = -0.81$) with a topological property of the protein's native structure, the native contact order CO. This quantity measures the average distance $|j - i|$ (in the primary sequence) between all pairs of residues i and j that form a contact (i, j) in the native state, including contacts which make little or no contribution to the energy (HP or PP contacts in the HP model). Proteins with mostly local native contacts (e.g. α-helical proteins) have a low CO, and fold faster than proteins with more non-local contacts (e.g. β-sheet proteins), which have a high CO. We find that our simple model of folding rates predicts a qualitatively similar trend.

Figure 10 shows that our model also predicts a significant correlation between native contact order and overall

[§§]The actual factor K depends on the geometry of the individual conformations.

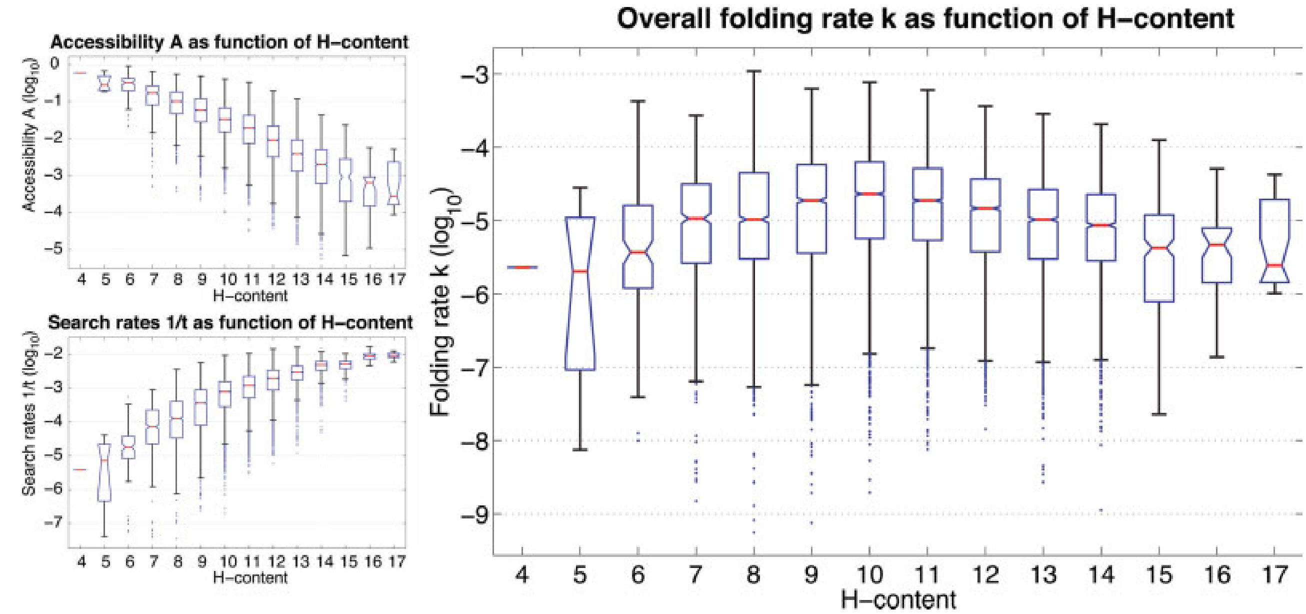

Fig. 9. The overall folding rate k is independent of the hydrophobic content of the HP sequence. The main graph shows the distribution of log(k) for all 20mers (sorted by H-content), as a box plot. The boxes indicate the interquartile range IQR: Q3–Q1 around the median (red). Data points beyond 1.5áIQR are shown as outliers. The graphs on the left show the corresponding distributions of log(A) and log(1/t). [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

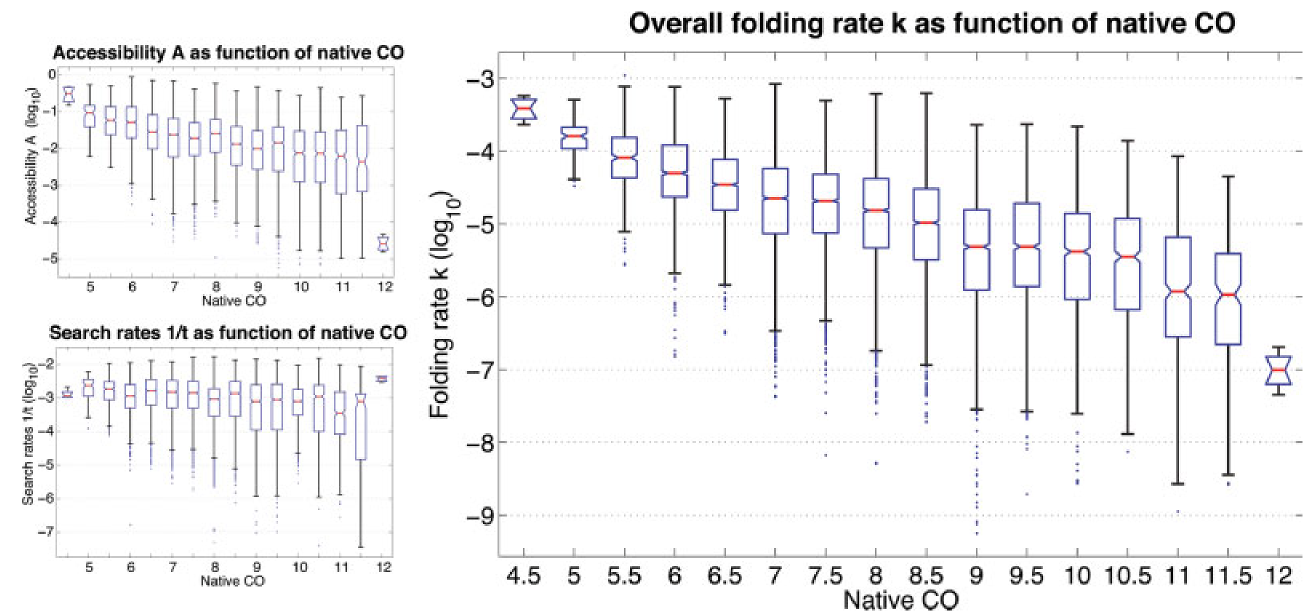

Fig. 10. The overall folding rate k decreases exponentially as the native contact order increases. The main graph shows the distribution of $\log_{10}(k)$ for all 20mers (sorted by native CO), as a box plot. The boxes indicate the interquartile range IQR: Q3–Q1 around the median (red). Data points beyond 1.5·IQR are shown as outliers. The graphs on the left show the corresponding distributions of log(A) and log(1/$\langle t \rangle$). [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

folding rates k ($r = -0.49$ at a P-value of $P = 0.0000$), even though k has only a weak negative correlation with accessiblity A ($r = -0.32$) and with per-route folding rates $1/\langle t \rangle$ ($r = (0.19)$. However, these two effects combine to give a much stronger overall correlation. In particular, all sequences with a low native CO have a high native accessibility A, corresponding to few energetic barriers and thus a low probability of misfolding. They also have very fast per-route folding rates $1/\langle t \rangle$, that is, low search entropy. Sequences with a high native contact order, however, may have a high probability of misfolding (low A, or many barriers), and may also require a lot of search.

Figure 11 (left), shows that all sequences with low native accessibility have high per-route folding rates, whereas sequences with high native accessibility have a wide range of folding rates. Figure 11 (right), indicates

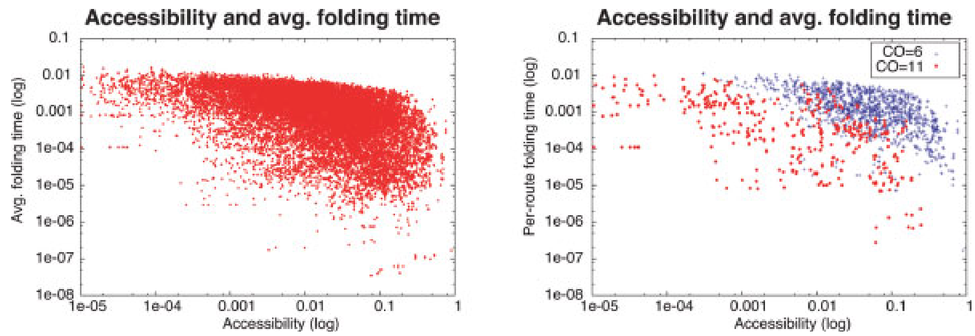

Fig. 11. Dependence of relation between accessibility and folding time on native CO (loglog plot). On the right, blue corresponds to sequences with a CO of 6, and red to a CO of 11.

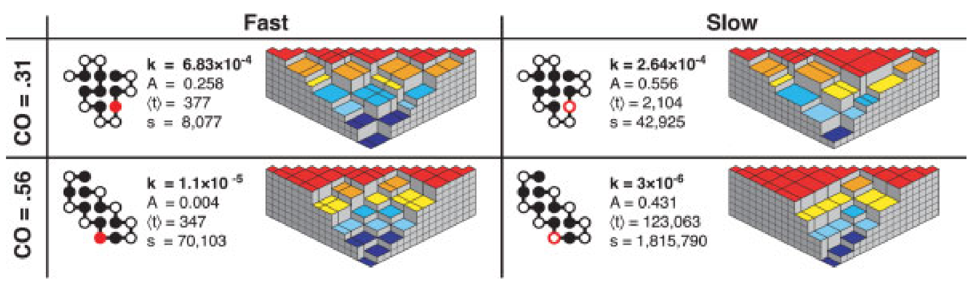

Fig. 12. The effect of single-point mutations.

that the relation between accessibility and per-route search rates depends on the native contact order. All sequences have a similar range of entropic search times, but sequences with low contact order (in blue) have typically lower probabilities of misfolding (i.e. higher accessibility). Low contact order sequences also tend to have lower search entropy than high contact order sequences with the same accessibility.

Single-point mutations can dramatically change folding rates

Despite the good correlation of folding rates with average CO shown by Plaxco et al., there are also situations in which two different proteins with the same chain fold can have very different folding kinetics,[42] sometimes with variances of more than an order of magnitude. Our CKY method gives a similar result, allowing us to interpret that variation. Figure 12 shows two different cases in which a protein sequence has only a single H to P mutation, and for which the wild-type and mutant sequences have exactly the same native structure. Yet, CKY finds the native state 5 times faster in one case and 25 times slower in the other case. The mutation causes a change in the folding rate by a factor of 2.59 in the low CO case, and by a factor of 3.67 in the high CO case. In both cases, the additional H monomer decreases accessibility, and decreases folding times.

Since the native structures are identical in both pairs, the CO cannot explain these differences. However, the explanation is clear from the shapes of the chart landscapes: in both cases, the slower-folder landscape is more golf course-like, and the faster-folder landscape is broader and more funnel-like.

In the fast folding, low-CO sequence, all possible $(i, i + 3)$ HH contacts are native. Thus misfolding is likely to occur only during later stages of folding. In the other low-CO sequence, there are fewer possible non-native HH contacts, but because the overall search entropy is higher, we predict a slower folding rate. The fast folding sequence among the low-CO proteins has a more funnel-like landscape than the slower folding sequence, but its native state has very low accessibility, since a single nucleating contact is required for this structure to zip up. In the slower folding high-CO sequence, misfolding is far less likely to occur, but the native routes are on average 354 times slower.

Kinetically Inaccessible Native States

Not all native states can be found by our greedy, hierarchical search. As seen above, kinetic accessibility A generally decreases as the hydrophobic content of the sequence increases. But sometimes a hydrophobic residue is necessary to make the native state accessible. Figure 13 shows a sequence for which CKY cannot find the native state if monomer 18 is changed from H to P. In most structures for which we cannot find the native state (and also in the example given here), one of the ends is buried. At the chain lengths we are currently dealing with, this typically requires the other end to form a single strand that wraps around the rest of the chain. Therefore, the native structure has to "grow" from the other end. This growth step can only succeed if there are sufficiently many HH contacts between this strand and the buried part.

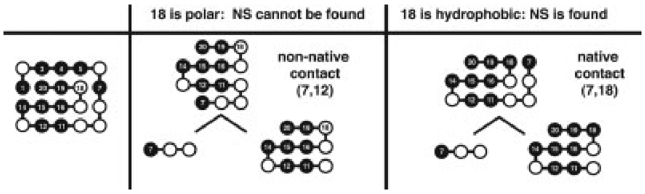

Fig. 13. Some native states are not accessible by our greedy, hierarchical search.

CONCLUSIONS

Based on the assumption that protein folding is guided by a hierarchical search for locally optimal structures, we propose that folding routes have an underlying tree structure and that the folding process can be viewed as a hierarchical, greedy search. Similar proposals have been made, for example, by Crippen,[4] Rose and Lesk,[5,6] and Tsai and Nussinov.[24] This allows us to apply CKY, a dynamic programming technique that is normally used to identify the grammatical structure of natural language sentences, to protein folding. By showing that CKY can predict the native states of sequences in the two-dimensional HP lattice model efficiently and accurately, we have provided evidence that our proposed hierarchical search strategy is a viable model of protein folding.

The main search methods of computational biology—Monte Carlo and molecular dynamics—are serial, and follow only a single trajectory at a time. We believe the CKY method is more physical in capturing the parallel and ensemble nature of the true folding process. CKY returns all the direct folding routes to the native state. This allows us to make predictions about macroscopic folding behavior that are based on the underlying microscopic processes. We find that distinct macroscopic pathways emerge, even though every sequence has multiple microscopic routes. We also predict that folding rates are different for different amino acid sequences, consistent with experiments showing that different proteins fold at different rates. In particular, we have shown that our estimated folding rates predict and provide an explanation for the experimental observation of Plaxco et al. that the topology of the folded structure determines the speed with which proteins fold. We have illustrated the principle of CKY here using a simplified protein model, but similar algorithms should also be applicable to other types of lattice models and to off-lattice models. The main ideas here—(1) the early stages involve many independent extensive searches of small parts of the chain having only a few degrees of freedom, (2) the best of which are then chosen and retained, (3) followed in later stages by bringing those parts together—are readily implemented in a wide range of types of models. If the full set of 20 amino-acid types (and a corresponding, more complex, energy function) are used, it may be better to use a beam search strategy which retains not only the lowest-energy conformations in each cell, but all those conformations whose energy lies within a certain threshold of the lowest energy found for the cell. Hence this method may be useful both for protein structure prediction and also for a better understanding of protein folding physics.

CKY has originally been developed for linguistic, not for biological, purposes. Its applicability to protein folding may therefore seem surprising, although connections between formal grammar theory and the structure of proteins and RNAs have been made before (e.g., the pioneering work by Searls[35]). Our work is slightly different from this line of research, in that it does not try to capture biological structures with a grammar, but instead aims to apply algorithmic ideas that were originally developed to analyze language to model the folding process. Any given sentence has a large number of possible interpretations, just as any amino acid sequence has an astronomical number of possible spatial conformations. Parsing a sentence is an exercise in global optimization, the goal of which is to find the one correct meaning of the sentence. Protein folding is also an exercise in global optimization, the goal of which is to find the one native chain fold. 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. In this way, early decisions are small, local, parallel, and independent, while later decisions are larger, less local, more serial, and dependent.

ACKNOWLEDGMENTS

We thank Vincent Voelz, Banu Ozkan, John Chodera, and David Chiang for helpful discussions.

REFERENCES

1. Levinthal C. Are there pathways for protein folding? J Chim Phys 1968;65:44,45.
2. Dill KA. Theory for the folding and stability of globular proteins. Biochemistry 1985;24:1501–1509.
3. Plaxco KW, Simons KT, Baker D. Contact order, transition state placement and the refolding rates of single domain proteins. J Mol Biol 1998;277:985–994.
4. Crippen GM. The tree structural organization of proteins. J Mol Biol 1978;126:315–332.
5. Rose GD. Hierarchic organization of domains in globular proteins. J Mol Biol 1979;134:447–470.
6. Lesk AM, Rose GD. Folding units in globular proteins. Proc Natl Acad Sci USA 1981;78:4304–4308.
7. Maity H, Maity M, Krishna M, Mayne L, Englander SW. Protein folding: the stepwise assembly of foldon units. Proc Natl Acad Sci USA 2005;102:4741–4746.
8. Dill KA, Fiebig KM, Chan HS. Cooperativity in protein folding kinetics. Proc Natl Acad Sci USA 1993;90:1942–1946.
9. Srinivasan R, Rose GD. LINUS: a hierarchic procedure to predict the fold of a protein. Proteins 1995;22:81–99.
10. Kasami T. An efficient recognition and syntax algorithm for context-free languages. Scientific report AFCRL-65-758, Air Force Cambridge Research Laboratory, Bedford, MA, 1965.
11. Younger DH. Recognition and parsing of context-free languages in time $O(n^3)$. Inf Contr 1967;10:189–208.
12. Lau K, Dill K. A lattice statistical mechanics model of the conformational and sequence s of proteins. Macromolecules 1989;22: 638–642.

13. Dill KA, Bromberg S, Yue K, Fiebig KM, Yee DP, Thomas PD, Chan HS. Principles of protein folding—a perspective from simple exact models. Protein Sci 1995;4:561–602.
14. Baldwin RL, Rose GD. Is protein folding hierarchic? I. Local structure and peptide folding. Trends Biochem Sci 1999;24: 26–33.
15. Baldwin RL, Rose GD. Is protein folding hierarchic? II. Folding intermediates and transition states. Trends Biochem Sci 1999; 24:77–83.
16. Karplus M, Weaver DL. Protein folding dynamics. Nature 1976; 260:404–406.
17. Kim PS, Baldwin RL. Specific intermediates in the folding reactions of small proteins and the mechanism of protein folding. Annu Rev Biochem 1982;51:459–489.
18. Fersht AR. Nucleation mechanisms in protein folding. Curr Opin Struct Biol 1997;7:3–9.
19. Weikl TR, Dill KA. Folding rates and low-entropy-loss routes of two-state proteins. J Mol Biol 2003;329:585–598.
20. White GW, Gianni S, Grossmann JG, Jemth P, Fersht AR, Daggett V. Simulation and experiment conspire to reveal cryptic intermediates and a slide from the nucleation-condensation to framework mechanism of folding. J Mol Biol 2005;350:757–775.
21. Krishna MMG, Englander SW. The n-terminal to c-terminal motif in protein folding and function. Proc Natl Acad Sci USA 2005;102:1053–1058.
22. Weinkam P, Zong C, Wolynes PG. A funneled energy landscape for cytochrome c directly predicts the sequential folding route inferred from hydrogen exchange experiments. Proc Natl Acad Sci USA 2005;102:12401–12406.
23. Przytycka T, Srinivasan R, Rose GD. Recursive domains in proteins. Protein Sci 2001;11:409–417.
24. Tsai CJ, Maizel JV, Jr, Nussinov R. Anatomy of protein structures: visualizing how a one-dimensional protein chain folds into a three-dimensional shape. Proc Natl Acad Sci USA 2000;97: 12038–12043.
25. Fiebig KM, Dill KA. Protein core assembly processes. J Chem Phys 1993;98:3475–3487.
26. Weikl TR, Dill KA. Folding kinetics of two-state proteins: effect of circulization, permutation and crosslinks. J Mol Biol 2003;332: 953–963.
27. Weikl TR, Palassini M, Dill KA. Cooperativity in 2-state protein folding kinetics. Protein Sci 2004;13:822–829.
28. Merlo C, Dill KA, Weikl TR. φ values in protein-folding kinetics have energetic and structural components. Proc Natl Acad Sci USA 2005;102:10171–10175.
29. Chomsky N. Syntactic structures. The Hague: Mouton; 1957.
30. Collins M. Head-driven statistical models for natural language parsing. PhD Thesis, University of Pennsylvania, 1999.
31. Charniak E. A maximum-entropy-inspired parser. In: Proceedings of the first meeting of the North American chapter of the Association for Computational Linguistics, Seattle, WA, 2000. pp 132–139.
32. Irbäck A, Troein C. Enumerating designing sequences in the HP model. J Biol Phys 2002;28:1–15.
33. Crescenzi P, Goldman D, Papadimitriou CH, Piccolboni A, Yannakakis M. On the complexity of protein folding. J Comput Biol 1998;5:423–466.
34. Berger B, Leighton FT. Protein folding in the hydrophobic-hydrophilic(HP) model is NP-complete. J Comput Biol 1998;5:27–40.
35. Searls DB. The language of genes. Nature 2002;420:211–217.
36. Durbin R, Eddy SR, Krogh A, Mitchison G. Biological sequence analysis. Cambridge: Cambridge University Press; 1998.
37. Booth TL, Thompson RA. Applying probability measures to abstract languages. IEEE Trans Comput 1973;22:442–449.
38. Jurafsky D, Martin JH. Speech and language processing. Upper Saddle River, NJ: Prentice-Hall; 2000.
39. Manning CD, Schütze H. Foundations of statistical natural language processing. Cambridge: MIT Press; 1999.
40. Wallin S, Chan HS. Conformational entropic barriers in topology-dependent protein folding: perspectives from a simple native-centric polymer model. J Phys: Condens Matter 2006;18:S307–S328.
41. Steedman M. The syntactic process. Cambridge, MA: MIT Press; 2000.
42. Myers JK, Oas TG. Mechanisms of fast protein folding. Annu Rev Biochem 2002;71:783–815.

APPENDIX A: OBTAINING THE TREES—UNPACKING THE CHART

In order to obtain the actual folding route trees from CKY's compact representation, the chart has to be "unpacked," since each item X in the chart represents a set of trees T_X with the same root label X. Although the size of this set is exponential, many interesting properties can be computed in polynomial time. We first illustrate this by showing how to calculate the number of folding routes that lead from the substring $i...j$ to an item X which spans $i...j$, $I_X = |T_X|$. We will refer to this quantity as the number of *inside routes* that lead to X. We are also interested in the number of complete routes that go through X and lead to the native state. This requires the introduction of an additional quantity, O_X, the *outside* *routes*, or the number of trees that lead to the native state from an item X (see Fig. A1 for an illustration).

Inside Routes

Here we define the quantity I_X, the number of native routes that lead from the substring $i...j$ spanned by X to X, or the number of routes "inside" X.

Items X in the initial cells *chart*[i][i] represent one tree, consisting of one (leaf) node. Therefore, $I_X = 1$. The number of trees represented by items X in cells *chart*[i][j] ($i \neq j$) can be obtained recursively from the pairs of backpointers $\langle L,R\rangle$ to items L in *chart*[i][k] and R in *chart*[$k+1$][j]. If L corresponds to I_L trees and R corresponds to I_R trees, the pair of backpointers $\langle L,R\rangle$ from X to L and R represents a set $T_{\langle L,R\rangle}$ of $I_L I_R$ trees, since every pair of trees in T_L and T_R can form a distinct new tree with root X, left child L and right child R. Each pair $\langle L,R\rangle$ represents a distinct set of trees, thus $I_X = \Sigma_{\langle L,R\rangle} I_L I_R$.

X has at most *j-i pairs* $\langle L,R\rangle$. For each pair of cells *chart*[i][k] and *chart*[$k+1$][j], there can be at most one pair of items L and R from which X can be formed, because no contact in L or R can be broken and our pruning strategy ensures that only conformations with the highest number of contacts survive in each cell. Therefore, the number of native routes in an $n \times n$ chart with $O(n^2)$ cells can be calculated in $O(n^3)$ time.

APPENDIX B: CALCULATING ROUTE-SPECIFIC FOLDING TIMES

Folding Times as Search Times

We can calculate the search times along specific folding routes. A node $\mathbf{P}$ with two children $\mathbf{L}$ and $\mathbf{R}$ can only be formed after both $\mathbf{L}$ and $\mathbf{R}$ are formed. Therefore, if it takes $\Delta t_{\langle L,R\rangle}$ time steps to combine conformations $\mathbf{L}$ and $\mathbf{R}$ then $t_{\mathbf{P}}$, the time at which $\mathbf{P}$ is formed, is given by $t\mathbf{P} = max(t_{\mathbf{L}},t_{\mathbf{R}}) + \Delta t_{\langle \mathbf{L},\mathbf{R}\rangle}$. Furthermore, all nodes $\mathbf{L}$ and $\mathbf{R}$ that are associated with the same items L and R correspond to the same set of conformations; therefore, $\Delta t_{\langle \mathbf{L},\mathbf{R}\rangle} = \Delta t_{\langle \mathrm{L},\mathrm{R}\rangle}$ for all trees represented by a pair of backpointers $\langle L,R\rangle$ in any item I. Assuming that it takes one time step to try one specific combination of two conformations, $\Delta t_{\langle L,R\rangle}$ equals

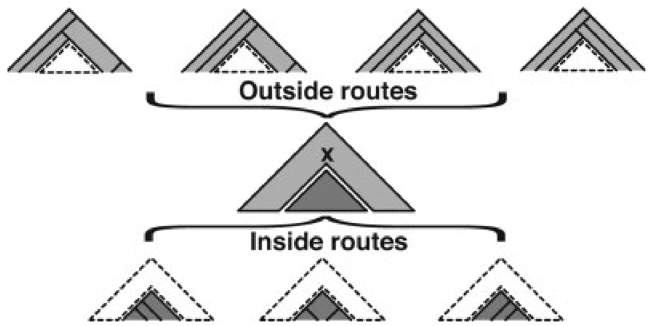

Fig. A1. Inside and outside folding route trees.

the number of conformations that are searched when the conformations in L and R are combined.

Computing Folding Time Distributions

The folding times distribution of the set of trees represented by an item I can be stored in two integer arrays *time* and *freq*, where *freq*[i] counts the number of trees with folding time *time*[i] (Fig. B1). Assuming that *time*[i] is ordered, the folding times distribution for a pair of backpointers $\langle L,R\rangle$ can be calculated in $O(L.freq.length() + R.freq.length())$ time.

Computing Folding Routes in an Interval $[t_{\min}, t_{\max}]$

Each pair of backpointers $\langle L,R\rangle$ from an item I represents a set of trees that take $\Delta t_{\langle L,R\rangle}$ time steps to form from L and R, but L and R themselves represent sets of trees with different folding times. In order to obtain the number of routes $O_I^{[t_{\min},t_{\max}]}$ that go through an item I and reach the native state within the interval $[t_{\min},t_{\max}]$, we need to calculate $I_I^{[t_1,t_2]}$, the number of inside routes that reach I within some interval $[t_1, t_2]$, and $O^{[t_1,t_2]}$, the number of corresponding outside routes of I that reach the native state N in the desired interval $[t_{\min},t_{\max}]$ if $t_1 \le t_I \le t_2$ (Fig. B2).

The trees represented by $\langle L,R\rangle$ lie within $[t_1 \ldots t_2]$ if the slower of L and R is formed within the interval $[t'_1 \ldots t'_2]$, where $t'_1 = t_1 - \Delta\, t_{\langle L,R\rangle}$ and $t'_2 = t_2 - \Delta t_{\langle L,R\rangle}$. This is the case if either $0 \le t_L \le t_1$ and $t'_1 \le t_R \le t'_2$ or if $t'_1 \le t_L \le t'_2$ and $0 \le t_R \le t'_2$. Therefore, we have to recurse twice on each the items L and R. In both cases, we need to know the number of outside routes, $O_L^{[0,t'_2]}$ and $O_R^{[t'_1,t'_2]}$ in the first case, and $O_L^{[t'_1,t'_2]}$ and $O_R^{[0,\,t'_2]}$ in the second case.

If $O^{[t_1,t_2]_I}$ is the number of I's outside routes that reach the native state within the desired interval $[t_{\min} \ldots t_{\max}]$ if the corresponding inside routes reach I within $[t_1 \ldots t_2]$, the outside routes $O_L([0 \ldots t'_2])$ are given by $O_I([t_1 \ldots t_2])\, I_R([t'_1 \ldots t'_2])$.

APPENDIX C: CALCULATING THE ACCESSIBILITY OF THE NATIVE STATE

CKY returns all native routes τ_i. Each folding route is a labeled binary tree with n leaf nodes. We do not know the number of possible labeled binary trees with n leaf nodes; but the number of unlabeled binary trees with n leaf nodes is given by the $(n-1)$th Catalan number C_{n-1}, which is defined as follows:

$$C_n = (2n)!/((n+1)!n!)$$

Since no contacts can be broken between a child node and its parent, each native route corresponds to a different

```
getFoldingTimes(I, L, R)
  i, j, k := 0;
  leftTotalFreq := sumFreq(L.freq);
  rightTotalFreq := sumFreq(R.freq);
  while i ≤ L.freq.size & j ≤ R.freq.size
    if L.time[i] ≥ R.time[j]
      leftFreq := L.freq[i];
      leftTotalFreq := leftTotalFreq − leftFreq
      rightFreq := rightTotalFreq;
      eTime := L.time[i] + Δt⟨L,R⟩;
      i := i + 1;
    else
      rightFreq := R.freq[j];
      rightTotalFreq := rightTotalFreq − rightFreq;
      leftFreq := leftTotalFreq;
      itemTime := R.time[j] + Δt⟨L,R⟩;
      j := j + 1;
    eFreq := leftFreq × rightFreq;
    E.freq[k] := eFreq;
    E.time[k] := eTime;
    k := k + 1;

sumFreq(freq)
  f := 0;
  for i := 0..freq.length
    f := f + freq[i];
  return f;
```

Fig. B1. Calculating folding time distributions.

```
countNativeFoldingRoutesInterval(t_min, t_max)
  incrNativeInterval(N, 1, t_min, t_max).

incrNativeInterval(I, O'_I, t_1, t_2)
  1. Increment the routes that go through I
  Ω_I^[t_min,t_max] = Ω_I^[t_min,t_max] + I_I([t_1...t_2]) × O'_I;
  2. Recurse on I's children
  for (i = 0; i ¡ I.children.size(); i++)
    ⟨L,R⟩ = I.getChild(i);
    incrNativeIntervalChild(⟨L,R⟩, O'_I, t_1, t_2);

incrNativeIntervalChild(⟨L,R⟩, O'_I, t_1, t_2)
  t'_1 = t_1 − Δt⟨L,R⟩
  t'_2 = t_2 − Δt⟨L,R⟩
  O'_R = O'_I I_L^[0,t'_1−1]
  O'_L = O'_I I_R^[t'_1,t'_2]
  incrNativeInterval(R, O'_R, t'_1, t'_2);
  incrNativeInterval(L, O'_L, 0, t'_2);
  O''_L = O'_I I_R^[0,t'_1−1]
  O''_R = O'_I I_L^[t'_1,t'_2]
  incrNativeInterval(L, O''_L, t'_1, t'_2);
  incrNativeInterval(R, O''_R, 0, t'_2);
```

Fig. B2. Computing folding time intervals.

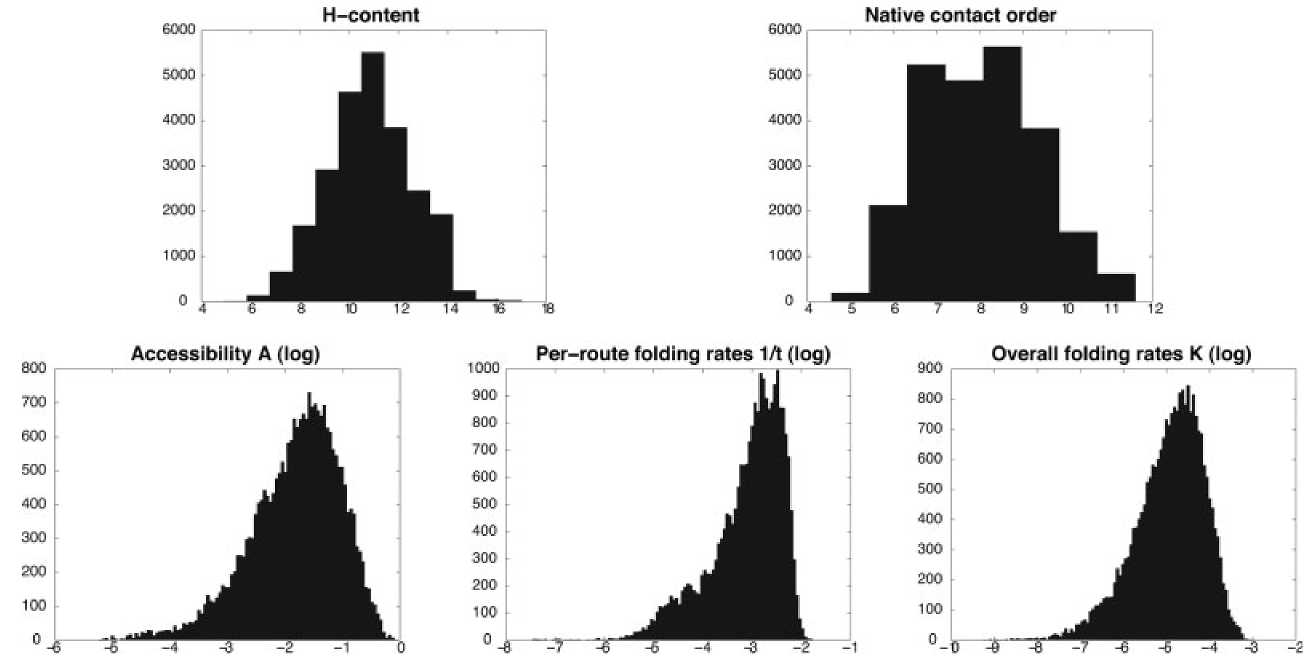

Fig. D1. Distributions of H-content and native contact order (top) and accessibility, per-route folding rates and overall folding rates (bottom).

unlabeled tree. If U_N is the set of unlabeled trees that the native routes correspond to, κ is equal to the probability that one unlabeled tree υ_{i} out of U_N is chosen and that subsequently this tree is labeled with the labels of the corresponding native route:

$$\kappa = \sum_i P(\mathrm{label}(\upsilon_i) = \tau_{\mathrm{i}} | \upsilon_i)/C_{n-1}$$

Every node $\mathbf{U}_j$ in υ_i has to be labeled with the label of the corresponding node $\mathbf{N}_j$ in τ_i. We assume that labels are chosen iteratively, in a bottom-up fashion, that is, starting with the leaf nodes. The leaf nodes of all trees have the same label, the empty contact map, therefore,

$$P(\mathrm{label}(\mathbf{U}_i) = \mathrm{label}(\mathbf{N}_i)) = 1 \quad \text{if } U_i \text{ is a leaf node}$$

An internal node $\mathbf{U}_{\mathrm{j}}$ can only be labeled like $\mathbf{N}_{\mathrm{j}}$ if its children have been assigned the native labels. The probability that two labeled nodes **L** and **R** are combined into one labeled parent node **N** corresponds to the probability that the substrings represented by **L** and **R** form the contacts represented by the label of **N**, given that they have already formed the contacts specified by the labels **L** and **R**, or, equivalently, it is the probability of the macrostate defined by **N**'s contact map, given the macrostate defined by the unified contact map of **L** and **R**. Our simple measure takes size, but not energy differences between macrostates into account. Therefore, if there are c_i conformations represented by the ith contact map label $\mathbf{N}_i$ that can be obtained by combining **L** and **R**, we define the probability that $\mathbf{U}_i$ is labeled with $\mathbf{N}_i$'s label as follows:

$$P(\mathrm{label}(\mathbf{U}_i) = \mathbf{N}_i | \mathbf{L}, \mathbf{R}) = c_{\mathrm{i}} / \sum_{j=1}^{n} c_j$$

We assume that only locally optimal contact maps $\mathbf{N}_i$ (corresponding to items which survive the pruning in their cell) can be chosen. The number of conformations that they correspond to is known, therefore, all that is required to compute this probability is to keep track of which contact maps $\mathbf{N}_i$ can be obtained from **L** and **R**.

The probability that υ_i is assigned τ_i's labels is

$$P(\mathrm{label}(\upsilon_i) = \tau_i | \upsilon_i) = \prod_{j:U_j \text{ is leaf}} P(\mathrm{label}(\mathbf{U}_j) = \mathbf{N}_j) \times \prod_{j:\mathbf{U}_j \text{ is parent of } \mathbf{L},\mathbf{R}} P(\mathrm{label}(\mathbf{U}_j) = \mathbf{N}_j | \mathbf{L}, \mathbf{R})$$

APPENDIX D: AN ANALYSIS OF OUR DATA SET

Figure D1 shows the distribution of H content, native contact order, accessibility, per-route rates and overall folding rates for all unique-folding 20mers for which CKY (with ICO = 11) finds a unique structure.