The Evolutionary Capacity of Protein Structures

Leonid Meyerguz[*]
Dept. of Computer Science
Cornell University, Ithaca NY
leonidm@cs.cornell.edu

David Kempe[†]
Dept. of Computer Science
University of Washington,
Seattle WA
kempe@cs.washington.edu

Jon Kleinberg[‡]
Dept. of Computer Science
Cornell University, Ithaca NY
kleinber@cs.cornell.edu

Ron Elber[§]
Dept. of Computer Science
Cornell University, Ithaca NY
kleinber@cs.cornell.edu

ABSTRACT

In nature, one finds large collections of different protein sequences exhibiting roughly the same three-dimensional structure, and this observation underpins the study of structural protein families. In studying such families at a global level, a natural question to ask is how close to "optimal" the native sequences are in terms of their energy. We therefore define and compute the *evolutionary capacity* of a protein structure as the total number of sequences whose energy in the structure is below that of the native sequence. An important aspect of our definition is that we consider the space of *all* possible protein sequences, i.e. the exponentially large set of all strings over the 20-letter amino acid alphabet, rather than just the set of sequences found in nature.

In order to make our approach computationally feasible, we develop randomized algorithms that perform approximate enumeration in sequence space with provable performance guarantees. We draw on the area of rapidly mixing Markov chains, by exhibiting a connection between the evolutionary capacity of proteins and the number of feasible solutions to the Knapsack problem. This connection allows us to design an algorithm for approximating the evolutionary capacity, extending a recent result of Morris and Sinclair on the Knapsack problem. We present computational experiments that show the method to be effective in practice on large collections of protein structures. In addition, we show how to use approximations to the evolutionary capacity to compute a statistical mechanics notion of "evolutionary temperature" on sequence space.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Non-numerical Algorithms and Problems; J.3 [**Life and Medical Sciences**]: Biology and genetics

General Terms

Algorithms

Keywords

protein structure, evolutionary networks, approximate counting, rapidly mixing Markov chains

1. INTRODUCTION

One of the more striking recent observations in the field of structural biology is the profound redundancy in the sequence-to-structure map for proteins: while it remains a general belief that the sequence of amino acids in a protein uniquely determines the protein's three-dimensional shape under physiological conditions, the converse (a given structure implying a unique sequence) is far from true. Rather, a very limited number (about 500) of fundamentally different families of protein structures have been found experimentally, and large numbers of different protein sequences, even with little sequence similarity, can adopt essentially the same three-dimensional structure [17, 20].

This redundancy reflects the way in which molecular evolution has "explored" the space of sequences for a given structure. How thorough has this process of exploration been? Is molecular evolution "incomplete" — has it only been able to sample from a small portion of the space of feasible sequences? At present, we have very little understanding of these types of questions, in large part because of the scale at which such questions must be addressed. To reason about the

[*]Supported by a GAANN Fellowship.

[†]Supported by an NSF Postdoctoral Fellowship.

[‡]Supported by a David and Lucile Packard Foundation Fellowship.

[§]Supported by an NSF grant.

RECOMB'04, March 27–31, 2004, San Diego, California, USA.

extent to which evolution has filled out the range of feasible proteins, we need techniques for working with the space of all possible protein sequences — not simply the set of all sequences observed in nature, but the vastly larger set of all n-letter strings over the 20-symbol amino acid alphabet, for n equal to the lengths of typical proteins.

In this paper, we develop techniques for analyzing sequence space at a global level, using physically realistic protein energy functions, and we report on findings about the organization of this space relative to the collection of known protein folds. Our methods utilize ideas from statistical mechanics, treating sequence space as a large ensemble of varying energies; we combine these ideas with algorithmic techniques from the area of approximate counting.

Underlying our analysis is the notion of a sequence–structure *fitness function*, which defines the energy of each sequence in a fixed structure σ. Letting $N(E)$ denote the number of amino acids sequences with energy at most E in the structure σ then defines a characteristic *sequence distribution function* $N(\cdot)$, which in turn lets us determine the following two fundamental quantities:

Evolutionary Capacity. Consider the native sequence X_σ for the structure σ, and define the *native energy* E_σ to be the energy of X_σ in the structure σ. The quantity $N(E_\sigma)$ is then simply the number of sequences that would have energy in σ no greater than that of the actual native sequence. We refer to $N(E_\sigma)$ as the *evolutionary capacity* of the structure σ, because it reflects how far the current state of molecular evolution on σ is from the "energetic optimum."

The Temperature of Evolution. There is more to the picture than just variation in energy: while individual sequences of extremely low energy may be more favorable, sequences of higher energy are much more numerous, and this trade-off is a qualitative reflection of a balance between energy and entropy in sequence space. Given the distribution function N, one can use methods from statistical physics to capture this trade-off quantitatively via an evolutionary analogue of "temperature," and thereby obtain insight into the probability of finding sequences of different energy values.

This analysis rests on evaluating the distribution function $N(\cdot)$, whose definition requires implicit enumeration of the full space of n-letter strings of amino acids. Finding ways of approximating N is thus a fundamental first step toward deeper analysis of the space.

The present work. Our first main result is a fully-polynomial, randomized approximation scheme for the sequence distribution function N, where the sequence-structure energy is computed according to an arbitrary *local fitness function* that uses only single-site terms. Such fitness functions, including the THOM2 function [14] that we focus on for our experiments here, are widely used in practice because they can be evaluated very quickly on a per-sequence basis.

More specifically, given a structure of length n, a target energy E, and an error parameter ε, we provide an algorithm that approximates $N(E)$ with high probability to within a factor of $1+\varepsilon$, in time that is polynomial in n and ε^{-1}. The problem, and our algorithm, are closely related to the problem of approximately counting the number of feasible solutions to a Knapsack problem — this latter problem was a well-known open question resolved only recently by Morris and Sinclair [16]. Evaluating N is a more general problem than counting Knapsack solutions, however, and the main theoretical contribution of our work is an extension of the Morris-Sinclair theorem to this more general setting. We then discuss computational experiments in which we examine the function N associated with different proteins; we use a large dataset of roughly 3400 protein structures that represent the folds in the Protein Data Bank [1], and we consider how N varies across different lengths and characteristic shapes.

We go on to approximately determine an evolutionary temperature, and associated sequence probabilities, for each protein in the collection. Our computational experiments here reveal the surprising finding that the distribution of temperatures, over all proteins of length greater than 200 amino acids, is very sharply concentrated. We suggest qualitative conclusions that one can draw from a roughly constant temperature across this collection of proteins, concerning the potential existence of a universal selection mechanism in sequence space.

We also consider the corresponding problems with respect to *pairwise fitness functions*, where the energy of a sequence in a given structure depends on pairwise interactions among residues that are physically close. While such fitness functions are more expressive, they are also less tractable computationally; in particular, the approximate evaluation of the distribution function N becomes provably intractable with general pairwise interactions. We develop computational heuristics that appear to be effective in practice for the specific pairwise fitness function TE13 [24]; the results are qualitatively similar to the case of the local fitness function THOM2. The fact that two very different fitness functions yield such similar results suggests that our results are not strongly dependent on particular modeling assumptions.

It should be noted that the fitness of sequences relative to a structure σ involves two qualitative issues: a sequence should be energetically favorable with respect to σ, *and* at the same time not be even more favorable in a competing structure $\sigma' \neq \sigma$. (These are sometimes referred to as *positive design* and *negative design*, respectively.) We make the positive design aspect explicit, and deal with the negative design aspect only implicitly, by using fitness functions which have been trained to favor sequences in their native structures. (In other words, explicit negative design was used in the construction of the fitness functions.) This is in keeping with the style of analysis adopted in all the related work discussed below, and hence allows for comparison with this related work. Addressing the negative design aspect more explicitly is an interesting direction for further research.

Related work. Influential early work on the use of statistical mechanics techniques for analyzing protein sequences was done by Shakhnovich and Gutin [20], who considered sequence optimization for lattice models and a small set of proteins. Further work on foldability and design in the context of sequence space was done by Shakhnovich [21], Saven and Wolynes [19], and Betancourt and Thirumalai [2]. A crucial distinction, however, is that while this earlier work focused primarily on model systems, and developed general principles

from a statistical physics perspective, we develop combinatorial algorithms with provable approximation guarantees, and apply these algorithms to the full set of known protein folds.

More recent work by Koehl and Levitt and by Larson et al. [9, 10] has also considered the sequence-structure relationship at the level of sequence space, using detailed atomic potentials. Again, this leads to a key difference with our work. While atomic potentials can be more accurate than the simplified fitness energy functions we use, they are more expensive to compute and less tractable to use in algorithms and analysis of the type we develop here; we focus on the more simplified functions so as to be able to sample the full sequence space with provable performance guarantees.

2. EVOLUTIONARY CAPACITY UNDER LOCAL FITNESS FUNCTIONS

We begin by describing the algorithm for approximately evaluating N with respect to a local fitness function. Let σ be a protein structure with n sites, and let $\mathcal{S}_n$ denote the set of all strings (i.e. protein sequences) of length n over the alphabet of amino acids. Thus $|\mathcal{S}_n| = 20^n$. For our purposes, a *fitness function* is any function $g : \mathcal{S}_n \to \mathbf{R}$ that assigns a real number to each protein sequence $X = x_1 x_2 \cdots x_n$, representing the energy of X in the structure σ. A fitness function is *local* if there exist functions g_i associated with each position $i = 1, 2, \ldots, n$ such that $g(X) = \sum_{i=1}^{n} g_i(x_i)$. This definition is sufficiently general to include any fitness function evaluating the energy as a sum of individual contributions from each site, including the THOM2 function that we adopt for our experiments. We refer the reader to [14] for a complete description of the THOM2 function.[1] The sequence distribution is defined in terms of g as $N(E) = |\{X : g(X) \leq E\}|$.

To provide some insight into the connection between evaluating N and the problem of counting feasible solutions to the Knapsack problem, we begin with a greatly simplified version of our main theorem, showing an equivalence between the Knapsack problem and evaluating the analogue of the function N over a two-letter amino acid alphabet. Specifically, let $\mathcal{S}'_n$ be the set of all n-bit *binary* strings, g a local fitness function defined over $\mathcal{S}'_n$, and N be defined in terms of g as above.

In the form of the Knapsack problem we will be using, there are n items of non-negative weights $a_1, \ldots, a_n$, and we want to count how many subsets of these items have total weight at most b. By using an indicator variable z_i that takes the value 1 if item i is included in a subset, and 0 otherwise, we see that this corresponds to counting the number of 0-1 vectors $\mathbf{z} = (z_1, \ldots, z_n)$ such that $\sum_i z_i a_i \leq b$. Morris and Sinclair give an FPRAS for this problem, i.e. an algorithm which, for any $\varepsilon > 0$, determines the number of feasible solutions to an instance of the Knapsack problem with high probability to within a multiplicative error of $(1+\varepsilon)$, in time polynomial in n and ε^{-1} [16]. Extending the terminology slightly, we will say that a FPRAS for the sequence distribution function N is an algorithm that, given an additional parameter $E \geq 0$, performs as a FPRAS for $N(E)$.

THEOREM 2.1. *There is a FPRAS for any sequence distribution function N with respect to a local fitness function g over a two-letter alphabet.*

[1] See also the URL cbsu.tc.cornell.edu/software/loopp/index.htm.

Proof. We begin by observing that for each sequence position i, there is a "better" and a "worse" choice of $x_i \in \{0, 1\}$ from the point of view of energy minimization; we say that 0 is *better* for position i if $g_i(0) \leq g_i(1)$, and we say that 1 is better otherwise. A sequence X^* minimizing g can be obtained simply by choosing the better symbol x_i for each position i; let $E^* = g(X^*)$ denote the energy of this sequence.

Given an energy bound E, we construct an instance of the Knapsack problem as follows. We want $z_i = 0$ in the Knapsack instance to correspond to the choice of the better symbol for position i, and $z_i = 1$ to correspond to the choice of the worse symbol, so we set $a_i = |g_i(0) - g_i(1)|$ and $b = E - E^*$. Now, a 0-1 vector $\mathbf{z}$ forms a feasible solution to the resulting instance of the Knapsack problem if and only if the sequence X obtained by choosing the better symbol in precisely those positions i for which $z_i = 0$ has an energy of at most E. The number of such sequences is $N(E)$, which is thus also the number of feasible solutions to the instance of the Knapsack problem we have constructed; hence, applying the Morris-Sinclair algorithm, we have a FPRAS for N. ■

Theorem 2.1 shows the connection to the Knapsack problem, but our goal is to extend the Knapsack algorithm to general functions over a 20-letter alphabet. (Or, more generally, over a k-letter alphabet.) This introduces new challenges: the proof of Theorem 2.1 is based on the observation that when there are just two letters, we can use the decision of whether or not an item is included in the knapsack to encode the choice between the letters. But when extending this idea to $k > 2$ letters, each position i has a lowest-energy symbol x_i^*, and then effectively $k-1$ ways of including item i in the knapsack to varying "extents." While Morris and Sinclair [16] in fact develop an extension to Knapsack problems in which each z_i can take values in some set of integers $\{0, 1, 2, \ldots, L\}$, this is not enough for our purposes; we are essentially dealing with the case in which variable z_i can take values in the set $\{g_i(0) - g_i(x_i^*), g_i(1) - g_i(x_i^*), \ldots, g_i(k) - g_i(x_i^*)\}$ (where again, x_i^* is the lowest-energy symbol for position i). We thus have a generalization where the domain of z_i is not uniformly spaced, and the domains of z_i and z_j may differ arbitrarily.

We now develop a FPRAS for this more general case. Like the Morris-Sinclair algorithm, our algorithm is based on a well-known equivalence between approximate counting and approximately uniform sampling, sketched out below. We refer the reader to the survey by Jerrum and Sinclair [5] for a very readable overview. (The basic idea is also related to the technique of *umbrella sampling* from statistical physics [25].)

First, notice that it is easy to determine $N(E^*)$ at the minimum energy E^* of any sequence: $N(E^*)$ is simply the product $\prod_{i=1}^{n} k_i^*$, where k_i^* is the number of distinct symbols that each minimize the function g_i. Suppose that for energies $E^* = E_0 < E_1 < E_2 < \cdots < E_m = E$, we could approximate each ratio $N(E_{i+1})/N(E_i)$ with high probability to within a multiplicative error of $1 + O(\varepsilon/m)$. Then we could approximate the value of $N(E)$ to within a multiplicative error of $1 + \varepsilon$ by evaluating the telescoping product

$$N(E) \quad = \quad N(E_0) \cdot \frac{N(E_1)}{N(E_0)} \cdot \frac{N(E_2)}{N(E_1)} \cdot \cdots \cdot \frac{N(E_m)}{N(E_{m-1})}.$$

To approximate the ratio $N(E_i)/N(E_{i-1})$, we sample almost uniformly from $\mathcal{S}_n^{(E_i)} = \{X : g(X) \leq E_i\}$, and estimate

$N(E_i)/N(E_{i-1})$ to be the reciprocal of the fraction of samples from $\mathcal{S}_n^{(E_i)}$ whose energy is bounded by E_{i-1}. For this approach to be efficient, E_{i-1} and E_i must be chosen close enough for a reasonable fraction of samples to fall below E_{i-1} in energy; but this is easily taken care of, so we are left with the problem of sampling almost uniformly from the set of sequences $\mathcal{S}_n^{(E_i)}$.

For this sampling, we define a Markov Chain on the set $\Omega = \mathcal{S}_n^{(E_i)}$ as follows (we also write $b = E_i$ for brevity and compatibility of notation). The start state is any sequence of minimum energy E^*. Then, for t steps, we perform the following update on the current sequence $X = x_1 \dots x_n$. Choose a position i and a symbol α, both uniformly at random, and let $X' = x_1 x_2 \dots x_{i-1} \alpha\, x_{i+1} \dots x_n$. If the energy $E(X')$ is bounded by b, then choose X' as the next state, otherwise stay at X. This Markov Chain can actually be restated as a simple random walk on a (directed) graph G_Ω: it has node set Ω, and an arc between any two states (X, X') as above whenever $E(X') \le b$; on the other hand, if $E(X') > b$, then we add a directed self-loop at X. The resulting graph G_Ω is strongly connected, aperiodic (due to self loops), and regular, so it has the uniform distribution as its unique stationary distribution. Our goal is now to show that the stationary distribution is approached after a polynomial number t of steps; then, we can produce fresh almost uniform random samples from $\mathcal{S}_n^{(E_i)}$ every polynomial number of steps. Chaining back through the consequences developed above, Markov Chain-based nearly uniform sampling lets us estimate the ratio $N(E_i)/N(E_{i-1})$ arbitrarily closely in polynomial time, and hence approximate $N(E)$. In summary, our approximation result follows from

THEOREM 2.2. *For any constant $\delta > 0$, there is a number t that is bounded by a polynomial in n and $\log \delta^{-1}$, such that the following holds: the variation distance between the uniform distribution and the distribution of the Markov chain after t' steps is bounded by δ, for all $t' \ge t$.*

Proof. Our proof closely follows the proof by Morris and Sinclair [16]. It relies on a well-known connection between the mixing time of a random walk and the edge congestion of an all-pairs multi-commodity flow. Let f be a multi-commodity flow on G_Ω routing one unit of flow between every ordered pair of vertices (X, Y). The *congestion* of f with respect to the state space size is $\mathcal{C}(f) = \frac{1}{|\Omega|} \cdot \max_e f(e)$. The *length* of the longest flow-carrying path is denoted by $\mathcal{L}(f)$. Finally, the *mixing time* τ_{mix} of the random walk is $\tau_{\mathrm{mix}} = \max_{X_0} \min\{t \mid \|P_{t'}(X_0) - U\| \le \frac{1}{4} \text{ for all } t' \ge t\}$, where $P_{t'}(X_0)$ denotes the probability distribution of the random walk starting at node X_0 after t' steps, U is the uniform distribution, and $\|\cdot\|$ is the total variation distance. The following theorem is a special case of a result by Sinclair [23].

THEOREM 2.3. *1. For any flow f, the mixing time τ_{mix} is bounded from above by $4n(n+1)\mathcal{C}(f)\mathcal{L}(f)$.*

2. Within $O(\tau_{\mathrm{mix}} \cdot \log \delta^{-1})$ steps, the variation distance is bounded by δ.

The crux of the proof in [16], and of our proof as well, is to define an appropriate flow f with congestion $\mathcal{C}(f)$ bounded by a polynomial in n, the number of positions in the string. In bounding $\mathcal{C}(f)$, we do not know the size $|\Omega|$ of the state space — after all, approximating this quantity is our goal in the first place. To circumvent this problem, we define a mapping from the units of flow that pass through any given node Z to the state space Ω, and show that each state is the image of only polynomially many units of flow.

Flow description. To define a flow, fix two states (sequences) $X = x_1 \dots x_n$ and $Y = y_1 \dots y_n$, and let $G_{X,Y} = \prod_{i=1}^n \{x_i, y_i\}$. All of the X-Y flow is routed only through the subgraph induced by $G_{X,Y}$. Notice that in $G_{X,Y}$, there are at most two different characters at each position, so by the same argument as in Theorem 2.1, we have a direct correspondence with the Knapsack problem, allowing us to apply the flow construction by Morris and Sinclair verbatim. However, the analysis has to be extended somewhat, as any one node in $G_{X,Y}$ may be in sets $G_{X',Y'}$ for many more states X', Y' than in the 2-letter case. Hence, the flow through any one node or edge may be larger.

If $X = x_1 \dots x_n$ is a sequence, then we write $\xi_i = g_i(x_i)$, and similarly for Y and Z (using η_i and ζ_i). We also introduce convenient set notation for states $Z \in G_{X,Y}$: for any index i, we write that $i \in Z$ iff z_i is not the minimum-energy choice between x_i and y_i (with an arbitrary tie-breaking toward x_i), i.e. iff $\zeta_i > \min\{\xi_i, \eta_i\}$, or $\xi_i = \eta_i$ and $z_i \ne x_i$. In the following, we give a brief summary of (a slightly simpler and less tight version of) the flow construction by Morris and Sinclair. The full motivation, details, and analysis are beyond the scope of this paper, and we refer the reader to [16].

For a fixed constant Δ, we let H be the set of 6Δ *heavy* indices $i \in X \cup Y$, the indices with largest values $\max\{\xi_i, \eta_i\}$ (if $|X \cup Y| < 6\Delta$, then $H = X \cup Y$), and let $X' = X \setminus H, Y' = Y \setminus H$. Writing $M = \max_{i \in X' \cup Y'} \{\xi_i, \eta_i\}$ (where the maximum is defined as 0 if X' and Y' are empty), this ensures that $g(X') + g(Y') \le 2b - 6\Delta \cdot M$, while $X' \subseteq X$ and $Y' \subseteq Y$. We first define a unit-flow from X' to Y', and then show how to turn it into a flow from X to Y.

The flow can be divided into three stages: The first and third stages define a flow from X' to X'' (and Y'' to Y', respectively), where the states X'' and Y'' satisfy $g(X'') \le b - \Delta M$ and $g(Y'') \le b - \Delta M$ (these states are called *not full*). Let $I = X' \oplus Y'$ be the set of indices in which X' and Y' differ, and $m = |I|$ their number. Choose T uniformly at random from $\{1, \dots, c_1 m\}$ (for some constant c_1), and perform a random walk for T steps on the set of pairs of sequences $(X'', Y'') \in G_{X,Y}$, starting from the pair (X', Y'). In each step, a position $i \in I$ is chosen uniformly at random. Then, the characters x_i'' and y_i'' at position i of the current strings (X'', Y'') are swapped unless this would result in either $g(X'')$ or $g(Y'')$ exceeding the bound b (in which case the random walk stalls for one step). Morris and Sinclair show that with constant probability c_2, the final states of this random walk will not be full. If $p(X'', Y'')$ denotes the probability of the non-full state (X'', Y'') being the final state of this random walk, then $\frac{p(X'', Y'')}{c_2}$ units of flow are routed from X' to X'', and then from Y'' to Y' (after being routed from X'' to Y'' in the second stage).

The second stage defines a flow from X'' to Y'', along paths that are obtained by changing characters x_i'' to y_i'' for all positions $i \in I$ (notice that $I = X' \oplus Y' = X'' \oplus Y''$). The key question is in which order to change the characters. For

this purpose, Morris and Sinclair show the existence of Δ-*balanced* $\mathrm{poly}(m)$-*uniform permutations*: distributions over all permutations $\hat{I}$ of I such that (1) $\min\{g(X''), g(Y'')\} - \Delta \cdot M \leq g(X'' \oplus \hat{I}_k) \leq \max\{g(X''), g(Y'')\} + \Delta \cdot M$ for any initial segment $\hat{I}_k$ of k elements of $\hat{I}$, and (2) for any set U, the probability that the first $|U|$ elements of $\hat{I}$ are exactly the set U is at most $\mathrm{poly}(m) \cdot \binom{m}{|U|}^{-1}$, i.e. at most by a polynomial factor larger than if $\hat{I}$ were chosen uniformly at random from among *all* permutations of I. Each path corresponding to a balanced permutation carries exactly the fraction of flow that is the permutation's probability under the distribution.

Finally, we construct X-Y paths from the X'-Y' paths described above. We want the paths to stay as close as possible to the hyperplane defined by $g(Z) \leq b$ — to this end, the (heavy) elements from H are repeatedly added and removed as necessary. Specifically, suppose that the X'-Y' path adds/removes elements in the order $j_1, \ldots, j_l$, and that after processing $j_1, \ldots, j_k$, the path under consideration is at some state $Z \in \Omega$. If j_{k+1} is added to Z to obtain the new state Z', then first remove one heavy element h from Z if necessary, then add j_{k+1} (i.e. route the flow from Z to $Z \cup \{j_{k+1}\}$ if possible, and from Z through $Z \setminus \{h\}$ to $Z \setminus \{h\} \cup \{j_{k+1}\}$ if not). If instead j_{k+1} is removed from Z, then first add one heavy element $h \in H$ to Z whenever possible, afterwards remove j_{k+1}. Finally, before processing j_1, add as many heavy elements to X as possible, and after all l indices j_k have been processed, add one element from $H \cap Y$ to Z if possible, then remove as many elements from $H \cap X$ as necessary to add from $H \cap Y$ again, and repeat until Y is reached.

Congestion analysis. This construction defines a feasible multicommodity flow between all pairs (X, Y) of vertices in G_Ω. The length of all flow-carrying paths is obviously $O(n)$, so it remains to bound the edge congestion. In fact, we bound the amount of flow through any node, which is clearly an upper bound on the amount of flow through any edge. We define a mapping from the state space to itself as follows: given a state $Z \in G_{X,Y}$, its preliminary encoding is $\hat{Z}$, where $\hat{z}_i = x_i$ whenever $z_i = y_i$, and $\hat{z}_i = y_i$ whenever $z_i = x_i$ ($\hat{z}_i$ is always well-defined). While $\hat{Z}$ itself may not be in Ω, its energy does not exceed b by much: by repeatedly adding heavy items above, we ensured that there is a heavy index $h \notin Z$ such that the addition of h to Z would make it impossible to use the next edge from Z on the path (which, let us say, adds an index j). The resulting lower bound $g(Z) \geq b - |\xi_h - \eta_h| - |\xi_j - \eta_j|$, together with the upper bounds $g(X), g(Y) \leq b$, implies that $g(\hat{Z}) = g(X) + g(Y) - g(Z) \leq b + |\xi_h - \eta_h| + |\xi_j - \eta_j|$. From $\hat{Z}$, we obtain $Z' \in \Omega$ by setting z'_h to be the one of x_h, y_h with smaller $g_h(\cdot)$ value, and similarly for z'_j. Our encoding of Z is $(Z', h, \hat{z}_h, j, \hat{z}_j)$, so for any Z, there are at most $O(n^2|\Omega|)$ different encodings. Conversely, given $Z, Z', h, \hat{z}_h, j, \hat{z}_j$, we can uniquely reconstruct $G_{X,Y}$ (although not necessarily X and Y themselves), by first reconstructing $\hat{Z}$; then, $G_{X,Y} = \prod_i \{z_i, \hat{z}_i\} = G_{Z,\hat{Z}}$.

Now, fix a node Z, and bound the flow through Z. We bound separately the contributions from the stages considered above. During the first and third stage (when there are random exchanges between X' and Y'), once we have $G_{X,Y}$, we can reconstruct X and Y uniquely if we know which are the heavy indices $\overline{H} \subseteq H$ that have been changed starting from X, and which are the steps $j_1, \ldots, j_k$ taken by the random walk starting at X' until it reached Z. For then, X can be obtained from Z by changing each character at one of the indices in $\overline{H}, j_1, \ldots, j_k$ to its other alternative in $G_{X,Y}$; Y is obtained by setting $y_i \neq x_i$ whenever possible. The amount of flow sent from X to Y through Z is at most $\frac{1}{c_2}$ times the probability that the random walk takes $j_1, \ldots, j_k$ as its first k steps, i.e. at most $\frac{1}{c_2} m^{-k}$. Summing over all choices of $Z', h, \hat{z}_h, j, \hat{z}_j, k, j_1, \ldots, j_k, \overline{H}$ now gives that the flow in first and third stage is bounded by

$$\sum_{Z',h,\hat{z}_h,j,\hat{z}_j} \sum_k \sum_{j_1,\ldots,j_k} \sum_{\overline{H} \subseteq H} \frac{m^{-k}}{c_2} \leq \sum_{Z',h,\hat{z}_h,j,\hat{z}_j} \sum_k \frac{2^{|H|}}{c_2} \leq O(\mathrm{poly}(n) 2^{|H|} |\Omega|).$$

In the second stage, we can apply a similar calculation. To reconstruct X, Y from $G_{X,Y}$, we need to specify the set $\overline{H} \subseteq H$ of heavy indices that have been changed, the steps taken by the first random walk (specified by the number T of steps, and the actual steps $j_1, \ldots, j_T$), and the set U of indices that have been changed in the first $u = |U|$ steps of the permutation used for routing the flow. The amount of flow sent along this path is the probability of choosing the number of steps to be T, choosing exactly $j_1, \ldots, j_T$ as those T steps of the random walk, and of having the set U be the first u elements of the permutation. By the almost-uniform property of the permutation, this probability is at most $(c_1 m)^{-1} \frac{1}{c_2} m^{-T} \cdot \left(\mathrm{poly}(m) \cdot \binom{m}{u}\right)^{-1}$. Summing over all such flow-carrying paths gives us the following bound on the total flow through Z:

$$\begin{aligned}
&\sum_{Z',h,\hat{z}_h,j,\hat{z}_j} \sum_T \sum_{j_1,\ldots,j_T} \sum_u \sum_{U:|U|=u} \sum_{\overline{H} \subseteq H} \\
&\qquad (c_1 m)^{-1} \frac{1}{c_2} m^{-T} \cdot \left(\mathrm{poly}(m) \cdot \binom{m}{u}\right)^{-1} \\
&\leq O\Big(\sum_{Z',h,\hat{z}_h,j,\hat{z}_j} \sum_T \sum_u 2^{|H|} (c_1 m)^{-1} \mathrm{poly}(m) \Big) \\
&\leq O(\mathrm{poly}(n) 2^{|H|} |\Omega|).
\end{aligned}$$

Adding the flow contributions from all three stages, and recalling that $|H| \leq 6\Delta$ is bounded by a constant, we obtain that the total flow through Z, and hence through any edge incident with Z, is at most $\mathrm{poly}(n) \cdot |\Omega|$. Therefore, $\mathcal{C}(f) = \mathrm{poly}(n)$, completing the proof. ∎

THEOREM 2.4. *There is a FPRAS for any sequence distribution function N with respect to a local fitness function g over a k-letter alphabet.*

Given Theorem 2.4, we also obtain a FPRAS for the evolutionary capacity of a given structure, simply by approximately evaluating the function N at the energy of the native sequence.

3. FURTHER COUNTING HEURISTICS

A Normalized Local Fitness Function. One concern about standard local fitness functions is that the low-energy regions of sequence space are dominated by simple homopolymers and other sequences of low complexity. A heuristic approach

to avoid this problem is to *normalize* the fitness function by evaluating, for a sequence X, the quantity $\hat{g} = g(X) - g(X^{rev})$, where X^{rev} denotes the reverse of sequence X. Such a strategy was proposed in the different context of hidden Markov models by Karchin et al. [7]. The idea in our case is for X^{rev} to play the role of a random sequence of the same composition as X. Then, homopolymers and other sequences of low complexity tend to have $\hat{g}$ values close to 0, while sequences that are highly adapted to the underlying structure will tend to have $g(X) \ll g(X^{rev})$.

A crucial point is that $\hat{g}$ is a local fitness function whenever g is, and hence also amenable to our approximation algorithm. To see this, observe that $\hat{g}(X) = \sum_{i=1}^{n} \hat{g}_i(x_i)$, where $\hat{g}_i(x_i) = g_i(x_i) - g_{n+1-i}(x_i)$. In some of the experiments, we will thus employ *normalized THOM2* as well as the standard THOM2.

Pairwise Fitness Functions. We also investigate the function N with respect to a class of fitness functions more general than local functions. We say that a fitness function is *pairwise* if there exist functions g_i associated with each position i, and functions g_{ij} associated with each pair of positions $i < j$, such that $g(X) = \sum_{i=1}^{n} g_i(x_i) + \sum_{i<j} g_{ij}(x_i, x_j)$. As before, we define $N(E) = |\{X : g(X) \leq E\}|$.

There is no hope of designing a FPRAS for a general pairwise fitness function, assuming $P \neq NP$. We establish this via the following theorem, which is similar in spirit to complexity results for Ising models on arbitrary graphs. The proof is by a reduction from graph k-coloring, and deferred to the full version.

THEOREM 3.1. *For each $k \geq 3$, there exist pairwise fitness functions g over k-letter alphabets for which, given a value E, it is NP-hard to determine whether $N(E) > 0$.*

Despite this hardness result, we have had success in practice with heuristics to approximate the evolutionary capacity using the pairwise fitness function TE13. We refer the reader to [24] for a complete description of TE13.[2] Our algorithm is still based on estimating a sequence of ratios of the form $N(E_{i-1})/N(E_i)$ via sampling. The Markov chain we use changes a symbol at a single position per step, as in the chain of Section 2, and while we find it to be effective in experiments, it is an open question whether it has a polynomial mixing rate.

A crucial contrast with the algorithm of Section 2 arises from the fact that we cannot compute the minimum energy E^*. We deal with this obstacle by using ratios that telescope *up* to a suitably chosen large energy, rather than *down* to the minimum. We first approximate the mean energy $\overline{E}$ over all sequences by direct sampling and averaging, then choose closely-spaced energies $E_\sigma = E_0 < E_1 < \cdots < E_m = \overline{E}$ and write

$$N(E_\sigma) \quad = \quad \frac{N(E_0)}{N(E_1)} \cdot \frac{N(E_1)}{N(E_2)} \cdot \cdots \cdot \frac{N(E_{m-1})}{N(E_m)} \cdot N(E_m).$$

Here, we can evaluate the final term since in practice, there is always a large constant fraction of sequences both above and below the mean energy, and hence these fractions can be estimated by direct sampling.

[2]See also the URL cbsu.tc.cornell.edu/software/loopp/index.htm.

4. EVOLUTIONARY CAPACITY: COMPUTATIONAL EXPERIMENTS

We have used the approximate counting algorithms from Sections 2 and 3 to determine evolutionary capacities for a large collection of protein structures, employing the local fitness function THOM2 and the pairwise fitness function TE13. We note that for the algorithm associated with Theorems 2.2 and 2.4, the bound in Theorem 2.2 is a conservative estimate of the number of Markov chain steps needed for approximate uniformity; while polynomial, it can be quite large for moderately-sized proteins. Thus, for reasons of efficiency in the computational experiments, we modified the algorithm to run the sampling procedure for fewer steps than the proof of Theorem 2.2 specifies, relying on standard heuristic tests to determine when the Markov chain was well-mixed. The results appear to be robust relative to the number of sampling steps used.

We chose a collection of protein structures selected to cover the space of all currently known folds. Since the Protein Data Bank (PDB) [1] has considerable redundancy, we followed previous work [15] and used a large subset of 3409 proteins that is less redundant but still represents the known PDB folds well. This subset differs from the one used in [15] only in that we omitted the longest proteins (of length greater than 500 amino acids) since some of the more expensive counting algorithms had poor convergence at this scale.[3]

To get an initial sense for the way in which $N(E)$ grows as a function of E for different proteins, we show in the left-hand side of Figure 1 plots of $\ln N(E)$ for 19 different proteins, each of length $n = 199$ or $n = 200$. At a coarse level, the curves are qualitatively similar (and of course, they meet at $N(E) = 20^n$ for sufficiently large energies E), but the actual shapes of the curves vary significantly across the different proteins. The curves with the sharper initial slopes are those for which the low-energy parts of sequence space are more densely populated, and it is an interesting question to find precise ways in which the three-dimensional shape of the protein translates into rough structural features of the $\ln N(E)$ curves.

Despite the variation in these curves across proteins of the same length, we find that evolutionary capacity is still strongly correlated with protein length. The right-hand plot in Figure 1 shows the evolutionary capacity as a function of length; each protein in our collection shows up as a single dot in the scatterplot, with all of the counting methods from Sections 2 and 3 superimposed. We note the striking amount of agreement between the capacities computed using THOM2 and TE13; although they are very different kinds of functions, the points associated with them in the plot lie approximately on a common line.

5. THE TEMPERATURE OF EVOLUTION

Given a means of approximately evaluating the function N, we can define further thermodynamic quantities on sequence space, including a notion of evolutionary temperature. We adopt a style of analysis used to estimate the Boltzmann factor for a configuration space in statistical mechanics (see e.g. Feynman [3]); in this context, it is useful to consider (discrete) difference quotients as playing the role of (continuous) deriva-

[3]The URL www.cs.cornell.edu/∼leonidm/jm_list.txt contains a full list of proteins used.

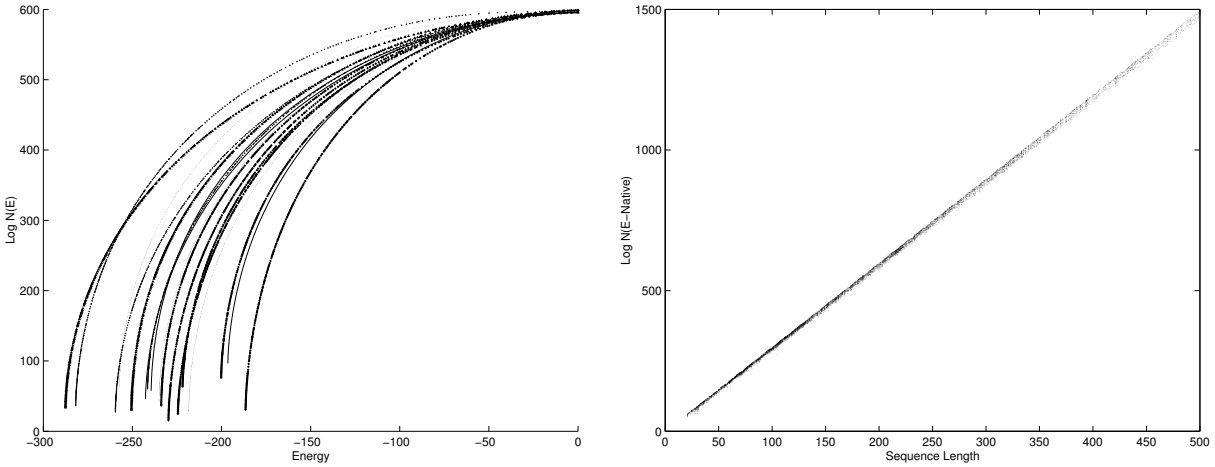

Figure 1: Left: $\ln N(E)$ as a function of E for 19 different proteins of (approximately) the same length: 1bcp_B, 1ber_A, 1iak_A, 1rgp, 4sbv_A, 1am2, 1pd2_1, 1qq2_A, 1qrj_B, 1nox, 3pcf_E, 1ais_B, 1bc9, 1iaa, 1ilm_B, 1qn8_A, 1qrn_D. Right: logarithm of evolutionary capacity as a function of length for each of THOM2, normalized THOM2, and TE13.

tives. Thus, for a small number $\Delta E > 0$, we write $\Omega(E) = \frac{N(E+\Delta E)-N(E)}{\Delta E}$; we refer to $\Omega(E)$ as the *number density of sequences*, and think of it as playing the role of the derivative $\frac{dN}{dE}$.

Now, consider a *selection function* G that takes a sequence at energy E and returns a *survival probability* $G(E)$. Such a selection function can be viewed as underpinning the evolution of sequences adapted to a given structure; using it, we can ask for the probability of seeing a sequence of energy between E and $E + \Delta E$. For small ΔE, this is approximately proportional to $G(E)(N(E + \Delta E) - N(E)) \approx G(E)\Omega(E)\Delta E$, and so we can write $P(E) = G(E)\Omega(E)$ as a probability density in terms of E.

We assume that sequences at the native energy have been selected because this is a highly probable value of $P(E)$; if we thus treat E_σ as the maximizer of $P(E)$ (and hence $\log P(E)$), we have

$$\left.\frac{d[\log P]}{dE}\right|_{E_\sigma} = \left.\frac{1}{\Omega(E)}\frac{d\Omega}{dE} + \frac{1}{G(E)}\frac{dG}{dE}\right|_{E_\sigma} = 0 \quad (1)$$

and hence

$$\left.-\frac{1}{G(E)}\frac{dG}{dE}\right|_{E_\sigma} = \left.\frac{1}{\Omega(E)}\frac{d\Omega}{dE}\right|_{E_\sigma}. \quad (2)$$

The point is that good approximations to $N(E)$ yield good approximations to $\Omega(E)$; in turn, this lets us approximate $\frac{d\Omega}{dE}$, and thus the right-hand side of Equation (2). In summary, we obtain an approximation for the left-hand side $-\frac{1}{G(E)}\frac{dG}{dE}$ at $E = E_\sigma$, and we refer to this quantity as β_σ.

From a statistical mechanics perspective (still in the spirit of [3]), we recognize the right-hand-side of Equation (2) as the standard definition of temperature: $\frac{1}{T} = \frac{dS}{dE}$, where $S = \log \Omega(E)$ is the entropy, and hence $\frac{1}{T} = \frac{1}{\Omega(E)}\frac{d\Omega}{dE}$. It is therefore natural to think of β_σ as an inverse temperature of selection: $\beta_\sigma = 1/T_\sigma$.

Using the above derivation, we can approximately compute T_σ for each structure σ in our collection of proteins. A key point here is that the temperature is a function of an individual protein structure, and there is no obvious reason why temperatures should be comparable for different proteins. In Figure 2 we plot temperature as a function of length for each protein in our collection, using the TE13 fitness function. (Results for the other functions are similar.) A striking observation that becomes clear from these plots is that the plot of the temperature in fact becomes relatively flat (i.e. roughly constant, though with large variance) for proteins of length 200 and greater. While approximately constant temperature was not *a priori* to be expected, we comment on some of its potential implications in the next, concluding section.

6. CONCLUSION

Our analysis of sequence space was based on the distribution function N, and by casting the evaluation of N as a combinatorial enumeration problem, we were able to develop Markov Chain-based techniques for approximating it with provable guarantees. We feel that this suggests the potential for methods from the area of approximate counting to shed light on further questions about the organization of the set of all amino acid sequences, a space that is much too large to be analyzed by more direct methods.

Armed with the ability to approximately evaluate N, we considered further analogues of thermodynamic quantities, including an evolutionary temperature. The sharply peaked distribution of temperatures across sufficiently long proteins is perhaps the most surprising finding to emerge from our computational experiments. The general conclusion is that the evolutionary selection function G appears to have the same dependence on energy in the neighborhood of the native state over an extensive set of proteins. It is not at all obvious that the "sequence-energy excitation" with respect to the lowest energy sequence should be comparable across protein families. This suggests that the mutation mechanism (at least as measured by stability energy for proteins longer than 200 amino acids) is approximately universal. Such universality is consis-

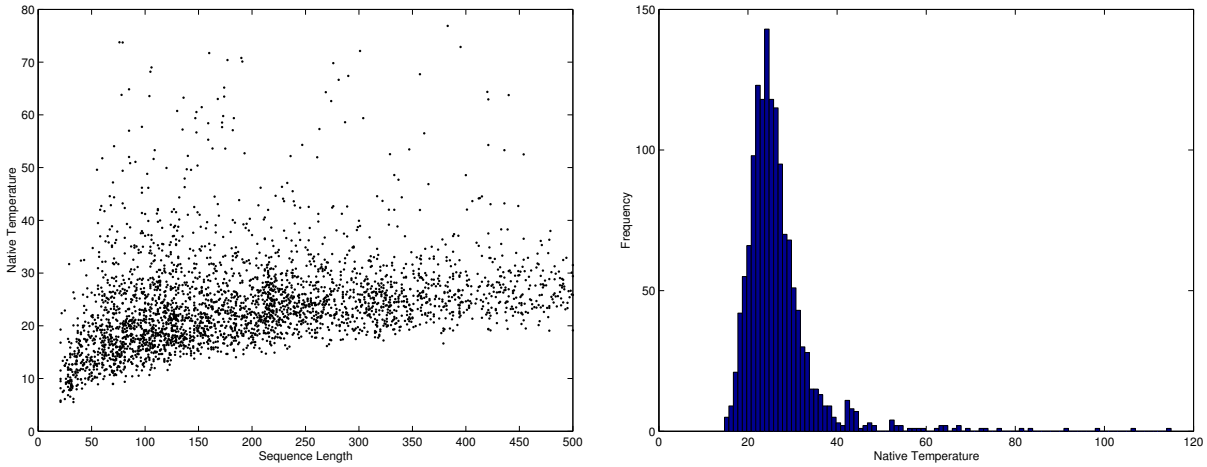

Figure 2: The plot on the left shows native sequence temperature under TE13 as a function of length. The plot on the right shows a histogram of native sequence temperatures under TE13 for all proteins in the collection of length greater than 200.

tent with two models: (1) connectivity between different clusters in protein space by small mutational steps [13], or (2) a single mutation mechanism that produces a similar statistical distribution of sequences in isolated islands of protein space. While evidence for extensive connectivity has been found in highly simplified models of protein structure [4, 8, 11, 12], it has been very difficult to demonstrate this connectivity for realistic models of proteins. Our method, in contrast, treats sequence space, which is far too large to analyze directly, by analogy with a physical system that can be "probed" at various points to test whether it is well-mixed. The uniformity of the temperature at these probe points offers evidence for a background mixing process on distinct protein families, and raises the question of whether this process is achieved by a uniform selection mechanism or by direct, though very slow, migration between protein families over the course of evolution.

Acknowledgments. We thank Catherine Grasso for her contributions to computational implementations that formed part of the background for the present work.

7. REFERENCES

[1] Berman, H. M., W. J., et al. (2000). "The protein data bank." Nucleic acids research 28: 235.

[2] Betancourt, M. R. and D. Thirumalai (2002). "Protein sequence design by energy landscaping." Journal of Physical Chemistry 106: 599-609.

[3] Feynman, R. P. (1982). *Statistical Mechanics: A set of lectures*, Benjamin, Reading MA.

[4] Huynen, M., P. Stadler, et al. (1996). "Smoothness within ruggedness: The role of neutrality in adaptation." Proceedings of the Natural Academy of Sciences USA 93: 397.

[5] Jerrum, M. and A. Sinclair (1996). The Markov chain Monte Carlo method: an approach to approximate counting and integration. Approximation Algorithms for NP-hard Problems. D. S. Hochbaum. Boston, PWS.

[6] Kabsch, W. (1978). "Discussion of solution for the best rotation to relate 2 sets of vectors." Acta Crystallographica Section A 34: 827-828.

[7] Karchin, R., M. Cline, et al. (2003). "Hidden Markov models that use predicted local structure for fold recognition: Alphabets of backbone geometry." Proteins, Structure, Function and Genetics 51(4): 504-514.

[8] Kleinberg, J. (1999). Efficient algorithms for protein sequence design and the analysis of certain evolutionary fitness landscapes. ACM RECOMB.

[9] Koehl, P. and M. Levitt (2002). "Protein topology and stability define the space of allowed sequences." Proceeding of the Natural Academy of Sciences USA 99: 1280.

[10] Larson, S. M., J. L. England, et al. (2002). "Thoroughly sampling sequence space: Large-scale protein design of structural ensembles." Protein science 11(12): 2804-2813.

[11] Lau, K. F. and K. Dill (1990). "Theory for protein mutability and biogenesis." Proceeding of the Natural Academy of Sciences USA 87: 638.

[12] Lipman, D. and W. Wilbur (1991). "Modeling the neutral and selective evolution of protein folding." Proceeding of the Royal Society London B 245: 7.

[13] Maynard, S. J. (1970). "Natural Selection and the concept of protein space." Nature 225: 563.

[14] Meller, J. and R. Elber (2001). "Linear Optimization and a double statistical filter for protein threading protocols." Proteins, Structure, Function and Genetics 45: 241.

[15] Meller, J. and R. Elber (2002). Protein recognition by sequence-to-structure fitness: Bridging efficiency and capacity of threading models. Advances in chemical physics. F. Richard, John Wiley and Sons. 120: 77-130.

[16] Morris, B. and A. Sinclair. (1999). Random walks on truncated cubes and sampling 0-1 knapsack solutions. *Proc. IEEE Foundations of Computer Science*, pp. 230-240.

[17] Orengo, C.A., D.T. Jones and J.M. Thornton. (1994). *Nature* 15, pp. 631-634.

[18] Saven, J. G. (2002). “Combinatorial protein design.” Current Opinion in Structural Biology 12: 453.

[19] Saven, J. G. and P. G. Wolynes (1997). “Statistical Mechanics of the Combinatorial Synthesis and Analysis of Folding Macromolecules.” Journal of Physical Chemistry B 101: 8375-8389.

[20] Shakhnovich, E. I. and A. M. Gutin (1993). “A new approach to the design of stable proteins.” Protein Engineering 6(8): 793-800.

[21] Shakhnovich, E. I. (1994). “Proteins with selected sequences fold into a unique native conformations.” Physical Review Letters 72(24): 3907-3911.

[22] Shindyalov, I. N. and P. E. Bourne (1998). “Protein structure alignment by incremental combinatorial extension (CE) of the optimal path.” Protein Engineering 11: 739-747.

[23] A. Sinclair (1992). “Improved bounds for mixing rates of Markov chains and multicommodity flow.” Combinatorics, Probability and Computing 1: 351-370.

[24] Tobi, D. and R. Elber (2000). “Distance dependent, pair potential for protein folding: Results from linear optimization.” Proteins, Structure, Function and Genetics 41: 40.

[25] Torrie, G. M. and J. P. Valleau (1977). “Non-physical sampling distributions in Monte-Carlo free energy estimation - umbrella sampling”.” Journal of Computational Physics 23: 187.