Massively parallel sampling of lattice proteins reveals foundations of thermal adaptation
Sergey V. Venev and Konstantin B. Zeldovich

Citation: The Journal of Chemical Physics 143, 055101 (2015); doi: 10.1063/1.4927565
View online: http://dx.doi.org/10.1063/1.4927565
View Table of Contents: http://scitation.aip.org/content/aip/journal/jcp/143/5?ver=pdfcov
Published by the AIP Publishing

Articles you may be interested in
Effects of knot type in the folding of topologically complex lattice proteins

Local packing modulates diversity of iron pathways and cooperative behavior in eukaryotic and prokaryotic ferritins

Mechanics of severing for large microtubule complexes revealed by coarse-grained simulations
J. Chem. Phys. 139, 121926 (2013); 10.1063/1.4819817

Order statistics inference for describing topological coupling and mechanical symmetry breaking in multidomain proteins
J. Chem. Phys. 139, 121913 (2013); 10.1063/1.4816104

Fold and sequence independent protein binding sites prediction algorithm
Massively parallel sampling of lattice proteins reveals foundations of thermal adaptation

Sergey V. Venev and Konstantin B. Zeldovich

Program in Bioinformatics and Integrative Biology, University of Massachusetts Medical School, 368 Plantation St, Worcester, Massachusetts 01605, USA

(Received 20 March 2015; accepted 17 July 2015; published online 5 August 2015)

Evolution of proteins in bacteria and archaea living in different conditions leads to significant correlations between amino acid usage and environmental temperature. The origins of these correlations are poorly understood, and an important question of protein theory, physics-based prediction of types of amino acids overrepresented in highly thermostable proteins, remains largely unsolved. Here, we extend the random energy model of protein folding by weighting the interaction energies of amino acids by their frequencies in protein sequences and predict the energy gap of proteins designed to fold well at elevated temperatures. To test the model, we present a novel scalable algorithm for simultaneous energy calculation for many sequences in many structures, targeting massively parallel computing architectures such as graphics processing unit. The energy calculation is performed by multiplying two matrices, one representing the complete set of sequences, and the other describing the contact maps of all structural templates. An implementation of the algorithm for the CUDA platform is available at http://www.github/kzeldovich/galeprot and calculates protein folding energies over 250 times faster than a single central processing unit. Analysis of amino acid usage in 64-mer cubic lattice proteins designed to fold well at different temperatures demonstrates an excellent agreement between theoretical and simulated values of energy gap. The theoretical predictions of temperature trends of amino acid frequencies are significantly correlated with bioinformatics data on 191 bacteria and archaea, and highlight protein folding constraints as a fundamental selection pressure during thermal adaptation in biological evolution. © 2015 AIP Publishing LLC. [http://dx.doi.org/10.1063/1.4927565]

I. INTRODUCTION

The theory of protein folding provides an exemplary link between statistical mechanics and evolutionary biology, and explosive growth of DNA sequencing data allows for robust statistical comparison of physical models with biological reality. While the basic concepts of protein physics such as the rugged energy landscape, the energy gap, decoy spectrum, and folding free energy are firmly supported by experimental data, protein thermal adaptation has received less biological attention so far. It is well established that organisms thriving in extreme environment, such as elevated temperatures found in hot springs and geothermal vents, evolved extremely thermostable proteins. On a case by case basis, specific molecular strategies to achieve thermostability have been identified, including formation of disulfide bonds, the hydrophobic effect, salt bridges and ionic interactions, and an expanded hydrophobic core. Nevertheless, the relative roles of those and other factors in conferring thermal stability vary depending on a specific protein. Therefore, we propose to address the question of thermal stability from a statistical viewpoint, by comparing statistical mechanics simulations with large scale genomic data.

Since the advent of the genomic era, it has been shown that proteome-averaged amino acid composition of thermophilic organisms differs from those of mesophilic ones, complementing the earlier findings about individual proteins. Specific trends of amino acid fractions with environmental temperature have been reported, making it possible to precisely predict the environmental temperature of an organism by analyzing amino acid frequencies in its proteome. These findings suggest that in addition to stabilization strategies unique to specific folds and sequences, a general biophysical phenomenon may drive the temperature shift of amino acid composition, at least in prokaryotes. Assuming that protein stability against thermal unfolding is under selective pressure during evolution, one can hypothesize that thermal adaptation results in proteins with larger energy gaps. Indeed, it has been shown that natural proteins increase their melting temperature by increasing their folding free energy $\Delta G$ at all temperatures. In the simplest physical framework, the larger gap can be achieved by selecting for specific (highly designable) native structures, or by adjusting effective amino acid interactions through biased composition. Alternatively, it has been suggested that protein stability is mainly controlled by protein chain length, omitting the specific molecular mechanisms that lead to the observed differences in enthalpy and entropy between idealized mesophilic and thermophilic proteins.

Surprisingly, we are not aware of a theory answering the very basic question of protein thermal adaptation: given the in-
interaction energies between the 20 types of amino acid residues, predict which amino acids will increase or decrease in frequency in more thermostable proteins. This problem was likely first approached by Dill, who showed (in a hydrophobic-polar (HP) model) that the free energy of folding has a pronounced minimum as a function of the fraction of solvophobic residues in a protein, in agreement with experimental observations. Unfortunately, to our knowledge, this statistical mechanics treatment has not been extended to the full 20-letter amino acid alphabet, despite the possibilities for quantitative comparisons afforded by present genomic data.

The closest result we are aware of comes from simulations. Based on the assumption that proteins in thermophiles have stabilities at high temperatures similar to those in mesophiles at lower temperatures, Berezovsky et al. used a 27-mer lattice protein model to design protein sequences that fold well at different temperatures, and found a good agreement between the amino acid frequency trends in the simulation and biological data. It was suggested that the temperature trends in amino acid composition can be qualitatively explained by positive and negative designs, so protein stability is increased by both decreasing the energy of the native state and increasing the energy of decoy structures. This concept, in turn, led to the “both ends of hydrophobicity spectrum” principle, according to which increased content of hydrophobic amino acids at elevated temperatures is responsible for a large hydrophobic core, while increased abundance of charged amino acids increases repulsion in the misfolded states. At the same time, 27-mer lattice proteins used in that work may introduce significant artifacts, as with a single residue at the center of the lattice they lack a true hydrophobic core. Although statistical mechanics of larger 3D lattice polymers proteins has been described, extended computational effort hampers their use in biophysical evolutionary simulations, which continue to use 2D or small 3D models.

To address these issues, we propose a very high performance, GPU-based algorithm for calculating energies of protein sequences in multiple conformations, and apply it to design thermostable proteins, and compare the simulated amino acid compositions with predictions of a novel, random distance-independent symmetric force field, as the typical number of contacts in a given conformation is proportional to protein length $L$ is $O(NL)$, Figure 1(a). Such an access pattern leads to inefficient memory accesses: for each pair of contacting residues $(m,n)$, one must look up residue types $A_m$ and $A_n$ in the protein sequence array, and then the interaction energy $e_{kmhn}$ must be retrieved from the force field matrix $e_{ij}$ and accumulated, Figure 1(a). Such an access pattern leads to inefficient use of memory bandwidth. Energy calculations for $K$ sequences in $N$ conformations each require $O(KNL)$ accesses to the force field, as the typical number of contacts in a given conformation is proportional to protein length $L$, Figure 2; the total number of arithmetic operations is also $O(KNL)$. Therefore, the serial algorithm is memory-bound, and its performance depends on the use of cache memory during force field lookup.

### B. GPU implementation

In simultaneous threading of many protein sequences into structural templates, also of length $L$, the threading energy $E^{(i)}$ of a sequence $A = A_1A_2A_3\ldots A_L$ in conformation $i$ is then defined as

$$E^{(i)} = \frac{1}{2} \sum_{m,n=1}^{L} e_{kmhn} \cdot c_{mn}.$$  

Threading of a sequence $A$ through all $N$ conformations yields its energy spectrum $E^{(i)}$, $i = 1 \ldots N$, which can be used to find its native state and folding free energy.

In the serial implementation, for each sequence, Eq. (2) is evaluated sequentially for all candidate conformations, forming an array of energies $E^{(i)}$, $i = 1 \ldots N$. Since contact maps are typically sparse, they can be stored as arrays of pairs of indices $(m,n)$ of residues in contact in a given structure, which provides the smallest memory footprint. As a result, the serial implementation requires multiple nested “jumping” memory accesses: for each pair of contacting residues $(m,n)$, one must look up residue types $A_m$ and $A_n$ in the protein sequence array, and then the interaction energy $e_{kmhn}$ must be retrieved from the force field matrix $e_{ij}$ and accumulated, Figure 1(a). Such an access pattern leads to inefficient use of memory bandwidth. Energy calculations for $K$ sequences in $N$ conformations each require $O(KNL)$ accesses to the force field, as the typical number of contacts in a given conformation is proportional to protein length $L$, Figure 2; the total number of arithmetic operations is also $O(KNL)$. Therefore, the serial algorithm is memory-bound, and its performance depends on the use of cache memory during force field lookup.
Calculation of the energy $E$ of a protein sequence in a given conformation. (a) Serial algorithm requires multiple “jumping” memory accesses (arrows): given an array of indices of contacting residues (top row), amino acid types are retrieved from the protein sequence (middle row), and then used to look up the interaction energies from the force field (bottom row), accumulated to the total energy $E$. (b) Alternatively, the energy can be calculated as an inner product of the sequence-energy matrix (left) and the structure’s contact map (right). The sequence-energy matrix contains the energies of all possible contacts between the residues in a sequence. As both matrices are stored as linear arrays, the calculation can be performed as a dot product of two vectors (below), representing the contact map and the energies of all possible contacts in a sequence. The dot product formalism is advantageous for massively parallel architectures, e.g., GPU.

A combination of sequence and structure are independent of each other and therefore can be parallelized. In a many-core device, such as GPU, it is tempting to assign a serial energy computation (2) of a sequence-structure combination to a dedicated core, expecting nearly linear performance scaling with the number of cores. However, such a brute force parallel implementation is suboptimal on GPUs due to nonlinear memory accesses while computing (2), and the competition between the cores for accessing the force field memory. To better map this problem to GPU hardware and to linearize memory accesses, we propose to calculate each energy $E$ of a sequence in a given conformation (2) as a dot product between two vectors, $E = \langle V|C \rangle$, where $V$ is a sequence-energy vector that combines force field $\epsilon_{ij}$ and primary sequence, and $C$ is a structural vector describing the conformation. This representation has a large memory footprint, because the lengths of both vectors $V$ and $C$ are $\sim L^2$, representing all possible elements in a contact map. However, the memory access pattern during dot product calculation is linear, allowing for efficient use of GPU resources and significant performance gains despite increased memory demands. Importantly, the sequence-energy vector $V$ can be prepared in advance, and reused for energy calculations of a sequence in many conformations, completely eliminating the memory access bottleneck during force field lookup. The matrix formalism of linking protein sequences, structures, and force field has been previously used in theoretical context.

To represent energy calculation (2) as a linear algebra operation, for each sequence $A = A_1 A_2 A_3 \ldots A_L$ of length $L$, we introduce the sequence-energy matrix $U$ of size $(L, L)$,

$$U = \begin{bmatrix} \epsilon_{A_1 A_1} & \cdots & \epsilon_{A_1 A_L} \\ \vdots & \ddots & \vdots \\ \epsilon_{A_L A_1} & \cdots & \epsilon_{A_L A_L} \end{bmatrix},$$

whose element $U_{ij}$ is energy of interaction $\epsilon_{A_i A_j}$ of two residues of types $A_i$ and $A_j$. The energy of a given sequence-structure combination is then given by the inner (Frobenius) product of the matrix $U$ and the contact map $c$,

$$E = \frac{1}{2} \sum_{m,n=1}^{L} U_{mn} c_{mn}. $$

In hardware, both $U$ and $c$ matrices are stored row by row as vectors of length $L^2$,

$$V = [\epsilon_{A_1 A_1} \cdots \epsilon_{A_1 A_L} \cdots \epsilon_{A_L A_1} \cdots \epsilon_{A_L A_L}],$$

$$C = [c_{11} \cdots c_{1L} \cdots c_{L1} \cdots c_{LL}],$$

so energy calculations (2) or (4) can be represented as a dot product of these two vectors,

$$E = \frac{1}{2} \langle V|C \rangle.$$

A schematic representation of this formalism is shown in Figure 1(b).

In Eq. (6), the vector $V$ represents the energies of all contacts between residues that the sequence can possibly produce (irrespective of structural constraints), and $C$ is a vector of 0 s and 1 s, corresponding to the contact map. Although during the calculation of the dot product many of the elements of the vector $V$ will be multiplied by zero, wasting arithmetic capacity,
the favorable sequential memory access pattern will result in superior performance compared to serial implementation (2).

In the general case of \(N\) conformations and \(K\) sequences, we can combine all conformational vectors into a matrix \(V\), where each column corresponds to a vector \(C_i, i = 1 \ldots N\), and further stack all sequence-energy vectors \(V\) into a matrix \(V\), where each row corresponds to a protein sequence. The product of these two matrices,

\[
E = \frac{1}{2} \cdot V \cdot C,
\]

provides a matrix \(E\), each row of which is the energy spectrum of a given sequence in the complete ensemble of conformations. Therefore, calculations of energies of \(K\) sequences of length \(L\) in \(N\) conformations can be represented by multiplying two large matrices \(V\) and \(C\), of sizes \((K, L^2)\) and \((L^2, N)\), respectively. This operation can be very efficiently performed on the GPU. In this algorithm, the number of force field lookups when threading multiple sequences is \(O(KL^2)\), and the total number of arithmetic operations is \(O(KNL^2)\).

To further improve the performance of the algorithm, we note that in a finite set of structures not all possible contacts are in fact present. Trivially, contacts between residues which are close in sequence are prohibited from geometric considerations. For the contacts between distant residues, lattice symmetry or stochastic nature of real protein structures provides additional constraints. Therefore, the number of contacts \(M\) that are encountered in at least one out of \(N\) structures is significantly less than \(L^2/2\). In cubic lattice structures, for example, only contacts between odd- and even-numbered residues are possible, so the number of distinct contacts in a large set of lattice structures is \(M \approx L^2/4\). For real protein structures, the randomness of long-range contacts decreases the gain, Figure 3. Specific values of \(M\) for various protein models are presented in Table I. For example, only 961 contacts are possible in compact cubic lattice 64-mers, rather than \(64^2/2 = 2048\). As the structural dataset is known before any energy calculations are attempted, the exact value of \(M\) can be determined empirically. Only those \(M\) contacts that are present in at least one of the structures are then included in the sequence-energy matrix \(V\) and the structural matrix \(C\), reducing their size to \((K, M)\) and \((M, N)\), respectively. With this optimization, the parallel algorithm uses \(O(KNM)\) rather than \(O(KNL^2)\) arithmetic operations when calculating (7), and the number of force field lookups is only \(O(KM)\), all during construction of \(V\).

Therefore, while the serial implementation uses \(O(KNL)\) force field lookups for threading of \(K\) sequences of length \(L\) onto \(N\) structures, our parallel algorithm uses only \(O(KM) \sim O(KL^2)\) lookups. Since in realistic applications \(L \sim 100\), and \(N \sim 10^4\), the parallel algorithm requires about 100 times fewer force field lookups, eliminating memory access collisions. This novel feature leads to a dramatic performance increase despite the greater number of arithmetic operations, \(O(KNM)\), compared to the serial algorithm’s \(O(KNL)\). Finally, if sequences and structures have different lengths, all sequences in the set can be extended to the length \(L_{\text{max}}\) of the longest sequence in the set, by padding with non-interacting gaps. The contact maps can be also zero-padded to form arrays of equal dimensions of \((L_{\text{max}}, L_{\text{max}})\), and the value of \(M\) can be found empirically as described above.

### III. ALGORITHM IMPLEMENTATION AND PERFORMANCE

The method of massively parallel energy function evaluation described above has been implemented in C++ using CUDA platform and matrix multiplication routines from cuBLAS library. The code is provided as C language API called GaleProt (GPU Accelerated Library for Evolution of Proteins), available at.github.com/kzeldovich/galeprot. The GaleProt API focuses on rapid computation of energy spectra, or energies of multiple protein sequences in multiple conformations, and transformation of the spectra \(E\) to biophysical quantities, such as probability of the native state, and folding free energy. Given the native state (index of the minimum

<table>
<thead>
<tr>
<th>Device</th>
<th>Dataset</th>
<th>Performance</th>
<th>Speed up</th>
<th>(M)</th>
<th>(L_{\text{L}}(L-1)/2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 CPU</td>
<td>27-mer</td>
<td>0.63</td>
<td>1.00</td>
<td>156</td>
<td>351</td>
</tr>
<tr>
<td>4 CPU</td>
<td>2.39</td>
<td>3.79</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tesla</td>
<td>332</td>
<td>526</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 CPU</td>
<td>64-mer</td>
<td>0.26</td>
<td>1.00</td>
<td>961</td>
<td>2016</td>
</tr>
<tr>
<td>4 CPU</td>
<td>0.92</td>
<td>3.53</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GeForce</td>
<td>45.2</td>
<td>174</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tesla</td>
<td>106</td>
<td>408</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 CPU</td>
<td>125-mer</td>
<td>0.12</td>
<td>1.00</td>
<td>3782</td>
<td>7750</td>
</tr>
<tr>
<td>4 CPU</td>
<td>0.47</td>
<td>3.91</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GeForce</td>
<td>12.5</td>
<td>104</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tesla</td>
<td>30.9</td>
<td>257</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 CPU</td>
<td>CATH</td>
<td>0.23</td>
<td>1.00</td>
<td>5653</td>
<td>5995</td>
</tr>
<tr>
<td>4 CPU</td>
<td>0.86</td>
<td>3.74</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GeForce</td>
<td>46.7</td>
<td>203</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tesla</td>
<td>111</td>
<td>484</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
energy conformation), and the complete energy spectrum of a sequence, one can estimate the partition function $Z$ and the probability of native state $P_{nat}$ of the sequence,

$$Z = \sum_{i=1}^{N} \exp \left(-\frac{E^n}{k_B T}\right),$$

$$P_{nat} = Z^{-1} \cdot \exp \left(-\frac{E^n}{k_B T}\right),$$

where $n$ is the index of the native conformation in the set of $N$ structures, $k_B$ is the Boltzmann constant, and $T$ is absolute temperature. Thus, assuming two-state folding mechanism, the folding free energy of a protein sequence can be expressed as

$$\Delta G = -k_B T \ln \frac{P_{nat}}{1 - P_{nat}}.$$  

GaleProt API provides functions to find the native state indices and energies, probability of the native state $P_{nat}$, and folding free energy $\Delta G$. The GPU implementation of the reduction algorithms uses tree-based reduction with sequential addressing and loop unrolling to achieve high efficiency. To serve evolutionary models that include protein-protein interactions, GaleProt API also provides functionality for GPU-accelerated calculation of binding energies in pairs of 64-mer compact cubic lattice proteins, enumerating the 144 mutual orientations of interacting cubes and selecting the lowest-energy orientation. GaleProt provides a 70-fold performance increase over a CPU when computing binding energies of lattice 64-mers on a Tesla K20 GPU.

To benchmark the performance of GaleProt for folding lattice proteins, we have generated sets of 10,000 random compact cubic structures of length $L = 27, 64, 125$, using an algorithm proposed by Jacobsen. This algorithm is believed to provide uniform sampling of conformational space without known biases. For each value of $L$, we have generated $10^3$ random protein sequences and found the native state of each sequence among the 10,000 conformations. Calculation speed (thousands of folding calculations per second, kfps, including finding the native structure and $P_{nat}$) was used as performance metric. The typical performance was about 106 kfps for 64mers using Tesla K20 GPU, or about 400 times faster than the serial implementation using a single core of a 3.7 GHz Intel Xeon E5-1620 v2 CPU. GaleProt output was validated by comparing the results obtained by GPU and serial CPU implementation. Small differences ($<10^{-3}$ of the $P_{nat}$ value) were observed due to non-commutative property of floating point summation, and the different ordering of summation operations in the CPU and GPU implementations. Additionally, some protein sequences have exactly the same energy in two different lattice conformations (degeneracy). The choice of one of the structures as the native state in this case is arbitrary, and depends on the implementation.

To test GaleProt on a real protein dataset, we applied it to a protein fold recognition problem, matching protein sequences from CATH database v3.4 to their native states using threading and a residue-based contact potential. We selected a subset of 1644 CATH domains of length between 90 and 110, and further reduced this list by excluding 50 structures with the largest and smallest numbers of contacts each, for a total of 1544 structures. Contact matrices were derived from structures using distance threshold $d = 11 \AA$ between $C_d$ atoms, and local contacts were excluded from consideration. To further assist in correct fold recognition, the contact potential $\epsilon_{i,j}$ was modified by subtracting the average energy from all elements, so that $\langle \epsilon_{i,j} \rangle = 0$. This modification makes fold recognition less sensitive to variations in the number of contacts or average density of template structures.

To calculate the energy of a sequence in a structural template of a different length, gaps have been added to both ends of the sequence in a sliding manner, transforming each input sequence into a set of sequences, with size determined by the difference between the length of the initial sequence, and the longest structural template. This procedure resulted in a set of 17,077 gapped sequences, forming 1544 groups corresponding to the original 1544 protein domains. Therefore, the problem of calculating the energy function in gapped threading is an excellent match for “many sequences—many structures” data model of GaleProt. GaleProt running on a Tesla K20 GPU achieved the performance of 70.1 kfps, while a single core of an Intel Xeon CPU could only yield 0.23 kfps. On a larger set of $10^5$ sequences which included randomized gapping, GaleProt performance on a Tesla K20 increased to 111.2 kfps. For the purposes of energy calculations, gaps did not carry a penalty, and did not interact with actual residues in the structure. To recognize the correct fold, for every group of sliding sequences, we selected the (gapped) sequence with the highest native state probability $P_{nat}$, considering the lowest energy structure as native, and the other 1543 as decoys. Although only the first sequence in the set aligns exactly to its native structure in the set of templates, other alignments may have comparably low energies, leading to imperfect fold recognition. Out of 1544 sequences, the native structures of 311 sequences, or 20% were exactly identified. An additional 21% of the sequences had the predicted native state structurally similar to the correct one (structural alignment TM-score is $>0.5$). Predicting the native structure of each sequence only took 14 $\mu$s on a Tesla K20 GPU. During these $14 \mu$s, on average, energies of 11 sequences in 1544 structures were evaluated, at the rate of 0.93 ns per sequence-structure combination for a ~100-residue protein. Of note, this test was designed to demonstrate the throughput of our algorithm, rather than to assess the quality of simplistic residue level threading for protein structure prediction.

The performance of GaleProt in the tests is summarized in Table I and Figure 4. Our GPU implementation of parallel protein threading yields a typical performance increase of 2.5-orders of magnitude (250-500 fold faster) compared to a single core, fast, modern CPU. Although most modern CPUs contain multiple cores, and energy function calculations are independent of each other, straightforward parallelization of the problem on a multicore CPU cannot reach GPU performance levels (Figure 4 and data not shown).

IV. RANDOM ENERGY MODEL OF PROTEIN THERMAL ADAPTATION

We used our novel algorithm for lattice protein folding to address the problem of thermal adaptation in proteins, specifically, prediction of amino acid types enriched or depleted in
thermostable proteins, if the amino acid interaction energies are known. We consider protein thermal adaptation in the context of the classical REM of protein folding. The REM assumes that the energy spectrum of a protein sequence in multiple structures follows a normal distribution, so the energy of the native (lowest energy) state can be found from the extreme value statistics, knowing the variance of the structural energy spectrum and the number of possible conformations. To link the energy spectrum of a sequence with the amino acid interaction potential, it is further assumed that the variance of the energy spectrum is proportional to the variance of the amino acid energy interaction matrix \( \mathcal{E} \). For simplicity, we can assume that \( \langle \epsilon_{ij} \rangle = 0 \), as adding a constant to the interaction energy does not affect the partition function in Eq. (8). The commonly accepted version of the REM assumes that all interactions in \( \mathcal{E} \) contribute equally to the protein’s energy spectrum. However, if the amino acid frequencies are sufficiently different, interactions between the more abundant amino acids will be inevitably sampled more frequently, affecting the variance of the energy spectrum. Denoting the amino acid frequencies by \( f_i, 1 \leq i \leq 20 \), we can approximate the effective interaction matrix by

\[
\mathcal{E}^* = \epsilon_{ij} f_i f_j,
\]

reflecting the probabilities of contact between the two amino acids and neglecting their correlations in sequences and structure. Therefore, we can approximate the energy gap of a protein sequence \( \Delta E = E_{\text{nat}} - \bar{E} \) by

\[
\Delta E \sim \sigma_{\mathcal{E}^*},
\]

where

\[
\sigma_{\mathcal{E}^*} = \sqrt{\langle (\epsilon_{ij} f_i f_j)^2 \rangle - \langle \epsilon_{ij} f_i f_j \rangle^2}.
\]

Here, in Eq. (11), we omitted all the prefactors accounting for the protein chain length, extreme value statistics of the native state due to the finite number of possible conformations, and entropic considerations.

To test the theoretical predictions of the random energy model of thermal adaptation, we used the GaleProt library to design 6 sets of 500,000 cubic lattice 64-mers each, so that in each set the average native state probability \( P_{\text{nat}} = 0.7 \) (Eq. (8)) for the temperatures \( T = 2, 3, 4, 5, 6, 7 \) (in arbitrary energy units). A residue-level potential was used in the simulation, and the structural dataset consisted of 10^5 structures drawn at random from 10^3 structures of compact 64-mers generated by the Jacobsen algorithm. A Monte Carlo algorithm was used for sequence design, accepting or rejecting random mutations according to the changes in \( P_{\text{nat}} \). In agreement with earlier findings, we found that average amino acid frequencies in each of these sets smoothly change with temperature, proving that selection of progressively more stable proteins imposes constraints on the amino acid composition. The energy gap of the designed sequences \( \Delta E = \bar{E} - E_{\text{nat}} \) was very strongly correlated with design temperature \( R = 0.99 \), data not shown, confirming that high design temperatures do select highly thermostable proteins.

Remarkably, we found that the average energy gap \( \Delta E \) of the designed protein sequences is very strongly correlated with its prediction according to Eq. (11), Figure 6. Therefore, changes in amino acid composition alter the effective amino acid interaction energy matrix, Eq. (10), and translate into quantitative predictions of the energy gap of model proteins. This finding naturally explains the previously proposed idea of “both ends of the hydrophobicity spectrum”: in a situation where hydrophobic and charged amino acids are overrepresented, the effective energy interaction matrix is more diverse compared to a balanced amino acid composition. The greater variance of the amino acid interactions in turn makes it possible to construct proteins with enhanced thermal stability.

Of note, these energetic effects lead to the decrease of sequence entropy

\[
S = - \sum_{i=1}^{20} f_i \log f_i
\]

with design temperature. The pronounced divergence of amino acid frequencies from their random 5% values, Figure 5, leads to the decrease of the sequence entropy with temperature, Figure 7. It is instructive to compare this finding with the previously hypothesized need of highly diverse protein sequences to achieve large energy gaps. While

FIG. 4. GaleProt performance (kfps) for different protein models and hardware in logarithmic scale. GaleProt on a Tesla K20 GPU outperforms the CPU implementation by 250-500 fold. Detailed metrics are presented in Table I.

FIG. 5. Model proteins designed to fold well at different temperatures \( T \) exhibit clear trends in amino acid frequencies; frequencies of leucine (L), isoleucine (I), arginine (R), and threonine (T) are shown as examples. At low temperatures, stability can be achieved by swapping amino acids within a random sequence (5% composition) elevated temperatures necessitate adjustment of the amino acid composition to reach the desired stability.
this expression does ensure a zero energy gap for homopolymers due to their zero sequence entropy, it assumes that the effective variance of amino acid interactions \( \sigma \) does not depend on the amino acid composition. Our simulations show that variation of effective amino acid interactions by changing the amino acid composition has a more profound impact on protein folding and stability than reduced sampling of sequence space at lower values of sequence entropy \( S \).

Successful prediction of protein energy gap by the variance of effective amino acid composition interaction matrix \( \sigma_{E^*} \) (Eq. (11) and Fig. 6) makes it possible to predict the temperature trends of amino acid frequencies knowing their design temperature, and the gap can be predicted by \( \sigma_{E^*} \), we can write

\[
\sigma_{E^*}(\vec{f}) = aT + b, \quad \vec{f} = \{f_i\}, \quad i = 1 \ldots 20. \tag{13}
\]

This approximation holds up numerically in our simulations, with the correlation coefficient \( R = 0.93 \). Therefore, we hypothesize that changes in amino acid composition observed in proteins with enhanced thermostability are due to the increase of \( \sigma_{E^*} \) by those composition trends, so

\[
\frac{df_i}{dT} \propto \frac{\partial \sigma_{E^*}}{\partial f_i}.
\tag{14}
\]

Additionally, since \( \sum_i f_i = 1 \), then \( \sum_i \frac{df_i}{dT} = 0 \), so finally

\[
\frac{df_i}{dT} \propto \frac{\partial \sigma_{E^*}}{\partial f_i} - C, \quad C = \frac{1}{20} \sum_k \frac{\partial \sigma_{E^*}}{\partial f_k}.
\tag{15}
\]

Although it is possible to derive an explicit expression for \( \frac{df_i}{dT} \), it is not particularly illuminating. In Figure 8(a), we compare the predictions of Eq. (15) with the temperature trends derived from the simulation as the slopes of the curves in Figure 5. Although the prediction of the slopes is imperfect, the strong, statistically significant correlation suggests that control of effective interaction energies via amino acid composition is among the major factors in protein thermal adaptation. The
noticeable split of the predictions into a hydrophobic and charged arms is reminiscent of the positive and negative design hypotheses, and may be also due to the specific structure of the Miyazawa-Jernigan amino acid interaction matrix, with only two major eigenvalues. To compare the prediction of our theory with biological data, we used a dataset of 191 bacteria and archaea with completely sequenced genomes and known optimum growth temperatures in the range between 7 and 103 °C. We have then performed the linear regression between the frequencies of each of the 20 amino acid types and optimum growth temperature, and plotted the slopes of that regression against the theoretical predictions from Eq. (15). Excluding one outlier, leucine, we have observed a significant positive correlation between the theoretical model and experimental data, Figure 8(b). The correlation suggests that biophysical constraints on protein stability have been among the selective pressures in place during adaptation of proteomes to different environmental temperatures.

V. DISCUSSION

Increasingly, data-intensive nature of modern biology presents challenges for storage, processing, and analysis of large datasets. These demands spurred the development of novel approaches, including crowdsourcing, high-performance molecular dynamics, specialized hardware, and a plethora of GPU accelerated versions of various existing algorithms including Rosetta-like RNA structure prediction algorithm and TM-score calculation. Several important problems, including protein structure prediction, protein design, and simulations of sequence evolution require calculations of protein energy functions for large numbers of sequences. Parallelizing these calculations on a GPU is a logical step in applying new commodity hardware to solving biological problems. However, as we demonstrate, straightforward porting of the traditional energy calculation to the GPU, under one core–one sequence paradigm, suffers from a severe memory bottleneck, as all compute cores compete for memory access during force field lookups. When calculating the energies of K sequences of length L in N conformations, such an algorithm performs \( O(KNL) \) arithmetic operations and \( O(KNL) \) force field lookups. Accesses to the force field array in memory occur in a random order, making this algorithm very inefficient when implemented on a GPU.

To address this issue, we propose a novel, very fast technique for calculating contact energies in proteins using a GPU. By introducing a reusable sequence-energy matrix \( \mathbf{V} \) independent of structures, and multiplying it by the structural matrix \( \mathbf{C} \), we were able to solve the same problem in just \( O(KM) \sim O(KL^2) \) force field lookups \( (M < L^2/2) \) is the number of unique contacts in the full set of structures). Since normally \( N \gg L \), the memory bottleneck during force field access is no longer significant. As a trade-off, the number of arithmetic operations increased from \( O(KNL) \) to \( O(KNM) \). However, these operations are now distributed across hundreds of cores in the GPU and are performed on perfectly ordered memory locations during matrix multiplication, allowing for coalesced memory accesses. As a result, our algorithm dramatically (two orders of magnitude) outperforms currently used CPU implementations of residue-level protein threading and lattice protein folding. Calculating energy as matrix multiplication allows for portability and scaling on any parallel hardware platform supporting BLAS linear algebra routines, such as supercomputer facilities.

Lattice protein calculations represent the most straightforward application of our method. These theoretical models received renewed attention in the context of modeling large-scale features of protein evolution, lattice protein models possess a realistic shape of energy spectrum and sequence space, while remaining computationally tractable. In evolutionary models, organism fitness or replication rate, is directly related to biophysical properties of its proteins, such as foldability and protein-protein interactions. Refinement of these models leads to increasingly complex fitness functions, exacerbating computational burden. Accordingly, either 2D or small 3D (27-mer) protein models are used, raising a possibility of lattice artifacts interfering with biologically relevant phenomena. While the use of off-lattice models is an important step towards higher realism, their combination with Brownian dynamics folding led to major approximations, such as short proteins and a reduced amino acid alphabet. High performance of GaleProt in folding large (64- and 125-mer) lattice proteins is very promising for improving the fidelity of evolutionary simulations, as these larger lattices are much closer to reality than previously used small lattice models, notably by surface to volume ratio. Dramatic performance increases achieved by GaleProt, up to 400-fold in folding free energy calculation (Table I), make complex protein evolution simulations easily accessible. Here, we used our technique to design \( 5 \times 10^5 \) thermostable 64-mer proteins for seven environmental temperatures. Such a design mimics thermal adaptation on the proteome level and represents a significant advance in terms of realism of the protein model and the scale of sampling, removing statistical uncertainty. This broad, rapid sampling of thermostable lattice proteins made it possible to make accurate comparisons between simulation and theory predictions. Our findings clearly show that control of effective amino acid interactions via amino acid composition is the key mechanism of thermal stabilization in heteropolymers. Increased diversity of the effective residue-residue interaction energies in thermostable proteins, in turn, leads to broader energy spectrum for individual sequences and ultimately to the larger energy gap.

The predicted trends in amino acid frequencies in thermostable proteins favorably compare with biological data, Figure 8(b). The positive correlation between purely theoretical prediction from Eq. (15) and experimental data from 191 genomes strongly suggests that physical factors such as protein thermostability did play a significant role in biological evolution, complementing previous findings. Our model successfully bridges multiple evolutionary scales, as it links intramolecular interactions in proteins with statistical properties of complete proteomes evolved under a wide variety of selective pressures. The deviations between theoretical and experimental trends in amino acid frequencies may be due to multiple factors not accounted for in the purely physical model, including the influence of the genetic code and
guanine-cytosine (GC) content of the genomes on amino acid frequencies, and the approximate nature of the Miyazawa-Jernigan interaction potential used for theoretical predictions. Inevitable differences between biochemical processes in bacteria and archaea both present in the dataset, aerobic and anaerobic lifestyles, and the limited control for phylogenetic similarity between the organisms from the dataset also contributes to differences between the theory and bioinformatics observations. In particular, both theory and simulations predict a strong increase of leucine content in the thermostable proteins, whereas it is only minimally increased in experimental data. While the origin of this discrepancy remains unclear, it is worth noting that leucine is encoded by six different codons and is the most abundant of the 20 amino acids, with ∼10% frequency in all proteomes. Additionally, changes in the hydrophobic effect and interaction potential at elevated temperatures may affect amino acid frequencies beyond our simplest physical model.

Selection for thermostable proteins affects not only the amino acid composition but also the structural repertoire, as stable sequences tend to fold into highly desirable structures, and structural proteomes of thermophilic organisms are enriched in dense and highly desirable protein domains. Although the compositional bias of sequences with highly desirable native folds has not been fully characterized, our results suggest that differences in structural desirability between thermophiles and mesophiles are coincident with significant changes in amino acid usage. Importantly, we demonstrated that thermal stabilization and change in folding free energy can be achieved by altering amino acid frequencies, without invoking the chain length argument. The method we propose is particularly efficient for “many sequences—many structures” problems, and it can be useful for genome-wide threading calculations, as in structural proteomics projects, or for rapid screening of protein sequences against large structural datasets. Importantly, as protein structure representation in our method relies exclusively on the indices of residues in contact, both compact and non-compact configurations can be treated in the same framework. The general nature of our algorithm makes it applicable to other types of contact map energy calculation in biophysics, including RNA structure prediction. Structural information on different levels of coarse graining may be useful in different applications. Residue burial profiles near the native protein conformation have been used to determine regions of protein that participate in the allostery interactions. Lattice protein models with realistic volume-to-surface ratio can be potentially used to extract similar type of information.

Protein design, or inverse folding problem, tightly links exploration of sequence space with precise energy calculations to find a sequence fitting a desired fold and/or forming a particular active site. Our algorithm can be potentially extended to protein design applications, as it allows for rapid calculations of energies of multiple sequences, so screening of an entire set of mutant sequences can be performed as a single operation. Efficient generation of protein contact maps with atomistic resolution will require computational optimization, especially when rotamer libraries are in use. Nevertheless, we believe that further development of our method will make it possible to perform fast energy calculations for highly realistic protein models.

ACKNOWLEDGMENTS

The authors acknowledge stimulating discussions with E. I. Shakhnovich, C. A. Schiffer, R. W. Finnberg, and members of the ALiVE working group. This work was supported by Defense Advanced Research Projects Agency [Nos. D13AP00041 and HR0011-11-C-0095].

54R. A. Goldstein, Protein Sci. 16, 1887 (2007).