Teaching computers to fold proteins

Ole Winther* and Anders Krogh†

Center for Biological Sequence Analysis, The Technical University of Denmark, Building 208, DK-2800 Lyngby, Denmark

(Dated: June 7, 2004)

A new general algorithm for optimization of potential functions for protein folding is introduced. It is based upon gradient optimization of the thermodynamic stability of native folds of a training set of proteins with known structure. The iterative update rule contains two thermodynamic averages which are estimated by (generalized ensemble) Monte Carlo. We test the learning algorithm on a Lennard-Jones (LJ) force field with a torsional angle degrees-of-freedom and a single-atom side-chain. In a test with 24 peptides of known structure, none folded correctly with the initial potential functions, but two-thirds came within 3Å to their native fold after optimizing the potential functions.

PACS numbers: 05.10.-a,07.05.Mh,87.15.Aa,87.15.Cc

It is one of the long-standing challenges of science to simulate protein folding in a computer and predict the three-dimensional structure – the native fold. According to Anfinsen’s hypothesis the native fold of a protein is the one with the lowest free energy [1]. To fold a protein in silico, it is therefore necessary to have a sufficiently good description of the energetics of the system. Even the most sophisticated all-atom potentials [2, 3] and statistical potential functions [4–6] will not usually give stability of an experimentally determined native structure. Furthermore, these potential functions have so many degrees of freedom that nano-second time-scale molecular dynamics simulations require of the order of months on even the fastest computers. To sample the state space of a protein in solution with present-day computers it is therefore necessary to use a simplified description of the protein and the solvent rather than an all-atom model. It is virtually impossible to calculate such potential functions from first principles.

In this paper we describe a method to estimate parametrized potential functions from a training set of known protein structures. Most previous work on estimation of potentials use statistical approaches [4–6], which are based on static structures. The main feature in our approach is that we optimize the potentials during simulation of the folding process, so as to maximize the thermodynamic probability of the native folds of the whole training set. This maximum likelihood estimation procedure, which is essentially Boltzmann learning [7], can be thought of as iteratively stabilizing the native structure on the one hand and ‘unlearn’ incorrect folds, which traps the protein during folding, on the other. Other approaches exist that, rather than optimizing the thermodynamic stability directly, optimize closely related measures such as the difference between the native energy and the energy of a set of alternative conformations [8, 9], the normalized difference between the native energy and the average energy over all alternative conformations [10–12], the thermodynamic average of the overlap to the native state in a contact map energy model [13, 14] and linear optimization methods for ensuring that the native state has lowest free energy [15, 16]. The overlap method is equivalent to optimizing the thermodynamic stability for a specific overlap contact map definition of the native fold. Optimizing the thermodynamic stability has also been suggested as an objective in theoretical protein design, see e.g. Ref. [17].

In the general setup we have a parameterized energy function \( E_\theta(R, seq) \) with parameters \( \theta \), which give the energy for an amino acid sequence \( seq \) with atomic coordinates \( R \). The probability of finding the \( i \)th training sequence in its native state is given by the Boltzmann weighted volume of conformation space compatible with the native structure divided by the total Boltzmann weighted volume of conformation space

\[
P(\text{nat}_i | seq_i, \theta) = \frac{\int_{\text{nat}_i} \exp(-\beta E_\theta(R, seq_i)) \, dR}{\int \exp(-\beta E_\theta(R, seq_i)) \, dR},
\]

where \( \beta = 1/kT \) and the integral in the numerator is only over the part of conformation space associated with the native structure. The definition and choice of the size of the native volume in conformation space should reflect all expected variability such as the loss of description accuracy due to the crudeness of the protein model, thermal variability of the native state and the uncertainty in the determination of the crystal/NMR structure. Optimizing the probability density of the native (crystal) structure, as suggested in Ref. [9], rather than a volume around the native structure ignores these sources of variability. In this study we define the native volume as all structures within a \( C_\alpha \) root mean square deviation (RMSD) of 1Å from the crystal structure. Note that although non-protein like structures (e.g. with steric overlaps) exist within the native volume, a successful training will assign a high energy and thus a small Boltzmann weight to these.

The objective is to maximize the joint probability \( \prod_{i=1}^N P(\text{nat}_i | seq_i, \theta) \) with respect to the parameters \( \theta \), where \( N \) is the number of sequences in the training set. We choose to perform the maximization by gradient as-
respectively. In neural computation context this is known
from the Boltzmann averages by a (generalized ensemble)
as the Boltzmann learning rule [7]. In simulations we per-
Monte Carlo method.

space and the part associated with the native structure,
noting Boltzmann averages over the total conformation
with
potential is close to the real physics. To demonstrate the
validity of the method we have applied it to a simple force
model steric constraints, and the sum over
of adjustable parameters for the hydrophobic energy is
increasing on a log-scale. Once every cycle, where a cycle
min
ing from

= 0
to being completely buried. The inner sum counts the
consists of
energy contains both angle dependence and a 12–10
Lennard-Jones potential with two adjustable parameters
local energy contains two types of terms introduced to model
side chain
of terms. The first one is a pure radial 12-6 Lennard
interactions are included. The
a
interaction, i.e.
where
both amino acid side chain, so
this introduces a total of
parameters.

Hydrogen bonding is only considered for backbone
NH and C’O pairs. The hydrogen bond energy contains both angle dependence and a 12–10
Lennard-Jones potential with two adjustable parameters

= \sum_{i,j} u_{ij} \left( 5 \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{12} - 6 \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{10} \right)
where

\sum_{i,j>i} e^\alpha(a_i,a_j) \left\{ \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{12} - 2 \left( \frac{\sigma_{ij}}{r_{ij}} \right)^6 \right\}
where

\sum_R \epsilon_i^{\text{surf}} f \left[ \frac{1}{H} \sum_{j=1}^{n} f \left( \frac{x_i - x_j}{\sigma_i^{\text{surf}}} \right) - \frac{1}{R} \right]
where

\epsilon_i^{\text{surf}} = \epsilon^\alpha(a_i) is the energy change induced by taking
side chain
from being completely exposed to solvent to being completely buried. The inner sum counts the
number of neighboring amino-acids and the adjustable
parameters
set the relevant scale and
for burial of each amino acid. The total number of adjustable parameters for the hydrophobic energy is
2*2*2*(20*19/2+20)=840 for Lennard Jones and 5*20=100
for surface.

The temperature scale is arbitrary since the Boltzmann
weight only depends upon the product \beta E and the scale
of E is set during training. In the test below we choose the
folding ‘temperature’ 1/\beta_{fold} = 0.1 for all training
examples and set the initial parameter values to be on
the same scale. The initial values are chosen such that
all amino-acids have the same parameter values, except
for different surface energy terms (\epsilon^\alpha(a_i)).

To get the learning algorithm eq. (2) to work properly
we need reliable estimates of thermodynamic averages.
To achieve this we use parallel tempering [21], where
the system is simulated independently at a number
of different temperatures. In this study N_{temp} = 15
ranging from T_{min} = 0.1 to T_{max} = 0.8 with equal spacing
on a log-scale. Once every cycle, where a cycle
consists of N_{conf} elementary conformation updates, the
temperature of two random systems (adjacent in temperature) are exchanged with the Metropolis probability
\( P(\text{accept}) = \min(1, \exp(\Delta \beta \Delta E)) \) where \( \Delta E \) and \( \Delta \beta \) are the energy and inverse temperature difference between the two systems respectively. In our case we use \( N_{\text{conf}} = 40 \) of which one quarter are pivot moves (rotation around a random torsional angle) and the rest are 'local' moves which is defined as choosing two torsional angles next to the same peptide bond (i.e. \( \psi_i \) and \( \phi_{i+1} \)) and rotate them opposite angles. We collect statistics for \( 10^4 \) cycles between each update of the adjustable parameters. We choose the folding temperature to be the minimum temperature \( T_{\text{fold}} = T_{\text{min}} \) and thus only use the statistics for this temperature to update the parameters. To ensure sampling of the native conformations, a number (typically 3) of the \( N_{\text{temp}} \) systems are initialized in the native state. The remaining systems are initialized in low energy states found in the previous update in different RMSD-distance intervals (0-1 Å, 1-2 Å, ...). This ensures fast focus on relevant regions of conformation space. Subsequent long runs starting from coil confirm that this procedure is sufficient close to generate equilibrated samples. As an alternative to the batch update rule eq. (2), one can use an online version where the parameters are updated using a single training example at a time. In this study, we use an intermediate approach, where we update the parameters using three batches each with one third of the example. The results presented below are obtained using approximately 500 parameter updates. The training set consists of a small set of 24 protein fragments (or peptides) of length 11-14 of mainly \( \alpha \)-helices and \( \beta \)-turns[24]. They have been suggested to adopt their native structure even as fragments [23]. Running the training on 8 processors on a Silicon Graphics Origin 3000 computer, 500 parameter updates take approximately 2 CPU weeks.

We tested the final potential by initializing the 24 training sequences in random coil. After an initial equilibration, conformations were saved with fixed intervals at the folding temperature \( T_{\text{fold}} = 0.1 \) in a long test run. These sampled conformations are called decoys below. Some results from the test run are shown in Fig. 1. The decoys are clustered by introducing a RMSD cut-off of 0.5 Å and assigning as the first cluster center the decoy with the most neighbors within the cut-off. We remove these decoys and repeat the procedure until all decoys have been assigned to a cluster. The clustering is not very sensitive to the specific choice of the cut-off indicating that the clusters are well-separated and that it makes sense to assign a free energy to each: \( F(\text{clus} i) = -T \ln P(\text{clus} i) \), where the probability \( P(\text{clus} i) \) of cluster \( i \) is the number of decoys in cluster \( i \) divided by the total number of decoys. The decoy plots reveal a complex free energy landscape with competing minima. In a few cases free energy minima both with small and large RMSD to the native fold exist simultaneously. In the subsequent analysis the cluster center of the cluster with the lowest free energy was chosen as the predicted fold.

To assess the performance of the trained potential, it is compared to the initial essentially homo-polymer potential and the results of folding with an all-atom potential [22]. Binning the RMSD results in bins of size 0.5 Å, i.e. 0-0.5, 0.5-1,..., 6-6.5 Å, we get \([7,4,2,0,2,1,2,3,3,0,0,0,0]\) for the trained potential, which should be compared to \([0,0,0,1,0,8,5,8,1,0,1,0]\) for the initial parameter setting and \([4,0,2,3,5,3,0,1,0,2,0,2,2]\) for the all-atom potential. We observe a clear improvement over the initial potential indicating that the training process actually works and similar results—although the comparison is biased—to the all-atom potential that requires many human expert man hours to derive.

Probing the significance of the folding temperature by performing the test run at a lower temperature \( T = 0.025 \) shows that the overall performance—in terms of RMSD for the largest cluster—is significantly worse. This ‘cold denaturation’ effect, which is illustrated in Fig. 1, shows that the solution found by the algorithm is using entropy to stabilize the native fold.

To get an understanding of the successes and failures of the potential we have visualized low energy structures (see Fig. 2) and made Ramachandran plots for the amino acids. The successful predictions are very native-like making the same hydrogen bonds as the native structure. However, some of the side chains are very close together. Although a side chain is ‘effective’ with degrees of freedom averaged out and not as an atom, the small distance means that the characteristic separation in the Lennard Jones potential is small and will be sensitive to small changes in the distance between the side chains. The structure for some of the failures are not ‘protein-like’ and some of the amino acids are not reproducing the Ramachandran behavior found for real proteins.
These findings show that the principle works, however, it is clear that the potential function model can be improved in many ways. One of the great advantages of the method is that many terms can be added and if they do not work well their weight would end up being very low. However, it should be kept in mind that the better the starting point, the more likely it is to reach a reasonable parameter set. The test suggests that the representation of side chains in the potential with just one pseudo-atom atom. This would probably make the Ramachandran behavior 'protein-like' and remove some of the false minima the model is currently struggling with. It is also possible to go in the opposite direction and use a more restricted conformational search space, e.g. by only sampling experimentally observed Ramachandran angles or using an I-Sites library to generate conformations [20]. The two different views are complementary and the results of the CASP exercise has shown that it is important to pursue both to generate good ab initio predictions [20].

The ultimate goal of optimizing potentials is to obtain reasonable predictions for sequences not in the training set (generalization). Preliminary runs on such test se-

sequences show poor generalization, which is primarily a result of the small training set. It is therefore important to now scale up to a more realistic size using more and longer sequences. We are currently working on ways to speed up the whole process to achieve this goal.

More details about parameter settings, data sets and results can be found at www.imm.dtu.dk/~owi/.

Acknowledgments. We thank the anonymous referees for useful comments. This work was sponsored by a grant to the Center for Biological Sequence Analysis (Soren Brunak) from the Danish National Research Foundation.

* Electronic address: owi@imm.dtu.dk; Present address: Informatics and Mathematical Modelling, Technical University of Denmark, 2800 Lyngby, Denmark.
1 Electronic address: krogh@binf.ku.dk; Present address: Bioinformatics Centre, University of Copenhagen, Universitetsparken 15, 2100 Copenhagen, Denmark.

24. Protein fragments used, PDB-code and amino acids: 1ALC 21-32, 1BGS 10-22, 1BSG 88-98, 1FKF 27-38, 1HGF 100-113, 1HRc 91-102, 1I1B 101-112, 1MBC 6-17, 1MBC 29-40, 1MBC 99-111, 1MBC 131-142, 1Pga 43-54, 1UBQ 3-15, 211B 69-82, 211B 103-112, 2MHR 51-62, 2MHR 67-78, 2MHR 102-113, 2PCY 18-29, 3LZM 24-35, 3LZM 99-111, 4PTI 22-33, 5CYT 88-101, 7RSA 2-13. All sequences and structures are listed in Ref. [22].