Teaching computers to fold proteins

Winther, Ole; Krogh, Anders Stærmose

Published in:
Physical Review E

Link to article, DOI:
10.1103/PhysRevE.70.030903

Publication date:
2004

Document Version
Publisher's PDF, also known as Version of record

PHYSICAL REVIEW E **70**, 030903(R) (2004)

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
(Received 26 September 2003; revised manuscript received 26 April 2004; published 27 September 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 3 *Å* to their native fold after optimizing the potential functions.

DOI: 10.1103/PhysRevE.70.030903 PACS number(s): 87.15.Cc, 07.05.Mh, 05.10.−a, 87.15.Aa

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 $E_\theta(\mathbf{R},\mathrm{seq})$ with parameters θ, which give the energy for an amino acid sequence seq with atomic coordinates $\mathbf{R}$. The probability of finding the ith 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

$$P(\mathrm{nat}_i|\mathrm{seq}_i,\theta) = \frac{\int_{\mathrm{nat}_i} \exp[-\beta E_\theta(\mathbf{R},\mathrm{seq}_i)]d\mathbf{R}}{\int \exp[-\beta E_\theta(\mathbf{R},\mathrm{seq}_i)]d\mathbf{R}}, \qquad (1)$$

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/nuclear magnetic resonace (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_α root mean square deviation (RMSD) of 1 Å 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.

*Present address: Informatics and Mathematical Modelling, Technical University of Denmark, 2800 Lyngby, Denmark. Electronic address: owi@imm.dtu.dk

†Present address: Bioinformatics Centre, University of Copenhagen, Universitetsparken 15, 2100 Copenhagen, Denmark. Electronic address: krogh@binf.ku.dk

The objective is to maximize the joint probability $\Pi_{i=1}^{N} P(\mathrm{nat}_i|\mathrm{seq}_i,\theta)$ with respect to the parameters θ, where N is the number of sequences in the training set. We choose to perform the maximization by gradient ascent $\theta^{\mathrm{new}} := \theta^{\mathrm{old}} + \Delta\theta$, where

$$\Delta\theta = \eta\nabla_\theta \sum_i \ln P(\mathrm{nat}_i|\mathrm{seq}_i,\theta)$$

$$= \eta\beta \sum_i [\langle \nabla_\theta E_\theta(\mathbf{R},\mathrm{seq}_i)\rangle - \langle\langle \nabla_\theta E_\theta(\mathbf{R},\mathrm{seq}_i)\rangle_{\mathrm{nat}_i}], \quad (2)$$

with η being the learning rate, $\langle\ldots\rangle$ and $\langle\ldots\rangle_{\mathrm{nat}_i}$ denoting Boltzmann averages over the total conformation space and the part associated with the native structure, respectively. In neural computation context this is known as the Boltzmann learning rule [7]. In simulations we perform the Boltzmann averages by a (generalized ensemble) Monte Carlo method.

The above learning rule applies to any differentiable potential function. The aim is to estimate a potential that gives a high probability to the correct fold under the given protein model and with the chosen simulation method, which of course does not guarantee that the potential is close to the real physics. To demonstrate the validity of the method we have applied it to a simple force field. The amino acid unit model has six atoms:

$$\begin{array}{ccccc} & & R_i & & \\ -\mathrm{N} & & \mathrm{C}_\alpha & & \mathrm{C}'- \\ & \mathrm{H} & & \mathrm{O} & \end{array}$$

where O and H are introduced to be able to define backbone hydrogen bonds and to give a more realistic local torsion potential. The whole side chain is represented by R_i, so the parameters relating to this is amino acid dependent, whereas the other atoms are treated more conventionally. The conformational degrees of freedom are the torsional angles ϕ and ψ (rotation around the NC_α bond and $\mathrm{C}_\alpha\mathrm{C}'$ bond). The $\mathrm{C}_\alpha R_i$ distance is adaptive (one parameter for each of the 20 amino acids), whereas all other bond lengths and angles are fixed to their average value as given in Ref. [18]. The angle to R_i is fixed to the average value for $\mathrm{C}_\alpha\mathrm{C}_\beta$.

The energy is a generalization of the one proposed in Ref. [19] which in turn is inspired by the classical force fields. The energy is split into local and non-local terms $E = E_{\mathrm{local}} + E_{\mathrm{non\text{-}local}}$. The local interactions are mainly introduced to model local steric constraints and the non-local consists of three types of terms introduced to model pairwise interactions, hydrophobic, surface and related effects (hp) and hydrogen bonding (hb): $E_{\mathrm{non\text{-}local}} = E_{\mathrm{hp}} + E_{\mathrm{hb}}$.

The local energy contains two types of terms $E_{\mathrm{local}} = \epsilon_\phi/2\Sigma_i(1+\cos 3\phi_i) + \epsilon_\psi/2\Sigma_i(1+\cos 3\psi_i) + \Sigma_i \Sigma_{X,Y} \epsilon_i^{XY}\{(\sigma^{XY}/r_i^{XY})^{12} - 2(\sigma^{XY}/r_i^{XY})^6\}$. The sums on i are over the individual amino acids. The first two terms–which are probably not so important–introduce a three fold symmetry and has two parameters independent of the side chains. The last term is mainly introduced to model steric constraints, and the sum over X, Y runs over a restricted set of pairs of local atoms in amino acid i, $i+1$ and $i+2$ (with terms set to zero when $i+1$ or $i+2$ are larger than the length of the protein). The following atom pairs are included: $\mathrm{H}(i)R(i)$, $R(i)\mathrm{O}(i)$, $\mathrm{H}(i)\mathrm{H}(i+1)$, $\mathrm{O}(i)\mathrm{H}(i+2)$ and $\mathrm{O}(i)\mathrm{O}(i+1)$. Only the first two pairs depend on the amino acid side chain, so this introduces a total of $2(20+20+2+2+2+) = 92$ 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 $E_{\mathrm{hb}} = \epsilon_{hb}\Sigma_{ij}\, u_{ij}\{5(\sigma_{hb}/r_{ij}^{\mathrm{HO}})^{12} - 6(\sigma_{hb}/r_{ij}^{\mathrm{HO}})^{10}\}$, where $u_{ij} = \cos^2\theta_{ij}^{\mathrm{NHO}}\cos^2\theta_{ij}^{\mathrm{HOC}'}$ for $\pi/2 < \theta_{ij}^{\mathrm{NHO}}$, $\theta_{ij}^{\mathrm{HOC}'} < 3\pi/2$ and zero otherwise, r_{ij}^{HO} is the distance between $\mathrm{H}(i)$ and $\mathrm{O}(j)$, $\theta_{ij}^{\mathrm{NHO}}$ is the $\mathrm{N}(i)\mathrm{H}(i)\mathrm{O}(j)$ angle and $\theta_{ij}^{\mathrm{HOC}'}$ the $\mathrm{H}(i)\mathrm{O}(j)\mathrm{C}'(j)$ angle.

The hydrophobic interaction E_{hp} consists of two types of terms. The first one is a pure radial 12-6 Lennard Jones potential that should take into account all non-local, hydrophobic and other forces between the ith amino acid a_i and the jth a_j. Both $\mathrm{C}_\alpha\mathrm{C}_\alpha$ and RR interactions are included. The $\mathrm{C}_\alpha\mathrm{C}_\alpha$ interaction, i.e. $\Sigma_{i>j}\, \epsilon^\alpha(a_i,a_j)\{(\sigma^\alpha(a_i,a_j)/r_{ij}^\alpha)^{12} - 2(\sigma^\alpha(a_i,a_j)/r_{ij}^\alpha)^6\}$, where r_{ij}^α is the $\mathrm{C}_\alpha(i)\mathrm{C}_\alpha(j)$ distance, is mainly introduced to model steric constraints. The second type of term, which plays a minor role, is a surface energy term inspired by Refs. [20] $\Sigma_i^R\, \epsilon_i^{\mathrm{surf}} f[1/s_i \Sigma_{j\neq i-1,i,i+1}^R f((\sigma_j^{\mathrm{surf},0} - r_{ij}^R)/\sigma_j^{\mathrm{surf}}) - b_i/s_i]$, where $f(x) \equiv 0.5(1+\tanh(x)) \in [0,1]$ is a sigmoid non-linearity. The surface energy term is chosen such that $\epsilon_i^{\mathrm{surf}} = \epsilon^{\mathrm{surf}}(a_i)$ is the energy change induced by taking side chain a_i from being completely exposed to solvent to being completely buried. The inner sum counts the number of neighboring amino-acids and the adjustable parameters $s_i, b_i, \sigma_i^{\mathrm{surf},0}, \sigma_i^{\mathrm{surf}}$ set the relevant scale and bias for burial of each amino acid. The total number of adjustable parameters for the hydrophobic energy is $2\times 2\times(20\times 19/2+20) = 840$ for Lennard-Jones and $5\times 20 = 100$ for surface.

The temperature scale is arbitrary since the Boltzmann weight only depends upon the product βE and the scale of E is set during training. In the test below we choose the folding "temperature" $1/\beta_{\mathrm{fold}} = 0.1$ 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 $[\epsilon^{\mathrm{surf}}(aa_i)]$.

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 N_{temp} of different temperatures. In this study $N_{\mathrm{temp}} = 15$ 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 $P(\mathrm{accept}) = \min(1, \exp(\Delta\beta\Delta E))$ where ΔE and $\Delta\beta$ are the energy and inverse temperature difference between the two systems respectively. In our case we use $N_{\mathrm{conf}} = 40$ 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., ψ_i and ϕ_{i+1}) and rotate

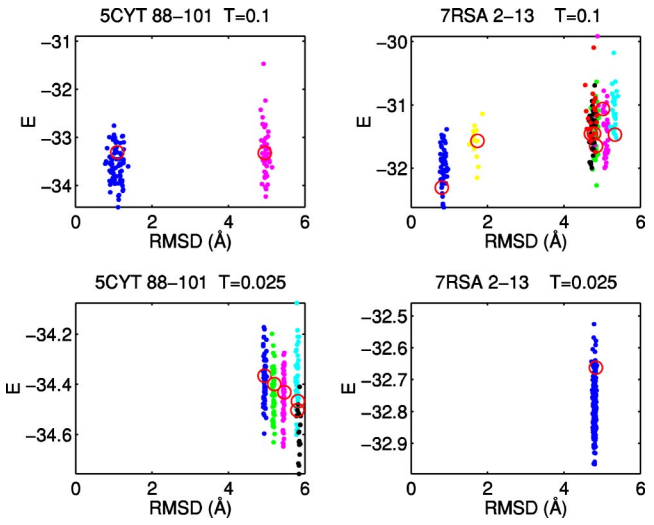

FIG. 1. (Color online) Decoy plots–energy vs RMSD (Å) for two peptides at the folding temperature and below. These two peptides "cold denaturate." Cluster centers are marked with a circle. The largest cluster (in blue) is, in all four cases, the one with the lowest RMSD.

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 $T_{\mathrm{fold}}=T_{\mathrm{min}}$ 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_{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 $(0-1\ \text{Å}, 1-2\ \text{Å}, \ldots)$. This ensures fast focus on relevant regions of conformation space. Subsequent long runs starting from coil confirm that this procedure is sufficiently 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 α-helices and β-turns [24]. They have been suggested to adopt their native structure even as fragments [23]. Running the training on eight 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 $T_{\mathrm{fold}}=0.1$ 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 cutoff of 0.5 Å and assigning as the first cluster center the decoy with the most neighbors within the cutoff. 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 cutoff

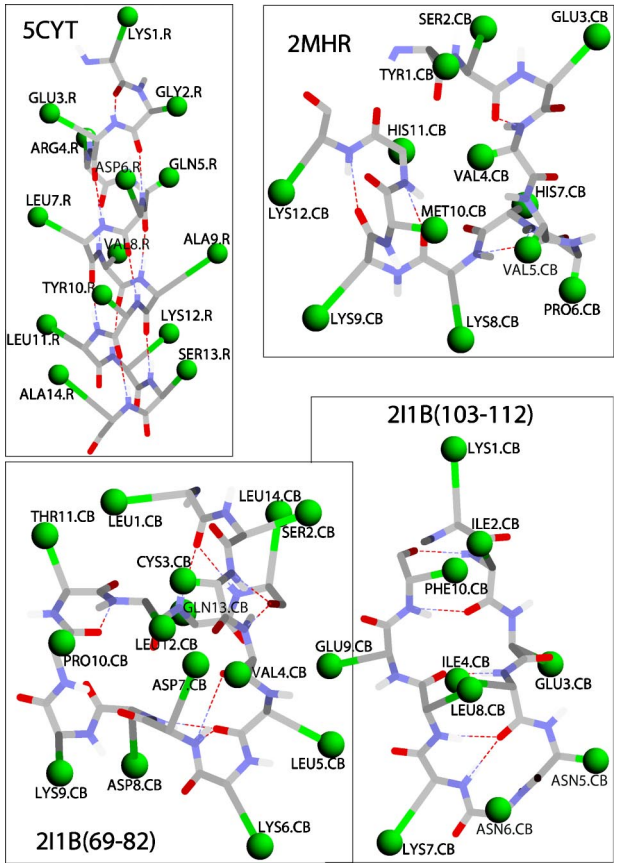

FIG. 2. (Color online) Representative decoys from the lowest free energy cluster for four peptides from top left to right (ID, RMSD from native, native structure type): 5CYT 88-101, 1.04 Å, α-helix; 2MHR 67-78, 4.74 Å, α-helix; 2I1B 69-82, 4.4 Å, β-turn and 2I1B 103-112, 1.63 Å, β-turn.

indicating that the clusters are well-separated and that it makes sense to assign a free energy to each: $F(\mathrm{clus}\ i)=-T\ \ln P(\mathrm{clus}\ i)$, where the probability $P(\mathrm{clus}\ i)$ 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 size 0.5 Å, i.e.,$0-0.5, 0.5-1, \ldots, 6-6.5$ 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 $[0,0,0,0,1,0,8,5,8,1,0,1,0]$ 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

$T=0.025<T_{\mathrm{fold}}$ 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 and a fixed angle is too crude. One remedy is to make the side chain model more realistic, e.g., by introducing an explicit C_β 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 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/

ACKNOWLEDGMENT

This work was sponsored by a grant to the Center for Biological Sequence Analysis (Søren Brunak) from the Danish National Research Foundation.

[1] C. B. Anfinsen, Science **181**, 223 (1973).
[2] B. R. Brooks *et al.*, J. Comput. Chem. **4**, 187 (1983).
[3] F. A. Mohany *et al.*, J. Phys. Chem. **79**, 2361 (1975).
[4] S. Miyazawa and R. L. Jernigan, J. Mol. Biol. **256**, 623 (1996).
[5] R. Samudrala and J. Moult, J. Mol. Biol. **275**, 895 (1998).
[6] M. J. Sippl, J. Mol. Biol. **213**, 859 (1990).
[7] G. E. Hinton and T. J. Sejnowski, IEEE Conf. Comput. Vision and Patt. Recog., 448 (IEEE Press, New York, 1983).
[8] V. N. Maiorov and G. M. Crippen, J. Mol. Biol. **227**, 876 (1992).
[9] F. Seno *et al.*, Proteins **30**, 244 (1998).
[10] R. A. Goldstein, Z. A. Luthey-Schulten, and P. G. Wolynes, Proc. Natl. Acad. Sci. U.S.A. **89**, 4918 (1992).
[11] M.-H. Hao and H. A. Scheraga, Proc. Natl. Acad. Sci. U.S.A. **93**, 4984 (1996).
[12] L. A. Mirny and E. I. Shakhnovich, J. Mol. Biol. **264**, 1164 (1996).
[13] U. Bastolla, M. Vendruscolo, and E.-W. Knapp, Proc. Natl. Acad. Sci. U.S.A. **97**, 3977 (2000).
[14] U. Bastolla *et al.*, Proteins **44**, 79 (2001).
[15] J. B. Rosen *et al.*, Biophys. J. **79**, 2818 (2000).
[16] C. Micheletti *et al.*, Proteins **42**, 422 (2001).
[17] A. Rossi *et al.*, J. Chem. Phys. **112**, 2050 (2000).
[18] T. E. Creighton, *Proteins: Structures and Molecular Properties*, 2nd ed. (W. H. Freeman & Co., New York, 1992).
[19] A. Irbäck *et al.*, Proc. Natl. Acad. Sci. U.S.A. **97**, 13614 (2000).
[20] R. Bonneau *et al.*, Proteins **45**, 119 (2001).
[21] C. J. Geyer and E. A. Thompson, J. Am. Stat. Assoc. **90**, 909 (1995).
[22] J. T. Pedersen and J. Moult, J. Mol. Biol. **269**, 240 (1997).
[23] J. Moult and R. Unger, Biochemistry **30**, 3816 (1991).
[24] Protein fragments used, PDB-code and amino acids: 1ALC 21-32, 1BGS 10-22, 1BGS 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, 2I1B 69-82, 2I1B 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].