Can a Pairwise Contact Potential Stabilize Native Protein Folds Against Decoys Obtained by Threading?

Michele Vendruscolo,* Rafael Najmanovich, and Eytan Domany
Department of Physics of Complex Systems, Weizmann Institute of Science, Rehovot, Israel

ABSTRACT **We present a method to derive contact energy parameters from large sets of proteins. The basic requirement on which our method is based is that for each protein in the database the native contact map has lower energy than all its decoy conformations that are obtained by threading. Only when this condition is satisfied one can use the proposed energy function for fold identification. Such a set of parameters can be found (by perceptron learning) if M_p, the number of proteins in the database, is not too large. Other aspects that influence the existence of such a solution are the exact definition of contact and the value of the critical distance R_c, below which two residues are considered to be in contact. Another important novel feature of our approach is its ability to determine whether an energy function of some suitable proposed form can or cannot be parameterized in a way that satisfies our basic requirement. As a demonstration of this, we determine the region in the (R_c, M_p) plane in which the problem is solvable, i.e., we can find a set of contact parameters that stabilize simultaneously all the native conformations. We show that for large enough databases the contact approximation to the energy cannot stabilize all the native folds even against the decoys obtained by gapless threading. Proteins 2000;38:134–148.**

Key words: protein folding; contact maps; protein potential; pairwise contact approximation

INTRODUCTION

Any numerical approach to protein folding involves three main choices: (a) the representation of protein structure; (b) an energy function for a sequence on a given structure; or (c) the set of alternative structures among which the native fold is selected. The underlying assumption is that there exists some energy function, from which the native fold can be derived by minimization.[1]

A conceptually straightforward way to implement this assumption to approaching the native fold is to solve, by Molecular Dynamics, Newton's equations of motion for a detailed atomistic model of a protein in solution. Although considerable advances have been made along this direction,[2–4] carrying this out all the way till folding lies beyond the capacities of present day computers. This indicates that perhaps one needs a more coarse-grained representation of a protein and its structure. In such representations amino acids are represented by one or a few interacting centers, from whose positions the eneregy of the protein is calculated.[5–12] The representation of protein structure that we decided to use[13] is that of contact maps.[14–17] Using this representation has numerous additional computational advantages[18] that are discussed in the following.

The energy function to be used is dictated by the choice of the representation of the structure. Nevertheless, there is still considerable freedom to choose one of a multitude of possible potentials and for each choice one has to select the parameters to be used. There are two main methods to derive energy parameters. The first method is to relate them to observed amino acid pairing frequencies. The idea was first introduced by Tanaka and Scheraga[19] and successively developed by Miyazawa and Jernigan.[7] A number of excellent papers have been published to review the many applications of this method[20–22] and to investigate its limitations.[23–27] Recently, Goldstein et al.[28] and Maiorov and Crippen[29] pioneered a new method in which energy parameters are optimized by the condition of stability of the native state. Maiorov and Crippen required that the observed native structures should have the lowest energy among a set of decoys obtained by threading. Our approach[30] and the present work is an extension of theirs; the differences in scope, method, and conclusions are explained below. Goldstein et al. maximized the ratio R between the width of the distribution of the energy and the average energy difference between the native state and the unfolded ones. Such an approach was developed also by other groups.[31–33] Hao and Scheraga[31] introduced the idea of optimizing energy parameters by an iterative procedure that uses at each epoch the current energy parameters to generate a new ensemble of alternative conformations. Mirny and Shakhnovich[32] followed the lesson learned from protein design[34] where a stable sequence of amino acids with a target fold is obtained by minimizing the Z score as a function of the sequence. They expressed the Z score as a function of the energy parameters that were then derived by optimization.

Having decided which representation and which energy function to use, the remaining step is to define the search

Grant sponsor: Minerva Foundation; Grant sponsor: Germany-Israel Science Foundation; Grant sponsor: Israeli Ministry of Science; Grant sponsor: BSF (Israel Binational Foundation).

*Correspondence to: Michele Vendruscolo, Oxford Centre for Molecular Sciences, New Chemistry Laboratory University of Oxford, South Parks Road, OX1 3QT Oxford, UK. E-mail: michelev@bioch.ox.ac.uk

Received 28 June 1999; Accepted 30 August 1999

problem for the native state, namely, how to generate alternative conformations. There are two main procedures to generate alternative conformations: by threading and by energy minimization. In threading[35–37] the native fold of a given sequence is searched in a library of known structures. Gapless threading, which will be used in this work, is a standard tool in deriving and testing energy parameters.[23,26,29,38–41] This straightforward procedure uses the contact map representation. Once the contact maps of the proteins in the database are obtained, we generate decoys for a given sequence of length N from the structures of proteins of lengths N' ($>N$) by selecting submaps of size $N \times N$ along the main diagonal of the contact map of the longer protein. The second way to build decoys is to use Molecular Dynamics simulations starting from the crystallographic structures[42,43] or by extensive Monte Carlo simulations of chains either on lattice[12,31,44] or off lattice.[41,45–47] More recently, decoys were produced by performing moves in the space of contact maps.[18]

The representation of the structure, the selection procedure, and the energy function are interconnected. The aim of this work is to identify the roles played by of each one of these choices in the successes and failures of the particular approach we decided to use.

Similarly to Maiorov and Crippen,[29] we formulate a basic requirement for a protein-folding potential and use it to derive energy parameters. The basic requirement expresses the condition that the energy should attain its lowest value at the *true native map.* The basic requirement can be stated by posing the following question: Is it possible to choose energy parameters so that for all the proteins in the database the native states have the lowest energy among all possible decoys? In this study we adopt the pairwise contact energy approximation and discuss how the answer to the above question depends on the following issues:

1. The definition of contact.
2. The assignment of the contact length R_c.
3. The number M_p of proteins in the database.

Other authors[25,31,32] recently pointed out the deficiency of the pairwise contact approximation. Mirny and Shakhnovich[32] argued that it is unlikely that any choice of parameters could stabilize simultaneously all the proteins in a large database. Their conclusion was based on the observation that their optimization procedure did not yield a low enough value for the Z score averaged over all the proteins. Hao and Scheraga[31] optimized an extended set of energy parameters including pairwise contact interaction to fold one particular protein (crambin); although their optimization procedure was able to produce energy parameters that were good enough to fold crambin within 2 Å from the experimental structure, they were unable to lower the energy of the native state below the continuous density of non-native states. Park and Levitt[25] tested several choices of parameters defined for some simple energy functions, including the pairwise contact approximation. They were also unable to assign the lowest energy to native states in a gapless threading experiment. This conclusion is somewhat in disagreement with the commonly held belief that gapless threading is a poor challenge for empirical energy functions.[23]

Our study *proves* that there is no possible choice of the 210 pairwise contact energy parameters that can solve the general problem of gapless threading. That is, for a large enough value of M_p the basic requirement cannot be met simultaneously by all the proteins in the database. We also show that one can be misled to the opposite conclusions by considering a small database, for a specific definition of contact and for particular values of R_c. This is for example the case of the work of Maiorov and Crippen,[29] who even though they used a different, more elaborate definition of contact, found that it was possible to stabilize simultaneously the native folds of a database of 69 proteins. Even though they used a different, more elaborate definition of contact, which makes direct comparison with our work difficult, we believe that if they had enlarged sufficiently their database, they also would have found that stabilizing all native folds is an impossible task. On the other hand, Mirny and Shakhnovich based their negative conclusions on the fact that a particular set of pairwise contact energy parameters, which was the outcome of their optimization procedure, failed to stabilize the native structures. Our work *proves* that for *any* choice of such energy parameters there are violations of the stability of some native structures.

RESULTS

Representation of Protein Structure

In our work we use the *contact map* representation[13–17,48] of protein structure. The contact map of a protein with N residues is an $N \times N$ matrix $\mathbf{S}$, whose elements are defined as

$$S_{ij} = \begin{cases} 1 & \text{if residues } i \text{ and } j \text{ are in contact.} \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Given all the inter-residue contacts or even a subset of them, it is possible to reconstruct a protein's structure, by means of either distance geometry,[49] Molecular Dynamics,[50] or Monte Carlo.[17]

One can define contact between two residues in different ways. One of the main purposes of this work is to analyze in detail the consequences of this choice. In particular we will analyze two cases:

- Two amino acids are considered in contact when their C_α atoms are closer than some threshold R_c.[17,30] We refer to this convention as the *C_α definition.*
- Two amino acids are in contact when any two heavy atoms (all except hydrogen ones) that belong to the two residues[13,44] are at a the distance lower than R_c (*all atoms definition*).

These two definitions are, in some sense, at the opposite extremes in a scale of detail in the description of the amino acids in terms of atomic coordinates. At an intermediate level we briefly discuss other possible definitions:

- A contact is assumed when any two C atoms of the two amino acids are closer than R_c (*all C definition*).
- Two amino acids are in contact when either the C_α or the C_β atom of the first amino acid is closer than R_c to the C_α or the C_β atom of the second (*C_β definition*).
- For each amino acid, two atoms are selected, the C_α and the heavy atom that is the most distant from the C_α. A contact is assumed when any pair of these atoms (one from each amino acid) is closer than R_c (*distant atom definition*).

For all these choices one has to specify R_c and to study how the results depend on R_c and on the number M_p of proteins in the database.

Generation of the Decoys

The number of decoys that can be generated by gapless threading is limited by M_p, the number of proteins in the database and by their lengths N_i, ($i = 1, \ldots, M_p$). Say we have protein i of length N_i and protein j of length $N_j > N_i$. The total number of conformations that can be generated by threading the sequence of protein i onto the structure of protein j is $N_j - N_i + 1$. Therefore, having ordered the proteins by increasing length, the total number P of decoys that can be generated is given by

$$P = \sum_{j>i}^{M_p} (N_j - N_i + 1) \tag{2}$$

where N_i and N_j are the numbers of amino acids of proteins i and j. One should be aware that the decoys generated by gapless threading are not guaranteed to be physical. This observation is true in general and is not limited to the contact map representation, using any of the definitions of contact given above. Because of the different sizes of the amino acids, a given sequence cannot in general be squeezed onto the structure assumed by another sequence. This problem could in principle be solved by allowing for local structural rearrangements for each decoy.

In this study, we selected five sets of protein structures from the Protein Data Bank (PDB).[51] The PDB is an archive of experimentally derived structures of proteins. To date, 8,295 coordinate entries are deposited. It is known that the information contained in the entire PDB is redundant and some are incorrect. Routine methods are available to select subsets of nonhomologous proteins whose experimental structures are determined reliably.[52–54] In Materials and Methods, we describe in detail the sets of proteins that we have used.

Pairwise Contact Energy Function

Consider a protein of sequence $\mathbf{A} = (a_1, a_2, \ldots a_N)$, where a_i identifies the amino acid species at position i along the chain. Denote by $\mathscr{C}$ a microstate of the system, specified by the coordinates of all the atoms of the proteins and of the water molecules of the solvent. Because many microscopic conformations share the same contact map $\mathbf{S}$, it is appropriate to define a *free energy* $\mathscr{H}(\mathbf{A}, \mathbf{S})$ associated with this sequence and map:

$$\text{Prob}\,(\mathbf{S}) \propto e^{-\mathscr{H}(\mathrm{A,S})} = \sum_{\mathscr{C}} e^{-E(\mathscr{C})/k_BT} \Delta(\mathscr{C},\mathbf{S}) \tag{3}$$

where $\Delta(\mathscr{C}, \mathbf{S}) = 1$ if $\mathbf{S}$ is consistent with $\mathscr{C}$ and $\Delta = 0$ otherwise, i.e., Δ is a "projection operator" that ensures that only those configurations whose contact map is $\mathbf{S}$ contribute to the sum (Eq. 3). For real proteins $E(\mathscr{C})$ is the unknown "true" microscopic energy.

Because it is impossible to evaluate this exact definition of the free energy of a map, we resort to a phenomenological approach, guessing the form of $\mathscr{H}(\mathbf{A}, \mathbf{S})$ that would have been obtained had the sum (Eq. 3) been carried out. The simplest approximation to the true free energy is the pairwise contact approximation

$$E^{pair}(\mathbf{A}, \mathbf{S}, \mathbf{w}) = \sum_{i<j}^{N} \mathbf{S}_{ij} w(a_i, a_j) \tag{4}$$

where $w(a_i, a_j)$ are energy parameters that represent the energy gain when a pair of amino acids is in contact.

It is important to further clarify the meaning of E^{pair}. The approximation E^{pair} to the true free energy $\mathscr{H}(\mathbf{A}, \mathbf{S})$ is not the function used by nature to fold proteins. We speak of $w(a_i, a_j)$ as "pairwise contact energies" for short, their meaning being explained in the discussion above. It should not be forgotten that this is only a possibly useful approximation; as a matter of fact, we shall prove in this work, for several different definitions of contact and for decoys obtained by gapless threading, that there is no set $\mathbf{w}$ that can be used to assign the lowest energy to the native states of all the proteins of a dataset, provided the number of its members, M_p, is large enough.

Using Equation (4) the basic requirement can be cast in the form

$$E^{pair}(\mathbf{A}, \mathbf{S}, \mathbf{w}) < E^{pair}(\mathbf{A}, \mathbf{S}_\mu, \mathbf{w}) \tag{5}$$

where $\mu = 1, \ldots, P$ runs over the entire ensemble of decoys. We ask whether it is possible to choose $\mathbf{w}$ in such a way, that the native contact map of every protein in the database has lower energy than all corresponding decoys. In this case we say that the problem is learnable. We found that if the number of proteins in the data set used for threading is large enough, the answer is *negative,* and the problem is *unlearnable.* We obtained this result by using perceptron learning (see Materials and Methods) to derive the energy parameters or to prove that they do not exist (for large M_p). We note that other techniques, such as linear programming[55] and in particular the simplex method[56] are in principle available to solve this problem (see also the method used by van Mourik and coworkers[57]).

Learning With the All Atoms Definition

The region of learnability

In this section we discuss the dependence of learnability on M_p and R_c. For each definition of contact that we considered we found two "phases." There is a region in the (R_c, M_p) in which the problem is learnable; that is, there is a set **w** of pairwise contact energy parameters that stabilize simultaneously all the native maps in the set. On the other hand, outside this region (e.g., for fixed R_c and large enough M_p) the problem is unlearnable, and no set **w** exists.

Without doing any calculation, we can predict a few general features of the (R_c, M_p) phase diagram. Having set a definition of contact (e.g., the all atoms one), one can plot the distribution of distances between amino acid pairs. Choosing R_c smaller than the smallest observed distance would result in contact maps with no contacts, independently of the conformation. No set of energy parameters can then discriminate the native map from the decoys. Similarly, choosing R_c larger than the largest observed distance would result in contact maps with all the entries set to 1. In this case again no discrimination is possible. Thus, we expect to find a window of learnability in R_c. It is also reasonable to expect that such a window will shrink with increasing M_p. The problem is thus reduced to investigate whether such window remains open for an arbitrary large value of M_p, or it closes for M_p large enough.

The boundaries of the region of learnability must be interpreted in a probabilistic sense. At given M_p and R_c, learnability depends on the particular choice of the proteins in the database. In principle one should define $P(R_c, M_p)$, the probability for a randomly extracted set to be learnable at (R_c, M_p). The boundary is then defined by $P(R_c, M_p) = const.$ In the present study, we chose not to give a precise numerical estimation of $P(R_c, M_p)$, which would require many extractions of sets of M_p proteins from the PDB and would be numerically intensive. We are interested in establishing the existence of the boundary rather than in its precise determination. Hence, we will show that it is possible to find unlearnable data sets when their size is large enough.

The approximate boundaries of the region of learnability are shown in Figure 1. The windows of learnability shrink to zero for M_p above 270. The precise value of the limit of learnability depends on the particular choice of proteins, and the fluctuations in M_p are of the order of a few tens of proteins. Such fluctuations are larger than expected; they are due to a few proteins that are markedly more difficult to learn than others, and their inclusion in the data set lowers the chances of learnability. For example, 1vdfA and 1gotG are two such "hard-to-learn" proteins. 1vdfA is a chain of 46 amino acids that forms a single α helix; 1gotG is a chain formed by 2 α helices hinging at 90°. One might expect that proteins that are not stable without a cofactor or a ligand and perhaps proteins whose structure is derived by nuclear magnetic resonance (NMR) could be more difficult to learn. This interesting issue deserves further study. The answer, however, is not crucial to the present discussion where we are interested in establishing whether the pairwise contact approximation is suitable to stabilize a generic set of proteins against decoys obtained by threading.

Each point shown in Figure 1 is derived from a few (1–3) randomly generated sets of M_p proteins in each. A full circle indicates that all the sets considered were learnable; open circles signal that at least one set was unlearnable. The largest fluctuations are found for small M_p at the right boundary. For example, for different sets with $M_p = 20$ we found that the maximal R_c of learnability varies between 28 Å and 31 Å.

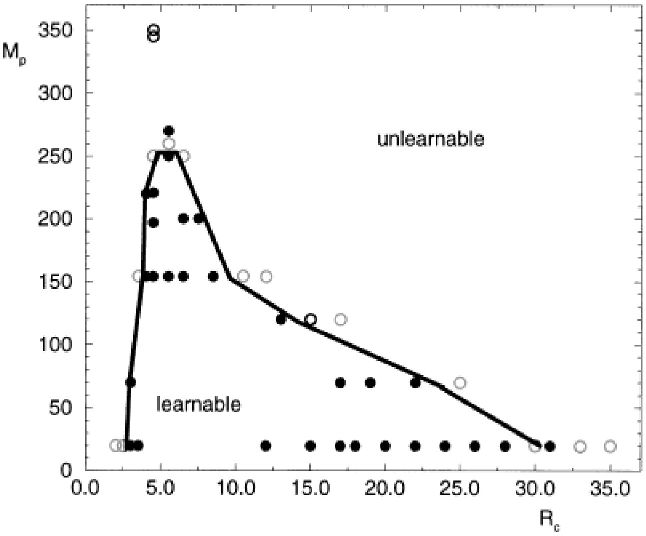

Fig. 1. Region of learnability for the all atoms definition of contact. Several sets of M_p proteins were generated for each value of R_c. Full circles indicate that all the sets considered for a particular value of (R_c, M_p) were learnable; otherwise, we use open circles.

SET_{350} is unlearnable at $R_c = 4.5$ Å. Even removing the 5 proteins that alone are responsible for 86,599 of the total of 125,345 misclassified examples, the problem remains unlearnable. A subset of 250 proteins randomly chosen from SET_{456} is learnable at $R_c = 5.5$ Å, but not at $R_c = 4.5$ and at $R_c = 6.5$ Å. A subset of 221 proteins randomly chosen from SET_{350} is learnable at $R_c = 4.5$ Å.

Correlations between solutions at different M_p and R_c ranged from 0.22 (for a solution at $R_c = 7.5$ Å and $M_p = 200$ with a solution at $R_c = 4.5$ Å and $M_p = 154$) to 0.94, which is the typical correlation between two solutions of maximal stability at $R_c = 4.5$ Å and $M_p > 150$.

These findings indicate that with the all atoms definition of contact and the physically motivated choice of $R_c = 4.5$ Å[13] it is possible to stabilize simultaneously, with high probability, about 200 randomly selected nonhomologous proteins against decoys generated by gapless threading.

Comparison with other potentials

In the "learnable" phase sets of contact potentials that satisfy the basic requirement do exist; our procedure terminates when such a solution is found. We turn now to compare these solutions with contact potentials that were obtained previously by using other methods.

When the problem is solvable, perceptron learning will yield one of a family of solutions (depending on the initial guess, the order in which the examples are presented, etc). Therefore, we first have to decide which of these solutions

TABLE I. Contact Definition Used by the Different Potentials Referred to in This Work

Potential	Acronym	Definition of contact
This work	VND_1	All atoms definition, $R_c = 4.5$ Å
This work	VND_2	All atoms definition, $R_c = 4.5$ Å
This work	VND_3	C_α definition, $R_c = 8.5$ Å
Hinds and Levitt[44]	HL	All atoms definition, $R_c = 4.5$ Å
Mirny and Domany[13]	MD	All atoms definition, $R_c = 4.5$ Å
Mirny and Shakhnovich[32]	MS	All atoms definition, $R_c = 4.5$ Å
Miyazawa and Jernigan[67]	MJ	Centers of mass closer that 6.5 Å
Skolnick et al.[26]	S	All atoms definition, $R_c = 4.5$ Å
Thomas and Dill[68]	TD	C_β definition, amino acid dependent R_c

TABLE II. Results of the Gapless Threading Fold Recognition Experiment

Potential	Misclassified proteins (%) (Total proteins = 218)		Misclassified decoys (%) (Total decoys = 3,518,869)	
VND_1	17	(7.8)	362	0.01
HL	19	(8.7)	7,406	0.2
MD	11	(5.0)	3,773	0.1
MS	19	(8.7)	7,566	0.2
MJ	74	(33.9)	136,870	3.9
S	29	(13.3)	36,097	1.0
TD	20	(9.2)	41,516	1.2

(which constitute the "version space") we choose as our representative set **w**. We rely on the following picture. For the physically motivated value of $R_c = 4.5$ Å, the version space shrinks for increasing M_p. For a learnable case the version space has a non-zero volume. The maximally stable solution is by definition the most distant point from the boundaries of version space; therefore, it is the most likely one to remain a solution for larger M_p. Hence, we chose solution of maximal stability (see Materials and Methods) as our representative. Our first potential (VND_1) was obtained from SET_{197}. From this set, which gives 2,467,150 decoys, 189,697 were selected to obtain a solution of maximal stability.

We considered the seven potentials listed in Table I. To measure the correlation between potentials, the parameters were shifted to have average $\Sigma_c w_c = 0$ and rescaled to have variance $\Sigma_c w_c^2 = 1$. The correlation between different potentials ranged from 0.56 (MJ-S) to 0.91 (HL-MS). The correlations between VND_1 and other potentials are all around 0.6. To compare the performances of these potentials, we selected a subset of 218 proteins from SET_{456} and used all seven to classify the decoys. The test set is nonoverlapping with the training set used to derive VND_1 (two other sets, VND_2 and VND_3, will be introduced below). Results are given in Table II. Apart from MJ, which uses a different definition of contact, the number of misclassified proteins is comparable for all the potentials used. What is perhaps more interesting is that the number of misclassified decoys is remarkably lower for our potential.

We mention here in passing that our potential gave, as expected, largely negative results on a fold recognition test on a standard benchmark.[58] The test requires threading 68 sequences on the structures of a nonredundant set of 301 proteins to identify the most compatible fold for each sequence. It is known that when proteins are distant homologous, as in the case of the benchmark, insertions and deletions are always present, and a gapless sequence-structure alignment is unsuitable. In a recent work, Mirny and Shakhnovich[59] showed that a subtle problem undermines threading as a method to perform protein fold predictions. In protein folding one needs an energy function that should single out the native state among all possible folds. In threading one searches for an approximate nativelike fold in a library of folds, and the energy function should discriminate between such an approximate fold and the ensemble of decoys. Mirny and Shakhnovich gave evidence that optimizing an energy function for threading is more difficult than for protein folding, because the energy gap above the true native state is larger than the energy gap above the approximate guess for the native state, which is the result obtained by threading. In the present study we proved that by using the pairwise contact approximation and decoys obtained from gapless threading it is not possible to create an energy gap above the native state for all the proteins in a database. In other words, at least for the cases presented in this work, the pairwise contact energy approximation is not suited to unmistakingly single out the native state of a protein among decoys obtained by threading. The chances to use such energy to identify an approximate fold are argued to be lower, and our failure on the benchmark is consistent with such expectations.

Learning With Other Definitions of Contacts

C *definition*

In this section we investigate the effect of changing the definition of contact on the conclusion about learnability of a given set of proteins. We found that using the C_α definition SET_{154} is unlearnable.

Which is then the maximal number of proteins of SET_{154} that can be stabilized together by using the C_α definition? By eliminating the 13 "worst" proteins that are responsible for 69,227 of the 78,866 misclassified examples, we obtained a subset of 141 proteins in SET_{154} that is learnable in the narrow window $11 < R_c < 12$. Next, we selected a subset of 123 proteins by eliminating the 28 worst proteins (that are responsible for 78,147 misclassified examples) and the 3 ones that are shorter than 46. In this case the window of learnability became larger—it extended to $7.5 < R_c < 15$.

The region of learnability in the (R_c, M_p) plane is shown in Figure 2. Although the upper part of the boundary was

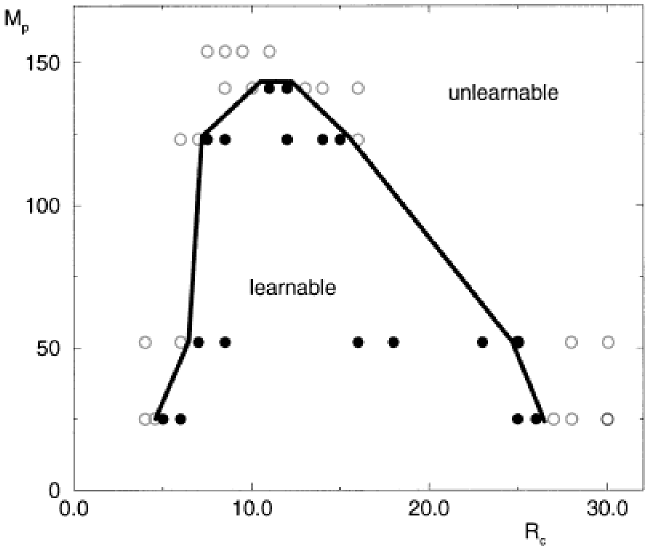

Fig. 2. Region of learnability of the C_α definition.

not derived by random selection of proteins (as explained above), we can argue that a random selection of M_p nonhomologous proteins has a good chance to be learnable if $M_p \sim 100$.

We now compare the C_α and the all atoms definitions of contact. We consider the extension of the region of learnability and the performance in fold self-recognition experiments. The region of learnability of the C_α definition is smaller than the one of the all atoms definition. To perform a comparison between the self-recognition performances of the two definitions, we optimized two potentials on the same training set and tested them on the same test set. We took a learnable subset of 123 proteins from SET_{154} and derived a solution of maximal stability (VND_3) by using the C_α definition at R_c = 8.5 Å. We also derived a solution of maximal stability (VND_2) by using the all atoms definition and R_c = 4.5 Å. Next, we extracted a subset of 100 proteins from SET_{456} and by using the C_α definition and R_c = 8.5 Å obtained an ensemble of decoys for each by gapless threading. Fold identification results are given in Table III. Then the native maps and decoys were also obtained using the all atoms definition of contact with R_c = 4.5 Å on the same test set. The results for these decoys are given in Table IV.

It is possible to make some observations on these results, a few of which are rather surprising.

1. The all atoms definition of contact gives rise to better performance overall than the C_α definition.
2. VND_3, which was optimized for a C_α definition, gives better results when used with an all atoms definition. Moreover, VND_1, which was optimized for an all atom definition, misclassifies less decoys than VND_3, even when tested on maps and decoys that were derived with a C_α definition.
3. The number of proteins in the training set influences the results; VND_1, which was trained on a larger set, performs better than VND_2.

TABLE III. Results of the Gapless Threading Fold Recognition Experiment[†]

Potential	Misclassified proteins (%) (Total proteins = 100)	Misclassified decoys (%) (Total decoys = 683,415)	
VND_1	48	14,228	(2.1)
VND_2	53	26,491	(3.9)
VND_3	33	16,097	(2.4)
HL	51	33,632	(4.9)
MD	42	19,313	(2.8)
MS	55	33,615	(4.9)
MJ	90	87,318	(12.7)
S	67	78,432	(11.5)
TD	32	27,415	(4.0)

[†] C_α definition of contact, R_c = 8.5 Å.

TABLE IV. Results of the Gapless Threading Fold Recognition Experiment[†]

Potential	Misclassified proteins (Total proteins = 100)	Misclassified decoys (Total decoys = 683,415)	
VND_1	7	43	(0.6)
VND_2	5	553	(8.1)
VND_3	12	6,205	(90.8)
HL	8	2,264	(33.1)
MD	4	1,562	(22.8)
MS	4	1,804	(26.4)
MJ	32	32,838	(473.8)
S	11	15,538	(227.4)
TD	9	17,981	(263.1)

[†] All atoms definition of contact. R_c = 4.5 Å. Last column is the number of misclassified decoys per ten thousand.

In the present study we do not attempt to explain why the C_α definition has worse performance than the all atom definition. Possible directions are to investigate statistics on amino acids pairs, information carried by amino acid pairs and correlation functions of pairwise contacts. Here we give only an indication, based on the distribution of the vectors of decoys. We computed the vector of the center of mass of all the decoys obtained from SET_{154} both for the C_α and for the all atoms definitions. We then considered the angle τ between the center of mass vector $\mathbf{x}_{\text{cm}}$ and the vector $\mathbf{x}_\mu$ of each decoy (therefore $\cos \tau = \mathbf{x}_{\text{cm}} \cdot \mathbf{x}_\mu$). In Figure 3 we show the histogram of the number of decoys with a given $\cos \tau$ for both definitions. The distribution for the C_α definition has a peak that is closer to zero than the distribution for the all atoms definition. The C_α decoys are on average "more orthogonal" to the center of mass than the all atoms decoys. Moreover, there is a longer negative tail for the C_α decoys. We argue that the vectors of the C_α decoys are more spread in the 210 dimensional space of examples. Hence, the problem is less likely to be linearly separable. The analysis of the covariance matrix of the set of decoys is consistent with this picture and reveals that the largest eigenvalue is larger for the C_α case than for the all atoms one.

Which decoys are more challenging? It is instructive to consider the overlap Q between the contact map of the native state and of the decoys, which is defined as

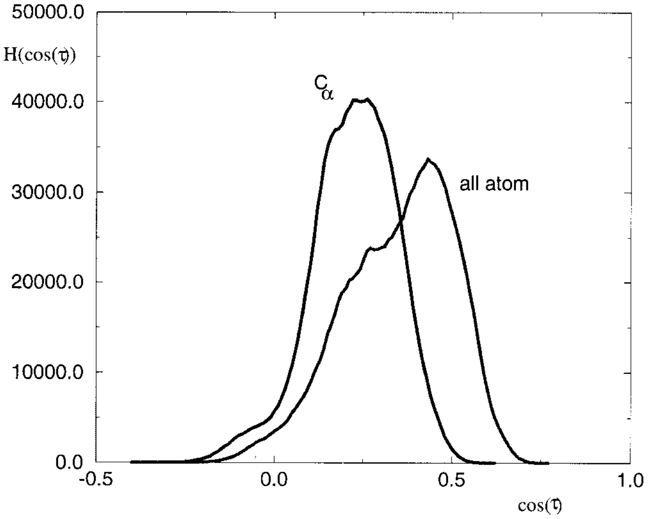

Fig. 3. Distribution of the directions of the vectors of the decoys. The scalar product with the center of mass vector is taken for all the examples. The peak in the C_α distribution is closer to zero than the one in the all atoms distribution. Moreover, there is a longer tail extending into the negative side.

$$Q = \frac{1}{\max(N_c^0, N_c^\mu)} \sum_{h>k+1}^{N} \mathbf{S}^0_{hk}\mathbf{S}^\mu_{hk} \qquad (6)$$

where N is the length of protein, and N_c^0 and N_c^μ are the number of contacts in $\mathbf{S}^0$ and $\mathbf{S}^\mu$, respectively. We considered, for the C_α definition and $R_c = 8.5$ Å, two sets of proteins. The first, of 123 proteins, is learnable and the second, of 141 proteins, is unlearnable. First, for each decoy we calculated, using VND_1 as initial weights, the energy difference ΔE with the native state and the overlap Q. In Figure 4a and b we present scatter plots of ΔE on Q for our decoys. The first question we asked is whether decoys of high overlap are the more challenging ones. To answer this question we repeated the learning procedure by considering only decoys with $Q < Q_t$, where Q_t is a threshold value for the overlap. The set of 141 proteins was still unlearnable for $Q_t = 0.6$. It seems that the "difficult" decoys are spread over the entire range of Q. Including all the decoys in the learning procedure we were able to learn the set of 123 proteins, but not the set of 141 proteins. As shown in Figure 4c, after learning the set of 123 proteins, decoys of low energy are present in the approximate range $0.2 < Q < 0.8$. The important finding is that the unlearnable case is not qualitatively different (see Fig. 4d). Also in this case decoys of low energy are present in the range $0.2 < Q < 0.8$, although now some of them have $\Delta E < 0$. The difference is that with the set of 123 proteins there are 805,938 decoys, whereas for the set of 141 proteins, 1,071,753. With 210 energy parameters it is possible to satisfy the smaller set of inequalities but not the larger. Decoys of arbitrary Q enter in the learning process, and one can argue that the lack of correlation between ΔE and Q is a major cause that renders the problem unlearnable for large enough M_p and, further, that an improved energy function should first of all provide such a correlation.

Other definitions of contact

In this section we explore other definitions of contact. We found that SET_{154} is unlearnable using the "C_β" definition and the "distant atom" definition. We did not investigate further these two definitions that do not seem to add much phenomenological information.

More interesting is the case of the "all *C*" definition. We used subsets extracted from SET_{350}, SET_{456}, and SET_{945}. We determined numerically the extension of the learnable region in the (R_c, M_p) plane in an approximate way, as shown in Figure 5. We proved the existence of a boundary at $R_c = 4.5$ Å by showing that a subset of 420 proteins of SET_{456} is learnable and that SET_{945} is unlearnable. For R_c between 4 and 20 Å, the location of the boundary is only tentative.

Detailed Analysis of a Learnable Case

In this section we analyze in detail the solutions in a learnable case, SET_{154} using the all atoms definition and $R_c = 4.5$ Å.

Learning subtasks

We considered several learning tasks of increasing difficulty. First, we searched for a set **w** to stabilize the native fold of a single protein against decoys obtained by threading it onto all the longer proteins in SET_{154}. Next, we investigated whether learning time is affected by the fold family as obtained by the CATH classification. In Table V we present the number of sweeps needed to learn each of the different sets of examples used in this work using the perceptron algorithm (see Materials and Methods). Finally, we learned the full set of 154 proteins. From SET_{154} we obtained by gapless threading $P = 2{,}497{,}334$ decoys. This number is doubled with respect to that given by Equation (2) because in this case we also thread backward. We invert the sequence of amino acids of protein i and thread it again into the structure of protein j. This procedure is valid because in the contact map representation there is no difference between the N-terminus or the C-terminus of the protein chain. That is, unlike real protein chains, there is no directionality in the contact map. The difficulty in learning a set of examples can be measured by the number of perceptron iterations needed to learn. This measure is independent of the number of examples and can be used to compare different proteins. Typically, one cycle was sufficient to learn independently any single protein, whereas when all proteins in the database were learned together, on the average 30 cycles were necessary. No clear size dependence is observed; long chains are as easily learned as short ones.

Learning different classes of protein structure

In this section we estimate the influence on the derived potential of the protein structures that are included in the training database. We derived three different potentials by using only proteins from one particular CATH class (mainly-α, mainly-β, and α-β). These three potential were then tested on the proteins from the other two classes. Results are summarized in Table VI. For example, the

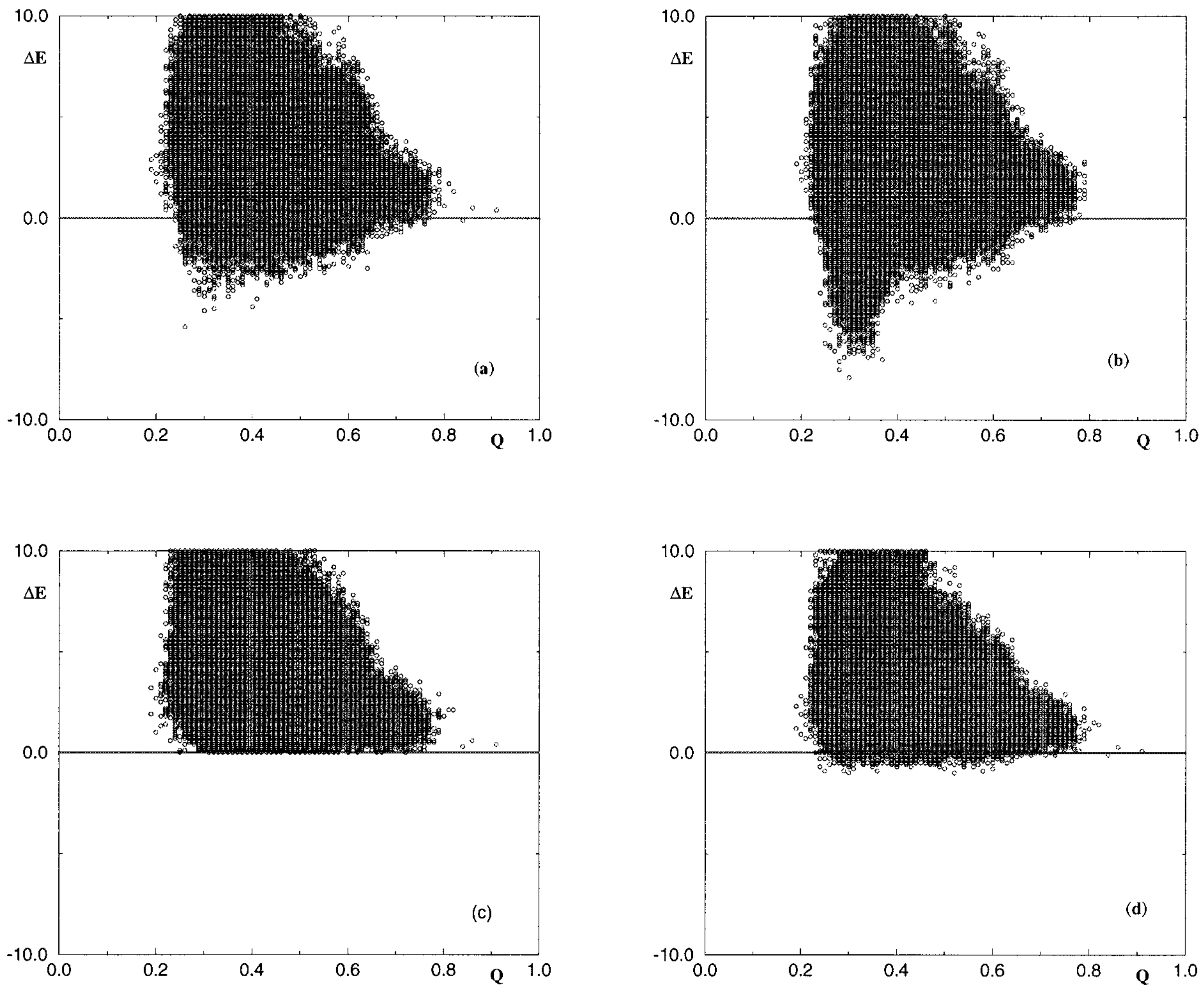

Fig. 4. Scatter plot of the energy difference *E* between decoys and native state and the overlap *Q* with the native state. **a:** Learnable set of 123 proteins using the solution VND_1. **b:** Unlearnable set of 141 proteins with the same initial set of energy parameters. **c:** The set of 123 proteins with the energy parameters obtained from learning. **d:** The set of 141 proteins with the energy parameters arrived at when unlearnability has been established.

α-potential, obtained by learning the 42 mainly-α proteins, was tested on 82 proteins of the other two classes. The native fold of 63 of these was identified as the map of lowest energy, and 19 proteins were misclassified. The failure rate of this experiment was close to the one in which the potential was obtained by learning 27 mainly-β proteins. On the other hand, the fraction of misclassified proteins was significantly lower when the $\alpha - \beta$ potential was tested on the other two classes. We found evidence that learning on proteins that contain both kinds of secondary structure elements yields a potential whose fold recognition capability is better than that of potentials trained on a single kind of secondary structure.

Statistical analysis of the version space

The solution to the learning problem is a 210-components vector **w** of pairwise contact energies. In a learnable case the solution is not unique, and the ensemble of solutions forms the "version space." In this section we study the size and shape of the version

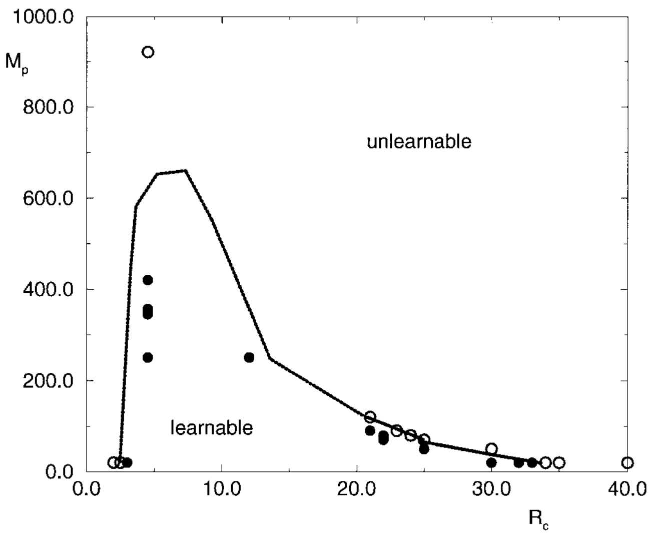

Fig. 5. Region of learnability of the "all *C*" definition.

TABLE V. Learning Time for Different Sets of Examples Used in This Study

Task	No. of proteins	No. of decoys	Learning time (sweeps)
1 protein	1	~15,000 (on average)	~1
beta	26	70,958	0.72
alpha	41	144,380	5.22
alpha-beta	55	328,144	1.63
whole database	153	2,497,334	~30

TABLE VI. Results of the Gapless Threading Fold Recognition Experiment†

Class potential	No. of proteins in test set/ No. of decoys in test set	Misclassified proteins/ misclassified decoys	Percentage of misclassification (%)
α-potential	82	19	23
	713,182	12,617	2
β-potential	97	21	22
	931,222	27,660	3
α-β-potential	68	9	13
	422,576	11,390	3

†Potentials derived by learning a single class of proteins were tested on decoys generated by threading (using only proteins from the other two classes).

space. Directions in which the extension of version space is small define combinations of the contact energies that are determined narrowly by our learning procedure. On the other hand, if there is considerable width of version space in a certain direction, this means that the corresponding linear combination of contact parameters is only weakly determined by our database and training set.

A straightforward way to sample the version space is to repeat the learning procedure several times. A more efficient way to obtain the same result is to use the following Monte Carlo procedure:

- At $t = 0$, initialize the weights as $\mathbf{w}(0) = \mathbf{w}^*$, where $\mathbf{w}^*$ is a solution previously obtained by learning.
- At time step t, set $\mathbf{w}' = \mathbf{w}(t) + v$, where $\mathbf{v}$ is a vector whose components are all zero except the i-th (chosen at random) which is a random number in $[-\epsilon, \epsilon]$.
- Set $\mathbf{w}(t + 1) = \mathbf{w}'/|\mathbf{w}'|$ if $\mathbf{w}'$ is a solution, otherwise $\mathbf{w}(t + 1) = \mathbf{w}(t)$.
- The vector $\mathbf{w}(t)$ is stored after every τ accepted steps, where τ is the autocorrelation time.

We store one vector only every τ acceptances to eliminate correlations. The autocorrelation time τ is estimated from the autocorrelation function $\rho(t)$ of $\mathbf{w}(t)$

$$\rho\Big(t) = \frac{\langle \mathbf{w}(s) \cdot \mathbf{w}(s + t)\rangle - \langle \mathbf{w}\rangle^2}{\langle \mathbf{w}^2\rangle - \langle \mathbf{w}\rangle^2} \sim e^{-t/\tau} \tag{7}$$

where the average $\langle \cdot \rangle$ is performed over s. In this way we obtained $\tau = 2{,}000$.

A practical way to save computer time is to select a relevant subset of decoys, as explained in Materials and Methods. In this case an additional step is required. The ensemble of solutions is screened to discard all those that fail to solve the entire training set. We generated a large number of solutions by using different values of ϵ and of initial weights; the ensemble that we studied and discuss below contains about 100 solutions.

As a first measure of the size of version space we calculated all the scalar products $\mathbf{w}(t) \cdot \mathbf{w}(s)$. The histogram of these scalar products, presented in Figure 6, is narrow and centered at 0.96. This indicates that there is not much freedom in choosing the values of the components of the weight vectors; solutions are fairly close to one another. The peak at 1.00 is an artifact of our Monte Carlo procedure. We have verified that the area of this peak decreases for increasing τ.

To further investigate the shape of version space, we obtained the covariance matrix C of our ensemble of independent solutions. The element C_{ij} of the covariance matrix is defined as

$$C_{ij} = \langle (w_i - \langle w_i \rangle)(w_j - \langle w_j \rangle) \rangle \tag{8}$$

where $\langle \cdot \rangle$ denotes the average taken over the ensemble of solutions of the components w_i and w_j of the vector of weights. In Figure 7 we present the normalized eigenvalues $\lambda*_i = \lambda_i/\lambda_1$ of the covariance matrix sorted in decreasing size. In all but a few directions, the version space is very tight.

DISCUSSION

The basic requirement that a free energy function should satisfy is that the native structure has the lowest energy among a set of decoys. In this study we focused our attention to the pairwise contact approximation. We presented a procedure to establish if there is a set of 210 contact energy parameters so that the basic requirement is satisfied for all the proteins in a database, when the decoys are obtained by threading.

The main result presented in this article is that the existence of such a set of energy parameters depends on several factors:

- The definition of contact.
- The assignment of the contact length R_c.
- The number M_p of proteins that are stabilized together.

In addition to these factors, learning is also influenced by the way in which decoys are obtained. We have shown in a separate publication[30] that if decoys are generated by energy minimization, it is not possible to satisfy the basic requirement even for one single protein. In that work[30] we used a particular definition of contact, namely when two C_α atoms are closer than $R_c = 8.5$ Å. This particular choice limited the generality of the conclusion of Vendruscolo and Domany[30] and motivated the present study.

Here we analyzed several different definitions of contact and several assignment of R_c for each one. We choose to generate decoys by gapless threading because it is an

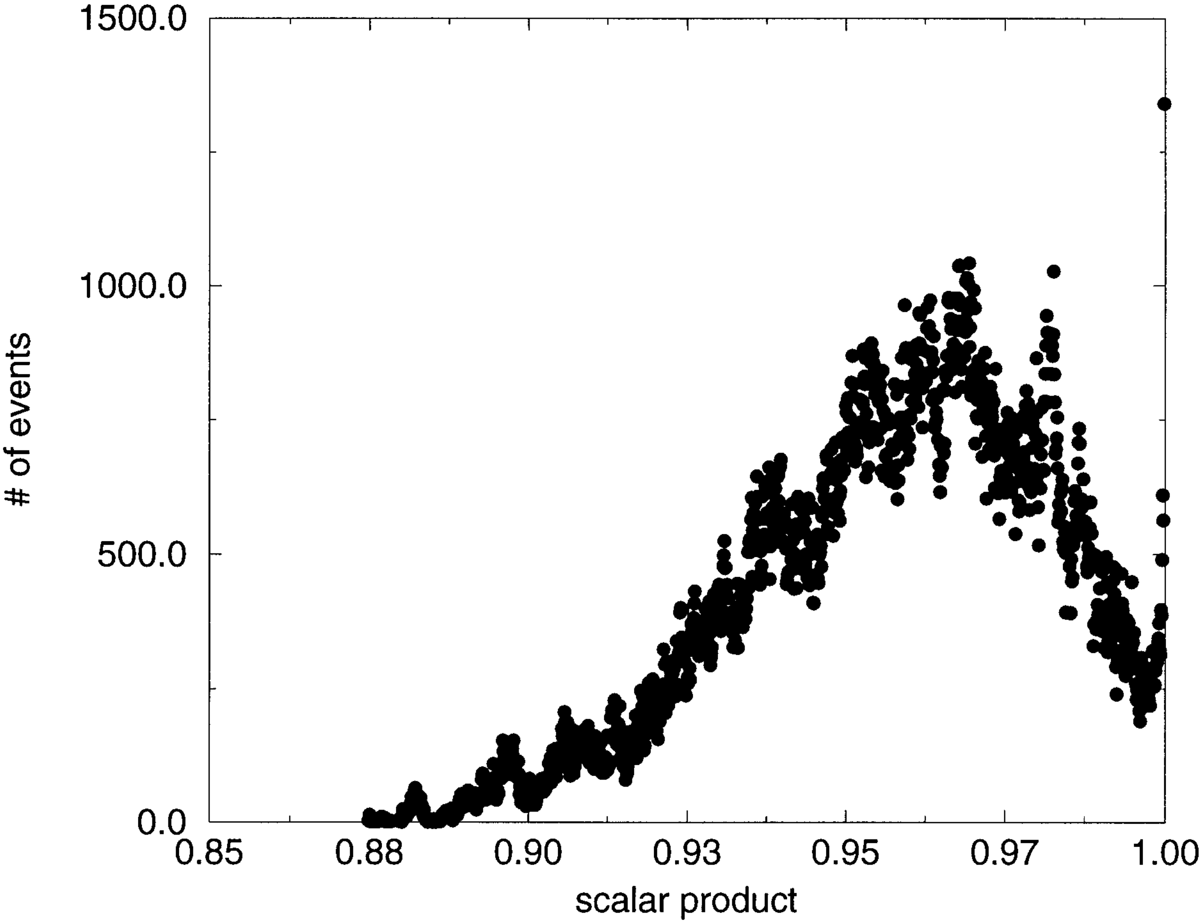

Fig. 6. Histogram of the scalar products between different solutions.

efficient way to obtain decoys, but our conclusion is not limited to threading and could only be reinforced by using more refined way to generate decoys, as shown by Vendruscolo and Domany.[30]

In this article we showed that, no matter which definition of contact and which R_c are chosen, there is always a maximal number (of the order of a few hundreds) of proteins that can be stabilized together. Is this number large or small? A definite answer is outside the limits of this study. It is small compared with the total number of proteins existing in nature (hundreds of thousands), but it is fairly large compared with the number of representative proteins in the PDB (also of the order of a few hundreds).[53]

Can we argue from our results that the C_α representation of the structure is "worse" than the all atoms one? The answer is negative. First, we restate our findings. With the all atoms definition of contact it is possible to accomplish a more demanding learning task and consequently that it is possible to stabilize the proteins in a larger database. We point out, however, that this statement involves both the representation of the structure and the approximation for the energy. We can conclude that the C_α representation of the structure is worse for the particular choice of the approximation of the energy that we have considered in this work. It is possible that an approximation of the energy beyond the pairwise contact one could capitalize better on the simplicity of the C_α representation of the structure.

It is important to study in which way protein structures are nonrandom and which representation of the structure is more able to capture such information. The aim is to understand which features of protein structure are more relevant and must be built in in the approximation of the energy. Our study constitutes a first step on this problem, and the precise message that must be taken is that *structural and energetic considerations based on amino acid pairs alone are unsuitable for protein folding and for fold recognition.*

MATERIALS AND METHODS

Protein Databases

First database: SET$_{154}$

The first set (SET$_{154}$) of proteins we used is selected from the list of 312 proteins (R-factor < 0.21 and Resolution < 2.0; list created from the PDB on July 23, 1997) as obtained by WHATCHECK,[54] which adopts the following criteria[60] (a) The keyword MUTANT does not appear in the COMPOUND name; (b) The structure is solved by X-ray crystallography; (c) The resolution is better than 2.0

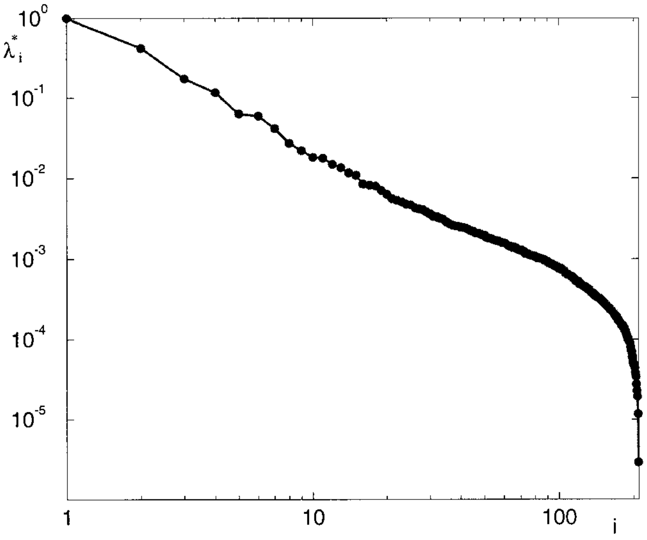

Fig. 7. Normalized eigenvalues $\lambda*_i$ of the covariance matrix ordered by size.

TABLE VII. Database of the 154 Protein Structures of SET_{154} With Their CATH Class Classification (Orengo et al., 1997)†

CATH class	Total	PDB code
Mainly α-helix	42	1351, 1531, 1aru, 1axn, 1cmb A, 1cot, 1cpc A, 1cpc B, 1csh, 1ctj, 1flp, 1hyp, 1mba, 1mbd, 1osa, 1pnk A, 1rro, 1thb A, 1xik A, 1ycc, 2abk, 2cyp, 2end, 2gdm, 2hbg, 2wrp R, 3sdh A, 451c, 4icb, 256b A, 2ccy A, 2spc A, 1bbh A, 1cpq, 1lis, 1mzm, 1poa, 1vls, 1htr P, 1lts C, 1rop A, 1lmb 3.
Mainly β-sheet	27	1ext A, 3ebx, 1cka A, 1ova A, 1arb, 1hbq, 1hyl A, 1ida A, 1ifc, 1lid, 4fgf, 1kap P, 1lop A, 1prn, 1vqb, 2cpl, 2por, 1amm, 1bdo, 1slt A, 1ten, 1tta A, 2rhe, 1gpr, 1xso A, 2aza A, 2bbk L.
α-β	56	1aay A, 1brn L, 1cyo, 1doi, 1fkj, 1frd, 1hpi, 1igd, 1mml, 1onc, 1ubi, 1frb, 1nfp, 1tml, 1tph 1, 2mnr, 1bhp, 1bur S, 1cse I, 1daa A, 1fxd, 1iro, 1kpt A, 1npk, 1otf A, 1ptf, 1ptx, 1puc, 1tgs I, 2bop A, 2chs A, 2nll A, 2phy, 2pii, 3cla, 1pot, 1rcf, 9wga A, 1abe, 1atl A, 1cus, 1dad, 1gca, 1gd1 O, 1iae, 1lkk A, 1mrj, 1pdo, 1sbp, 1tad A, 2abh, 2dri, 5p21, 1reg X, 1wad, 2cy3.
Few secondary structure	2	2psp A, 2ltn B.
Not classified	27	1beb A, 1cei, 1cyd A, 1dut A, 1dxy, 1edm B, 1ept A, 1ept B, 1fle I, 1gnd, 1hrd A, 1kuh, 1kve A, 1kve B, 1lbu, 1mai, 1nox, 1onr A, 1pmi, 1rie, 1tfe, 1whi, 1yas A, 2arc A, 2cbp, 2mbr, 2tsc A.

†The last row presents the entries which were not classified on CATH or were still under processing.

Å; (d) The R factor is lower than 0.21; (e) Chains with "abnormal features" were excluded; (f) No more than 749 or less than 32 amino acids; (g) No more than 30% sequence identity; (h) No more than 1 chain break; (i) No more than two C_α-only residues. This list was further reduced to 154 proteins, by removing proteins according to the following criteria:

- C_α distance between consecutive residues outside the interval of 4 SD ς from the mean d ($d = 3.81$, $\varsigma = 0.05$). In this way we remove the chains with CIS-peptide bonds (including cis-PRO) or backbone chain breaks.
- Any residue (ASX, GLX, UNK, ACE, PCA, etc.) that does not match the 20 standard amino acid names (in the case where the first and/or last residues are undefined, the residues were removed, not the protein).
- Any chain for which the C_α or the backbone-N atoms' coordinates are not present.
- Any unexplained mismatch between the sequence of amino acids presented in SEQRES and the actual sequence appearing in the coordinates section.
- In case of multiple locations of amino acids we keep the protein and consider only the location for which altloc = "A."

The three-letter PDB code (with a chain identifier where existent) as well as the CATH classification of the 154 protein chains of the set are presented in Table VII.

To get a representative database of protein structures, the structural similarity among the proteins that are included must be as small as possible. We followed the commonly used practice, excluding proteins whose sequence identity with any one of the set's members exceeds some threshold. To keep track of the degree of remaining structural similarity, we made a note of the CATH classification of protein domains.[61] For the 126 chains of SET_{154} that have a CATH code, comparison of the first domain shows that the proteins of SET_{154} belong to one of 21 groups, with at least two proteins in each group, for which the CATH classification codes are identical. These 21 groups comprise a total of 49 chains. Hence, 28 chains could in principle be removed, leaving only one representative from each group. The 21 groups are the following: {1abe, 1gca, 1gd1 O, 1tad A, 2dri}; {1flp, 1mbd, 2gdm, 2hbg, 3sdh A}; {1pot, 1sbp, 2abh}; {1arb, 1hyl A}; {1aru, 2cyp}; {1cot, 1ctj}; {1ifc, 1lid}; {1mba, 1thb}; {1tgs I, 9wga A}; {1wad, 2cy3}; {1atl A, 1iae}; {1bbh A, 2ccy A}; {1bdo, 2bbk L}; {1cka A, 1ptx}; {1cpc A, 1cpc B}; {1cyo, 2ltn B}; {1iro, 1otf A}; {1prn, 2por}; {1rro, 4icb}; {1ycc, 451c}; {2gdm, 2hbg}.

This structural analysis was performed by comparing only the first domain of each protein chain; some proteins may have other domains that were not taken into account.

Because two different protein chains might have similar structures on their first domains but have completely different structures on other domains, it is questionable whether in such a case one should classify the two chains as similar (as we do here). With this caveat we can say that 28 chains have an "important level" of structural similarity with the remaining 21 chains that represent each group.

With this list we generated $P = 1{,}248{,}667$ decoys.

Second database: SET$_{197}$

We chose a second, larger set of proteins (SET$_{197}$) according to the Obstruct criteria.[52] The list has 197 entries taken from the 203 of Obstruct Prerun 3, which is compiled considering proteins with <25% sequence similarity. We further restricted this list by the additional criteria listed above and removing all proteins of length $N < 46$. With this list we generated $P = 2{,}467{,}150$ decoys.

Third database: SET$_{350}$

The third set of proteins (SET$_{350}$) was again chosen according to the Obstruct criteria.[52] The list has 350 entries taken from the 455 of Obstruct Prerun 7, which is compiled considering proteins with <40% sequence similarity. We restricted this list by the additional criteria listed above and removing all proteins of length $N < 46$. With this list we generated $P = 6{,}719{,}783$ decoys.

Fourth database: SET$_{456}$

The fourth set of proteins (SET$_{456}$) contains 456 proteins, selected from the WHATIF database with 478 PDB chains, created from the PDB on April 17, 1998[54] again by removing proteins of length $N < 46$. With this list we generated $P = 15{,}112{,}792$ decoys.

Fifth database: SET$_{945}$

Our largest set of proteins, (SET$_{945}$) contains the 945 representative protein chains listed in the most recent release of PDB_SELECT (the list can be downloaded from ftp://ftp.embl-heidelberg.de/pub/databases/protein_extras/pdb_select/recent pdb_select/).

We refer to the original paper of Hobohm and Sander[53] for the detailed criteria of selection. (The detailed lists of all the protein used are available from the authors on request.)

The Perceptron Learning Algorithm

The perceptron is the simplest neural network. In this section we give a practical notion of what a perceptron is and how it works. For more exhaustive treatments, we refer to existing excellent reviews[62,63] about the vast literature on the statistical mechanics of perceptron learning.

A perceptron is an algorithm that associates a binary output (answer) to any input ("example") that is presented. Hence, the perceptron is a binary classifier that assigns every possible input to one of two classes. The example is in the form of an n-component vector, $\mathbf{x}$, whose components quantify in mathematical terms qualities of the object to be classified. The value of the answer produced for a given example depends on the parameters (weights) of the perceptron. The weights (sometimes referred to as *synapses*) are the components of another vector, $\mathbf{w}$. The output f, corresponding to $\mathbf{x}$, depends on h, the scalar product between $\mathbf{x}$ and $\mathbf{w}$

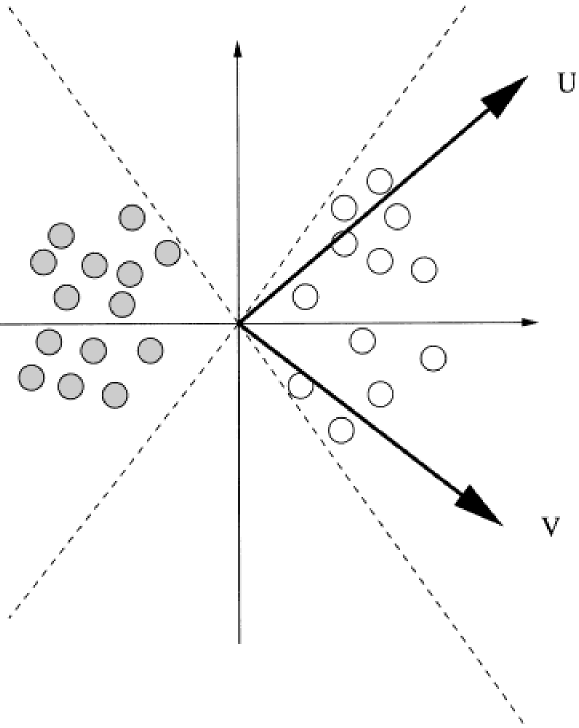

Fig. 8. Graphical representation of the learning process in the case $n = 2$.

$$h = \mathbf{x} \cdot \mathbf{w} = \sum_{i=1}^{n} x_i w_i \tag{9}$$

and is given by

$$f(h) = \begin{cases} 1 & \text{if } h \geq 0 \\ 0 & \text{if } h < 0 \end{cases} . \tag{10}$$

A typical problem that can be solved by using a perceptron is to find a set of weights $\mathbf{w}$ from the knowledge of the correct output for each one of the P examples in a given set (the "training set"). When a vector $\mathbf{w}$ exists, the problem is said to be "linearly separable." In geometrical terms, the goal is to find a $(n - 1)$-dimensional hyperplane (whose normal is the n-components weight vector) that separates the examples (points in this hyperspace) into two subspaces, so that each subspace contains only examples that give rise to the same output. This is shown in two dimensions in Figure 8, where the different circles denote the different examples, to be classified according to their "color." Any weight vector that lies between the vectors $\mathbf{u}$ and $\mathbf{v}$ is a solution. In the case of presenting only one kind of examples, which is the case considered in this work, the desired hyperplane is such that all examples are located on one side.

Such solutions are determined in the course of a learning session; examples from the training set are presented sequentially and each time the perceptron produces an output by using Equations (9) and (10), it is compared to the known correct answer. In case of agreement we move to the next example; otherwise, the weights are changed to correct the mistake. (In general the output $f(h)$ could be any real number but in a boolean perceptron, to classify an example this real number must be mapped onto a binary value, for example, by using $f(h) = sign(h)$.) We cycle through the training set again and again. To update the weights, we use here the simplest perceptron learning rule,

$$\mathbf{w}^{new} = \begin{cases} (\mathbf{w} + \eta\mathbf{x})/\,|\,\mathbf{w} + \eta\mathbf{x}\,| & \text{if } h \geq 0 \\ \mathbf{w} & \text{if } h < 0 \end{cases} \tag{11}$$

or in short, $\mathbf{w}^{new} = \mathbf{w} + f(h)\eta\mathbf{x}$ where η is an arbitrary parameter. In the case of existence of a solution a convergence theorem guarantees that in a finite number of training steps the perceptron will find it.[64] Such a solution will be reached irrespective of the value of η and of the way in which the weight vector is initialized. In general, for different initializations the perceptron reaches different points in the space of solutions (called the *version space*). The perceptron learning rule generates a walk in the space of weight vectors, with a bias in the direction of the version space. As soon as this walk penetrates version space, the procedure stops; the task is learned. Thus, the simple learning procedure usually terminates at a solution that lies near the boundary of version space. In the example of Figure 8, these boundaries are defined by the vectors **u** and **v**. If the set of examples is not linearly separable, there is no solution to the learning problem; such a training set is unlearnable. In the geometrical picture of Figure 8 this means that the circles of different color are mixed in such a way that no straight line can separate them. In such a case the perceptron learning algorithm described above will keep running indefinitely.

Equation (5) can be expressed as

$$\mathbf{w} \cdot \mathbf{x}_\mu > 0. \tag{12}$$

To see this, note that for any map $\mathbf{S}_\mu$ the energy Equation (4) is a linear function of the 210 contact energies that can appear and it can be written as

$$E(\mathbf{A}, \mathbf{S}_\mu, \mathbf{w}) = \sum_{c=1}^{210} N_c(\mathbf{S}_\mu)w_c. \tag{13}$$

Here the index $c = 1, 2, \ldots 210$ labels the different contacts that can appear and $N_c(\mathbf{S}_\mu)$ is the total number of contacts of type c that actually appear in map $\mathbf{S}_\mu$. The difference between the energy of this map and the native one is therefore

$$\Delta E_\mu = \sum_{c=1}^{210} x_c^\mu w_c = \mathbf{w} \cdot \mathbf{x}_\mu \tag{14}$$

where we used the notation

$$x_c^\mu = N_c(\mathbf{S}_\mu) - N_c(\mathbf{S}_0) \tag{15}$$

and $\mathbf{S}_0$ is the native map. We denote the normalization factor of the vector $\mathbf{x}^\mu$ by

$$X_\mu = \sum_{c=1}^{210} (x_c^\mu)^2 \tag{16}$$

and from here on we set $x_c^\mu = [N_c(\mathbf{S}_\mu) - N_c(\mathbf{S}_0)]/X_\mu^{1/2}$, so that both **x** and **w** are normalized to 1.

The Nabutovsky-Domany algorithm

In this section we present the algorithm we used, based on the one introduced by Nabutovsky and Domany.[65] The main advantage of this algorithm is that it allows to rigorously prove the absence of a solution in case of a unlearnable task.

Each candidate map $\mathbf{S}_\mu$ is represented by a vector $\mathbf{x}_\mu$ and hence the question raised in the introduction becomes whether one can find a vector **w** so that condition (12) holds for all $\mathbf{x}_\mu$?

We take a learning step when

$$h_\mu = \mathbf{w} \cdot \mathbf{x}_\mu < 0. \tag{17}$$

The learning step consists of updating **w** according to Equation (11) and η and d in the following way:

$$d^{new} = \frac{d + \eta}{\sqrt{1 + 2\eta h_\mu + \eta^2}} \tag{18}$$

where

$$\eta = \frac{-h_\mu + 1/d}{1 - h_\mu/d} \tag{19}$$

and the initial value of d is set to 1. The training session can terminate with only two possible outcomes. Either a solution is found (that is, no pattern that violates condition Eq. (17) is found in a cycle), or unlearnability is detected. The problem is unlearnable if the despair parameter d exceeds a critical threshold

$$d > d_c = \sqrt{n}[2X_{max}]^{n/2} \tag{20}$$

where X_{max} is the maximal value of the normalization factors (see Eq. 16) and $n = 210$ is the number of components of **w**. This value of d_c is easily derived for examples of the type of Equation (15) by using the same method as given Nabutovsky and Domany.[65]

Maximal stability algorithm

Among all the possible solutions **w** we can single out one of special significance, that of *maximal stability*. This is the point that lies deepest in version space, i.e., its distance to the closest boundary is maximal. In the example of Figure 8, the solution of maximal stability is in the direction of **u** + **v**. For any solution **w** there is a particular example $\mathbf{x}_\nu$ which has the minimal scalar product $h_\nu = \min_\mu \mathbf{w} \cdot \mathbf{x}_\mu$ among all possible $\mathbf{x}_\mu$. The problem is to find the vector $\mathbf{w}^*$ so that

$$h^* = \max_{\mathbf{w}} \min_{\mu} h_\mu \tag{21}$$

is maximal for $\mathbf{w}$ taken in the version space.

We use a modified version of the minover algorithm,[66] which is one of several that was proved to converge to the maximal stability solution when the set of examples is linearly separable. The algorithm consists of three modules: initialization, learning loop, and stop procedure. The initialization consists of:

- At $t = 0$, initialize a weight vector $\mathbf{w}(0)$ (for example, one can use a particular solution obtained beforehand).
- Find $h_\nu(0)$ and $\mathbf{x}_\nu(0)$.
- Set $h^* = h_\nu(0)$.
- Use the example $\mathbf{x}_\nu$ to update $\mathbf{w}(0)$: $\mathbf{w}(1) = (\mathbf{w}(0) + \eta\mathbf{x}_\nu(0))/ | \mathbf{w}(0) + \eta\mathbf{x}_\nu(0) |$, where η is an independent parameter.

At time t of the learning loop the following steps are taken:

- Find $\mathbf{x}_\nu(t)$ and $h_\nu(t)$.
- Set $\mathbf{w}(t + 1) = (\mathbf{w}(t) + \eta\mathbf{x}_\nu(t))/ | \mathbf{w}(t) + \eta\mathbf{x}_\nu(t) |$.
- If $h_\nu(t) > h^*$, set $h^* = h_\nu(t)$ and $\mathbf{w}^* = \mathbf{w}(t + 1)$.

The field $\mathbf{h}^*$ asymptotes to a limiting value, and we use the condition $| h_\nu(t) - h^* | < \epsilon$ to stop the algorithm, where ϵ is a given parameter (typically $\epsilon = 0.001$).

Selection of a relevant subset of decoys

In a typical threading experiment discussed in this work we have to deal with millions of decoys. Not all of them are relevant for the learning process. During a learning session one can count how many times each example violates the inequality (Eq. 12), thus entering in the learning process. We found that only a small fraction of decoys is used in a learning session (typically, a few percent of the total). These decoys are referred to as *low-energy decoys.* This name is justified by the following picture. The version space is a connected region in parameter space that shrinks to zero volume in unlearnable cases. If a problem is nontrivially learnable, one can still assume that in some sense this region is "small." The meaning of this is that if one uses two points in version space to rank the energy of all the decoys, the two rankings do not differ dramatically. In this sense, a low-energy decoy is of low energy in all the version space. From the computational point of view, it is much more convenient to consider only low-energy decoys. We present in detail the selection procedure.

1. Say that we have a set of M_p proteins. For each protein i ($i = 1, \ldots, M_p$), we use all the longer proteins in the database as templates to generate decoys. We use some existing set of energy parameters to rank the decoys according to their energy.
2. For each protein i, we select all the p_i decoys that violate inequality (Eq. 12) plus other q_i of lowest energy (typically $q_i = 1{,}000$).
3. We subunit the subset of $P = \Sigma_i^{M_p} p_i + q_i)$ decoys collected in this way to the learning procedure. If this subset is unlearnable, we proved that also the entire set is unlearnable. If this subset is learnable, we use the solution to reclassify the entire set. If we find no violations of the inequality (Eq. 12) (which is typically the case if q_i is not too small) we found a solution for the entire set.

ACKNOWLEDGMENTS

We are grateful to Ron Elber for discussing with us a similar approach, based on Equation 5 (unpublished), and to Ido Kanter and Gaddy Getz for discussions.

REFERENCES

1. Anfinsen C. Principles that govern the folding of protein chains. Science 1973;181:223–230.
2. Levitt M. Molecular dynamics of native protein. II. Analysis and nature of motion. J Mol Biol 1983;168:595–620.
3. Lazaridis T, Karplus M. "New view" of protein folding reconciled with the old through multiple unfolding simulations. Science 1997;278:1928–1931.
4. Brooks CL. Simulations of protein folding and unfolding. Curr Opin Struct Biol 1998;8:222–226.
5. Levitt M. A simplified representation of protein conformations for rapid simulation of protein folding. J Mol Biol 1976;104:59–107.
6. Ueda Y, Taketomi H, Go N. Studies of protein folding, unfolding and fluctuations by computer simulations. II. A three-dimensional lattice model of lysozyme. Biopolymers 1978;17:1531–1548.
7. Miyazawa S, Jernigan RL. Estimation of effective interresidue contact energies from protein crystal structures: quasi-chemical approximation. Macromolecules 1985;18:534–552.
8. Skolnick J, Kolinski A. Simulations of the folding of a globular protein. Science 1990;250:1121–1125.
9. Šali A, Shakhnovich EI, Karplus M. Kinetics of protein folding: a lattice model study of the requirements for folding to the native state. J Mol Biol 1994;235:1614–1636.
10. Dill KA, Bromberg S, Yue K, et al. Principles of protein folding—a perspective from simple exact models. Protein Sci 1995;4:561–602.
11. Guo Z, Thirumalai D. Kinetics and thermodynamics of folding of a do novo designed four-helix bundle protein. J Mol Biol 1996;263:323–343.
12. Hao MH, Scheraga HA. Molecular mechanisms for cooperative folding of proteins. J Mol Biol 1998;277:973–983.
13. Mirny L, Domany E. Protein fold recognition and dynamics in the space of contact maps. Proteins 1996;26:391–410.
14. Lifson S, Sander C. Antiparallel and parallel beta-strands differ in amino acid residue preferences. Nature 1979;282:109–111.
15. Chan HS, Dill KA. Origins of structure in globular proteins. Proc Natl Acad Sci USA 1990;87:6388–6392.
16. Godzik A, Skolnick J, Kolinski A. Regularities in interaction patterns of globular proteins. Protein Eng 1993;6:801–810.
17. Vendruscolo M, Kussell E, Domany E. Recovery of protein structure from contact maps. Fold Des 1997;2:295–306.
18. Vendruscolo M, Domany E. Efficient dynamics in the space of contact maps. Fold Des 1998;3:329–336.
19. Tanaka S, Scheraga HA. Medium- and long-range interaction parameters between amino acids for predicting three-dimensional structures of proteins. Macromolecules 1976;9:945–950.
20. Wodak S, Rooman M. Generating and testing protein folds. Curr Opin Struct Biol 1993;3:247–259.
21. Sippl M. Knowledge based potentials for proteins. Curr Opin Struct Biol 1995;5:229–235.
22. Jernigan RL, Bahar I. Structure-derived potentials and protein folding. Curr Opin Struct Biol 1996;6:195–209.
23. Kocher JP, Rooman MJ, Wodak SJ. Factors influencing the ability of knowledge-based potentials to identify the native sequence-structure matches. J Mol Biol 1994;235:1598–1623.

24. Thomas PD, Dill KA. Statistical potentials extracted from protein structures: how accurate are they? J Mol Biol 1996;257:457–469.
25. Park B, Levitt M. Energy functions that discriminate x-ray and near-native folds from well-constructed decoys. J Mol Biol 1996; 258:367–392.
26. Skolnick J, Jaroszewski L, Kolinski A, Godzik A. Derivation and testing of pair potentials for protein folding: when is the quasichemical approximation correct? Protein Sci 1997;6:676–688.
27. Rooman M, Gilis D. Different derivations of knowledge based potentials and analysis of their robustness and context-dependent predictive power. Eur J Biochem 1998;254:135–143.
28. Goldstein R, Luthey-Schulten ZA, Wolynes PG. Optimal protein folding codes from spin glass theory. Proc Natl Acad Sci USA 1992;89:4918–4922.
29. Maiorov VN, Crippen GM. Contact potential that recognizes the correct folding of globular proteins. J Mol Biol 1992;227:876–888.
30. Vendruscolo M, Domany E. Pairwise contact potentials are unsuitable for protein folding. J Chem Phys 1998;109:11101–11108.
31. Hao MH, Scheraga HA. How optimization of potential function affects protein folding. Proc Natl Acad Sci USA 1996;93:4984–4989.
32. Mirny L, Shakhnovich EI. How to derive a protein folding potential? A new approach to an old problem. J Mol Biol 1996;264: 1164–1179.
33. Seno F, Maritan A, Banavar JR. Interaction potentials for protein folding. Proteins 1998;30:244–248.
34. Abkevich VI, Gutin AM, Shakhnovich EI. Impact of local and non-local interactions on thermodynamics and kinetics of protein folding. J Mol Biol 1995;252:460–471.
35. Bowie D, Luthy JU, Eisenberg D. A method to identify protein sequences that fold into a known 3-dimensional structure. Science 1991;253:164–170.
36. Jones DT, Taylor WR, Thornton JM. A new approach to protein fold recognition. Nature 1992;358:86–89.
37. Fisher D, Rice D, Bowie JU, Eisenberg D. Assigning amino acid sequences to 3-dimensional protein folds. FASEB J 1996;10:126–136.
38. Hendlich M, Lackner P, Weitckus S, et al. Identification of native protein folds amongst a large number of incorrect models: the calculation of low energy conformations from potentials of mean force. J Mol Biol 1990;216:167–180.
39. Sippl M, Weitckus S. Detection of native-like models for amino acid sequences of unknown three-dimensional structure in a database of known protein conformations. Proteins 1992;13:258–271.
40. Ouzonis C, Sander C, Scharf M, Schneider R. Prediction of protein structure by evaluation of sequence-structure fitness: aligning sequences to contact profiles derived from 3D structures. J Mol Biol 1993;232:805–825.
41. Park B, Huang ES, Levitt M. Factors affecting the ability of energy functions to discriminate correct from incorrect folds. J Mol Biol 1997;266:831–846.
42. Wang Y, Zhang H, Li W, Scott RA. Discriminating compact nonnative structures from the native structure of globular proteins. Proc Natl Acad Sci USA 1995;92:709–713.
43. Huang ES, Subbiah S, Tsai J, Levitt M. Using a hydrophobic contact potential to evaluate native and near-native folds generated by molecular dynamics simulations. J Mol Biol 1996;257:716–725.
44. Hinds DA, Levitt M. Exploring conformational space with a simple lattice model for protein structure. J Mol Biol 1994;243:668–682.
45. Cohen FE, Richmond TJ, Richards FM. Protein folding: evaluation of some simple rules for the assembly of helices into tertiary structures with myoglobin as an example. J Mol Biol 1979;132:275–288.
46. Elofsson A, Le Grand SM, Eisenberg D. Local moves: an efficient algorithm for simulation of protein folding. Proteins 1995;23:73–82.
47. Yue K, Dill KA. Protein folding with a simple energy function and extensive conformational searching. Protein Sci 1996;5:254–261.
48. Holm L, Sander C. Protein structure comparison by alignment of distance matrices. J Mol Biol 1993;233:123–138.
49. Crippen G, Havel TF. "Distance Geometry and Molecular Conformation." New York: John Wiley, 1988.
50. Brünger AT, Adams PD, Rice LM. New applications of simulated annealing in X-rays crystallography and solution NMR. Curr Opin Struct Biol 1997;5:325–336.
51. Bernstein FC, Koetzle TF, Williams GJB, et al. The protein data bank: a computer-based archival file for macromolecular structures. J Mol Biol 1977;112:535–542. The PDB database is available on-line at http://www.pdb.bnl.gov
52. Heringa J, Sommerfeldt H, Higgins D, Argos P. OBSTRUCT: a program to obtain largest cliques from a protein sequence set according to structural resolution and sequence similarity. CABIOS 1992;8:599–600. The on-line material is available at http://www.embl-heidelberg.de/obstruct.
53. Hobohm U, Sander C. Enlarged representative set of protein structure. Protein Sci 1994;3:522–524.
54. Hooft RW, Vriend G, Sander C, Abola EE. Errors in protein structures. Nature 1996;381:272.
55. Jurs PC. "Computer Software Applications in Chemistry." New York: John Wiley, 1986.
56. Press WH, Teukolsky SA, Wtterling WT, Flannery BP. "Numerical Recipes in Fortran: The Art of Scientific Computing." New York: Cambridge University Press, 1992.
57. Van Mourik J, Clementi C, Maritan A, Seno F, Banavar JR. Determination of interaction potentials of amino acids from native protein structures: tests on simple lattice models. J Chem Phys 1999;110:10123–10133.
58. Fisher D, Elofsson A, Rice D, Eisenberg D. Assessing the performance of fold recognition methods by means of a comprehensive benchmark. Proceedings of the 1st Pacific Symposium on Biocomputing 1996:300–318.
59. Mirny L, Shakhnovich EI. Protein structure prediction by threading: why it works and why it does not. J Mol Biol 1998;283:507–526.
60. Hooft RW, Sander C, Vriend G. Verification of protein structures: side-chain planarity. J App Crystallogr 1996;29:714–716.
61. Orengo CA, Michie AD, Jones S, Swindells MB, Thornton JM. CATH—a hierarchic classification of protein domain structures. Structure 1997;5:1093–1108. The CATH database is available on-line at http://www.biochem.ucl.ac.uk/bsm/cath/
62. Hertz J, Krogh A, Palmer RG. "Introduction to the Theory of Neural Computation. Santa Fe Institute Studies in the Science of Complexity." Lecture Notes v. 1 (Computation and Neural Systems Series). Redwood City, CA: Addison-Wesley Publishing Company, 1991.
63. Watkin TLH, Rau A, Biehl M. The statistical mechanics of learning a rule. Rev Mod Phys 1993;65:499–556.
64. Minsky ML, Papert SA. "Perceptions." Cambridge, MA: MIT Press, 1969.
65. Nabutovsky D, Domany E. Learning the Unlearnable. Neural Comput 1991;3:604–616.
66. Krauth W, Mezard M. Learning algorithms with optimal stability in neural networks. J Phys A 1987;20:L745–L752.
67. Miyazawa S, Jernigan RL. Residue-residue potentials with a favorable contact pair term and an unfavorable high packing density term, for simulation and threading. J Mol Biol 1996;256: 623–644.
68. Thomas PD, Dill KA. An iterative method for extracting energy-like quantities from protein structures. Proc Natl Acad Sci USA 1996;93:11628–11633.