Evidence against the Paradigm of Energy Minimization in Protein Design

A. Bazzoli[1] **and A. G. B. Tettamanzi**[1]
[1]Department of Information Technologies, University of Milan, Crema (CR), Italy

Abstract— *Stating protein design as an energy minimization in the space of amino acid sequences and rotamer combinations is usual. In the current paper an evolutionary algorithm built on this paradigm is presented, where each candidate sequence is threaded by SCWRL onto a fixed backbone structure and the energy of the resulting protein is estimated by FOLD-X. The high accuracy of such an estimate provides grounds for the hypothesis that the lowest-energy sequences may not fold into the input backbone structure, since the algorithm easily finds sequences that have a lower FOLD-X energy than the native sequence and are very different from it.*

Keywords: protein design, fixed-backbone energy minimization, native recapitulation, evolutionary algorithms

1. Introduction

The possibility of computationally designing amino acid sequences to adopt a predetermined native fold has drawn the attention of bioinformaticians for more than two decades [1], [2]. In a classic formulation of this problem, the input to a design algorithm are the Cartesian coordinates of a fixed protein backbone, while the degrees of freedom at each residue position are the type of amino acid and the orientation of its side chain, which can be chosen among a set of few, statistically-significant alternatives, called rotamers. The task is then to minimize, over all amino acid sequences and rotamer combinations, an energetic function that estimates protein free energy; as a matter of fact, it is widely accepted that the sequences which should most plausibly fold into the given backbone structure are those that would display there, through favourable side-chain orientations, the lowest free energy.

Both deterministic and stochastic methods have been developed for protein design [3], which, stated in the above terms, has been proved to be *NP*-hard [4]. Deterministic methods based on the dead-end elimination theorem [5] exclude from the search those rotamers and rotamer pairs that are guaranteed not to belong to the optimal solution; these methods always find, in theory, the optimal solution, yet they are so computationally expensive that their applicability is limited to the shortest proteins. On the other hand, stochastic methods such as Monte Carlo [6] or evolutionary algorithms [7] can be applied to proteins of any size; the solution they return is, however, not expected to be optimal, since the search space is probed only partially.

Evolutionary algorithms [8] are biologically-inspired stochastic methods used for optimization problems. Like genetic algorithms [9], of which they are a generalization, they create a population of chromosomes, each one representing a possible solution to the problem, and evolve it for a number of iterations, commonly known as generations. As in nature, a chromosome is a collection, typically a string, of genes that encode specific traits of the solution. The mechanisms by which the population is evolved mimic those that drive the natural evolution of organisms: under a sort of survival-of-the-fittest principle, chromosomes are selected to reproduce based on their quality, or fitness, and may occasionally be struck by random mutation. By the end of the evolution the population is expected to converge to chromosomes that represent optimal solutions to the problem or, at least, high-quality sub-optimal ones.

When it can be afforded, the most reliable way to assess the suitability of predicted sequences is to synthesize them and to verify, by such experimental techniques as crystallography or nuclear magnetic resonance, that they indeed fold into the input backbone structure. If, on the contrary, only a computational assessment is possible, it is a common practice to do design strictly on the backbone of known proteins — like, for example, those stored in the Protein Data Bank (PDB) [10] — so that one can directly quantify the extent to which predicted sequences reproduce, or recapitulate, native sequences. The rationale behind this kind of validation is that a design algorithm that obtains for known proteins a high recapitulation of native sequences may likely predict suitable sequences even for novel (engineered) backbone structures; nonetheless, native recapitulation in protein design is still relatively low: around 35% of correctly-predicted residue positions [6], [11].

This paper stands as a critique of the paradigm of energy minimization in protein design, by presenting an evolutionary algorithm built on it. The next section describes how the algorithm works; Section 3 goes over the approach that was taken to choose the energetic function; Section 4 gives the results of the algorithm, preluding to a final discussion, in Section 5, on how they contradict the paradigm.

2. Outline of the Algorithm

The evolutionary algorithm takes as chromosomes amino acid sequences expressed in one-letter codes, meaning that each gene is the one-letter code of an amino acid type. Its execution relies on the output of a start filter procedure, which rapidly scores the amino acid types at each residue position in order to bias the creation of the initial population toward realistic sequences. In particular, the score assigned to an amino acid type at a residue position is

$$\text{score} = 0.8\, f_{\mathrm{T}} + 0.2\, f_{\mathrm{S}} \ , \qquad (1)$$

where f_{T} and f_{S} are a torsional-angle and a solvation term, respectively; either of them is a normalization into range $[0, 1]$ of a corresponding term from the Victor/FRST function [12], implying that a score is also always between 0 and 1; and the closer a score is to 1, the more suitable the amino acid type is deemed to be. The chromosomes in the initial population are then assembled through position-specific random extractions, where amino acid types are sampled with a probability proportionate to their score.

Two central operations of the evolutionary algorithm — threading an amino acid sequence onto the input backbone structure and evaluating the resulting energy, respectively — were not implemented from scratch, but were taken from existing tools whose reliability has extensively been tested over the years. In particular, the former is effected by SCWRL [13], a program that, after mapping each residue position to the corresponding amino acid type, chooses one optimal combination of rotamers by taking into account their steric interactions both with the backbone and with each other; the latter is accomplished by using the FOLD-X algorithm [14], which estimates free energy, accurately and rapidly, as a linear combination of eight terms.

Figure 1 shows the flowchart of the evolutionary algorithm, where block EVAL is assumed to take as input a population of amino acid sequences, then to invoke SCWRL and FOLD-X for each of them, and finally to return the list of their energies. The algorithm evolves a population of 50 sequences for a total of 100 generations: these are quite small numbers compared to a typical configuration, but it would not be reasonable to increase them very much as long as SCWRL, considered in its current version, remains the chosen threading method; as a matter of fact, on the 3.2-GHz processor employed in the present work, SCWRL can take hours even for relatively short sequences, appearing to be especially slow during the initial phases of the evolution, when it encounters sequences having a more randomized amino acid composition.

After the population is randomly initialized from the scores of the filter procedure, the evolutionary algorithm makes it become the parent population of the first generation; this is carried out in three steps by means of three

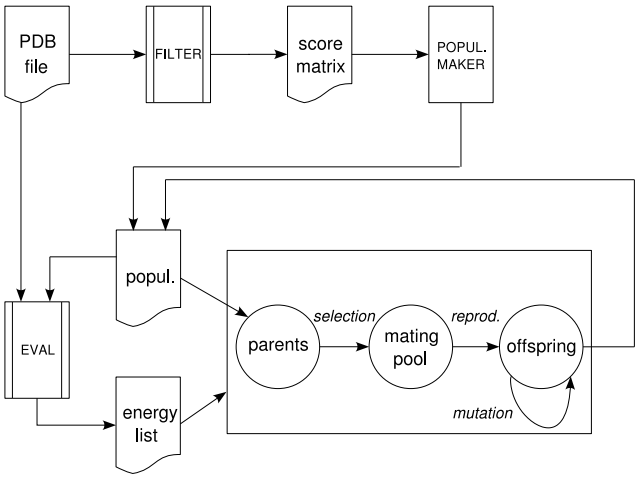

Figure 1: Flowchart of the evolutionary algorithm.

populations of 50 sequences each: a parent population, a mating pool, and an offspring population.

The first step is fitness-proportionate selection from the parents to the mating pool. Fitness is a mapping of energy into a non-negative range, in such a way that the lower is energy the higher is fitness; more precisely, the fitness of a parent chromosome with energy e is taken to be $\mathrm{M} - e$, where M denotes the maximum energy among the parents.

The second step is reproduction, whereby chromosomes in the mating pool pair up and generate offspring chromosomes. Any two paired chromosomes, or mates, undergo with probability 0.7 one-point crossover, which consists of an exchange of their right-hand parts starting at a gene position that is extracted at random; the two resulting chromosomes are the mates' offspring. On the other hand, when crossover is not applied, either mate is simply copied to an offspring chromosome.

In the third step the offspring population is mutated based on the same gene-sampling mechanism that was used to create the initial population. For each mutated offspring, mutation hits at a single, randomly-extracted gene position, and the replacing gene (amino acid type) is sampled with a probability proportionate to the score it was assigned by the start filter procedure. The purpose of this operator is to guarantee a certain amount of variability over the population, which, due to its small size, can easily suffer from premature convergence to a unique, high-energy chromosome. However, such a variability is also biased toward amino acid types that are good for the filter procedure, and is therefore expected to be less disruptive, as to the quality of chromosomes, than that induced by a uniform-probability sampling; for this reason, a rather high mutation probability, 0.1 per chromosome, was chosen.

Following mutation, the offspring become the actual

population after one generation, and are provided, with their energies, as input to the next generation. Now the algorithm is again ready to perform the three steps as explained above, and will do so on and on until the end of generation 100; eventually, an amino acid sequence that was found to have, over the whole run, the lowest energy is returned as the most suitable for the input backbone structure.

3. Search for the Energy Function

3.1 Test of Native-Sequence Discrimination

Having decided to state protein design as a problem of energy minimization, it was clear that, if the assessment of the evolutionary algorithm had to be based on native recapitulation, it was necessary to find an energy function whose global minimum and low-energy local minima lay at amino acid sequences that were similar to the native sequence. To this purpose, all energy functions candidated to be the one used in the algorithm were subjected to a preliminary test, in which their ability to discriminate the native sequence was measured. Such a test involved a unique backbone structure — that of entry 1CTF in the PDB, a protein domain of 68 residues with three alpha helices and a three-stranded beta sheet — and considered a pool of 100 random sequences, that had been collected in the same way as for an initial population (see Section 2); the energy of each sequence in the pool and the energy e_n of the native sequence were then evaluated by means of the function under study, so that two statistical quantities could be computed: the native rank, defined as one plus the number of sequences in the pool that have a higher energy than the native sequence, and the native Z score, defined as

$$\mathrm{Z} = \frac{e_n - \mu}{\sigma} \ , \tag{2}$$

where μ and σ are, respectively, the average and standard deviation of energies over the pool; in words, this second quantity expresses how the energy function discriminated the native sequence from the pool by normalizing the discrimination signal (numerator) with respect to the noise in the pool (denominator). Assuming the pool were a fairly representative sample of sequence space, the discovery of an energy function with a high rank and a low Z score, preferably far lower than -1, would have made minimization of that function a sensible approach to discovering native-like sequences.

3.2 Tested Energy Functions

One of the first functions to be tested was taken from GROMACS [15], a package that is particularly indicated for simulating proteins by molecular dynamics. This function evaluated a threaded sequence by first solvating it in a box of water, typically containing between one and two thousand molecules, and then taking the energy of the protein-water system after 200 steps of steepest-descent energy minimization, which had the purpose of setting the system in a local energy minimum. Native discrimination was definitely poor, with a rank of 63 and a Z score of -0.34. The major source of perplexity was the huge number of water molecules in the box, since it was likely that the accumulation of their reciprocal interactions rendered negligible the interactions, both intra- and intermolecular, involving the protein, thereby making pointless any comparison between amino acid sequences. The computed energy value was made even more unreliable by the fact that water molecules were added to the box in a disordered manner, changing with the amino acid sequence, and steepest-descent minimization might be too short a process to adjust them into a physiological state.

Since a whole water box was thought to hamper native-sequence discrimination, actions were needed toward reducing the number of water molecules evaluated by the energy function, and a thorough removal of water from the system was chosen as the simplest, and most extreme, of these actions. The resulting function, which just returned the energy of the protein after 200 steps of steepest-descent minimization, clearly displayed a better discrimination, since the native rank reached the maximum value of 101 and the Z score dropped to -1.87. Although it was expected that a longer local minimization would yield a better energy estimation, allowing to capture more reliably the differences between amino acid sequences in suitability to the input backbone structure, doubling the length of steepest descent, from 200 to 400 steps, caused a general worsening of native discrimination. Things got even worse when steepest descent was followed by conjugate-gradient minimization: while the former is known to reach very rapidly the neighbourhood of a local minimum but to lack, once there, the capacity to find it precisely, the latter is just specialized in refining a minimization, though it needs to be started not too far from the local minimum; coupling the two techniques in a unique energy function did likely result in an improvement of energy estimation, but, again, this improvement was harmful to native discrimination.

A following series of GROMACS tests was characterized by a mild reintroduction of water into the system. To take into account the protein-solvent interface, the energy functions under test placed a shell of water molecules around the protein and then returned the energy of the whole system after steepest-descent minimization, whose length, on the basis of what previously learned, was kept at 200 steps. A shell thickness of 3 Å, counting about 250 water molecules, afforded the best discrimination up to then, with a rank of 101 and a Z score of -2.29.

The improvement in discriminating the native sequence became dramatic upon adopting the FOLD-X energy func-

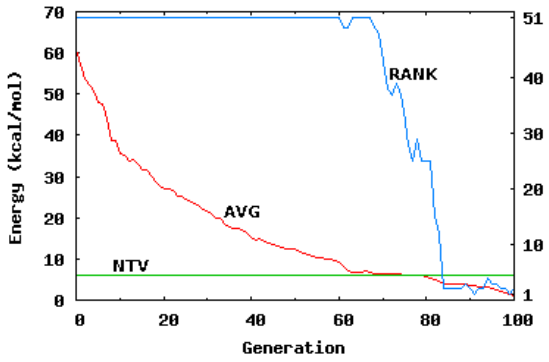

Figure 2: Run of the evolutionary algorithm under the FOLD-X energy function chosen from the native-discrimination tests — left axis: comparison at each generation between average energy in the population (AVG) and energy of the native sequence (NTV); right axis: native rank as a function of generation (RANK). See text for a description of the energy function.

tion, which had previously demonstrated to be able to model the energy change undergone by a protein when a residue is mutated into a different amino acid type [14]. Furthermore, since FOLD-X evaluates energy quite rapidly, it is also especially indicated for protein design, where returning a plausible solution can take thousands of evaluations. Only two FOLD-X candidates were compared, which differed in the way they computed the solvation energy; one adopted an implicit-solvent model, while the other used a thin shell of predicted water molecules, numbering around 50. For either function, the test of native-sequence discrimination was repeated several times, by varying one FOLD-X parameter, temperature, at increments of 1 K over the $[295\,\mathrm{K}, 369\,\mathrm{K}]$ range. Results were really encouraging, since both functions had, at all temperatures, a rank of 101 and a Z score less than -4; explicit solvation yielded always the better Z score, reaching a minimum as low as -4.73. Interestingly, both functions had the minimum Z score at a temperature of $354\,\mathrm{K}$, which is well beyond the highest temperatures normally tolerated by living organisms.

Notwithstanding a so remarkable native discrimination, when integrated into the evolutionary algorithm, the 354-K explicit-solvent FOLD-X energy function returned a predicted sequence, on the backbone of 1CTF, with only 10 native amino acid types out of 68. While the small population size may suggest that a possible reason for this outcome was premature convergence, a comparison of energies between the evolved population and the native sequence reveals that, contrary to what is assumed by protein design, it was very easy for the algorithm to discover sequences with an energy lower than native. As shown in Fig. 2, such sequences started to be discovered after 60 generations, upon which the native rank — defined for a population just like it was defined, above, for the test pool — fell to a value of 2 in only 24 generations. Not surprisingly, the average energy in the population tended to decrease throughout the whole run, indicating that the algorithm promised to minimize energy, even though such a minimization could not possibly lead to the native sequence.

The fact that, compared to GROMACS, a so significant increase in native discrimination had been poorly rewarded in terms of native recapitulation by the evolutionary algorithm (data not shown) led to the decision of closing there the quest for an energy function. Instead, to broaden the insight into the native-recapitulation skills of the algorithm, it was chosen to run it on a set of ten proteins, which are representative of the main conformational classes found in nature and had already been used in a previous protein-design study [16]. Moreover, since the indications provided by the native-discrimination tests were regarded, at that point, as anything but crucial, it was preferred to rely on a default-like FOLD-X energy function, with implicit solvent and room temperature (300 K).

4. Results

When run on the set of ten proteins, the evolutionary algorithm confirmed to be unable to recapitulate native sequences. As shown in Tables 1 and 2, the native amino acid type was correctly predicted at most in 16% of the residue positions (protein 4ICB), and for seven proteins the percentage was lower than 10%.

SCWRL proved to take much more time in the initial generations of an evolution, perhaps due to some correlation between threading time and energy of a sequence. This initial overhead got more and more intolerable with increasing protein size; in one extreme case, that of protein 5MBN, since SCWRL had been stuck for weeks in threading an early amino acid sequence, termination of the evolutionary algorithm was enforced before it could make its prediction; this is the reason why no results are given in Tables 1 and 2 for protein 5MBN.

If one considers the nine predicted sequences, six of them have a lower energy than the native sequence (Table 2), indicating that the algorithm was very effective, at least in those six cases, at minimizing the FOLD-X energy function, although such an excellent performance did not translate into a proper native recapitulation. By contrast, the higher-than-native energy of the other three predicted sequences is likely to be an effect of premature convergence, rather than evidence that the native sequence lies at the global energy minimum.

The experience gathered with the evolutionary algorithm inspired a parallel study whose purpose was to explore,

Table 1: Native recapitulation by the evolutionary algorithm for the reference protein set. For each protein, the alignment of the native (top line) and predicted (bottom line) sequences is shown, where a "*" denotes identity of amino acid type between two aligned residues. Where necessary, alignments are split into multiple rows, each one labelled (column 2) with the least residue position it covers. No prediction could be made for protein 5MBN.

```
 1PGB      1 MTYKLILNGKTLKGETTTEAVDAATAEKVFKQYANDNGVDGEWTYDDATKTFTVTE
           1 IQVSLLIWHLDQRNEYHVAVDDFTKFKQFPLPALESHKLSGTFTWKYFRWVWLHVD

 5PTI      1 RPDFCLEPPYTGPCKARIIRYFYNAKAGLCQTFVYGGCRAKRNNFKSAEDCMRTCGGA
           1 PRDPLYKRPPEGQHGSLYRCLLCDPASHYACGRDTYGVNPVDLFFGQQCTTYTVTLMS

 2CI2      1 NLKTEWPELVGKSVEEAKKVILQDKPEAQIIVLPVGTIVTMEYRIDRVRLFVDKLDNIAEVPRVG
           1 PPPYLYADVLGIYEQMPTSFIMHMIPQIVLIRQAKQRRRLPDQFSPFCVFVVTYMAKLIVFPIRC

 1CTF      1 EFDVILKAAGANKVAVIKAVRGATGLGLKEAKDLVESAPAALKEGVSKDDAEALKKALEEAGAEVEVK
             WYLMYIINVYGVFFPIVNLMIQFFGPSAPVIIVYNRKTGITVISNVPENITPAMFQAYMLVMCEVIII

 2HSP      1 GSPTFKCAVKALFDYKAQREDELTFIKSAIIQNVEKQEGGWWRGDYGGKKQLWFPSNYVEEMVNPEGIHRD
           1 YCYAQFIWDKYPPTNMFYDCMFLIPLPTDYNYKAQDETTGHFRRLFRYCYMCYTPSETTSVMYAQGPTDRG

 4ICB      1 MKSPEELKGIFEKYAAKEGDPNQLSKEELKLLLQTEFPSLLKGPSTLDELFEELDKNGDGEVSFEEFQVLV
           1 PYDPIPVAEYYSLYQITSNVAQLWPWPPLTSHILEAYPLWLHRKLDKKHIMSCMMPYGEGLVSKEFCVPWW
          72 KKISQ
          72 WKMMQ

1LMB-3     1 PLTQEQLEDARRLKAIYEKKKNELGLSQESVADKMGMGQSGVGALFNGINALNAYNAALLAKILKVSVEEF
           1 KMNMPFFIVCAFTSYAFFSVLPPYDLTYYICAVFYPQDIPIIIWLFTRLHMFHPPIEVKLMITWDSLIPCW
          72 SPSIAREIYEMYEAVS
          72 TPIMCLCIVDLCKVYI

 7PCY      1 AAIVKLGGDDGSLAFVPNNITVGAGESIEFINNAGFPHNIVFDEDAVPAGVDADAISAEDYLNSKGQTVVR
           1 DDHHRMSDNKSPPFIQDLKTTSPPKNYYKIFLLLSPDYGLWWIGPDFTNSLTAERSRQKGVMRCHGPVRER
          72 KLTTPGTYGVYCDPHSGAGMKMTITVQ
          72 VIVPFDIFMWFCWYYKRLALIVCIMIT

 5MBN      - -

9WGA-A     1 RCGEQGSNMECPNNLCCSQYGYCGMGGDYCGKGCQNGACWTSKRCGSQAGGATCPNNHCCSQYGHCGFGAE
           1 EITRYNDHGHKTEDVCPPWHGKVSENCTFCHRFVEIYASQIEELGCPYWNHVWFPDRPYKSFEAETKEKEK
          72 YCGAGCQGGPCRADIKCGSQSGGKLCPNNLCCSQWGFCGLGSEFCGGGCQSGACSTDKPCGKDAGGRVCTN
          72 TIKIKCCPGPAENETRAKPLEGSMECYRPYPVIPGVTRMCGPGVSCPREFCMFSLNAPIMKVSQCLNVNPP
         143 NYCCSKWGSCGIGPGYCGAGCQSGGCDA
         143 GCEAAAEKSRCYPILIMHQGHPDQRKVY
```

from an energetic perspective, the neighbourhood of the native sequence. By keeping the set of ten proteins and the FOLD-X energy function used in the final runs of the algorithm, the native sequences were subjected to an exhaustive series of single-point mutations where, for each residue position, the native amino acid type was replaced, in turn, by all other 19 amino acid types; as a consequence, given a protein of n residues, a total of $19\,n$ mutated sequences, or mutants, were generated. Figure 3 shows, as an example representative of the entire set, the profile of average energy variation along the native sequence of protein 2HSP. Although mutations are, in general, energetically unfavourable — as indicated by a positive average of energy variations in most residue positions — it is clear that the native sequence is not an energy minimum even in its own neighbourhood. Remarkably, in all proteins except one (2CI2, data not shown) there is at least one position at which mutation is so commonly favourable that the average energy variation is lower than 0 by more than one standard deviation; why at these positions nature has instead selected the native amino acid type is a question that can find an answer in evolutionary pressures, like optimization of biological function, that antagonize energy minimization.

5. Discussion

This paper has presented an evolutionary algorithm that attempts to solve protein design by minimizing a FOLD-X energy function in amino acid sequence and rotamer space. On a test set of ten proteins it achieved poor native recapitulation (see Tables 1 and 2), by far worse than that obtained by previous methods [6], [11]. Two

Table 2: Runs of the evolutionary algorithm on the reference protein set — summary. Column 2 reports native recapitulation as the fraction of residue positions at which the predicted sequence has the native amino acid type; columns 3 and 4 report, in kcal/mol, the energy of the native and predicted sequence, respectively. No prediction could be made for protein 5MBN.

protein	recap	e_n	e_p
1PGB	0.09	14.02	-7.59
5PTI	0.09	-1.52	10.72
2CI2	0.09	7.84	4.59
1CTF	0.09	0.88	-20.94
2HSP	0.11	74.73	50.02
4ICB	0.16	15.64	-8.87
1LMB-3	0.08	5.92	-20.33
7PCY	0.06	1.86	15.77
5MBN	-	-5.20	-
9WGA-A	0.09	72.04	154.47

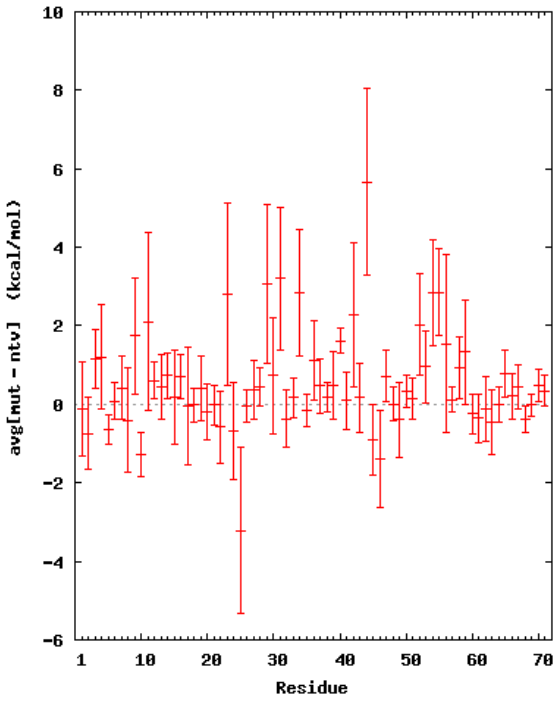

Figure 3: Mutation of protein 2HSP — profile of average energy variation. The profile was obtained by plotting, at each residue position, the average difference between the energy (mut) of a mutant at that position and the energy (ntv) of the native sequence; the corresponding standard deviation over the 19 mutants was also plotted, in the form of half the error bar centred around the average difference.

reasons for such an outcome may be identified: first, SCWRL is too slow to be integrated in the evolutionary algorithm for threading sequences onto the input backbone structure; despite its reported high speed [13], it proved particularly sensitive to the initial phases of the evolution, where threading a single sequence could take hours of computation even for the shortest proteins in the set. To compensate for this overhead, the entire evolution process had to be scaled down compared to what is considered a normal configuration, with the number of generations reduced to 100 and the population size to 50. Therefore, it is likely that the algorithm suffered from premature convergence and that predicted sequences are quite unrelated to the global energy minimum. However, the fact that some predicted sequences have a lower energy than the native sequence, as shown in Table 2, rejects the hypothesis that premature convergence and, more in general, ineffective energy minimization be sufficient to justify the poor native recapitulation.

The second reason that could justify it is that the energy function minimized by the evolutionary algorithm cannot discriminate native-like sequences, or, in other words, that the native sequence and sequences similar to it are assigned relatively high energies. The test for optimizing native discrimination, described in Section 3, proved unable to select energy functions that, once integrated in the evolutionary algorithm, would allow to achieve a high native recapitulation. Arguably, the pool of 100 random sequences collected in the test was not a representative sample of the entire sequence space, thus explaining why a clear native discrimination in the pool was not reflected into a concrete rewarding of native-like sequences during the evolution. Nonetheless, the final runs of the algorithm were effected under an energy function that provided reliable guarantees of an accurate estimation of protein energy [14]; hence, sequences that the function estimated to be lower in energy than the native sequence are likely to indeed be so in reality, thereby questioning that an approach based on energy minimization is adequate to protein design. In the light of this objection, a somewhat surprising result in the reduced problem of hydrophobic-core design [17] — namely, that different algorithms using different energy functions converged to the same predicted sequences, which were assigned, though, a better score than the native sequence — may suggest that those algorithms did really find the global energy minimum, but it was not the native sequence.

Anfinsen's principle [18] states that amino acid sequences fold into a three-dimensional structure that minimizes their (free) energy. This does not imply that native sequences are the global energy minimum for their structure. However, most protein-design algorithms do assume it, encouraged by successful works in which energy

minimization yielded sequences with an appreciable native recapitulation [6], [11], [16]. Whereas such an assumption is not compatible with the results obtained in the present work, looking at protein design from a different perspective could reconcile these results with those successes. The main idea is that, given a backbone structure, the distribution of energies in sequence and rotamer space cannot indicate whether that structure is native for any of the sequences. Even the sequence with the lowest energy could, if synthesized, fold to a totally different structure that afforded it a still lower energy. Accordingly, when evaluating some sequence in protein design, one should consider, instead of energy upon threading, a measure of how much the input structure represents an arrival point in that sequence's folding process. But the issue is bound to overlap the field of protein structure prediction, which is also a very difficult task.

References

[1] C. Pabo, "Molecular technology. Designing proteins and peptides," *Nature*, vol. 301, p. 200, Jan. 1983.

[2] O. Alvizo, B. D. Allen, and S. L. Mayo, ''Computational protein design promises to revolutionize protein engineering," *Biotechniques*, vol. 42, no. 1, pp. 31–39, Jan. 2007.

[3] C. A. Voigt, D. B. Gordon, and S. L. Mayo, "Trading accuracy for speed: a quantitative comparison of search algorithms in protein sequence design," *J. Mol. Biol.*, vol. 299, no. 3, pp. 789–803, Jun. 2000.

[4] N. A. Pierce and E. Winfree, "Protein design is *NP*-hard," *Protein Eng.*, vol. 15, no. 10, pp. 779–782, Oct. 2002.

[5] D. B. Gordon, G. K. Hom, S. L. Mayo, and N. A. Pierce, "Exact rotamer optimization for protein design," *J. Comput. Chem.*, vol. 24, no. 2, pp. 232–243, Jan. 2003.

[6] B. Kuhlman and D. Baker, "Native protein sequences are close to optimal for their structures," *Proc. Natl. Acad. Sci. USA*, vol. 97, no. 19, pp. 10383–10388, Sep. 2000.

[7] S. M. Larson, A. Garg, J. R. Desjarlais, and V. S. Pande, "Increased detection of structural templates using alignments of designed sequences," *Proteins*, vol. 51, no. 3, pp. 390–396, May 2003.

[8] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Berlin, Germany: Springer-Verlag, 1992.

[9] J. H. Holland, *Adaptation in Natural and Artificial Systems*, Ann Arbor, Michigan, USA: University of Michigan Press, 1975.

[10] F. C. Bernstein, T. F. Koetzle, G. Williams, E. F. Meyer, Jr., M. D. Brice, J. R. Rodger, O. Kennard, T. Shimanouchi, and M. Tasumi, "The Protein Data Bank: a computer based archival file for macromolecular structure," *J. Mol. Biol.*, vol. 112, no. 3, pp. 535–542, May 1977.

[11] F. Ding and N. V. Dokholyan, "Emergence of protein fold families through rational design," *PLoS Comput. Biol.*, vol. 2, no. 7, pp. 725–733, Jul. 2006.

[12] S. C. E. Tosatto, "The Victor/FRST function for model quality estimation," *J. Comput. Biol.*, vol. 12, no. 10, pp. 1316–1327, Dec. 2005.

[13] A. A. Canutescu, A. A. Shelenkov, and R. L. Dunbrack, "A graph-theory algorithm for rapid protein side-chain prediction," *Protein Sci.*, vol. 12, no. 9, pp. 2001–2014, Sep. 2003.

[14] R. Guerois, J. E. Nielsen, and L. Serrano, "Predicting changes in the stability of proteins and protein complexes: a study of more than 1000 mutations," *J. Mol. Biol.*, vol. 320, no. 2, pp. 369–387, Jul. 2002.

[15] E. Lindahl, B. Hess, and D. van der Spoel, "GROMACS 3.0: a package for molecular simulation and trajectory analysis," *J. Mol. Model.*, vol. 7, no. 8, pp. 306–317, Aug. 2001.

[16] P. Koehl and M. Levitt, "De novo protein design. II. Plasticity in sequence space," *J. Mol. Biol.*, vol. 293, no. 5, pp. 1183–1193, Nov. 1999.

[17] S. Ventura and L. Serrano, "Designing proteins from the inside out," *Proteins*, vol. 56, no. 1, pp. 1–10, Jul. 2004.

[18] C. B. Anfinsen, "Principles that govern the folding of protein chains," *Science*, vol. 181, pp. 223–230, Jul. 1973.