Evolutionary design of the energy function for protein structure prediction

Paweł Widera Jonathan M. Garibaldi Natalio Krasnogor

Abstract— **Automatic protein structure predictors use the notion of energy to guide the search towards good candidate structures. The energy functions used by the state-of-the-art predictors are defined as a linear combination of several energy terms designed by human experts. We hypothesised that the energy based guidance could be more accurate if the terms were combined more freely. To test this hypothesis, we designed a genetic programming algorithm to evolve the protein energy function. Using several different fitness functions we examined the potential of the evolutionary approach on a set of candidate structures generated during the protein structure prediction process. Although our algorithms were able to improve over the random walk, the fitness of the best individuals was far from the optimum. We discuss the shortcomings of our initial algorithm design and the possible directions for further research.**

I. Introduction

The most widely accepted hypothesis explaining the process of protein folding was formulated by Christian Anfinsen. In a Nobel prize winning experiment he found that a refolded protein always forms the same native structure [1]. He therefore concluded that all the information needed to fold a protein has to be contained in its sequence and nature is applying a “folding algorithm” with a protein sequence as an input and native state as an output. Anfinsen’s thermodynamic hypothesis stated that the native configuration is in the thermodynamic equilibrium and explained the algorithm of folding as a process of minimisation of the protein’s free energy .

The free energy is defined as a function of structure and is used in so called *ab initio* folding, where the prediction cannot rely on sequence similarity to previously solved structures. The model of the structure has to be built from “scratch” and is based on the physical principles of folding, namely, the protein inter-atomic interactions [2].

For the last few decades several models of protein force fields have been proposed such as AMBER99, CHARMM22 or OLPS-AA [3]. However, due to the high computational cost of the all-atom energy functions, their practical use is limited to the molecular dynamic simulation of short protein chains. Consider for example massively distributed projects such as Folding@home [4] or Rosetta@home [5] that are able to gather vast computational resources. Even though Folding@home is currently the most powerful computing system on Earth (it operates at 4 peta FLOPS performance level [6]), a simulation of $10\mu s$ of protein folding uses 10 000 CPU days while proteins usually fold in a millisecond timescale [7].

To lower the computational cost of prediction, high level simplified models of proteins are commonly used, such as SICHO [8], UNRES [9], CABS [10] or CAS [11]. Instead of an exact atomic representation these models use a reduced representation, where coordinates of groups of atoms are replaced with a single high level entity (pseudo-atom), e.g. the group center of mass.

Due to the loss of details, the protein free energy in the reduced models cannot directly rely on intermolecular forces. The knowledge-based potentials are used instead. They are derived from an analysis of known structures and represent the likelihood of observing a specific feature in the native state. As a consequence, the energy function does not capture the physical free energy explicitly but represents a probability that a given structure is native-like. Because this extra knowledge is being used, the prediction process is no longer considered to be *ab initio*. Instead, the structure prediction community gathered around CASP experiment [12] uses the term “template free”.

In the CASP (Critical Assessment of Techniques for Protein Structure Prediction) experiment, participants are given the protein sequences of unknown three dimensional structure to be determined computationally. In parallel the structures are determined experimentally and used to assess the prediction accuracy of the methods. Structure prediction methods are divided in two categories: template based modelling (target sequence has close homologue - template, or adopts a known fold) and template free modelling (targets with a new topology).

The two most successful prediction methods in the “template free” category of the CASP7 experiment [13][14][15] are Robetta [16] and I-TASSER [17]. Both methods build the initial protein models from short fragments of known structures similar on a sequence level. Small random changes are applied to these models and the Monte Carlo method is used to find a structure with minimal energy. In both methods the energy is formulated as a weighted sum of knowledge-based potentials.

To determine the optimal set of weights both methods generate a set of candidate structures, so called decoys, by applying small random changes to a known native structure. The optimisation objective is to maximise the correlation between the value of energy function and the similarity of decoys to the native structure. Therefore, the energy function is expected to have the lowest value for the decoys that are most similar to the native structure. Similarity is measured as

Pawel Widera, Jonathan M. Garibaldi and Natalio Krasnogor (corresponding author) are with the School of Computer Science, University of Nottingham, Nottingham, NG8 1BB, UK (e-mails: {plw,jmg,nxk}@cs.nott.ac.uk)

the root mean square deviation (RMSD) of euclidean distance between C_α atoms of a decoy and the native structure.

In the weight optimisation process, Robetta used a training set of 21 proteins. For each protein 30 000 decoys were generated and the linear regression against RMSD was used as an objective function [18]. I-TASSER used 30 proteins, with 60 000 decoys each, and maximised complex objective function with correlation to RMSD as its main element [19]. Both prediction methods are able to distinguish between native-like (RMSD value $< 0.4nm$) and non-native decoys (RMSD value $> 0.8nm$). However, the actual correlation coefficient between the energy and similarity is not too high, eg. Zhang et al. [19] report it to be 0.54 for naive combination of terms and 0.65 for the optimised one.

A critical analysis of the approach described above reveals two drawbacks. Firstly, the set of decoys created by randomisation of the native structure is biased towards that structure, resulting in potential overfitting of the energy function. The process itself is also the exact opposite of what predictors do in practice, where the native structure is unknown and decoys have to be built from scratch. Secondly, the linear combination of energy terms is a very simple but potentially very restrictive approach to construct the energy function.

With this paper we tried to address both issues. Using the set of decoys generated during the prediction process we designed a genetic programming (GP) algorithm to test the hypothesis that a more general functional combination of energy terms will result in higher correlation of the energy function with the similarity to the protein native structure.

We have selected a subset of eight energy terms used by I-TASSER and pre-calculated their values for each decoy. In a number of experiments we applied genetic programming to evolve non-linear energy functions featuring a range of basic algebraic operators and transcendental functions. Using several different fitness definitions we tried to determine how difficult it is to evolve an energy function that is highly correlated with structural similarity to the native state.

In Section II, we present the methodology of our research, a detailed description of the data sets used and the GP parametrisation. In Section III, we describe the experiments carried out and the results obtained. We then present a discussion of the results and concluding remarks.

II. Methods

A. Energy terms

We have implemented eight I-TASSER energy terms. Three short-range potentials between C_α atoms E_{13}, E_{14} and E_{15}, long-range pairwise potential between side chain centres of mass E_{pair}, environment profile potential E_{env}, local stiffness potential E_{stiff} and electrostatic interactions potential $E_{electro}$ as described in [19][11] and the hydrogen bonds potential E_{HB} as explained in supplementary materials to [20].

The first five of these energy terms use a distribution of structural features that is derived from the data base of known protein structures. Stiffness and hydrogen bond potentials represent structural bias towards regular arrangements of predicted secondary structure and penalise irregularities. Electrostatic potential uses the Debye-Hückel equation to calculate the interaction energy of ions in the solution.

We left out potentials using data from the threading process (e.g. distance map or contact order) and the hydrophobic potential introduced in [17] using neural network [21] as they depend on external feature predictors which were not available for local use at the time of writing this paper.

B. Data preprocessing

In our experiments we used 54 protein chains used by Zhang et. al [17]. For each protein we used a set of decoys generated by I-TASSER along the Monte Carlo optimisation process [22] (available online [17]). To eliminate highly similar decoys we have taken a 10% sample of each set (one decoy from every 10th I-TASSER iterations), resulting in a training set of 1250–2000 decoys per protein. For each decoy we have pre-calculated the values of all eight energy terms mentioned above.

For each protein we have measured the similarity of generated decoys to the known native structure. As a measure we used the root mean square deviation (RMSD) between 3D coordinates of C_α atoms of two structures minimised with respect to the rotation using Kabsch algorithm [23][24]. As a non-weighted average of all C_α–C_α distances the RMSD is sensitive to local errors and might return high values of distance even if global topology is correct. Despite known limitations of RMSD as a measure of protein structural similarity [25], we decided to use it to make the fair comparison to the previous work [19][17].

To compensate for the noise introduced by RMSD, we decided not to rely on the absolute RMSD values directly, but rather on the relative rank order. That is, for given decoys A and B we decide only if $RMSD(A, native) < RMSD(B, native)$ ignoring the scale of absolute difference in the distance to a native $\delta = RMSD(A, native) - RMSD(B, native)$. This approach simplifies the optimisation objective, as ranking is more robust than a matrix of exact distances between all pairs of decoys.

For each protein we sorted the decoys in increasing order of the RMSD to generate the reference ranking R_r. In case of ties, we used the original I-TASSER energy as a second criterion (lower energy corresponds to lower index in the ranking). A tie between decoys was called when RMSD values were the same up to the first two decimal places. This gives us a precision of a 1 picometer (for reference, the radius of hydrogen atom is 25 pm).

For the same set of protein chains we ran the Rosetta ab initio prediction [16] and obtained the same number of decoys as in I-TASSER case. These decoys were only used for visual assessment of correlation between Rosetta energy and RMSD (see Section III).

C. Genetic programming experiment

We used a set of 16 terminals and 8 operators. Half of the terminals were the energy terms T_1–T_8 described in II-A (see Table II for the mapping to I-TASSER terms), half were ephemeral random constants in range [-1,1]. Operators were both binary (addition, subtraction, multiplication, division) and unary (sine, cosine, exponential function, natural logarithm). We did not impose any selection bias towards any of the primitives.

The fitness function used to evaluate the energy function was based on a comparison of the reference ranking R_r (obtained in the preprocessing stage) to the evolutionary ranking R_e. For each protein the evolved energy function was used to rank the decoys and obtain ranking R_e. A normalised distance between rankings $d(R_r, R_e)$ was calculated for each protein, and then averaged for all proteins to produce the total fitness.

We used several different methods to calculate the distance between rankings (see examples in Table I):

- Levenshtein edit distance [26] - a popular string metric where distance is given by the minimum number of operations (insertion, deletion or substitution of a character) needed to transform one string into the other,

$$L(a,b) = d_{n,n}$$

$$d_{i,0} = d_{0,i} = i \text{ for } i = 0 \dots n$$

$$d_{i,j} = min\{d_{i-1,j} + 1, d_{i,j-1} + 1, d_{i-1,j-1} + c(i,j)\}$$

$$c(i,j) = \begin{cases} 0 & \text{if } a(i) = b(i) \\ 1 & \text{if } a(i) \neq b(i) \end{cases}$$

- Kendall τ distance [27] - the number of inversions between two permutations also known as the bubble-sort distance,

$$K(a,b) = |\{(i,j) : i < j \wedge a(i) < a(j) \wedge b(i) > b(j)\}|$$

- Spearman footrule distance [28] - the sum of differences between the ranks of elements.

$$S(a,b) = \sum_{i}^{n} |a(i) - b(i)|$$

TABLE I

EXAMPLES OF USE OF SELECTED RANKING DISTANCE MEASURES.

Levenshtein distance = 2	**Kendall distance = 3**	**Spearman distance = 10**	**weighted Spearman distance = 4.6 ($\frac{23}{5}$)**
1 2 3 4 5	1 2 3 4 5	4 3 2 1 5	4 3 2 1 5
1 **3** 4 5 **2**	1 **3** **4** **5** 2	3 4 1 5 2	3 4 1 5 2
		1 1 1 4 3	**1 1 1 4 3**
			$\frac{5}{5}$ $\frac{4}{5}$ $\frac{3}{5}$ $\frac{2}{5}$ $\frac{1}{5}$

Notice that the measures differ in the computational cost. For the Levenshtein distance a dynamic programming algorithm has to be used with a complexity of $O(n^2)$. Kendall distance can be computed in $O(nlogn)$ time by counting inversions during the merge sort procedure. Spearman distance is the simplest measure of these three and can be calculated in linear time.

Both Kendall and Spearman distances are bounded by $O(n^2)$, having the maximum possible distance equal to respectively $\frac{n(n-1)}{2}$ and $\frac{1}{2}n^2$ for the reversed ranking. Levenshtein distance, similar to many other editing distance metrics on permutations such as Hamming metric, Cayley distance or Ulam metric, is bounded by the $O(n)$.

An additional weighting mechanism was applied to the Spearman distance to promote correct order at the beginning (more native-like) of the ranking and to be less sensitive to differences in the order at the end (less native-like). We used two weighting functions:

- linear function decreasing from 1 to 0 along the position in the ranking,

$$w(i) = 1 - i/N, \text{ for } 0 \leq i < N$$

- sigmoid function with inflection point (weight 0.5) at 25% of the ranking length.

$$w(i) = \frac{1}{1 + \exp(\frac{i - 0.25N}{scale * N})}$$

$$scale = \begin{cases} \frac{0.25N}{width} & \text{if } i < 0.25N \\ \frac{0.75N}{width} & \text{if } i \geq 0.25N \end{cases}$$

We have implemented the genetic programming algorithm using the Open BEAGLE framework [29]. In all experiments we used two replacement strategies: generational and steady-state [30], with the tournament selection [31] and 1000 generations. The population size was set to 100, crossover probability was 0.8 with 0.1 probability that the crossover point is a leaf and 0.05 probability of reproduction without modification. Three mutation operators were used with 0.05 probability each: sub-tree replacement with a random tree, tree shrink where node is replaced by one of its child nodes, sub-tree swap with 0.5 probability that mutation point is a leaf. The initial exploratory trial from which this configuration is derived is not reported here due to space limitations.

We have run two rounds of experiments. In the first round we used Levenshtein, Kendall and non-weighted Spearman distances. In the second round we drop the worst performing Levenshtein distance and applied the linear and sigmoid weighting to the Spearman distance. To have higher selection pressure we increased the tournament size from 2 in the previous round to 4, 6 and 8. We added one additional run configuration: generational replacement with strong elitism (keeping 5 best individuals from each generation), single mutation operator (replacement with a random tree) applied with probability of 0.1 and the crossover with probability 0.9.

Additionally, a random walk was performed for reference. In each generation the population was created using the half-and-half initialization operator [32][33].

Each experiment was repeated 5 times with different random seeds. In the next section we report results of the best run, as we are interested in obtaining the best possible GP-designed energy function that could perhaps be human-competitive.

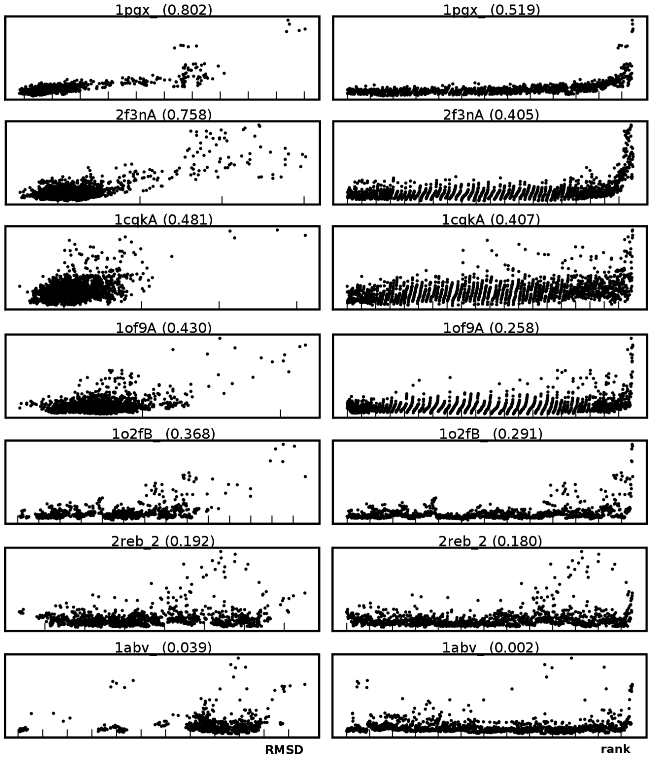

Fig. 1

Scatter plots of I-TASSER original energy (vertical axis) vs. RMSD (left column) and rank (right column). Each plot represents decoys for a single protein. The correlation coefficients are given in brackets.

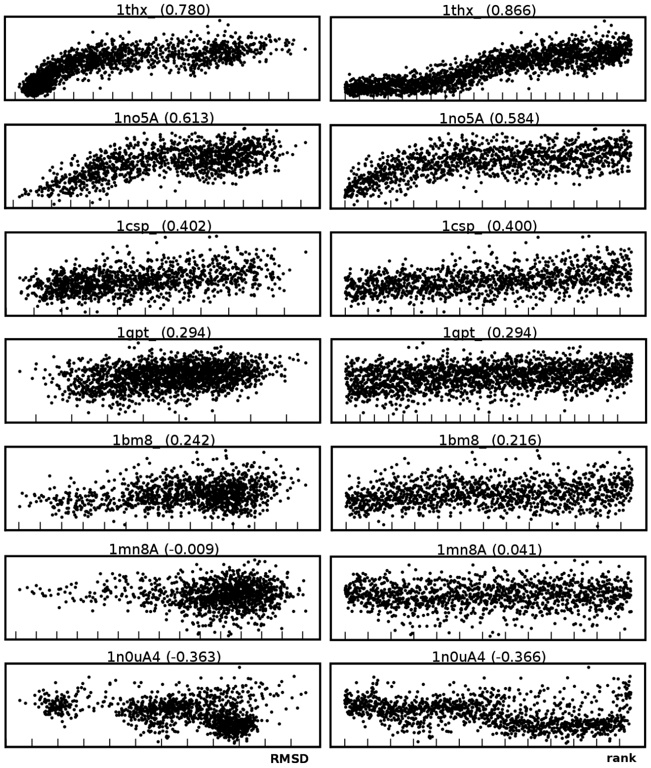

Fig. 2

Scatter plots of Rosetta original energy (vertical axis) vs. RMSD (left column) and rank (right column). Each plot represents decoys for a single protein. The correlation coefficients are given in brackets.

III. Main results

A. Energy correlation

To compare the ability of the original energy functions used by I-TASSER and Robetta to distinguish between native-like and non-native structures, we plot the relation between the energy and similarity to the native for all decoys. Figure 1 shows the I-TASSER energy correlation to RMSD (left column) and rank (right column) for selected proteins with a high, average and low correlation coefficient (given in brackets). The average correlation coefficient for all proteins was 0.44 ± 0.23 (second value is a standard deviation). Interestingly, even a high correlation coefficient (*1pqx_*), was not enough to point to the most native-like structure as we observe a flat cloud in the lowest energy region stretched over a distance of 0.1–0.2nm. This cloud becomes wider with a decrease of the correlation coefficient and its center tends to shift towards greater values of RMSD.

This difficulty in selecting the most native-like decoys is even more visible when the energy is plotted against the rank (right column of Figure 1). Instead of a clear trend with the energy decreasing along the decreasing rank, the trend line is very flat and thick. Several slightly slanted vertical stripes are visible in regions were a number of decoys equally distant from the native have a different energy (*2f3nA*, *10f9A*). Overall, the correlation to the rank was almost half as low as in the case of RMSD with the total average of 0.25 ± 0.16.

Similar plots for decoys generated by Rosetta are shown in Figure 2. As the decoys cover a wider RMSD range and are not concentrated in a single region, the total average correlation coefficient to the rank, equal to 0.28, is only 0.02 lower than the coefficient of correlation to RMSD. However, similar to the I-TASSER energy, pointing out the native-like decoys using the value of Rosetta energy is in most cases very difficult.

B. I-TASSER energy terms

Coefficients for individual energy terms are shown in Table II. The values for our decoys are significantly lower than the one reported by Zhang et al. [19]. Notice the negative correlation of selected terms which decreases the average correlation nearly to zero. The low values of the ρ_2 coefficient could, however, be somewhat misleading as they are hiding the spread amongst different proteins. The relative standard deviation for ρ_2 ranged from 82% for T_2 to 942% for T_6.

The average correlation coefficient for the naive sum of energy terms $E_N = \sum_{i=1}^{8} T_i$ was 0.12 ± 0.16. Correlation between the naive sum of energy terms and the rank was lower as in the case of original I-TASSER energy, and the coefficient value was 0.07 ± 0.16.

TABLE II

BOTH ρ_1 AND ρ_2 SHOW THE AVERAGE CORRELATION BETWEEN A SINGLE ENERGY TERM AND THE SIMILARITY TO THE NATIVE STRUCTURE MEASURED BY RMSD. ρ_1 IS THE COEFFICIENT ORIGINALLY REPORTED BY ZHANG ET AL. [19] AND ρ_2 IS THE COEFFICIENT CALCULATED FOR OUR IMPLEMENTATION OF I-TASSER ENERGY TERMS ON 54 PROTEINS USED IN OUR EXPERIMENT. ρ_E IS THE COEFFICIENT OF CORRELATION BETWEEN A SINGLE ENERGY TERM AND THE ORIGINAL I-TASSER ENERGY.

energy term	ρ_1	ρ_2	ρ_E
T_1 (E_{13})	0.27	0.03 ± 0.11	0.08 ± 0.15
T_2 (E_{14})	0.56	0.20 ± 0.17	0.38 ± 0.16
T_3 (E_{15})	0.33	0.15 ± 0.15	0.34 ± 0.19
T_4 (E_{stiff})	0.25	0.24 ± 0.22	0.44 ± 0.24
T_5 (E_{HB})	0.51	-0.16 ± 0.20	-0.36 ± 0.23
T_6 (E_{pair})	0.38	0.01 ± 0.14	0.12 ± 0.13
T_7 ($E_{electro}$)	0.27	-0.20 ± 0.23	-0.34 ± 0.26
T_8 (E_{env})	0.34	0.04 ± 0.16	0.03 ± 0.15
average	0.40	0.06	0.09

C. Fitness distance correlation

The optimisation objective was to minimise the distance $d(R_r, R_e)$ between the reference ranking R_r and the ranking R_e produced by the evolved function. The range of distances was normalised to the [0,1] fitness range, where maximum distance (comparison with reversed R_r) gives fitness equal to 0 and zero distance corresponds to fitness equal to 1.

To compare the landscape difficulty of different fitness functions we have measured the fitness distance correlation on the phenotype level. Starting from a random reference ranking R_r of length 100, for each of $t \in \{1, \ldots, 1000\}$ steps 20 new permutations were generated by applying a random transposition to the permutations from previous step $t - 1$. The correlation of the fitness functions to the distance to R_r measured in number of applied transpositions, as well as the direct correlation between the fitness functions, is shown in Figure 3.

Notice that the minimum number of transpositions needed to transform n-element permutation a to b, known as Cayley distance, is bounded by $O(n-1)$ and equal to $n-c$, where c is the number of cycles in the disjoint cycle decomposition of ab^{-1}. Because of that, for $n > 100$ a fluctuation of the fitness value is observed in plots A–C of Figure 3. As the range of values that the fitness function based on the Levenshtein distance can obtain is a square of the range obtainable for Spearman and Kendall distances, the fluctuation range is also lower.

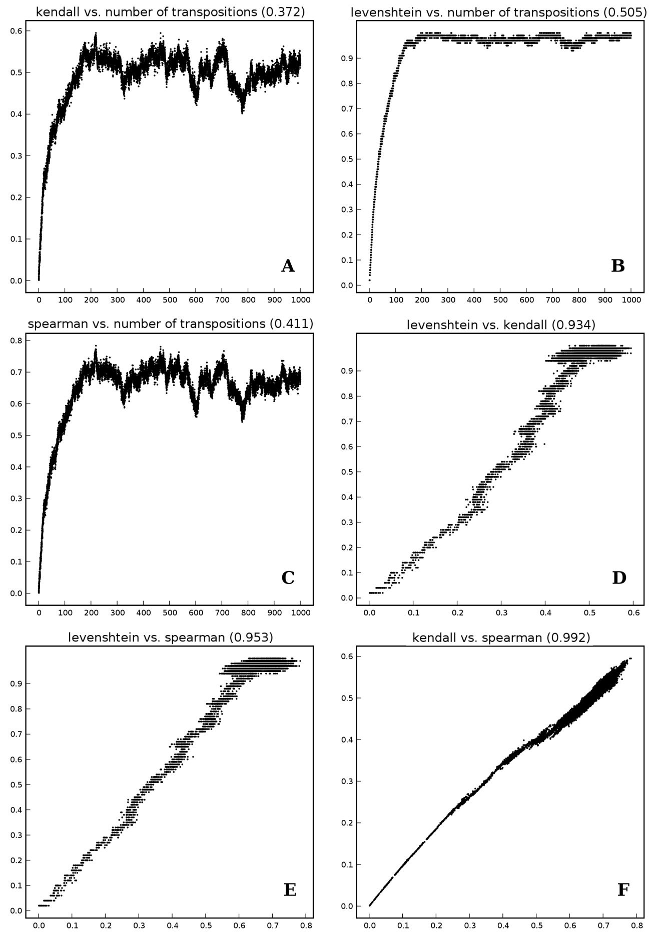

Fig. 3

CORRELATION BETWEEN FITNESS AND NUMBER OF TRANSPOSITIONS APPLIED TO THE REFERENCE RANKING (PLOTS A–C) AND DIRECT CORRELATION BETWEEN FITNESS FUNCTIONS (PLOTS D–F). THE CORRELATION COEFFICIENTS ARE GIVEN IN BRACKETS.

This limitation is visible even more clearly in plots D and E of Figure 3. The distinct horizontal stripes appear for groups of permutations equally distant from R_r in the Levenshtein metric space but easily distinguishable by the other two distances. The gaps between the stripes are another indicator of sparse space of values of the Levenshtein distance. The horizontal stripes become longer near the maximum of Levenshtein distance, showing inablity of this measure to distinguish between many samples slightly above the middle of other two distances.

D. First round of experiments

The average fitness for the Levenshtein distance diverged in a tiny range very close to the maximum distance (see Figure 4. For Spearman distance we observed a fast improvement of the average fitness in the first 50–100 generations and the later saturation around 40% of the maximum fitness. For the Kendall distance the initial improvement seems to be more rapid but the spread of fitness was equal only to a very small 3% range of the maximum fitness. The improvement in fitness of the best found individual over the random walk was only 1.3% for the Kendall distance and almost 5.5% for the Spearman distance (see Table III).

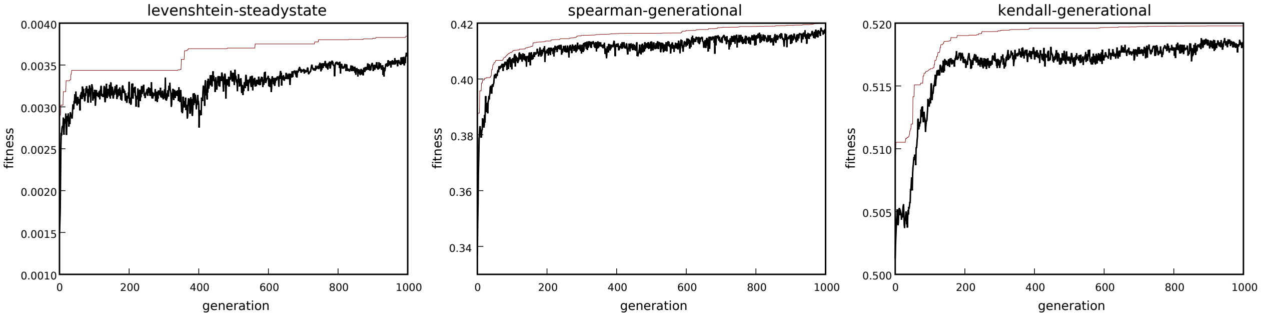

Fig. 4

Fitness throughout the generations in the first round of experiments for the best run for each fitness function. The lines show average (thick black line) and the maximum (thin red line) fitness in the population.

TABLE III

Maximum fitness and improvement over the random walk for two run configurations and three fitness functions used in the 1st round of experiments.

	steady-state		**generational**	
measure	**max**	**improvement**	**max**	**improvement**
Levenshtein	0.004	25.91%	0.003	13.01%
Spearman	0.417	4.72%	0.420	5.49%
Kendall	0.520	0.97%	0.522	1.34%

E. Second round of experiments

The linear weighting mechanism did not change the fitness landscape but the sigmoid weighting did. As shown in Figure 6 the Spearman distance with sigmoid weights reached over 20% higher average and maximum fitness value than the non-weighted Spearman distance. It also seem that the evolutionary progress for the best runs of both linear and sigmoid weighted Spearman distance continues steadily without the early saturation observed in the first round of experiments (see Figure 5). However the spread of fitness values is again low, covering only about 10% of the fitness range (see Figure 6).

We did not observe a significant change in the evolutionary improvement over the random walk for the Kendall distance. In case of the Spearman distance with linear weights the improvement seems to be even twice as big (up to 11%) as in the previous round. However, the maximum fitness values are still in the 0.4–0.5 range, so similarly far away from the maximum as in the first round of experiments.

The run configuration with elitism and single mutation operator performs best (in terms of improvement over the random walk) with the sigmoid weighted Spearman distance. For the overall best evolved function (with fitness 0.530) we checked the correlation to RMSD and found the coefficient to be 0.26 ± 0.17, which is over two times higher than the correlation of the naive sum of terms (see Section III-B) but only 0.02 greater than the highest correlation of a single term (see Table II).

TABLE IV

Maximum fitness and improvement over the random walk for three run configurations with different tournament size and three fitness functions used in the 2nd round of experiments.

		steady-state		**generational**		**elitism**	
measure	**ts**	**max**	**impr**	**max**	**impr**	**max**	**impr**
Kendall	4	0.516	0.19%	0.522	1.49%	0.514	-0.08%
	6	0.515	0.01%	0.517	0.42%	0.513	-0.25%
	8	0.520	1.03%	0.514	-0.12%	0.517	0.45%
Spearman linear	4	0.416	6.24%	0.429	9.47%	0.418	6.67%
	6	0.430	9.79%	0.404	3.10%	0.408	4.18%
	8	0.419	6.78%	0.436	11.34%	0.423	7.80%
Spearman sigmoid	4	0.518	7.49%	0.503	4.38%	0.527	9.47%
	6	0.516	7.11%	0.514	6.71%	0.511	6.04%
	8	0.515	6.98%	0.508	5.38%	0.530	9.96%

IV. Discussion

To be useful, the energy function has to guide the search process towards the region of native-like proteins. It seems reasonable to measure this usefulness with a correlation coefficient between energy and similarity to native. However, as we have shown in Figures 1–2, even the high coefficient (> 0.7) does not mean that the native-like structure would be easy to distinguish from the others. This is reflected in an even lower correlation to rank, where decoys within the same energy range are spread across many ranks.

The difference in correlation of single energy terms between our implementation and the original work by Zhang et al. (see TableII) shows the difference in difficulty of choosing native-like structure between different decoys sets. While we have used a sample from the conformational search process that is initialised with fragments of other proteins similar on the sequence level and has no knowledge of the native structure, I-TASSER and Robetta used the decoys generated by randomisation of the native resulting in a biased set. Moreover, the decoys we used are often very similar to each other, whereas Zhang kept them separated by large 0.35nm RMSD distance. Our results show that decoys generated by the predictor are more difficult to assess and it might be inadequate to optimise the energy based on the randomised

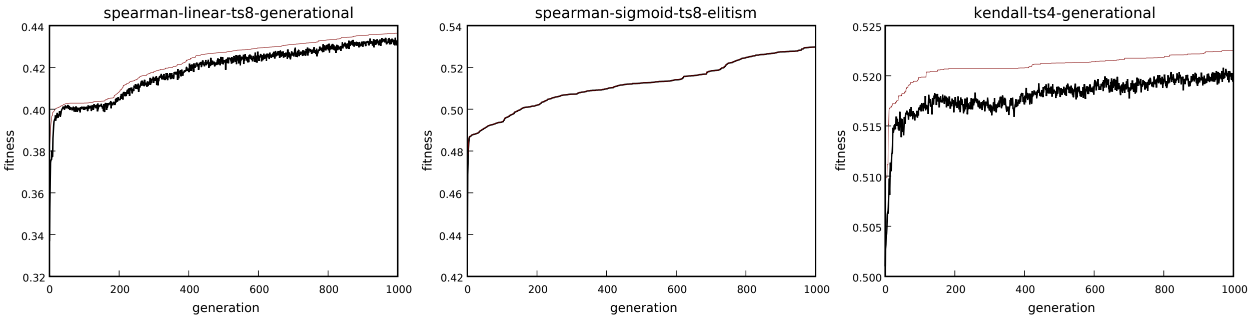

Fig. 5

Fitness throughout the generations in the second round of experiments for the best run for each fitness function. The lines show average (thick black line) and the maximum (thin red line) fitness in the population.

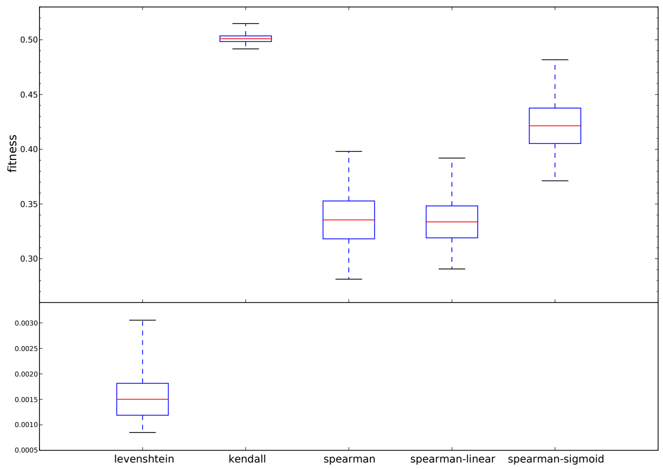

Fig. 6

Box plot of the fitness distribution achieved by a random walk for different fitness functions. Middle line is the median of the average fitness in population across all generations. Box size represents the median of the population fitness standard deviation. Top and bottom whiskers marks maximum and minimum fitness across all individuals.

and highly separated set of decoys, as this is not what predictors have to deal with in practise.

The main reason why the correlation to RMSD of the naive combination of energy terms compared to the original I-TASSER energy were much lower, is probably in the choice of energy terms. In future work we may extend the set of energy terms adding data from protein features predictors e.g. distance maps, contact order, contact restraints or solvent accessibility [34][35][36] to make it more comparable to what I-TASSER is using.

We decided to build the ranking with a picometer RMSD precision, since in the structure optimisation process it is important to be able to measure the energetic outcome of each structural change. Still, this caused many ties in the rank. The permutational approach, when the tie is decided by the original I-TASSER energy might not be the best choice as the I-TASSER energy itself was not highly correlated with RMSD. In the future work we might average the ranks in case of ties not to enforce any arbitrary ordering.

Regardless of a fitness function used the average fitness saturated around the maximum after only 50–200 generations. The major factor that may cause this early saturation is the polynomial bound on the number of possible values that the fitness function could take, which was in the best case limited by $O(n^2)$. As a result, many different energy functions had the same value of the fitness function. This explains the poor performance of the the fitness function using the $O(n)$ bounded Levenshtein distance.

Another important factor might be related to the amount of data we used in the evolutionary process. Maybe with a smaller number of decoys or a smaller number of proteins we could obtain better evolutionary improvement. Moreover, as the decoy sets seem to be very uneven and noisy (in terms of original energy) we could apply a filtering method to sample only those decoys, for which the original energy is highly correlated to RMSD. Perhaps in this way, a good energy function could be evolved more easily and as long as the filtered sample is sufficiently representative, it could be applied successfully to the set of all decoys.

V. Conclusions

In this paper we have proposed the use of genetic programming to evolve novel forms of energy function for protein structure prediction. We have demonstrated an initial GP design which, although not perfect yet, might lead in the future to significant improvements in the quality of protein structure prediction with perhaps human-competitive results.

We hope to address some of the limitations discussed in the text in the near future and extend the scope of the experiment to both I-TASSER and Rosseta generated decoys.

Acknowledgment

We would like to thank Yang Zhang for making the decoys data available online and for explaining the details of I-TASSER energy terms implementation.

This research was supported by the Marie Curie Action MEST-CT-2004-7597 under the Sixth Framework Programme of the European Community.

References

[1] C. Anfinsen, "Principles that Govern the Folding of Protein Chains," *Science*, vol. 181, pp. 223–30, July 1973. doi:10.1126/science.181.4096.223.

[2] P. E. Bourne, *Structural Bioinformatics*, ch. CASP and CAFASP experiments and their findings, pp. 499–505. Wiley-Liss, 2003. doi:10.1002/0471721204.ch24.

[3] J. A. MacKerell, "Empirical force fields for biological macromolecules: Overview and issues," *Journal of Computational Chemistry*, vol. 25, no. 13, pp. 1584–1604, 2004. doi:10.1002/jcc.20082.

[4] V. S. Pande, I. Baker, J. Chapman, S. P. Elmer, S. Khaliq, S. M. Larson, Y. M. Rhee, M. R. Shirts, C. D. Snow, E. J. Sorin, and B. Zagrovic, "Atomistic protein folding simulations on the submillisecond time scale using worldwide distributed computing," *Biopolymers*, vol. 68, no. 1, pp. 91–109, 2003. doi:10.1002/bip.10219.

[5] R. Das, B. Qian, S. Raman, R. Vernon, J. Thompson, P. Bradley, S. Khare, M. D. Tyka, D. Bhat, D. Chivian, D. E. Kim, W. H. Sheffler, L. Malmström, A. M. Wollacott, C. Wang, I. Andre, and D. Baker, "Structure prediction for CASP7 targets using extensive all-atom refinement with Rosetta@home," *Proteins: Structure, Function, and Bioinformatics*, vol. 69, no. S8, pp. 118–128, 2007. doi:10.1002/prot.21636.

[6] "Folding@home client statistics.," [online]. 2008 [cited 2008-10-10].

[7] S. E. Jackson, "How do small single-domain proteins fold?," *Folding and Design*, vol. 3, pp. R81–R91, Aug. 1998. doi:doi:10.1016/S1359-0278(98)00033-9.

[8] A. Kolinski and J. Skolnick, "Assembly of protein structure from sparse experimental data: An efficient Monte Carlo model," *Proteins: Structure, Function, and Genetics*, vol. 32, pp. 475–494, Jan. 1998. http://dx.doi.org/10.1002/(SICI)1097-0134(19980901)32:4¡475::AID-PROT6¿3.0.CO;2-F doi:10.1002/(SICI)1097-0134(19980901)32:4<475::AID-PROT6>3.0.CO;2-F.

[9] A. Liwo, S. Oldziej, C. Czaplewski, U. Kozlowska, and H. Scheraga, "Parametrization of Backbone-Electrostatic and Multibody Contributions to the UNRES Force Field for Protein-Structure Prediction from Ab Initio Energy Surfaces of Model Systems," *J. Phys. Chem. B*, vol. 108, no. 27, pp. 9421–9438, 2004. doi:10.1021/jp030844f.

[10] A. Kolinski, "Protein modeling and structure prediction with a reduced representation.," *Acta Biochimica Polonica*, vol. 51, no. 2, pp. 349–371, 2004 [cited 2007-08-06].

[11] Y. Zhang and J. Skolnick, "Tertiary Structure Predictions on a Comprehensive Benchmark of Medium to Large Size Proteins," *Biophys. J.*, vol. 87, pp. 2647–2655, Oct. 2004. doi:10.1529/biophysj.104.045385.

[12] J. Moult, "A decade of CASP: progress, bottlenecks and prognosis in protein structure prediction," *Current Opinion in Structural Biology*, vol. 15, pp. 285–289, June 2005. doi:10.1016/j.sbi.2005.05.011.

[13] D. Chivian. "CASP7 server ranking for FM category (GDT MM)," [online]. 2006 [cited 2007-08-06].

[14] Y. Zhang. "CASP7 server ranking for FM category (TM-Score)," [online]. 2006 [cited 2007-08-06].

[15] J. N. D. Battey, J. Kopp, L. Bordoli, R. J. Read, N. D. Clarke, and T. Schwede, "Automated server predictions in CASP7," *Proteins: Structure, Function, and Bioinformatics*, vol. 69, no. S8, pp. 68–82, 2007. doi:10.1002/prot.21761.

[16] C. A. Rohl, C. E. M. Strauss, K. M. S. Misura, and D. Baker, "Protein Structure Prediction Using Rosetta," in *Numerical Computer Methods, Part D* (L. Brand and M. L. Johnson, eds.), vol. Volume 383 of *Methods in Enzymology*, pp. 66–93, Academic Press, Jan. 2004. doi:10.1016/S0076-6879(04)83004-0.

[17] S. Wu, J. Skolnick, and Y. Zhang, "Ab initio modeling of small proteins by iterative TASSER simulations.," *BMC Biol*, vol. 5, p. 17, May 2007. doi:10.1186/1741-7007-5-17.

[18] K. T. Simons, I. Ruczinski, C. Kooperberg, B. A. Fox, C. Bystroff, and D. Baker, "Improved recognition of native-like protein structures using a combination of sequence-dependent and sequence-independent features of proteins," *Proteins: Structure, Function, and Genetics*, vol. 34, no. 1, pp. 82–95, 1999. http://dx.doi.org/10.1002/(SICI)1097-0134(19990101)34:1¡82::AID-PROT7¿3.0.CO;2-A doi:10.1002/(SICI)1097-0134(19990101)34:1<82::AID-PROT7>3.0.CO;2-A.

[19] Y. Zhang, A. Kolinski, and J. Skolnick, "TOUCHSTONE II: A New Approach to Ab Initio Protein Structure Prediction," *Biophys. J.*, vol. 85, pp. 1145–1164, Aug. 2003 [cited 2007-03-13].

[20] Y. Zhang, I. A. Hubner, A. K. Arakaki, E. Shakhnovich, and J. Skolnick, "On the origin and highly likely completeness of single-domain protein structures," *PNAS*, vol. 103, pp. 2605–2610, Feb. 2006. doi:10.1073/pnas.0509379103.

[21] H. Chen and H.-X. Zhou, "Prediction of solvent accessibility and sites of deleterious mutations from protein sequence," *Nucleic Acids Research*, vol. 33, pp. 3193–3199, June 2005. doi:10.1093/nar/gki633.

[22] Y. Zhang, D. Kihara, and J. Skolnick, "Local energy landscape flattening: Parallel hyperbolic Monte Carlo sampling of protein folding," *Proteins: Structure, Function, and Genetics*, vol. 48, no. 2, pp. 192–201, 2002. doi:10.1002/prot.10141.

[23] W. Kabsch, "A discussion of the solution for the best rotation to relate two sets of vectors," *Acta Crystallographica Section A*, vol. 34, pp. 827–828, Sep 1978. doi:10.1107/S0567739478001680.

[24] E. A. Coutsias, C. Seok, and K. A. Dill, "Using quaternions to calculate RMSD," *Journal of Computational Chemistry*, vol. 25, no. 15, pp. 1849–1857, 2004. doi:10.1002/jcc.20110.

[25] D. Barthel, J. D. Hirst, J. Blazewicz, and N. Krasnogor, "ProCKSI: A Decision Support System for Protein (Structure) Comparison, Knowledge, Similarity and Information," *BMC Bioinformatic*, vol. 8, no. 1, p. 416, 2007. doi:10.1186/1471-2105-8-416.

[26] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics Dokl.*, vol. 10, pp. 707–710, Feb. 1966.

[27] W. R. Knight, "A Computer Method for Calculating Kendall's Tau with Ungrouped Data," *Journal of the American Statistical Association*, vol. 61, pp. 436–439, June 1966.

[28] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar, "Rank aggregation methods for the Web," in *Proceedings of the 10th international conference on World Wide Web*, (Hong Kong), pp. 613–622, ACM, 2001. doi:10.1145/371920.372165.

[29] C. Gagné and M. Parizeau, "Genericity in Evolutionary Computation Software Tools: Principles and Case-study," *International Journal on Artificial Intelligence Tools*, vol. 15, no. 2, pp. 173–194, 2006. doi:10.1142/S021821300600262X.

[30] G. Syswerda, "A Study of Reproduction in Generational and Steady State Genetic Algorithms," in *Foundations of Genetic Algorithms* (G. J. E. Rawlins, ed.), pp. 94–101, Morgan Kaufmann, 1990.

[31] D. E. Goldberg and K. Deb, "A Comparative Analysis of Selection Schemes Used in Genetic Algorithms," in *Foundations of Genetic Algorithms* (G. J. E. Rawlins, ed.), pp. 69–93, Morgan Kaufmann, 1990.

[32] J. R. Koza, *Genetic programming: on the programming of computers by means of natural selection and genetics*. MIT Press, 1992.

[33] S. Luke and L. Panait, "A survey and comparison of tree generation algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)* (L. Spector, E. D. Goodman, A. Wu, W. B. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. H. Garzon, and E. Burke, eds.), (San Francisco, California, USA), pp. 81–88, Morgan Kaufman, July 2001 [cited 2008-04-08].

[34] J. Bacardit, M. Stout, N. Krasnogor, J. Hirst, and J. Blazewicz, "Coordination Number Prediction using Learning Classifier Systems: Performance and Interpretability," in *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation (GECCO '06)*, pp. 247–254, ACM Press, July 2006. doi:10.1145/1143997.1144041.

[35] M. Stout, J. Bacardit, J. Hirst, R. Smith, and N. Krasnogor, "Prediction of topological contacts in proteins using learning classifier systems," *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, vol. 13, pp. 245–258, Feb 2009. doi:10.1007/s00500-008-0318-8.

[36] M. Stout, J. Bacardit, J. D. Hirst, and N. Krasnogor, "Prediction of recursive convex hull class assignments for protein residues," *Bioinformatics*, vol. 24, no. 7, pp. 916–923, 2008. doi:10.1093/bioinformatics/btn050.