Noname manuscript No.
(will be inserted by the editor)

GP challenge: evolving the energy function for protein structure prediction

Paweł Widera · Jonathan M. Garibaldi · Natalio Krasnogor

the date of receipt and acceptance should be inserted later

Abstract One of the key elements in protein structure prediction is the ability to distinguish between good and bad candidate structures. This distinction is made by estimation of the structure energy. The energy function used in the best state-of-the-art automatic predictors competing in the most recent CASP (Critical Assessment of Techniques for Protein Structure Prediction) experiment is defined as a weighted sum of a set of energy terms designed by experts. We hypothesised that combining these terms more freely will improve the prediction quality. To test this hypothesis, we designed a genetic programming algorithm to evolve the protein energy function. We compared the predictive power of the best evolved function and a linear combination of energy terms featuring weights optimised by the Nelder-Mead algorithm. The GP based optimisation outperformed the optimised linear function. We have made the data used in our experiments publicly available in order to encourage others to further investigate this challenging problem by using GP and other methods, and to attempt to improve on the results presented here.

Keywords genetic programming · protein structure prediction · protein energy function

P. Widera
School of Computer Science, University of Nottingham, Nottingham, NG8 1BB, UK
E-mail: plw@cs.nott.ac.uk

J. M. Garibaldi
School of Computer Science, University of Nottingham, Nottingham, NG8 1BB, UK
E-mail: jmg@cs.nott.ac.uk

N. Krasnogor (corresponding author)
School of Computer Science, University of Nottingham, Nottingham, NG8 1BB, UK
E-mail: nxk@nottingham.ac.uk

1 Introduction

Proteins are polymers, linear chains made of series of 20 different amino acids encoded in the genetic material (DNA or RNA sequence). Each amino acid includes an $\alpha - carbon$ (C_α) with bonds to amino (NH) and carboxyl (COOH) groups and a variable side chain (different for each type of amino acid). The amino acids in the protein chain are connected with a peptide bonds (CO-NH) formed between the amino and carboxyl groups, as shown at the top of Figure 1. The linked carbon, oxygen and nitrogen atoms form a protein backbone. This backbone forms several repeating local structures such as alpha helices, beta sheets or loops, known as secondary structure elements (see Figure 1). These elements and their spatial interrelations define the tertiary structure of a protein (a fold). A protein, under physiological conditions, spontaneously folds into a specific shape known as its native state.

The prediction of the tertiary structure of a protein in its native state is one of the greatest challenges in the field of structural bioinformatics. This field was launched in earnest over thirty years ago with the Nobel prize winning experiment of Christian Anfinsen [1]. He found proteins to always form the same native structures and concluded that a "folding algorithm" has to exist that uses only information contained in the protein sequence. To explain this phenomena Anfinsen assumed that a protein in its native state has a minimum free energy and described the process of folding as a minimisation of this energy. His explanation became later known as the thermodynamic hypothesis.

1.1 The state of the art methods in protein structure prediction

Since Anfinsen's refolding experiment it is widely believed that the native state corresponds to the thermodynamic equilibrium. Thus, *ab initio* prediction methods, which cannot rely on sequence similarity to known structures, use a concept of a free energy to find the (near) native state of a protein. This energy is defined from physical "first" principles as a function of structure and a structural model minimising it is searched for [6].

To represent the forces affecting protein molecules, several empirical force fields have been designed such as AMBER99, CHARMM22 or OLPS-AA [33]. Since the computational cost of these all-atom energy functions is very high, their practical applicability is limited to massively distributed projects like Folding@home (using 10 000 CPU days for $10\mu s$ of simulation) [36] or Rosetta@home (using 500 000 CPU hours per protein domain [1]) [14]. To reduce this heavy computational cost, several simplified models of proteins have been proposed such as SICHO [25], UNRES [31], CABS [24] or CAS [52]. In these models, groups of atoms are usually represented by a single group centroid resulting in a more coarse representation of a protein and lower processing times.

[1] Protein domain is a independent part of a protein chain that folds into distinct structural region. Its average size is around 100 amino acids in length. [45]

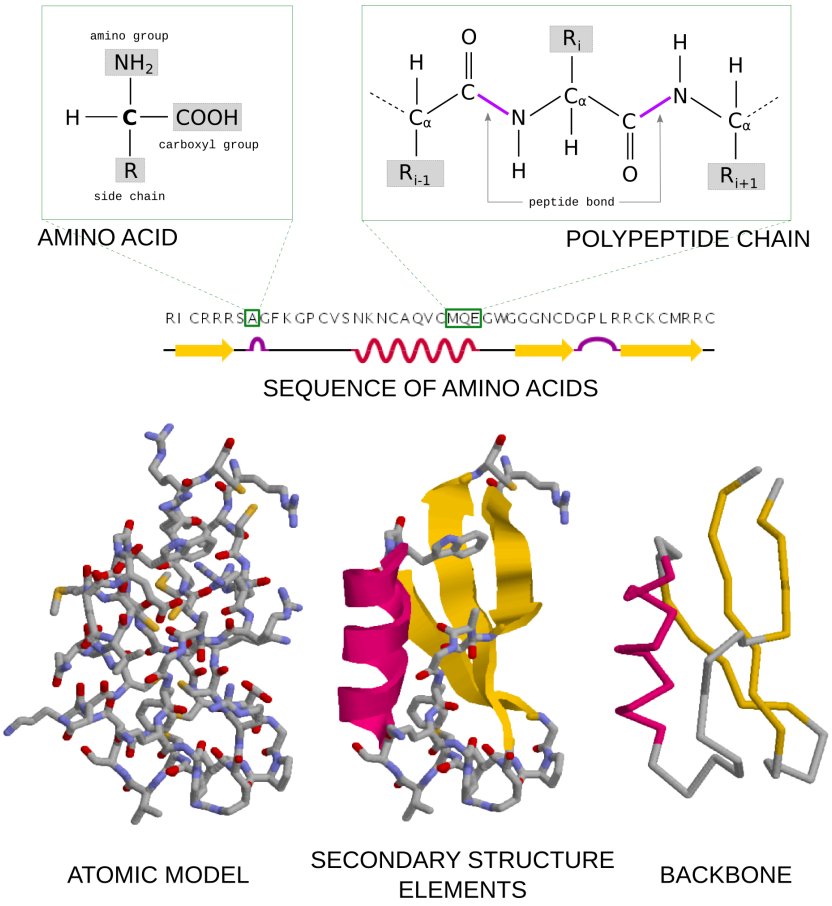

Fig. 1 The atomic structure of a single amino acid and the chain of amino acids is shown on top of a protein sequence. An all-atom representation of the protein colored by atom type, cartoon representation of secondary structure elements and a simplified representation illustrating the protein backbone colored by secondary structure is shown below.

The simplified models also use the notion of protein energy differently, to compensate for the loss of detail. In these models, the energy function incorporates knowledge-based potentials derived from a statistical analysis of the regularities seen in protein structures, as they actually occur in nature. This function does not capture the physical free energy explicitly. Instead, it represents the probability that a given structure is native-like.

The knowledge captured in potential terms is then algebraically combined into a single objective function. The weighted sum is used and the weights are selected through an optimisation process. To optimise these weights, a set of candidate protein structures, so called decoys, is generated by applying small random changes to a known native structure. For the decoys that are most similar to the native, the values of the energy function are expected

to be the lowest. Therefore, the optimisation objective is a maximisation of the correlation between the energy function and the similarity to the native structure. Similarity is usually measured by calculating the root mean square deviation (RMSD) between a decoy and the native structure.

This procedure is used in the two most successful prediction methods in the "template-free" category of the CASP7 experiment [10][48][4]: Robetta [37] and I-TASSER [46]. Robetta used a training set of 21 proteins (30 000 decoys each) and linear regression optimisation against RMSD [39]. I-TASSER used 30 proteins (60 000 decoys each) and maximised complex objective function with correlation to RMSD as its main element [51]. 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. [51] report it to be 0.54 for the naive sum of terms and 0.65 for the optimised weighted sum. This means that the energy function does not reflect the similarity very well and might be unable to distinguish between two similar models.

1.2 Evolving the energy function

The application of evolutionary algorithms (EA) to protein structure prediction is not new. Many methods have been studied in the past [43] using a variety of protein structure and energy models, ranging from a very simple H–P lattice model [16] in works of Krasnogor et al. [29][28] and Santana et al. [38] to all-atom force fields such as CHARMM used by Day et al. [15] and Cutello et al. [13] or AMBER used by Djurdjevic et al. [17]. All these methods, however, use the EA framework to optimise the model of protein structure with respect to a fixed, i.e. given, energy function. In our work, we focus instead on the improvement of the energy function itself.

We do this, because even the optimal model would be as good as the guidance of the energy functions used to optimise it. And as our analysis of the most successful prediction methods (I-TASSER and Robetta) suggests, the guidance of the energy function is far from perfect, as it is not highly correlated with similarity to the native, and thus it might be improved.

The aim of this paper is twofold: firstly, to test the hypothesis that a more general functional combination of energy terms will result in higher correlation between the energy function and the similarity of the candidate protein structures to the true native structure; and, secondly, to increase awareness of this challenging problem in order to encourage other researchers to attempt to further improve our results using GP or other optimisation methods.

To test our hypothesis we conducted a large number of experiments applying genetic programming to evolve the energy function used to evaluate protein structures. As a test set we used real decoys generated by I-TASSER predictor during the structure optimisation process [46]. We believe that this is more accurate than using decoys generated by randomisation of the native

structure (as in the original work of Zhang et al. [51]), because in practise, predictors have no a priori knowledge of the native structure. We have selected a subset of eight energy terms used by I-TASSER and pre-calculated their values for all decoys. Experiments were then carried out to evolve non-linear energy functions featuring a range of basic algebraic operators and transcendental functions. Using several different fitness measures we tried to determine how difficult it is to evolve an energy function that is highly correlated with structural similarity to the native state. As a baseline control experiment, we compared the best evolved energy functions with a linear combination of energy terms where weights were optimised using the Nelder-Mead downhill simplex method.

We hope that this paper will also encourage the GP community to engage in research aimed at evolving better energy functions. This is without doubts a very challenging problem for which even a modest advances could have great repercussions. To facilitate the adoption of this challenge all data used in our experiments are available online with detailed annotations (see Section 2.2).

2 Methods

2.1 Energy terms

We have implemented eight I-TASSER energy terms. These include: 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 [51][52] and the hydrogen bonds potential E_{HB} as defined in supplementary materials to [49].

The three $E_{i,i+n}$ energy terms represent $C_\alpha - C_\alpha$ interactions of the i-th residue with its n-th next neighbour. Each term measures the correlation of the local structure with the distribution of structural features extracted from the known structures (i.e. the negative logarithm of the frequency histogram derived from PDB database [5]). It depends on the amino acid type, predicted secondary structure and the distance between C_α atoms.

Stiffness potential E_{stiff} represents structural tendency towards a formation of the predicted secondary structure:

$$E_{stiff} = \sum_i \left(-\lambda \hat{v}_i \cdot \hat{v}_{i+4} - \lambda \left| \hat{b}_i \cdot \hat{b}_{i+2} \right| - \lambda \Theta_1(i) + \Theta_2(i) + \Theta_3(i) \right) \qquad (1)$$

$\hat{v}_i$ is a normalized (unit) vector between two consecutive carbon atoms $C_{\alpha,i}$ and $C_{\alpha,i+1}$. $\hat{b}_i$ is a unit bisector of the angle between $\hat{v}_{i-1}$ and $\hat{v}_i$ (see Figure 2). λ is a stiffness factor that differs for residues inside or outside of the protein radius of gyration (i.e. mean square distance from the center of mass). Three Θ functions represent structural bias in favour of predicted secondary structures and penalise irregularities.

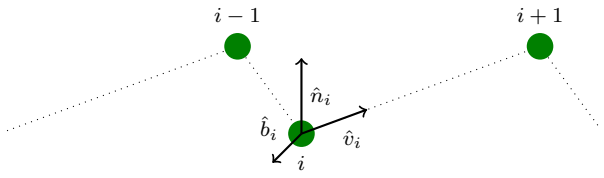

Fig. 2 Backbone vectors orientation.

Due to a lack of exact positions of all atoms in the CAS model (it uses only position of C_α atom and the side chain geometrical center of mass), hydrogen bonds are represented as a C_α packing preference supplemented with statistical parameters derived from known structures:

$$E_{HB_{ij}} = \begin{cases} \lambda_\alpha \frac{\left(1-\left|\hat{b}_i\cdot\hat{b}_j-b_{\alpha 0}\right|\right)(1-|\hat{n}_i\cdot\hat{n}_j-n_{\alpha 0}|)}{(1+||\epsilon\hat{n}_i-\mathbf{r}|-nr_{\alpha 0}|)(1+||\epsilon\hat{n}_j-\mathbf{r}|-nr_{\alpha 0}|)} & \text{if } i,j \text{ in } \alpha\text{-helix,} \\ \lambda_\beta \frac{|\hat{n}_i\cdot\hat{n}_j|\times\left(\hat{b}_i\cdot\hat{b}_j\right)}{\left(1+\frac{1}{2}|\epsilon\hat{n}_i-\mathbf{r}|\right)\left(1+\frac{1}{2}|\epsilon\hat{n}_j-\mathbf{r}|\right)} & \text{if } i,j \text{ in } \beta\text{-sheet.} \end{cases} \quad (2)$$

The location of hydrogen bonds in a specific secondary structure region is determined based on the contact order and relative orientation of $\hat{v}_i$, $\hat{v}_{i-1}$ and $\hat{v}_j$, $\hat{v}_{j-1}$ backbone vectors. $\hat{n}_i$ is a unit normal vector perpendicular to the plane containing $\hat{v}_i$ and $\hat{b}_i$ ($\mathbf{n}_i = \hat{b}_i \times \hat{v}_i$). $\lambda_{\alpha/\beta}$ is set to 1 if secondary structure prediction for residues i and j is the same (both α-helix or both β-sheet) and to 0.5 otherwise. *epsilon* is equal to 5Å for α-helix and to 4.6Å for β-sheet. Parameters $b_{\alpha 0}$, $n_{\alpha 0}$, $nr_{\alpha 0}$ are based on statistics derived from a set of 50 α-proteins and 50 β-proteins.

Long-range interactions are calculated between the side chain centers of mass. The potential is represented here by a hard-core sphere of excluded volume interactions and a soft-core $1/r$ type potential outside the sphere:

$$E_{pair} = \sum_{j>i} E_{ij}(s_{ij}) \quad (3)$$

E_{ij} is set to 0 outside the soft-core ($s_{ij} > R_{max_{ij}}$), to 4 inside the hard-core ($s_{ij} < R_{min_{ij}}$) or to statistical pairwise potential e_{ij} if between.

Electrostatic effects are included using a form of Debye-Hückel equation:

$$E_{electro} = \sum_{j>i} \frac{\exp(-ks_{ij})}{s_{ij}} \quad (4)$$

k is the inverse Deybe length, where $\frac{1}{k} \sim 15\mathring{A}$ was experimentally chosen by Zhang et al.

Potential describing the contact environment is based on bonds geometry and is derived from the set of PDB structures:

$$E_{env} = \sum_i V_i \quad (5)$$

V_i represents amino acid specific potential and depends on number of residues that are in contact with the ith residue for which their bisectors are parallel ($\hat{b}_i \cdot \hat{b}_j > 0.5$), anti-parallel ($\hat{b}_i \cdot \hat{b}_j < -0.5$) or perpendicular (dot product in $[-0.5, 0.5]$ range). Residues are considered to be in contact if $s_{ij} < R_{min_{ij}}$.

We left out potentials using data from the threading process (e.g. distance map or contact order) and the hydrophobic potential introduced in [46] using neural network [9] as they depend on external feature predictors which were not available for local use at the time of writing this paper and would have made pre-calculation much slower (see Section 5).

2.2 Construction of the ranking

In our optimisation experiments we have used 54 small non-homologues protein chains used by Zhang et. al [46] [2]. From the set of decoys generated by I-TASSER during the Monte Carlo based structure optimisation process [50] (available online [46]) we have taken a 10% sample (one decoy every 10 I-TASSER iterations) to eliminate highly similar decoys. This resulted in a training set of 1250–2000 decoys per protein. For each decoy we have pre-calculated the values of energy terms described in Section 2.1.

For each protein we have measured the similarity between the generated decoys and 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 [22][11]. We decided to use RMSD despite its known problems with accuracy [12][44][47] mainly for the sake of comparison to the previous work.

To compensate for the inaccuracies of the RMSD as a similarity measure, we decided not to take into account the absolute value of the similarity, but just the relative rank. For given decoys A and B we decide only if $RMSD(A, native) < RMSD(B, native)$ and we ignore the scale of absolute difference in the distance to a native $\delta = RMSD(A, native) - RMSD(B, native)$. By doing this, we also simplify the optimisation objective, as linear hierarchy is easier to reflect in energy function than exact distances between all pairs of decoys.

For each protein we sorted all decoys in increasing order of the original I-TASSER energy to obtain the initial ranking R_0 (see Figure 3). That is, decoys with low ranks (near the begining of the ranking) have lower energy, while those with high ranks have higher energy. Based on R_0 we created two variants of the final ranking R_1 and R_2. Ranking R_1 was constructed by sorting R_0 by RMSD (in ascending order). In case of a tie, the rank from R_0 was used as a second sorting criterion. Effectively R_1 is a permutation of R_0. R_2 in contrast, is a list of ranks assigned to each element of R_0 according to its position in R_1. The ranks in R_2 were averaged in case of a RMSD tie. A tie between decoys was called when RMSD values were the same up to the first

[2] From the original set of 56 protein we have excluded *1ogwA* (it contains LEF - a non-standard amino acid) and *1cy5A* (by omission)

two decimal places. This gave us a precision of a 1 picometer (for reference, the radius of hydrogen atom is 25 pm).

To ensure reproducibility and to encourage other researchers to work on this challenging problem, we made all data used in GP optimisation, i.e. energy terms and RMSD distance to native for each decoy, available at: `http://www.infobiotics.org/gpchallenge/`.

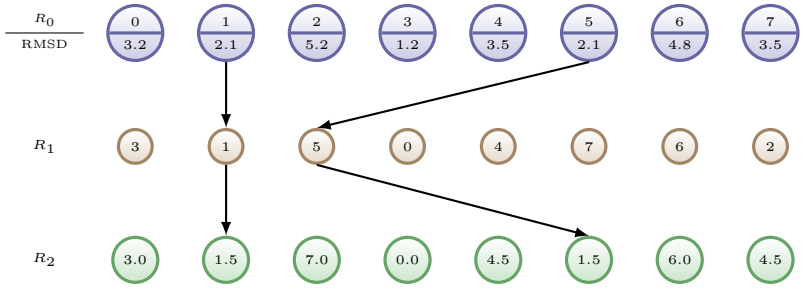

Fig. 3 The figure shows the two approaches we used to construct the ranking. R_0 is the initial ranking where decoys are sorted by the original I-TASSER energy (leftmost decoy has the lowest energy). R_1 is a permutation of R_0 where decoys are sorted by RMSD (leftmost decoy is most similar to a native structure) and a decoy index from R_0 is used to break ties (e.g. decoys 1 and 5 share the same RMSD value but as $1 < 5$, decoy 1 precedes 5 in R_1). R_2 is a list of ranks based on the order of decoys in R_1 (e.g. decoy 0 is in position 3 in R_1, so $R_2[0] = 3$). In case of RMSD ties the ranks in R_2 are averaged (e.g. decoys 1 and 5 have the same RMSD values, therefore they share the average rank of $\frac{1+2}{2} = 1.5$).

2.3 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 Section 2.1 (see Table 3 for the mapping to I-TASSER terms), half were ephemeral random constants in range [-1,1]. Half of the operators were binary (addition, subtraction, multiplication, division), half were unary (sine, cosine, exponential function, natural logarithm). We did not impose any selection bias towards any of the primitives.

The key element of our evolutionary mechanism is the objective function used to calculate the fitness of the candidate energy functions in the population. For each protein the objective function was used to rank the decoys using the evolved energy function. Then this ranking was compared to the reference ranking (obtained in preprocessing stage) and the normalised distance between the two was averaged for all proteins in the training set, producing the total fitness.

We used several different methods to calculate the distance between rankings:

- *Levenshtein edit distance* [30], 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,
- *Spearman footrule distance* [18], the sum of differences between the ranks of elements,
- *Kendall tau distance* [23], the number of inversions between two permutations also known as the bubble-sort distance.

For the Spearman distance we have applied an additional weighting mechanism to promote correct order at the beginning of the ranking (more native-like) and to be more forgiving for 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,

$$w(i) = 1 - i/N, \text{ for } 0 \leq i < N \tag{6}$$

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

$$w(i) = \frac{1}{1 + \exp(\frac{i - 0.25N}{scale * N})}, \text{ where } scale = \begin{cases} \frac{0.25N}{width} & \text{if } i < 0.25N \\ \frac{0.75N}{width} & \text{if } i \geq 0.25N \end{cases} \tag{7}$$

Additional experiments were performed with the reduced data sets. Instead of using all decoys for each protein, we used a small sample of decoys. We have used two sampling methods: simple selection of k decoys and noise filtering. The first method was used with $k = 100$ in three variants: random selection, uniform selection (every k/nth decoy), decoys with the lowest original I-TASSER energy. The goal of the noise filtering method was to obtain a uniformly sampled set of decoys for which the original I-TASSER energy is correlated with similarity. Starting from the decoys most similar to the native, the set of all decoys was divided into bins. Two variants of bins were used: equal size bins (same number of decoys) and equal distance bins (same RMSD range). From each bin a single most similar decoy was selected such as its original I-TASSER energy value was greater than for the decoy selected from the previous bin. If none such decoy existed no sample was selected from the bin. Along arbitrary selected number of bins $b = 100$ we also used b for which the average number of samples obtained for all proteins was the greatest. We found it to be 42 for equal size bins and 58 for equal distance bins.

We have implemented the genetic programming algorithm using the Open BEAGLE framework [19]. Base GP configuration used in all our experiments included two replacement strategies: generational and steady-state [42], the tournament selection [20] and the population size set to 100. Mutation was done using three different operators: sub-tree replacement with a random tree, sub-tree swap or tree shrink where a tree node is replaced by one of its child nodes. Table 1 shows the probabilities of all evolutionary operators used in our experiments. This configuration is derived from an initial exploratory trial (not reported here).

operation	operator	probability
crossover	non-leaf crossover point	0.70
	leaf crossover point	0.10
mutation	sub-tree replacement	0.05
	tree shrink	0.05
	sub-tree swap	0.05
reproduction	copy without modification	0.05

Table 1 Summary of the evolutionary probabilities used in all experiments.

We have run our experiments in three rounds changing the factors of the next round based on the results of the previous one. As a result, the first two rounds are of an exploratory nature, while the third round is more aggressive towards increasing the correlation. Configuration of all rounds is shown in Table 2.

In the first round we used Levenshtein, Kendall and non-weighted Spearman distances. In the second round the ineffective Levenshtein distance was rejected and the linear and sigmoid weighting was added to the Spearman distance. To have more selection pressure we changed the tournament size from 2 in previous round to 4, 6 and 8. In the third round we used only sigmoid weighted Spearman's distance together with the list of ranks R_2 instead of permutational ranking R_1 used in both previous rounds. The tournament size has been set to 8 for generational and 6 for steady-state replacement. Both these strategies were used alone, with strong 5-elitism or with automatically defined functions (ADF) operators [27]. The set of all decoys was extended with 5 reduced sets and the number of generations was increased to 2000 from 1000 used in previous rounds.

	Round I (6)	**Round II (18)**	**Round III (36)**	
measure of distance	Kendall Spearman Levenshtein	Kendall Spearman linear Spearman sigmoid	Spearman sigmoid	
replacement strategy	steady-state generational	steady-state generational	steady-state +elitism +ADF	generational +elitism +ADF
tournament size	2	4, 6, 8	6	8
generations	1000	1000	2000	
ranking type	R_1	R_1	R_2	
decoy sets	all	all	all, top, random, uniform, equal size, equal distance	

Table 2 Configuration of all the three rounds of experiments. The number in brackets is the number of experiments run in each round.

In all experimental rounds, a random walk was performed as a baseline for comparison. At each generation the population was created using the half-and-half initialization operator [26][32].

In total we have conducted 60 different experiments using over 100 CPU days to perform 300 GP runs.

3 Results

3.1 I-TASSER energy terms

The average correlation between the original I-TASSER energy and the similarity to the native structure measured by RMSD is shown in Figure 5. Each panel represents a set of decoys for a single protein and the Pearson correlation coefficient $\rho(x, y) = \frac{cov(x,y)}{\sigma_x \sigma_y}$ is given in brackets. The average correlation coefficient for all proteins was 0.44 ± 0.23 (second value is a standard deviation). Interestingly, even the highest correlation coefficient e.g. for *1f05A* or *2f3nA* (see the close-up in Figure 4), is not enough to point to the most native-like structure as we observe a flat cloud in the lowest energy region stretched over distance of 0.1–0.2nm.

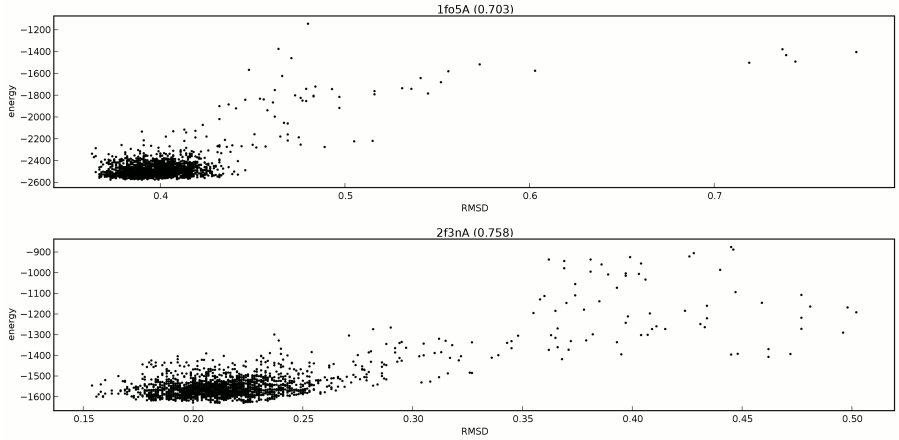

Fig. 4 Scatter plots of I-TASSER energy vs. RMSD. Close-up for two proteins with high correlation coefficient (given in brackets). Distance is given in nanometers.

The average correlation coefficient for the naive sum of energy terms $E_N = \sum_{i=1}^{8} T_i$ was 0.12 ± 0.16. Coefficients for individual terms are shown in Table 3. 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 correlation between original I-TASSER energy and rank is shown in Figure 6. It was almost 50% lower than in case of RMSD with the total average

energy term	ρ_1	ρ_2	σ_2	ρ_E	σ_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	0.36	0.04	0.17	0.09	0.19

Table 3 Both ρ_1 and ρ_2 represent the average correlation between a single energy term and the similarity to native structure measured by RMSD. ρ_1 is the coefficient originally reported by Zhang et al. [51] 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 correlation coefficient between a single energy term and the original I-TASSER energy. σ_2 and σ_E represent the standard deviation of ρ_1 and ρ_2 coefficients. In case of the hydrogen bonds potential E_{HB}, ρ_1 and ρ_2 cannot be directly compared, as the latter apply to the new implementation of this term [49].

of 0.25 ± 0.16. The vertical stripes are visible in regions were several decoys which are equally distant from the native have different energy. Correlation between the naive sum of energy terms and the rank was also lower with the average coefficient value of 0.07 ± 0.16.

3.2 First round of experiments

As could be seen in Figure 8 the average fitness for the Levenshtein distance oscillated in a tiny range just above zero (the maximum distance). For the Spearman distance the average fitness improved quickly in the first 50–100 generations and saturated later around 40% of the maximum fitness. The initial improvement was more rapid in case of the Kendall distance but the spread of the fitness was very small, covering only 3% range of the maximum fitness. The best evolved functions were only slightly (1.3% and 5.5% for the Kendall and Spearman distances respectively) better than the best function found by the random walk.

To check the statistical significance of the results we have performed the Kruskal-Wallis one-way analysis of variance to test the null hypothesis that medians of two average fitness distributions shown in Figure 8 are equal. The hypothesis was rejected for both 9 vertical (across measures of distance) and 9 horizontal (across configuration) pairwise comparisons at the 99.9% confidence level ($p < 0.001$).

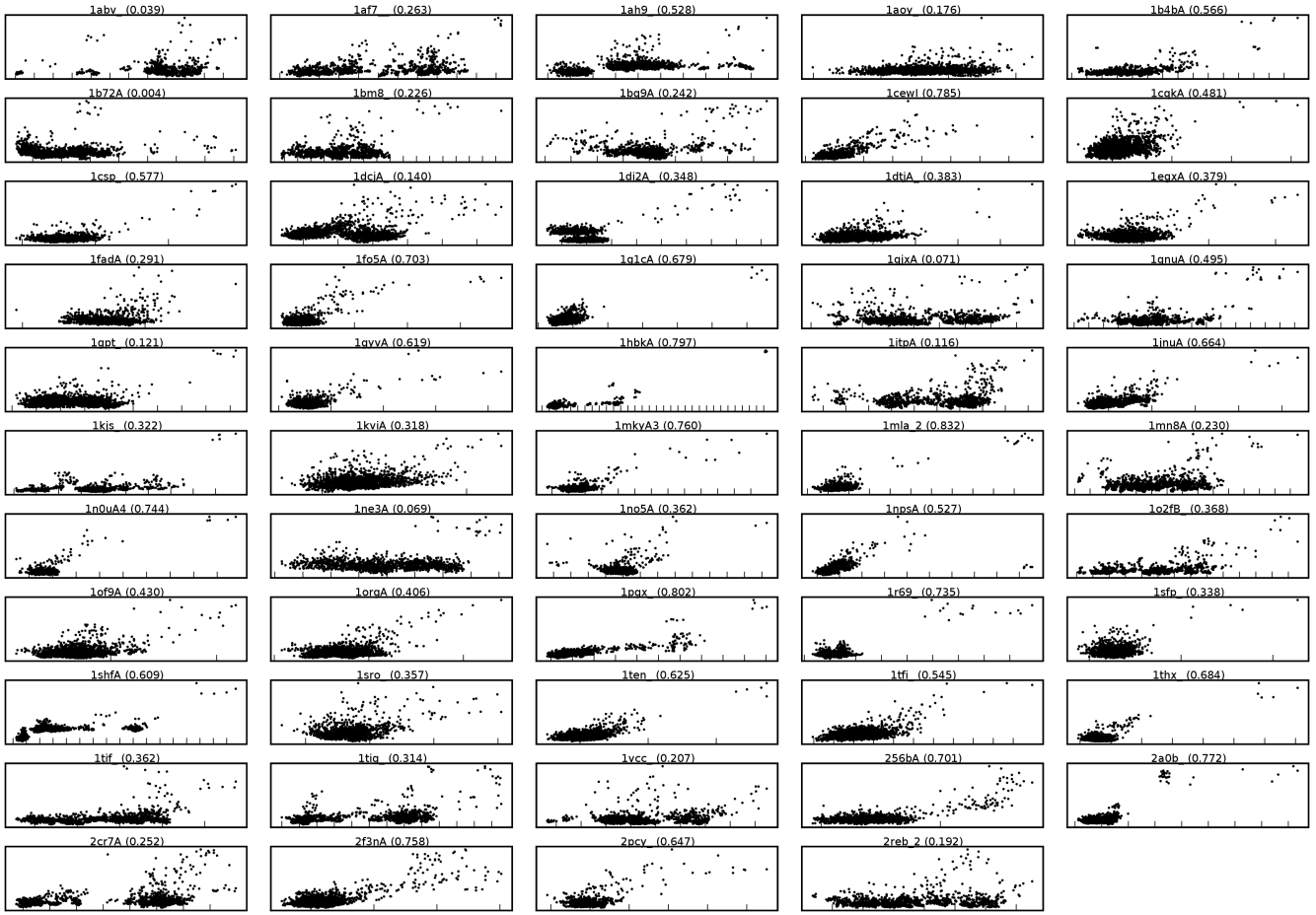

Fig. 5 Scatter plots of I-TASSER energy (vertical axis) vs. RMSD (horizontal axis). Each plot contains all decoys for a single protein. Correlation for each plot is given in brackets. Distance between ticks on horizontal axis is 0.1nm. A high resolution version of this figure is available at `http://www.infobiotics.org/gpchallenge/scatter-plots/`.

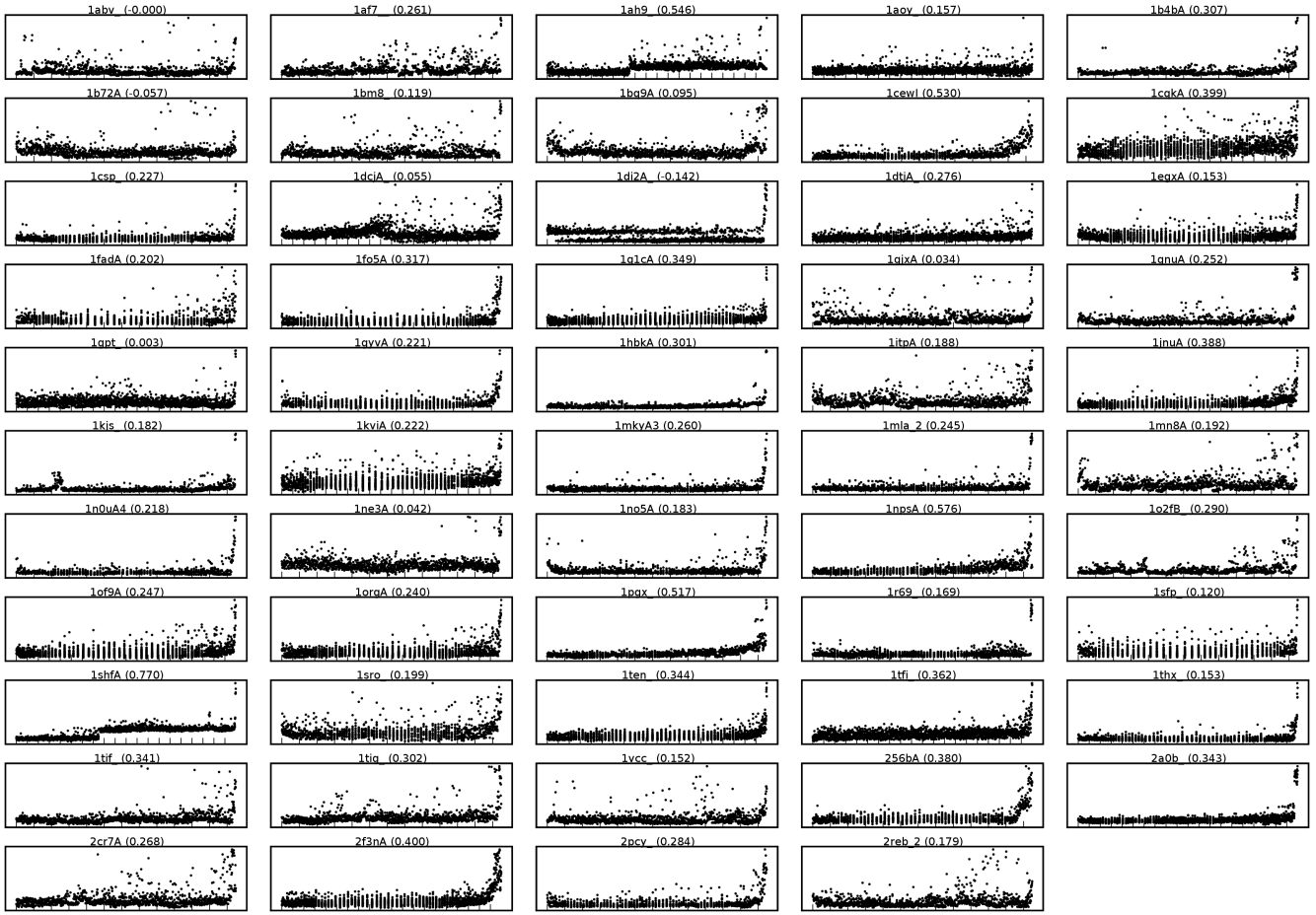

Fig. 6 Scatter plots of I-TASSER energy (vertical axis) vs. rank (horizontal axis). Each plot contains all decoys for a single protein. Correlation for each plot is given in brackets. Distance between ticks on horizontal axis is 100 ranks. A high resolution version of this figure is available at `http://www.infobiotics.org/gpchallenge/scatter-plots/`.

3.3 Second round of experiments

The fitness landscape was not changed much by the linear weighting. But for the sigmoid weights both average and maximum fitness values were over 20% higher than for the non-weighted Spearman distance as shown in Figure 7.

The evolutionary progress for the best GP configurations with both weighted Spearman distances continued slowly, in contrast to the early saturation observed in the first round of experiments. Despite that and a greater improvement over the random walk, the maximum fitness values were still in the 0.4–0.5 range.

We did not observe a significant change in the evolutionary process for bigger tournaments. As a result, for the next round of experiments we arbitrarily picked the tournament size 6 for the steady-state replacement and 8 for the generational one, just by the visual assessment of the fitness vs. generation plots.

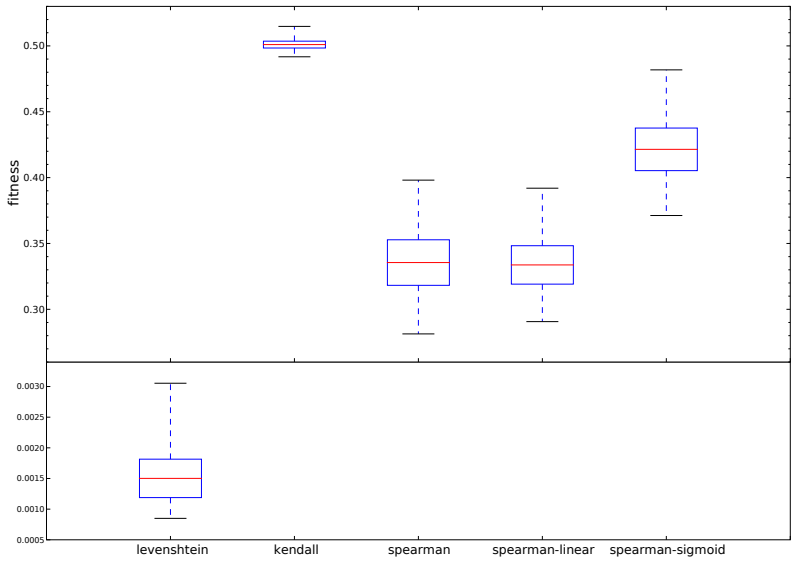

Fig. 7 Box plot of the fitness distribution achieved by a random walk for ranking distances used in first two rounds of experiments. Middle line is the median of the average fitness in population across all generations. Box size represents the median of the population fitness standard deviation. Top and bottom whiskers marks maximum and minimum fitness across all individuals.

3.4 Third round of experiments

Since in this round we used averaged ranks and in consequence the ranking was not a permutation any more, we did not use the Kendall distance. The fitness

function based on the sigmoid weighted Spearman distance was used in all experiments. The addition of ADFs or strong elitism improved the best fitness in several cases but we did not find any tendency that would be common for all sets of decoys.

For the set of all decoys, the value of the average and maximum fitness was around 40% higher compared to the permutational ranking. However, as shown in Table 4 the evolution process could not improve significantly over the random walk, despite the fact that the best randomly found energy function was as simple as $f = 0.412/T_6$. Similar results were obtained for the three sets of 100 selected decoys: random, uniform and top. For the noise filtered set of decoys, where the GP trees of the best functions were over two times larger, the improvement over random walk was greater than 7%.

		best fitness		**best tree size**		**best tree depth**	
decoys set	**improvement**	**max**	**avg**	**max**	**avg**	**max**	**avg**
all	0.78%	0.714	0.710	380	186.2	18	16.8
uniform-100	0.96%	0.716	0.710	289	151.5	18	17.3
random-100	1.28%	0.720	0.713	289	151.5	18	17.3
top-100	1.93%	0.709	0.701	266	118.7	18	15.5
f-42	7.76%	0.729	0.713	1110	485.0	18	17.7
f-100	7.64%	0.788	0.772	629	414.7	18	17.5
d-58	8.21%	0.809	0.780	752	388.7	18	17.3
d-100	10.88%	0.835	0.804	793	329.2	18	17.5

Table 4 Comparison of the best evolved functions for different sets of decoys. Second column shows percentage fitness improvement over the random walk. The next columns show the maximum and the average value of fitness, tree size and tree depth for the best functions of all six run configurations.

The change in fitness landscape for the noise filtered sets of decoys is shown in Figure 11. The range of fitness values for the random walk increased up to 40% of the maximum fitness from 15% observed previously for all decoys. Although the average fitness was lower, the fitness of the best individuals has improved reaching 0.75 for the set of decoys created using 100 equal distance bins (d-100).

The overall fittest evolved individual and the greatest evolutionary improvement over the random walk was observed again for the d-100 set. The best evolved energy function had almost 11% greater maximum fitness (0.835) than $f = 0.3433 * T1 + T3 - T6$ found by chance. However, the GP tree was difficult to analyse because of a bloat. As we did not introduce any size penalty in the fitness function, the average size of a GP tree was increasing through generations as shown in Figure 10.

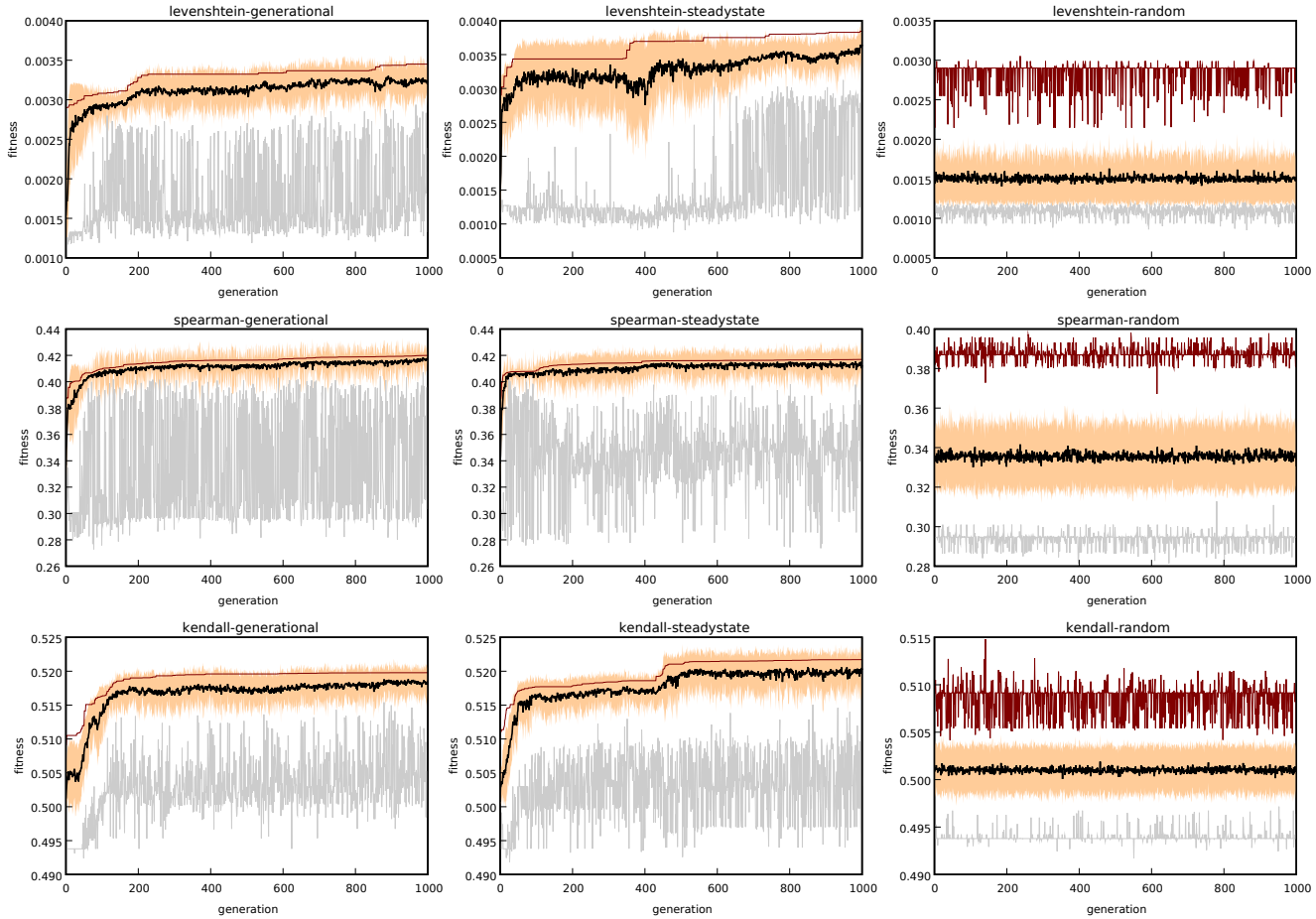

Fig. 8 Fitness throughout the generations in first round of experiments. Lines show the average (thick black), minimum (thin grey) and maximum (thin red) fitness in the population. Filled area around the average represents the standard deviation. Each row corresponds to a single fitness function and each column corresponds to a single GP configuration.

Fig. 9 Scatter plots of the best energy function evolved for d-100 set (vertical axis) vs. RMSD (horizontal axis). Red horizontal line marks the energy of the native structure. Each plot contains all decoys for a single protein. Correlation for each plot is given in brackets. Distance between ticks on horizontal axis is 0.1nm. A high resolution version of this figure is available at `http://www.infobiotics.org/gpchallenge/scatter-plots/`.

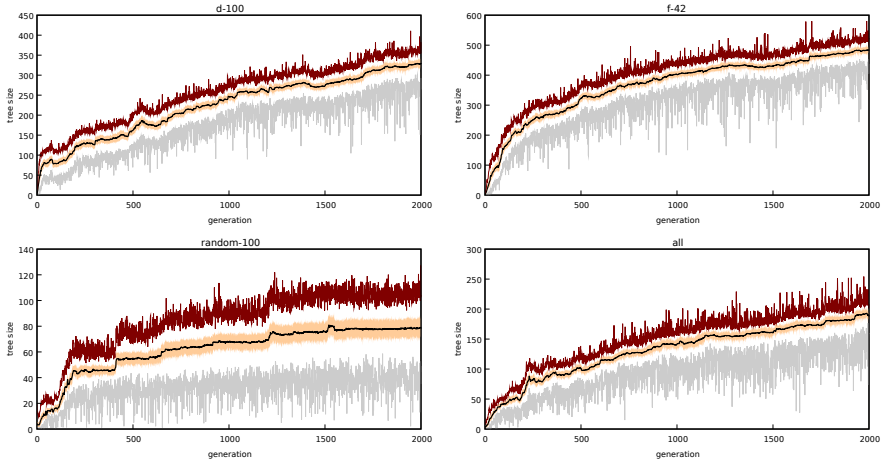

Fig. 10 Tree size throughout the generations. Sizes are averaged over all six GP configurations for a selected set of decoys from the third round of experiments. Lines show the average (thick black), minimum (thin grey) and maximum (thin red) tree size in the population. Filled area around the average represents standard deviation.

The distribution of the terminals and operators used in the best functions evolved for each set of decoys is summarised in Table 5. The most frequently used energy terms were T_4 and T_5. Interestingly, T_4 had the highest correlation to RMSD and T_5 the second lowest one (see Table 3). Similarly, the least frequently used energy terms, T_1 and T_6, were the ones with the correlation to RMSD closest to zero. Therefore, the GP optimisation based on the distance between ranks was able to discover an analogous hierarchy of the energy terms. Across the operators, the most frequent were addition and division with transcendental functions (sine, cosine and natural logarithm) being the least frequent.

decoys set	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	**add**	**sub**	**mul**	**div**	**log**	**exp**	**sin**	**cos**	**total**
all	0	0	10	18	19	3	10	1	99	2	0	0	8	104	4	4	323
uniform-100	4	0	0	1	2	1	3	4	0	15	1	0	4	21	0	1	59
random-100	1	4	5	3	1	5	14	6	6	1	27	46	12	24	2	4	203
top-100	0	0	0	0	0	3	9	9	0	0	9	61	4	48	64	3	260
f-42	11	49	76	84	26	3	105	19	310	38	25	174	5	8	1	1	1110
f-100	1	42	45	50	48	14	3	33	158	39	65	22	14	32	13	1	629
d-58	26	8	1	40	69	32	21	3	193	34	26	90	4	28	6	27	752
d-100	6	6	12	60	41	17	19	22	52	102	37	78	0	34	0	17	590

Table 5 The distribution of terminals and operators in the best evolved functions for different sets of decoys. Ephemerals are not shown.

The best energy function evolved for the d-100 set was correlated to RMSD with the coefficient of 0.76 ± 0.19 and to the rank with coefficient of $0.74 \pm$

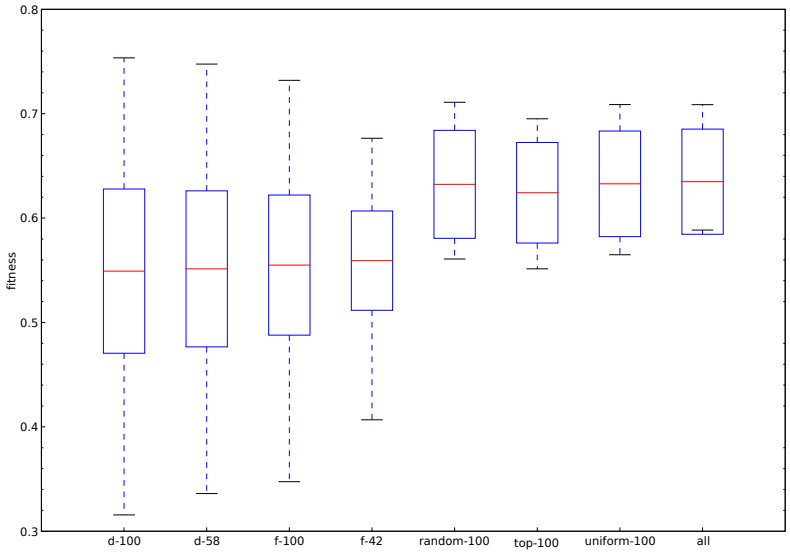

Fig. 11 Box plot of the fitness distribution achieved by a random walk with sets of decoys used in the third round of experiments. Middle line is the median of the average fitness in population across all generations. Box size represents the median of the population fitness standard deviation. Top and bottom whiskers marks maximum and minimum fitness across all individuals.

0.18. When the best evolved function was applied to the set of all decoys, the corresponding correlation coefficients dropped to 0.30 ± 0.18 (shown in Figure 9) and 0.20 ± 0.17. Still, compared to naive combination of terms the evolutionary optimisation improved the correlation coefficients 2.5 and 2.85 times respectively.

3.5 Comparison to the linear combination of terms

As we shown before, the best energy function found by our GP algorithm provide significantly better prediction guidance than the naive combination of terms or best functions found by the random walk. Moreover, the GP algorithm was able to automatically discover the most and the least useful energy terms without having any knowledge how these terms alone are correlated to RMSD.

To put these results in context, we used the Nelder-Mead downhill simplex method [35][34] to find the best weights of the energy function given as a linear combination of terms $E_L = \sum_{i=1}^{8} w_i T_i$, similar to what have been done in the original work by Zhang et al. [51]. We ran the SciPy [21] implementation of the algorithm using a vector of weights $\mathbf{w} = [w_1, \ldots, w_8]$ as a variable and minimising either the sigmoid weighted Spearman distance or the correlation coefficient between energy and rank directly. The method converged in a fraction of time allowed for GP optimisation (minutes vs. hours) performing

on average only about 500 objective function evaluations. Table 6 shows the maximum objective function values obtained for d-100 and all decoy sets compared with the results of the best evolved functions. The fitness of GP-evolved functions was in all cases over 10% higher.

	spearman-sigmoid		**correlation**	
method	**d-100**	**all**	**d-100**	**all**
simplex	0.734	0.638	0.650	0.166
GP	0.835	0.714	*0.740	*0.200

Table 6 The results of the simplex method optimisation of the weighted sum of terms compared to the best GP-evolved functions. Notice that the correlation coefficient for GP marked with star was calculated after the evolutionary optimisation, while in case of the downhill simplex method it was directly used as an objective function.

3.6 Population diversity analysis

The mapping between the tree representation of a function and it's fitness is very complex as it involves an evaluation of the energy of thousands of decoys and a comparison of the evolved ordering with a reference ranking for several proteins. It would not be surprising if this mapping will result in the loss of diversity between the levels as shown on Figure 12.

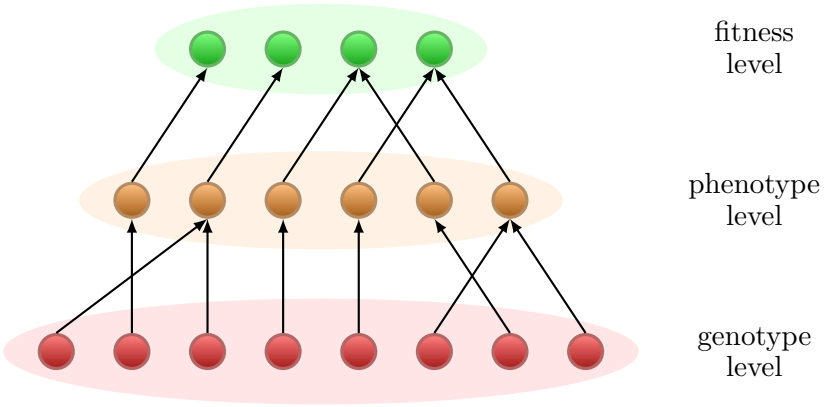

Fig. 12 Illustration of possible mapping between GP trees, decoys ordering created by the evolved function and the total fitness.

To gain more insight into the evolutionary process and usefulness of the proposed fitness functions, we have collected a wide range of population diversity statistics for the best configurations from the last round of experiments. Burke et al. [8][7] have shown that to understand the evolutionary dynamics

the diversity should be measured on several levels. As suggested there, we measured the population diversity on three levels:

- *genotype*, we measured a number of unique trees using a pseudo-isomorphism measure [7], where each tree is described by a triple <# terminals, # non-terminals, tree depth>
- *phenotype*, for each of n individuals in the population we generated decoy rankings and measured the average root mean square distance between them (using the Spearman footrule distance), which we then averaged for all m proteins obtaining phenotype rmsd:

$$\frac{1}{m}\sum_{p=0}^{m}\frac{2}{n(n-1)}\sqrt{\sum_{i=0}^{n}\sum_{j>i}^{n}d(r_{pi},r_{pj})^2} \tag{8}$$

- *fitness*, we measured the entropy in the population based on the frequency of occurrence of fitness values (using a precision of three decimal places):

$$\sum_{i=0}^{n}p_i * log(p_i), \text{ where } p_i = \frac{1}{n}duplicates_i \tag{9}$$

For individual runs we have observed a rapid loss of the diversity on both genotype and phenotype levels after a few initial generations. However, it was not the case for the diversity on the fitness level, which usually did not decrease even for late generations. Moreover, for the best individual runs the maximum fitness is not stagnating but slowly improving throughout 2000 generations (see Figure 13). Hence, the early saturation of the average fitness does not seem to be related to a converged population.

We have analysed the common diversity characteristics of a group of the most successful runs and we found the best evolutionary progress to be related to a gradual decrease of the phenotype diversity and high or increasing diversity on the fitness level (see runs A and C in Figure 13). Interestingly, a high diversity on the fitness or phenotype level alone did not result in good evolutionary progress (see runs D and E in Figure 14). A high diversity on tree level for late generations seems to indicate cases when evolution was trapped around a low quality local optima. Additional diversity plots are available at `http://www.infobiotics.org/gpchallenge/diversity/`.

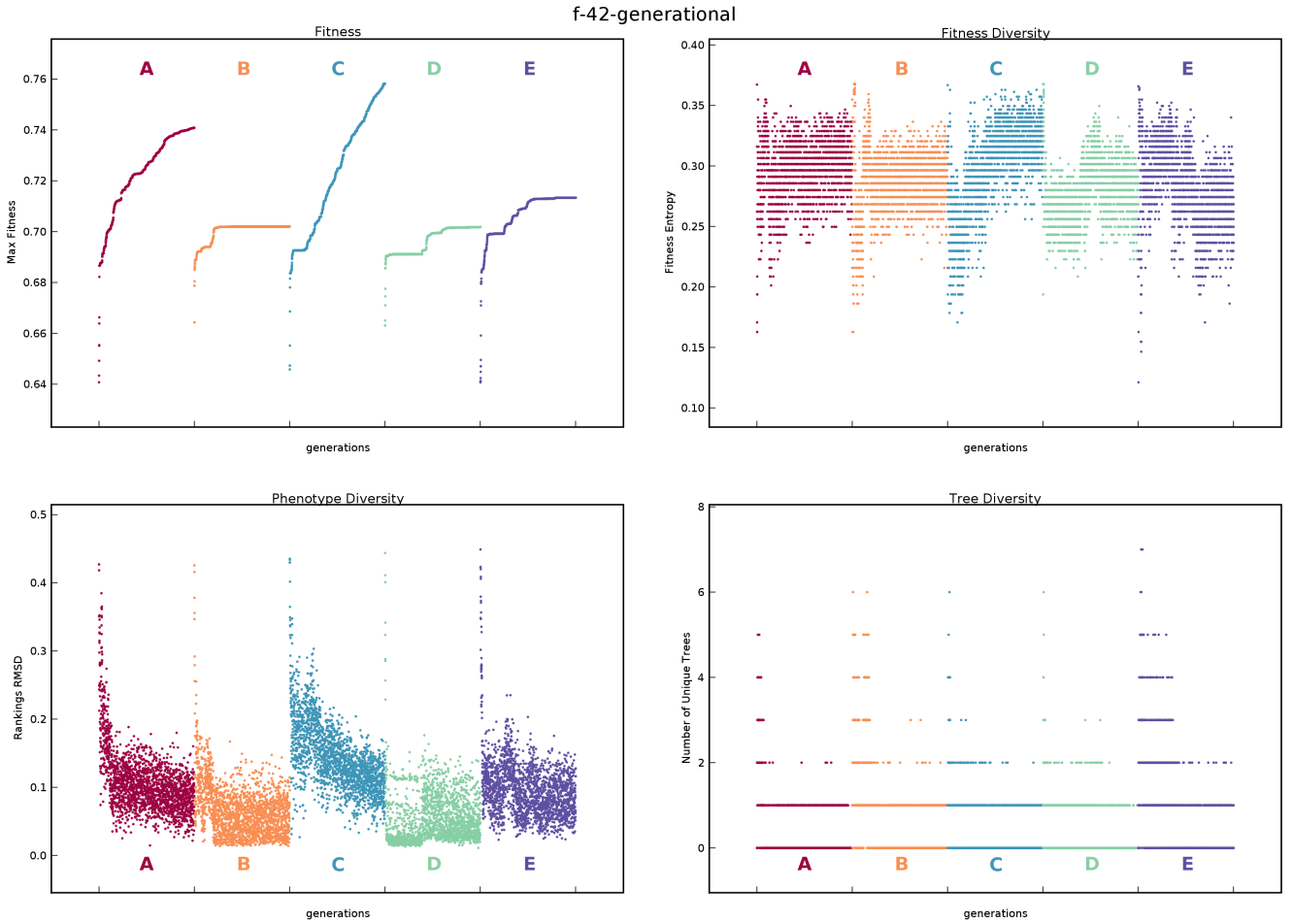

Fig. 13 Maximum fitness throughout generations compared to population diversity on fitness, phenotype and GP tree levels. Each plot shows side by side five different runs (A-E) for a selected GP configuration.

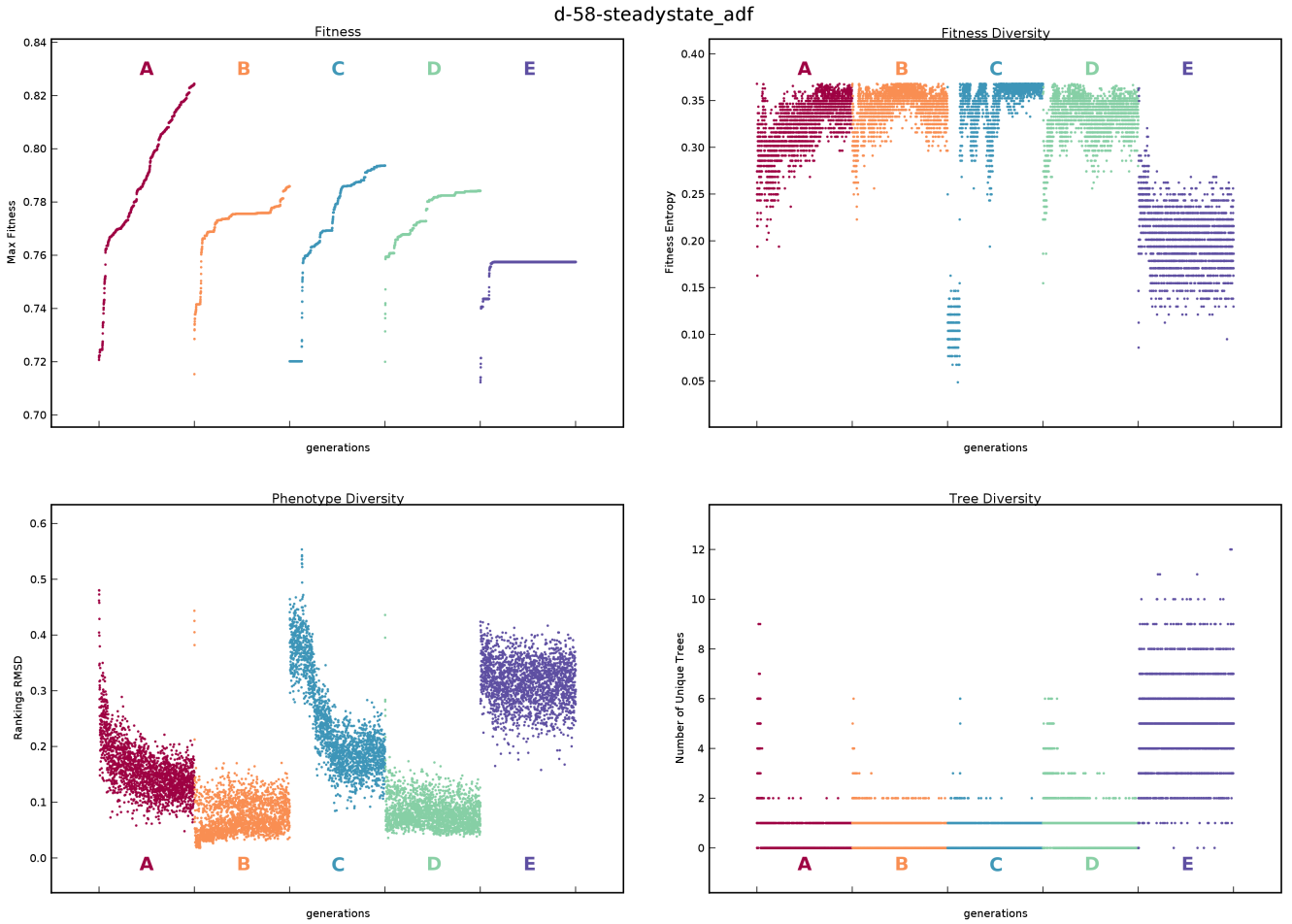

Fig. 14 Maximum fitness throughout generations compared to population diversity on fitness, phenotype and GP tree levels. Each plot shows side by side five different runs (A-E) for a selected GP configuration.

4 Discussion

In protein structure prediction a useful energy function is the one which guides the structural optimisation process towards the region of native-like structures. Therefore, it seems natural to measure this usefulness with a correlation coefficient between the energy and similarity to native. However, as we have shown, even a high correlation coefficient (> 0.7) does not guarantee that distinguishing the native-like structure from the others would be easy. This is reflected in the lower correlation to rank, since ranking ignores the scale. The lack of power to differentiate between the decoys is best observed on the energy vs. rank plots, where for several consecutive rank bins the assigned decoys are within the same energy range.

The difference between the correlation of single energy terms in our experiments and in the original work by Zhang et al. shows the difference in difficulty of the decoy sets used. It seems to be more difficult to choose a native-like structure from the set of decoys sampled from the structural optimisation process, than from a set generated by randomisation of the native used by Zhang. The former starts from fragments of other proteins similar to the target on a sequence level and has no knowledge of its native structure. The latter is using the native structure directly resulting in a biased set. Moreover, the decoys used in our experiments are often very similar to each other, whereas Zhang kept them separated by large 0.35nm RMSD distance. As our results show, decoys generated by the predictor are more difficult to assess and thus optimising the energy based on the randomised and highly separated set of decoys might be inadequate as this is not what predictors have to deal with in practise.

Although the naive combination of energy terms used in our experiments compared to the original I-TASSER energy was much less correlated to RMSD, the genetic programming optimisation was able to evolve energy functions significantly decreasing this gap. Considering the fact that this is only initial work in which we used a basic set of knowledge-based potentials, the results are encouraging.

One of the two biggest difficulties in our research was deciding how to build the ranking of decoys in a way that would lead to learning of the energy function. The second difficulty was a design of the fitness function that would result in an easy to search fitness landscape. These are discussed next.

4.1 Decoy ranking

Since in the structural optimisation process it is important to be able to measure the energy difference even for small structural changes, we decided to build the ranking with a picometer RMSD precision. Despite the high precision, we were not able to avoid ties in the ranking. Our initial permutational approach, in which the tie was decided by the original I-TASSER energy has been shown less efficient in terms of the fitness distribution than averaging the

ranks. It is not really surprising, as the I-TASSER energy itself was not highly correlated with RMSD.

The use of the ranking has been proven to be a good method to avoid the constraints of the direct comparison. This linear normalisation frees the evolution process from the need to reflect the scale of similarity or the direct differences between pairs of decoys.

In experiments with reduced sets of decoys we have shown that a wise selection of the sample representing the whole set improved the learning. But evolved functions were not proven useful when applied to the set of all decoys. Another way of pre-selecting the decoys, that would not depend on the original I-TASSER energy but rather purely on the decoys similarity might give better results. The similarity itself might be also measured differently. The RMSD as a non-weighted average of all C_α–C_α distances is sensitive to local errors and might return high values of distance even if global topology is correct.

The overall conclusion is that the influence of the parametrisation of the evolutionary process on the final result was not as important as the choice of the method to build the ranking. This leads to further work on how the decoys are ranked by the evaluation operator. Introduction of equal rank bins based on the distribution of the evolved energy values should possibly make the rankings comparison more accurate.

4.2 Fitness function

For all fitness functions used in our experiments the average fitness saturated around the maximum after initial 50–200 generations. Although experiments with increased number of generations have shown a continuous improvement even after 2000 generations, the scale of this improvement was very small.

There are several factors that may cause this early saturation. The major one is a polynomial bound on the possible values of fitness functions. The maximum distance between two rankings of length n for Levenshtein distance is n (substitution of all characters). For Spearman and Kendall distance it is bounded by $O(n^2)$ being respectively $\frac{1}{2}n^2$ and $\frac{1}{2}n(n-1)$ for the reverse ranking. As a result, many different energy functions have the same value of the fitness function. This is why for the Levenshtein distance all individuals had the fitness very close to minimum and why for the Kendall distance the fitness variety amongst the population was so limited.

The analysis of population diversity for the best configurations has revealed that higher diversity on the fitness and phenotype level leads to a better evolutionary progress. It might be then useful to design a mechanism similar to the fitness sharing to promote the population diversity on both these levels.

Another important factor is the use of non-weighted average to calculate the total fitness. A very low fitness value for a single protein also significantly lowers the total fitness. To overcome this we may exclude a k outliers from the total score. It might also be a good idea to use more complex averaging scheme, for example weighted by the native structures similarity, so that more

frequent but similar structures in the training set will have lower impact on the total fitness.

5 Conclusions and future work

In this paper we have proposed the use of genetic programming to evolve novel forms of energy function for protein structure prediction. We have shown that this problem is very challenging, mainly due to the need of complex mapping between a GP tree and the total fitness, large amount of data to process and the requirement to generalise over different proteins, and that evolving a high quality energy functions is not an easy task. We have demonstrated a GP design generating functions that outperform the optimised weighted sum of terms used in previous works. We believe that this new GP-based approach might lead to significant improvements in the quality of protein structure prediction.

However, there is still plenty of scope for improvement and several questions remain unanswered. Firstly, the definition of the fitness function needs improvement to better handle the problem of equal ranks and to relax the polynomial bound on the measure of distance between rankings. Moreover, additional objectives could be added to the fitness function to evolve energy functions that are compact and easy to compute.

Secondly, it is not known how the use of protein structure similarity measures other than RMSD will influence the landscape. It is reasonable to expect that more reliable measure of structural comparison might make it smoother, by eliminating some noise. The similarity consensus [3] that is derived from a set of different similarity measures and combines strengths of the individual methods might be a much more robust alternative.

We may also extend in future work the currently limited set of energy terms with data from protein feature predictors such as distance maps, contact order, contact restraints or solvent accessibility [2][40][41]. Especially the last one might be meaningful, since the hydrophobic effect is considered to be one of the main forces in protein folding.

Finally, it is not yet understood how general the evolved functions are or whether the use of different decoys (either generated by different prediction methods or for different set of proteins) will increase their ability to select near-native structures. We hope to address some of these issues in the future work and to see other researchers contributing to this important area of research.

Acknowledgements 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 and by the UK Engineering and Physical Sciences Research Council under grant GR/T07534/01.

References

1. Anfinsen, C.: Principles that Govern the Folding of Protein Chains. Science **181**(4096), 223–30 (1973). DOI 10.1126/science.181.4096.223
2. Bacardit, J., Stout, M., Krasnogor, N., Hirst, J., Blazewicz, J.: 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 (2006). DOI 10.1145/1143997.1144041
3. Barthel, D., Hirst, J.D., Blazewicz, J., Krasnogor, N.: ProCKSI: A Decision Support System for Protein (Structure) Comparison, Knowledge, Similarity and Information. BMC Bioinformatic **8**(1), 416 (2007). DOI 10.1186/1471-2105-8-416
4. Battey, J.N.D., Kopp, J., Bordoli, L., Read, R.J., Clarke, N.D., Schwede, T.: Automated server predictions in CASP7. Proteins: Structure, Function, and Bioinformatics **69**(S8), 68–82 (2007). DOI 10.1002/prot.21761
5. Berman, H.M.: The Protein Data Bank: a historical perspective. Acta Crystallographica Section A **64**(1), 88–95 (2008). DOI 10.1107/S0108767307035623
6. Bourne, P.E.: Structural Bioinformatics, chap. CASP and CAFASP experiments and their findings, pp. 499–505. Wiley-Liss (2003). DOI 10.1002/0471721204.ch24
7. Burke, E., Gustafson, S., Kendall, G.: Diversity in genetic programming: an analysis of measures and correlation with fitness. Evolutionary Computation, IEEE Transactions on **8**(1), 47–62 (2004). DOI 10.1109/TEVC.2003.819263
8. Burke, E., Gustafson, S., Kendall, G., Krasnogor, N.: Advanced Population Diversity Measures in Genetic Programming. In: H.G.B.J.L.F.V.H.P.S. J.J. Merelo Guervós P. Adamidis (ed.) 7th International Conference Parallel Problem Solving from Nature, *Springer Lecture Notes in Computer Science*, vol. 2439, pp. 341–350. PPSN, Springer Berlin / Heidelberg, Granada, Spain (2002). DOI 10.1007/3-540-45712-7_33
9. Chen, H., Zhou, H.X.: Prediction of solvent accessibility and sites of deleterious mutations from protein sequence. Nucleic Acids Research **33**(10), 3193–3199 (2005). DOI 10.1093/nar/gki633
10. Chivian, D.: CASP7 server ranking for FM category (GDT MM) (2006). URL `http://robetta.bakerlab.org/CASP7_eval/CASP7.FR_A-NF.Best-GDT_MM.html`
11. Coutsias, E.A., Seok, C., Dill, K.A.: Using quaternions to calculate RMSD. Journal of Computational Chemistry **25**(15), 1849–1857 (2004). DOI 10.1002/jcc.20110
12. Cristobal, S., Zemla, A., Fischer, D., Rychlewski, L., Elofsson, A.: A study of quality measures for protein threading models. BMC Bioinformatics **2**(1), 5 (2001). DOI 10.1186/1471-2105-2-5. URL `http://www.biomedcentral.com/1471-2105/2/5`
13. Cutello, V., Narzisi, G., Nicosia, G.: A multi-objective evolutionary approach to the protein structure prediction problem. Journal of The Royal Society Interface **3**(6), 139–151 (2006). DOI 10.1098/rsif.2005.0083. Applies MOO for CHARMM27 energy (computed with TINKER).
14. Das, R., Qian, B., Raman, S., Vernon, R., Thompson, J., Bradley, P., Khare, S., Tyka, M.D., Bhat, D., Chivian, D., Kim, D.E., Sheffler, W.H., Malmström, L., Wollacott, A.M., Wang, C., Andre, I., Baker, D.: Structure prediction for CASP7 targets using extensive all-atom refinement with Rosetta@home. Proteins: Structure, Function, and Bioinformatics **69**(S8), 118–128 (2007). DOI 10.1002/prot.21636
15. Day, R.O., Lamont, G.B., Pachter, R.: Protein Structure Prediction by Applying an Evolutionary Algorithm. In: Proceedings of the 17th International Symposium on Parallel and Distributed Processing, p. 155.1. IEEE Computer Society (2003). DOI 10.1109/IPDPS.2003.1213291
16. Dill, K.A.: Dominant forces in protein folding. Biochemistry **29**(31), 7133–7155 (1990). DOI 10.1021/bi00483a001
17. Djurdjevic, D.P., Biggs, M.J.: Ab initio protein fold prediction using evolutionary algorithms: Influence of design and control parameters on performance. Journal of Computational Chemistry **27**(11), 1177–1195 (2006). DOI 10.1002/jcc.20440
18. Dwork, C., Kumar, R., Naor, M., Sivakumar, D.: Rank aggregation methods for the Web. In: Proceedings of the 10th international conference on World Wide Web, pp. 613–622. ACM, Hong Kong (2001). DOI 10.1145/371920.372165

19. Gagné, C., Parizeau, M.: Genericity in Evolutionary Computation Software Tools: Principles and Case-study. International Journal on Artificial Intelligence Tools **15**(2), 173–194 (2006). DOI 10.1142/S021821300600262X
20. Goldberg, D.E., Deb, K.: A Comparative Analysis of Selection Schemes Used in Genetic Algorithms. In: G.J.E. Rawlins (ed.) Foundations of Genetic Algorithms, pp. 69–93. Morgan Kaufmann (1990)
21. Jones, E., Oliphant, T., Peterson, P., et al.: SciPy: Open source scientific tools for Python (2001–). URL `http://www.scipy.org/`
22. Kabsch, W.: A discussion of the solution for the best rotation to relate two sets of vectors. Acta Crystallographica Section A **34**(5), 827–828 (1978). DOI 10.1107/S0567739478001680
23. Knight, W.R.: A Computer Method for Calculating Kendall's Tau with Ungrouped Data. Journal of the American Statistical Association **61**(314), 436–439 (1966)
24. Kolinski, A.: Protein modeling and structure prediction with a reduced representation. Acta Biochimica Polonica **51**(2), 349–371 (2004). URL `http://www.actabp.pl/html/2_2004/349.html`
25. Kolinski, A., Skolnick, J.: Assembly of protein structure from sparse experimental data: An efficient Monte Carlo model. Proteins: Structure, Function, and Genetics **32**(4), 475–494 (1998). DOI 10.1002/(SICI)1097-0134(19980901)32:4⟨475::AID-PROT6⟩3.0.CO;2-F
26. Koza, J.R.: Genetic programming: on the programming of computers by means of natural selection and genetics. MIT Press (1992)
27. Koza, J.R.: Scalable learning in genetic programming using automatic function definition. In: K.E.J. Kinnear (ed.) Advances in Genetic Programming, chap. 5, pp. 99–117. MIT Press (1994)
28. Krasnogor, N., Blackburnem, B., Hirst, J., Burke, E.: Multimeme Algorithms for Protein Structure Prediction. In: J.J. Merelo, P. Adamidis, H.G. Beyer (eds.) Parallel Problem Solving from Nature - PPSN VII, *Springer Lecture Notes in Computer Science*, vol. 2439, pp. 769–778. Springer (2002). DOI 10.1007/3-540-45712-7_74
29. Krasnogor, N., Hart, W., Smith, J., Pelta, D.: Protein Structure Prediction With Evolutionary Algorithms. In: Banzhaf, Daida, Eiben, Garzon, Honovar, Jakiela, Smith (eds.) International Genetic and Evolutionary Computation Conference (GECCO99), pp. 1569–1601. Morgan Kaufmann (1999)
30. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Dokl. **10**(8), 707–710 (1966)
31. Liwo, A., Oldziej, S., Czaplewski, C., Kozlowska, U., Scheraga, H.: 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 **108**(27), 9421–9438 (2004). DOI 10.1021/jp030844f
32. Luke, S., Panait, L.: A survey and comparison of tree generation algorithms. In: L. Spector, E.D. Goodman, A. Wu, W.B. Langdon, H.M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M.H. Garzon, E. Burke (eds.) Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001), pp. 81–88. Morgan Kaufman, San Francisco, California, USA (2001). URL `http://en.scientificcommons.org/453130`
33. MacKerell, J.A.: Empirical force fields for biological macromolecules: Overview and issues. Journal of Computational Chemistry **25**(13), 1584–1604 (2004). DOI 10.1002/jcc.20082
34. McKinnon, K.I.M.: Convergence of the Nelder-Mead simplex method to a nonstationary point. SIAM Journal on Optimization **9**, 148–158 (1999)
35. Nelder, J., Mead, R.: A simplex method for function minimization. The Computer Journal **7**, 308–313 (1964)
36. Pande, V.S., Baker, I., Chapman, J., Elmer, S.P., Khaliq, S., Larson, S.M., Rhee, Y.M., Shirts, M.R., Snow, C.D., Sorin, E.J., Zagrovic, B.: Atomistic protein folding simulations on the submillisecond time scale using worldwide distributed computing. Biopolymers **68**(1), 91–109 (2003). DOI 10.1002/bip.10219
37. Rohl, C.A., Strauss, C.E.M., Misura, K.M.S., Baker, D.: Protein Structure Prediction Using Rosetta. In: L. Brand, M.L. Johnson (eds.) Numerical Computer Methods, Part D, *Methods in Enzymology*, vol. Volume 383, pp. 66–93. Academic Press (2004). DOI 10.1016/S0076-6879(04)83004-0

38. Santana, R., Larranaga, P., Lozano, J.: Protein Folding in Simplified Models With Estimation of Distribution Algorithms. Evolutionary Computation, IEEE Transactions on **12**(4), 418–438 (2008). DOI 10.1109/TEVC.2007.906095
39. Simons, K.T., Ruczinski, I., Kooperberg, C., Fox, B.A., Bystroff, C., Baker, D.: Improved recognition of native-like protein structures using a combination of sequence-dependent and sequence-independent features of proteins. Proteins: Structure, Function, and Genetics **34**(1), 82–95 (1999). DOI 10.1002/(SICI)1097-0134(19990101)34:1⟨82::AID-PROT7⟩3.0.CO;2-A
40. Stout, M., Bacardit, J., Hirst, J., Smith, R., Krasnogor, N.: Prediction of topological contacts in proteins using learning classifier systems. Soft Computing - A Fusion of Foundations, Methodologies and Applications **13**(3), 245–258 (2009). DOI 10.1007/s00500-008-0318-8
41. Stout, M., Bacardit, J., Hirst, J.D., Krasnogor, N.: Prediction of recursive convex hull class assignments for protein residues. Bioinformatics **24**(7), 916–923 (2008). DOI 10.1093/bioinformatics/btn050
42. Syswerda, G.: A Study of Reproduction in Generational and Steady State Genetic Algorithms. In: G.J.E. Rawlins (ed.) Foundations of Genetic Algorithms, pp. 94–101. Morgan Kaufmann (1990)
43. Unger, R.: Applications of Evolutionary Computation in Chemistry, *Structure & Bonding*, vol. 110, chap. The Genetic Algorithm Approach to Protein Structure Prediction, pp. 2697–2699. Springer (2004). DOI 10.1007/b13936
44. Wallin, S., Farwer, J., Bastolla, U.: Testing similarity measures with continuous and discrete protein models. Proteins: Structure, Function, and Genetics **50**(1), 144–157 (2003). DOI 10.1002/prot.10271
45. Wheelan, S.J., Marchler-Bauer, A., Bryant, S.H.: Domain size distributions can predict domain boundaries. Bioinformatics **16**(7), 613–618 (2000). DOI 10.1093/bioinformatics/16.7.613
46. Wu, S., Skolnick, J., Zhang, Y.: Ab initio modeling of small proteins by iterative TASSER simulations. BMC Biol **5**(1), 17 (2007). DOI 10.1186/1741-7007-5-17
47. Zemla, A.: LGA: a method for finding 3D similarities in protein structures. Nucl. Acids Res. **31**(13), 3370–3374 (2003). DOI 10.1093/nar/gkg571
48. Zhang, Y.: CASP7 server ranking for FM category (TM-Score) (2006). URL `http://zhang.bioinformatics.ku.edu/casp7/24.html`
49. Zhang, Y., Hubner, I.A., Arakaki, A.K., Shakhnovich, E., Skolnick, J.: On the origin and highly likely completeness of single-domain protein structures. PNAS **103**(8), 2605–2610 (2006). DOI 10.1073/pnas.0509379103
50. Zhang, Y., Kihara, D., Skolnick, J.: Local energy landscape flattening: Parallel hyperbolic Monte Carlo sampling of protein folding. Proteins: Structure, Function, and Genetics **48**(2), 192–201 (2002). DOI 10.1002/prot.10141
51. Zhang, Y., Kolinski, A., Skolnick, J.: TOUCHSTONE II: A New Approach to Ab Initio Protein Structure Prediction. Biophys. J. **85**(2), 1145–1164 (2003). URL `http://www.biophysj.org/cgi/content/full/85/2/1145`
52. Zhang, Y., Skolnick, J.: Tertiary Structure Predictions on a Comprehensive Benchmark of Medium to Large Size Proteins. Biophys. J. **87**(4), 2647–2655 (2004). DOI 10.1529/biophysj.104.045385