Evolutionary Algorithms for the Inverse Protein Folding Problem

33

Sune S. Nielsen, Grégoire Danoy, Wiktor Jurkowski, Roland Krause, Reinhard Schneider, El-Ghazali Talbi, and Pascal Bouvry

Contents

S. S. Nielsen (✉) · G. Danoy · P. Bouvry
Computer Science and Communications (CSC) Research Unit, FSTC, University of Luxembourg, Luxembourg City, Luxembourg
e-mail: sune.nielsen@uni.lu; sune.nielsen.pro@gmail.com; gregoire.danoy@uni.lu; pascal.bouvry@uni.lu

W. Jurkowski
The Genome Analysis Centre (TGAC), Norwich Research Park, Norwich, UK
e-mail: wiktor.jurkowski@tgac.ac.uk

R. Krause · R. Schneider
Luxembourg Centre for Systems Biomedicine (LCSB), University of Luxembourg, Luxembourg City, Luxembourg
e-mail: roland.krause@uni.lu; reinhard.schneider@uni.lu

E.-G. Talbi
Université des sciences et technologies de Lille, INRIA Lille Nord Europe, Villeneuve d'Ascq, France
e-mail: el-ghazali.talbi@inria.fr; el-ghazali.talbi@lifl.fr

R. Martí et al. (eds.), *Handbook of Heuristics*,
https://doi.org/10.1007/978-3-319-07124-4_59

Abstract

Protein structure prediction is an essential step in understanding the molecular mechanisms of living cells with widespread application in biotechnology and health. The inverse folding problem (IFP) of finding sequences that fold into a defined structure is in itself an important research problem at the heart of rational protein design. In this chapter, a multi-objective genetic algorithm (MOGA) using the diversity-as-objective (DAO) variant of multi-objectivization is presented, which optimizes the secondary structure similarity and the sequence diversity at the same time and hence searches deeper in the sequence solution space. To validate the final optimization results, a subset of the best sequences was selected for tertiary structure prediction. Comparing secondary structure annotation and tertiary structure of the predicted model to the original protein structure demonstrates that relying on fast approximation during the optimization process permits to obtain meaningful sequences.

Keywords

Genetic algorithm · Diversity preservation · Inverse folding problem

Introduction

The relation between the amino acid sequence of a protein and its three-dimensional structure is a principal research effort of structural biology. Obtaining the folded structure of a protein allows functional studies in silico and has given rise to the field of protein engineering.

Proteins are responsible for the majority of molecular functions in a cell. A simplified illustration of a real protein is provided in Fig. 1. Understanding protein folding has immense implications from health to biotechnology applications. Protein engineering in general aims at designing molecules with desired properties, and a method that allows to successfully design such molecules would find applications in a number of areas. For example, it could allow to design improved enzymes for biotechnology applications such as wastewater treatment or biomass production [7]

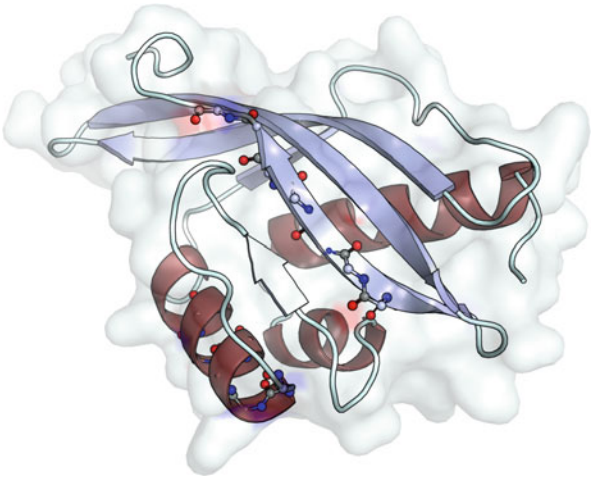

Fig. 1 Protein example *1OH0* with its surface shown semitransparent. *Helix* and *sheet* secondary structure segments are shown in dark red and light blue, respectively. Selected atoms are displayed for further clarification

or new antibodies specific toward already known targets, e.g., a given pathogen like HIV, by binding to its envelope spikes to neutralize the virus [19]. Since the advent of genome sequencing, all protein-coding genes of an organism can be obtained with ease, but structure prediction capabilities were only slightly improved over the last two decades and remain poor. If no homologous structure to a given sequence exists (the ab initio problem), finding the correct structure remains an essentially intractable, which hampers even the comparably easy task of classifying protein sequences into families.

Amino Acids and Protein Structure

A protein sequence is the code that describes the linear combination of any of the 20 common amino acids, also referred to as residues. The amino acid residues are basic organic building blocks consisting mainly of carbon (C), hydrogen (H), oxygen (O), and nitrogen (N) atoms. Common for all amino acids are their *amine* and *carboxylic acid* functional groups which bind through peptide bonds to form the protein backbone of $N - C_\alpha - C$ atoms as shown in Fig. 2. When ordered from left to right, as in the figure, the *amine* group, here represented by its nitrogen (N) atom for simplicity, is situated to the left of the amino acid, respectively, at the beginning of the chain. The side chains, noted as R_i, vary with each of the possible amino acids and can vary both in size and other properties, such as charge, acidity, and hydropathy. A typical protein sequence is 50–300 residues long. Due to the rotational freedom of the atom bonds and the molecular forces acting between the residues, it folds into one canonical three-dimensional structure. These intermolecular forces are the sum of a number of complex interaction forces largely depending on the mentioned properties of the residues, but also on the distance and orientation of interacting atoms and structures. In general the protein

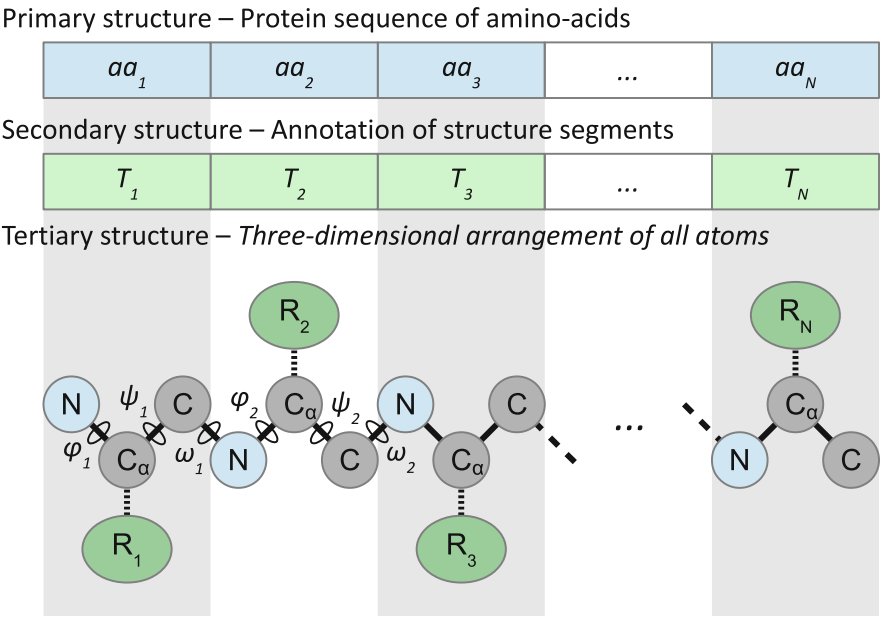

Fig. 2 Three levels of protein structure

structure will try to adapt a lower-energy configuration like a bolder that will roll down a mountain into the valley due to the gravitational force. In the case of proteins, such a more relaxed state corresponds to parts of the protein being either stacked or curled together referred to as *sheets* or *helices* as seen in Fig. 1. The remaining unstructured segments are commonly referred to as *loops* and serve as flexible connections between the other segments. The structure of a protein can be defined in different levels (see Fig. 2). The primary structure is the protein sequence of N amino acids $\{aa_i\}$ where $1 \leq i \leq N$ is the residue position. The secondary structure defines the organization of *helices*, *sheets*, and *loops* of the tertiary structure and can be expressed by a type $\{T_i\} \in \{H, E, L\}$ for each position i in the protein. If, for example, a protein consists of a *helix* and two *sheets*, its secondary structure would look like this: $\{L, L, H, H, H, H, L, E, E, L, E, E, L\}$. The tertiary structure completely describes the arrangement of all atoms of a protein in the three-dimensional space. The ensemble of three-dimensional positions of C_α atoms is commonly referred to as the alpha-trace which provides a rough residue type- and rotation-independent view of the protein configuration. Similar protein sequences generally obtain the same configuration or fold, but sequences not recognizable by similarity can nevertheless fold into 3D structures that are easily brought into congruence. Recommended reading for more in-depth information about proteins and their function in cells is the book by Alberts et al. [2].

Inverted Protein Folding

Conventional protein folding prediction research is concerned with finding or predicting the folded structure of a given amino acid sequence. As the problem is not solved, even to the present day, scientists have early on sought to simplify the task by solving the inverse problem. With the hierarchical definition of Fig. 2 in mind, the inverse folding problem (IFP) can be defined as follows: given a primary structure (protein sequences) and its corresponding tertiary structure, find alternative sequences that will result in the same tertiary structure. The inverted problem is thought of as a simplification because the structure is given, and sequence to structure compatibility becomes the main difficulty. When the structure is unknown (the ab initio case), the number of possible configuration solutions is enormous. A central part of any protein design process is to obtain, or come close to, a target tertiary structure with a certain degree of freedom in the choice of protein sequence. Hence solving the IFP would be a key to successfully engineer proteins. Furthermore, the IFP is of general scientific interest to study the size, shape, and characteristics of the sequence space that matches a given target structure.

Diversity Preservation as a Tool

In this chapter, the fact that matching secondary structures is a necessary, but not a sufficient condition for proteins to have the same tertiary structures, is exploited to reduce the IFP to its simplest formulation: given a protein's secondary structure and its corresponding protein sequence as input, find a set of highly dissimilar protein sequences that could result in the most similar secondary structure. The multi-objective genetic algorithm (MOGA) variant presented here is hence designed for maintaining high diversity, which in turn allows it to explore the decision space of sequences more efficiently and find better solutions than a conventional algorithm. This essentially makes the diversity preservation characteristic central in two aspects: (1) it increases the algorithm's performance in that it continuously pushes the exploration to new areas of the search space while (2) addressing the part of the problem statement of finding a set of protein sequences (i.e., problem solutions) that show large diversity among each other. The latter aspect is thoroughly covered in ▸ Chap. 32, "Diversity and Equity Models", though it should be noted that the representation of solutions and distance among them is different from this work.

An extended validation test is run predicting the final folded structure of as many as 300 generated sequences which are then analyzed in terms of secondary and tertiary structure. This test aims at answering the question of how well the target tertiary structure can be matched solely by taking the secondary structure into account.

The remainder of this chapter is organized as follows: in section "Related Work" the current work is situated in related literature; then a detailed description

of the problem and the biological background is introduced in section "Problem Description" and modeling thereof described in section "Problem Model". In section "Algorithm Design" the methodology achieving adjustable level of diversity in the genetic algorithm is presented. Section "Algorithm Experiments" describes the experiments conducted and the results obtained in terms of algorithm performance, with a validation study of secondary and tertiary structure in sections "Primary and Secondary Structure Validation Results" and "Tertiary Structure Validation Results". Finally the results and perspectives are summarized in section "Conclusion".

Related Work

This section reviews some of the most relevant works related to the two main areas covered in this chapter: protein design and diversity preservation in metaheuristics.

Protein Design

Most applied work of the IFP is concerned with protein design. Since the first design of a peptide by Gutte et al. [14] using secondary structure rules, numerous works have described different approaches to the IFP problem. The earliest reference to the inverted approach is found in an article by Pabo [24] referring to Drexler [11] stating that protein design engineers could in theory choose from a vast subset of possible sequences containing strategically placed groups that would have a predictable fold. Another early attempt at tackling the IFP is done by Ponder and Richards [25] who used a systematic exhaustive approach of enumerating a selected subset of residue positions. Central to the approach is the focus on packing criteria of core residues, taking a latest available side-chain rotamer library into account. Core residues are internal or buried residues not in contact with solvent. They contribute to the general structure of the protein and rather seldom to its primary function. A rotamer library is a library of known side-chain arrangements in 3D for each residue which is important to consider when evaluating the space filling of the core structure.

A few years later, Bowie et al. [5] introduced a 3D to 1D score for each secondary structure type and six environmental classes determined by (1) area buried in the protein structure and (2) fraction of polar side-chain area. By analyzing 16 known structures, the overall relative probability of observing a residue in a defined environment class is computed. From this and the target tertiary structure, a 3D profile can be generated taking the environment at each residue position into account. The 3D to 1D score is calculated by matching a sequence to the 3D profile of a structure. The result is expressed relatively using the Z-score, indicating the number of standard deviations above the mean of other sequences of same length. Using this method they were able to clearly separate homologs (evolutionary-related proteins) in terms of Z-score from a large set of sequences. Kuhlman and Baker [21] used a Monte Carlo approach of residue and rotamer substitution at 11 nonadjacent core positions, evaluating a free energy function. The lowest energy sequence of five

algorithm runs was chosen, and as a final result half of the generated residues were identical to the reference protein, referred to as "wild type."

The first to use a genetic algorithm (GA) was Jones in [17]. To assess 3D-1D compatibility and define an objective function, a set of statistically determined potentials known from fold recognition are used: pairwise potential and solvation potential. To prevent the generation of unlikely sequences, a residue composition term with an arbitrary weight is added corresponding to the target folding class ($\alpha\alpha$, $\alpha\beta$, $\beta\beta$). Jones concluded that there is no way to be sure the resulting sequences have not been overdesigned as the optimal sequence scores significantly better than the reference. He speculates that the energy optimal shape might be very steep and too hard for the real-world protein to fold into. Therefore, the algorithm should possibly be stopped earlier.

Mayo et al. [29] successfully used backbone flexibility in the design process by generating a set of perturbed backbones and applying enumeration of ten varying residue positions applying dead-end elimination to cut the search space. Similarly, Harbury et al. [15] incorporated such backbone freedom in their design approach. Both latter approaches were evaluated by synthesizing the proteins in the lab. Isogai et al. [16] used a recursive approach searching the 3D profile of the target structure keeping two residues fixed and applying a penalty to residues that protrude into the space with a repulsive function. Collisions among side chains were removed manually by replacing residues with smaller ones. The design was successfully synthesized, but the binding site did not stably bind oxygen.

Wernisch et al. [33] sought to combine the latest approaches into an automatic software solution named DESIGNER. The CHARMM package [6] is used for force-field calculation among side chains and backbone taking all hydrogen atoms bonded and nonbonded into account as well as adding van der Waals forces and electrostatic interaction. Both an exact enumeration approach (branch and bound) and a simple heuristic selecting the optimal rotamer for one random position at the time until a local optimum has been reached were tested. Different experiments aimed at analyzing different setting effects on the results were conducted. One test compares the effect of neglecting the reference energy and solvation energy terms, respectively, when redesigning 11 buried positions in the core. The choice of energy terms largely impacts the amount of polar amino acids, and neglecting the solvation term produced better packing with less cavities. Another test aimed at optimizing the protein surface with its larger proportion of polar amino acids. Again 11 positions are variable and varying settings are tested. First backbone and rotamers are kept fixed, and then alternative rotamers were allowed. Wernisch et al. consider that the energy calculations are approximations. Therefore, the software allows for outputting multiple solutions within a user-defined energy window. When packing constraints apply, DESIGNER generated sequences close to the reference.

Voigt et al. [32] combined the field of directed evolution with that of computational design and seek to benefit from both. Directed evolution is concerned with improving specific protein properties or functions mainly by applying a series of mutations to the target as mutagenesis in nature. In their computational method, energy was used to predict structural stability, and residues with low

entropy are detected as more tolerant to mutations. They also argued that coupled residues should be substituted together as several replacements need to take place to demonstrate improvement. High variability was observed on the exposed residues, and in general the variability should guide mutagenesis to allow the generation of a family of divergent sequences with structural integrity intact.

Klepeis et al. [20] presented a two-stage approach where an integer program is first used to generate a list of low-energy sequences which are then evaluated in terms of their fold. Using a force field based on pairwise C_α, distance-dependent interaction potential gives a more relaxed backbone flexibility constraint with less empirically tuned parameters. Validation was done by improving the activity of Compstatin, a 13-residue-long peptide fundamental in inhibiting complement activation. Certain residue positions and types were restricted based on knowledge about the functional nature and with the goal of increasing activity. Experimental results on 14 designed sequences showed significant activity improvements in most cases, one analogue was six to seven times more active than the wild-type underlining. This two-stage approach with small variations is used to design a template for human β-defensiv-2 in [12] and with more advanced second stage in [3,4].

Smadbeck et al. [28] have recently streamlined the two-stage process and present a server implementation with a usage example. The web interface allows for specifying all inputs: template (rigid/flexible), energy function (C_α, centroid, or any), and biological constraints (on charge and content). Stage two workflow consists of two independent fold specificity and approximate binding affinity modules. These include programs such as CYANA, TINKER, and AMBER for the, first, Rosetta (ab initio, dock, and design) and OREO for the latter.

Finally, Mitra et. al [23] used templates of structure families in combination with a force field to guide the search rather than physics-only-based force fields. Due to shortcomings of the latter, evolutionary-based designs have been demonstrated to be more stable. Experiments were conducted with one of the leading protein structure prediction frameworks, I-TASSER[37]. Previous works have shown that I-TASSER is able to distinguish successful designs from unsuccessful ones and is therefore used as validation of the results also in this work.

The research of the last three decades on the IFP problem has produced many methods, but their complexity and exhaustive nature effectively limits the size of the sequence or decision space that can be sampled. In addition, the final output of these methods consists of a single or few sequences close to the input sequence, where a larger and more diverse set of sequences would be desirable for practitioners.

Multimodal Optimization and Niching

In metaheuristics the subject of exploration vs. exploitation characteristics has been thoroughly studied. For population-based optimization algorithms, it is well known that a higher level of population diversity results in more exploration at the expense of exploitation. An elevated population diversity is especially desirable for *multimodal*, *deceptive*, and/or *dynamic* problems. In general, if diversity tends

toward zero, it indicates that the algorithm has converged toward a single solution, which might be an undesired behavior if it occurs too early. A number of works have focused on maintaining and controlling diversity, such as crowding methods by DeJong [8], fitness sharing by Goldberg and Richardson [13], cellular algorithms by Alba and Dorronsoro [1], and diversity-preserving selection strategies based on hamming distance by Shimodaira [27] and on altruism by Laredo et al. [22]. Another approach consists in designing new objectives through multi-objectivization, with which the problem is transformed into a bi- or multi-objective one. Extending problems with an objective designed specifically for diversity preservation has been proposed by Toffolo and Benini [30], by Deb and Saha [9], and most recently by Wessing et al. [34]. In these works, objectives have been designed based on the hamming distance to the closest neighbor, the distance to the nearest better, and number of individuals in the neighborhood.

In this chapter, the diversity-preserving objective is based on the average distance of each individual to all others which directly targets the global diversity measure stated by the problem, contrary to the pairwise local view of existing works. Given the discrete nature, complexity, and multimodality of the problem, an effective diversity limiting mechanism is required. The proposed approach achieves this with the added value of making the population diversity highly variable depending on a single algorithm setting.

Problem Description

The focus in this work is on finding multiple and diversified solutions to the inverse folding problem (IFP).

A simplified model is developed to match solely the reference secondary structure – a requirement for the tertiary structure; see Fig. 3 for a schematic representation. This is motivated by the fact that computing the tertiary structure of a given input sequence is computationally very expensive which would prevent the usage of a metaheuristic on the entire sequence. The found solutions should be a collection of very dissimilar sequences, as well as dissimilar to the input sequence or its homologs, the naturally occurring, evolutionary-related sequences of the input sequence.

A single solution is represented as a sequence $A = \{aa_i\}$ composed of N residue positions, where $1 \leq i \leq N$ and $aa_i \in \{1, 2, \ldots 20\}$ correspond to the set of 20 possible amino acids. As the solution space consists of a total of 20^N different combinations, considering that N is around 50–200 for typical design targets, alternatives to exhaustive exploration are mandatory.

Sequence Identity

Sequence identity is a common measure designed to assess the similarity of proteins occurring in nature in terms of their primary structure only. When computing

sequence identity, gaps are taken into account during the alignment of the sequences to be able to detect evolutionary relations among the compared proteins even if their sequences are of different lengths. In this work, all sequences being compared have the length of the target sequence and are generated by a random process. The chances of the same subsequence to occur in two different sequences with an offset diminish quickly as the subsequence length increases, which justifies ignoring gaps in the model. For the comparison of final results, the generally accepted approach with taking gaps into account is used.

Problem Model

This section presents the corresponding optimization problem. Two objective functions are first defined for integer encoded solutions $A = \{aa_i\}$. The first function estimates the similarity of secondary structure, in which definition and estimation are provided in sections "Secondary Structure Definition" and "Secondary Structure Estimation," respectively. The second function presented in section "Diversity Measure" is designed to address the diversity requirements of the problem and of the algorithm.

Secondary Structure Definition

Secondary structure refers to the annotation of structure segmentation as seen in Figs. 2 and 3. These segments are the result of the protein naturally folding so that different parts of its 3D structure connect through bonds between amino acids on separated residue positions in the sequence. Tertiary structure annotations are

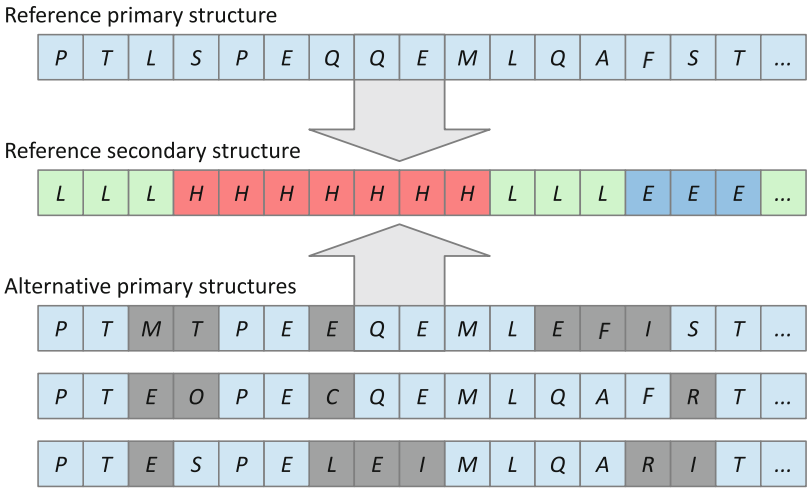

Fig. 3 Primary and secondary structure in the inverted folding problem

done using the "Define Secondary Structure of Proteins" (DSSP) tool [18]. As only the three structure types, *helices (H)*, *sheets (E)*, and *loops (L)*, are considered throughout this work, some simplification is required. In the documentation of DSSP, the following possible annotation types are found:

- G = 3-turn helix (310 helix). Min length three residues.
- H = 4-turn helix (α helix). Min length four residues.
- I = 5-turn helix (π helix). Min length five residues.
- T = hydrogen-bonded turn (three, four, or five turn)
- E = extended strand in parallel and/or antiparallel β-sheet conformation. Min length two residues.
- B = residue in isolated Îš-bridge (single pair β-sheet hydrogen bond formation)
- S = bend (the only non-hydrogen bond-based assignment).
- C = coil (residues which are not in any of the above conformations).

With *helices* characterized by a corkscrew shape, *sheets* as parallel-connected segments, and *loops* as everything else, the above structure types are simplified as follows:

$$G, H, I \Rightarrow H;\ E, B \Rightarrow E;\ T, C, S \Rightarrow L$$

Secondary Structure Estimation

The goal of this objective is to distinguish sequences by assigning better score to sequences that may match the reference secondary structure better. Using the tool PROFphd, updated to ReProf [26], the likely secondary structure type $T_{\text{pred}}(i)$ can be predicted per amino acid aa_i in A with a reliability $R_{\text{pred}}(i) \in \{0 \ldots 9\}$ by means of posterior neural network training. With $T_{\text{ref}}(i)$ the actual type found at position i of the reference secondary structure, the estimated similarity score $F_{\text{sec}}(A)$ is calculated as a sum of reliability weighted (mis)matches:

$$F_{\text{sec}}(A) = \frac{\Sigma_{\max} - \sum_{i=1}^{N} s_i \cdot (C^R_{\text{pred}} + R_{\text{pred}}(i))}{\Sigma_{\max}},\ F_{\text{sec}}(A) \in \{0 \ldots 2\}\ . \tag{1}$$

where

$$s_i = \begin{cases} 1 & \text{if } T_{\text{pred}}(i) = T_{ref}(i) \\ -1 & \text{if } T_{\text{pred}}(i) \neq T_{ref}(i) \end{cases}$$

and

$$\Sigma_{\max} = (C^R_{\text{pred}} + \max R_{\text{pred}}) \cdot N$$

Equation 1 is normalized by the maximum possible sum, $\Sigma_{\max}$, which may occur if all positions are perfectly matched with the highest possible probability.

In this case the score becomes 0 and it can never become negative. C^R_{pred} is a constant which purpose is to increase the contribution to the score of a matching type prediction that has a low reliability R_{pred}. In the current work, it was chosen such that $C^R_{\text{pred}} + \max R_{\text{pred}} = 20$. The reference types $T_{\text{ref}}(i)$ are extracted from the reference structure S_{ref} per residue position i as described in section "Secondary Structure Definition".

Diversity Measure

As a requirement stated in the problem description, the algorithm should not only find a single very good solution, but rather a number of good solutions as different as possible from each other and from the reference sequence. This diversity requirement is closely related to the models described in the ▸ Chap. 32, "Diversity and Equity Models". However, as the problem solutions in this work represent protein sequences, not binary selection of elements, a slightly different approach to the distance measure is taken. An effective and simple measure of distance between two sequences is the Hamming distance, defined as the number of single-point permutations necessary to convert one into the other. Not taking gaps or varying sequence lengths into account, for two sequences $A = \{aa_i\}$ and $A' = \{aa'_i\}$ where $1 \leq i \leq N$, the Hamming distance between them is defined as:

$$d_{\text{Hamm}}(A, A') = \sum_{i=1}^{N} d_i, \; d_i = \begin{cases} 0 & \text{if } aa_i = aa'_i \\ 1 & \text{otherwise} \end{cases} . \tag{2}$$

To obtain a *nonnegative* objective value for minimization, the average Hamming distance to all other $M - 1$ individuals in the current population minus the sequence length N is computed:

$$F_{\text{div}}(A) = N - \frac{1}{M-1} \sum_{i=1}^{M-1} d_{\text{Hamm}}(A, A_i), \; F_{\text{div}}(A) \in \{0 \ldots N\} . \tag{3}$$

This function favors individuals farthest away from the rest of the population. In addition, if a sequence similar to the input sequence exists in the population, the function will have a mutually repulsive effect and penalize it. In summary the function addresses two problem requirements: (1) promoting diversity and (2) promoting sequences which are not equal to the reference sequence.

Algorithm Design

In this chapter the DAO-QC NSGA-II algorithm proposed to tackle the IFP is presented. The modification of the NSGA-II [10], a well-known multi-objective

genetic algorithm, includes the diversity objective (DAO) $F_{\text{div}}(A)$ that enhances the explorative characteristic of the algorithm.

This favorable feature for solving *multimodal* problems is complemented by two modifications of the original algorithm highlighted in Algorithm 1: removal of doubles described in section "Removal of Doubles" and quantile constraint to promote good individuals in section "Quantile Constraint".

Removal of Doubles

In the context of diversity preservation, having two or more identical individuals in the population is undesired. Especially as in [30] when diversity for a sequence A is defined as the minimal distance to any other sequence A', a sequence $A = A'$ must be avoided. With the diversity calculation proposed in section "Diversity Measure", this issue has less impact, but nevertheless doubles are removed in this work. The procedure is executed in line 6 of Algorithm 1 after the application of genetic operations and before non-dominated sorting and crowding-based truncation of the unified population R_t takes place in NSGA-II.

When two identical sequences are detected, one of them is mutated with a probability of $\frac{5}{N}$ to distance the individual with a Hamming distance of 5 on average.

Quantile Constraint

A consequence of the nature of the objectives $F_{\text{sec}}(A)$ and $F_{\text{div}}(A)$ is that the latter is much easier to optimize; hence, the population quickly consists of very diversified individuals with poor fitness according to $F_{\text{sec}}(A)$. To counter this effect, the quantile constraint (QC) is introduced at the end of every generation, in line 9 of Algorithm 1. Given a quantile size C_q, the population P_t at time t is divided according to $F_{\text{sec}}(A)$ into a C_q%-sized partition and a $100 - C_q$%-sized partition. All individuals in the former, less fit, partition are assigned a constraint penalty that prevents the constrained individuals from mating and surviving the next generations. Hence, the population is cleaned from individuals far spread in

Algorithm 1: DAO-QC NSGA-II

1: $Initialize(P_0)$ {randomly generated individuals}
2: $t \leftarrow 0$
3: **while** $t < t_{\max}$ **do**
4: $\quad Q_t \leftarrow makeNewPop(P_t)$ {selection, mutation, recombination}
5: $\quad R_t \leftarrow P_t \cup Q_t$
6: $\quad mutateDoubles(R_t)$ {eliminate doubles by mutation}
7: $\quad F \leftarrow fastNonDominatedSort(R_t)$
8: $\quad P_t \leftarrow truncate(F)$ {based on domination and crowding}
9: $\quad setQuantileConstraint(P_t)$ {to penalize worst quantile}
10: **end while**

the solution space, but with poor $F_{\text{sec}}(A)$ score. The selection pressure can then be selectively adjusted by changing the size of the quantile C_q, which has been tested using $C_q \in \{0\%, 5\%, 10\%, 25\%\}$.

Algorithm Experiments

This section presents and compares the experimental results obtained with the proposed DAO-QC MOGA to a standard generational GA on two protein samples. The experimental setup is first introduced, starting with the two protein samples used and followed by the algorithms' parameters. These initial experiments focus on analyzing the effect of different quantile constraint settings on the proposed algorithms' performance. To this end, the diversity, convergence, and final fitness are compared to a standard generational GA for both of the test samples.

Protein Samples

The two chosen protein samples, namely, *1OAI* and *1URR*, are illustrated in Fig. 4a, b, respectively. *1OAI* is characterized by a length of 59 residues and a secondary structure that consists of 4 helices. *1URR* is 97 residues long, and its secondary structure is composed of 2 helices and 6 beta-sheets.

Experimental Setup

Table 1 summarizes the settings of both the standard generational GA and the proposed DAO-QC MOGA, i.e., the GA extended by multi-objectivization with diversity as objective (DAO) and quantile constraint (QC). Both algorithms use a population of 100 individuals, a binary tournament selection, 1-point crossover

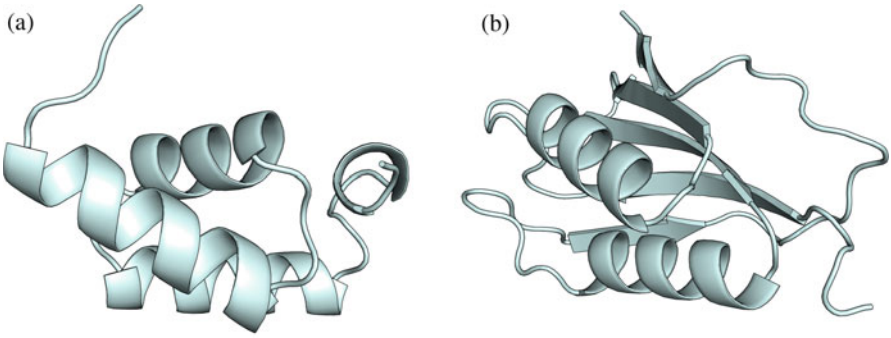

Fig. 4 Three-dimensional structure of the samples. (**a**) *1OAI*. (**b**) *1URR*

Table 1 Algorithm settings

Setting	Value
Population size	100
Algorithm	NSGA-II and std GA
Termination condition	30,000 function evaluations
Selection	Binary tournament (BT)
Crossover operator	1-point, $p_c = 1.0$
Mutation operator	Uniform, $p_m = \frac{1}{N}$
Quantile constraint	$C_q \in \{0\%, 5\%, 10\%, 25\%\}$

Table 2 *1OAI* average fitness cross-comparison

	GA	DAO-QC0	DAO-QC5	DAO-QC10	DAO-QC25
GA	⁄	−0.101∇	−0.0272∇	0.000138 -	0.00896▲
DAO-QC0		⁄	0.0743▲	0.102▲	0.110▲
DAO-QC5			⁄	0.0273▲	0.0361▲
DAO-QC10				⁄	0.00882▲
DAO-QC25					⁄

Table 3 *1OAI* average diversity cross-comparison

	GA	DAO-QC0	DAO-QC5	DAO-QC10	DAO-QC25
GA	⁄	−46.996∇	−45.068∇	−36.946∇	−14.065∇
DAO-QC0		⁄	1.928▲	10.050▲	32.931▲
DAO-QC5			⁄	8.122▲	31.003▲
DAO-QC10				⁄	22.880▲
DAO-QC25					⁄

with probability p_c=1.0, and uniform mutation with probability $p_m = \frac{1}{N}$. The termination condition was set to 30,000 fitness function evaluations, and each experiment was repeated 30 times. Four different values of quantile constraint C_q are considered for DAO-QC NSGA-II: 0%, 5%, 10%, and 25% of the population.

Algorithm Results

In the following the results of the standard GA and the DAO-QC NSGA-II with four different C_q settings are presented and compared in terms of average population fitness, population diversity, and convergence of these indicators based on 30 individual runs.

Tables 2, 3, 4, and 5 show all pairwise comparisons of the algorithm mean value difference. The Wilcoxon test indicator [35] with a 5% significance level provides statistical confidence in comparing the sets with symbols "▲," "∇," and "-" indicating superior, inferior, and no difference. In terms of fitness, the algorithms

Table 4 *1URR* average fitness cross-comparison

	GA	DAO-QC0	DAO-QC5	DAO-QC10	DAO-QC25
GA	／	−0.117∇	−0.026∇	0.0229▲	0.0321▲
DAO-QC0		／	0.0909▲	0.140▲	0.149▲
DAO-QC5			／	0.0489▲	0.058▲
DAO-QC10				／	0.00911▲
DAO-QC25					／

Table 5 *1URR* average diversity cross-comparison

	GA	DAO-QC0	DAO-QC5	DAO-QC10	DAO-QC25
GA	／	−46.611∇	−42.754∇	−25.651∇	−1.389 -
DAO-QC0		／	3.857▲	20.959▲	45.221▲
DAO-QC5			／	17.102▲	41.365▲
DAO-QC10				／	24.262▲
DAO-QC25					／

are ordered in the following way: DAO-QC25 > DAO-QC10 ∼ GA > DAO-QC5 > DAO-QC0 for sample *1OAI* and DAO-QC25 > DAO-QC10 > GA > DAO-QC5 > DAO-QC0 for sample *1URR* with statistical confidence. In terms of diversity, the order becomes DAO-QC0 > DAO-QC5 > DAO-QC10 > DAO-QC25 > GA and DAO-QC0 > DAO-QC5 > DAO-QC10 > DAO-QC25 ∼ GA for samples *1OAI* and *1URR*, respectively. As expected, the higher diversity of the DAO-QC0 approach comes at the expense of a lower average fitness due to the exploration/exploitation trade-off. However, an increase of C_q to 10% or 25% leads to increased exploitation, allowing the DAO-QC NSGA-II algorithm to be constantly ahead of the GA in terms of average fitness until depletion of the evaluation budget as seen in Figs. 5 and 6. Further, the appropriate setting ($C_q = 25\%$ for *1OAI*, $C_q = 10\%$ for *1URR*) allows the DAO-QC NSGA-II to outperform the GA in terms of fitness and diversity *at the same time*. Remaining observations to mention are steeper final fitness slopes for the sample *1URR* with settings $C_q \in \{10\%, 25\%\}$, than the standard GA and specifically for the sample *1OAI*; the diversity is observed to clearly start increasing once the fitness has converged. The steeper final slopes and the increased performance in fitness can be partially explained by the constantly high, and at times increasing, diversity combined with the highly *multimodal* nature of the problem. An elevated diversity clearly increases the chances of the algorithm discovering good new solutions in the rugged fitness landscape of this type of problem.

Table 6 shows the final average fitness and diversity values of all algorithms on both samples with their respective standard deviation. In each column the darker background emphasizes the best result, while the lighter background emphasizes the worst result. With $C_q = 25\%$ the proposed algorithm clearly outperforms the GA with statistical confidence for both samples with average values 0.105 vs. 0.136 and 0.193 vs. 0.242, respectively. From the figure and the table, it is also evident that the

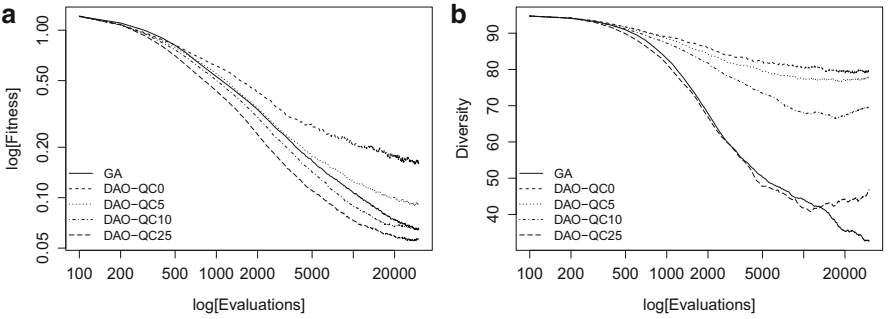

Fig. 5 Convergence of *1OAI*. (**a**) Average fitness convergence. (**b**) Average diversity convergence

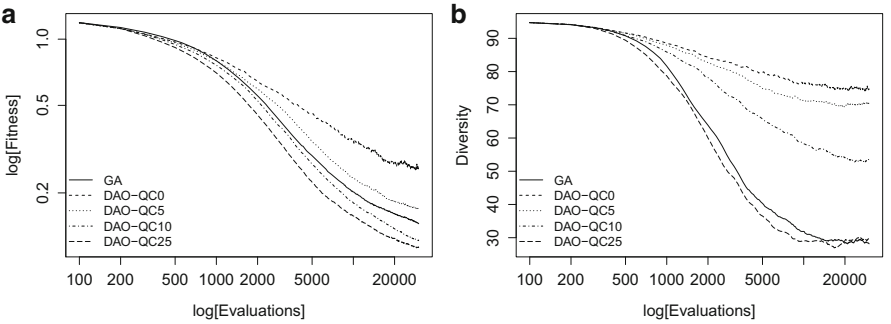

Fig. 6 Convergence of *1URR*. (**a**) Average fitness convergence. (**b**) Average diversity convergence

Table 6 Summary of final values

	1OAI		1URR	
	Average fitness	Average diversity	Average fitness	Average diversity
GA	$0.136_{\pm 1.52e-01}$	$43.578_{\pm 1.32e+01}$	$0.242_{\pm 1.77e-01}$	$35.565_{\pm 1.42e+01}$
DAO-QC0	$0.235_{\pm 1.41e-01}$	$80.992_{\pm 3.23e+00}$	$0.363_{\pm 1.62e-01}$	$77.163_{\pm 4.29e+00}$
DAO-QC5	$0.156_{\pm 1.45e-01}$	$78.598_{\pm 3.69e+00}$	$0.271_{\pm 1.80e-01}$	$72.763_{\pm 5.74e+00}$
DAO-QC10	$0.124_{\pm 1.45e-01}$	$70.564_{\pm 6.56e+00}$	$0.218_{\pm 1.75e-01}$	$59.400_{\pm 1.00e+01}$
DAO-QC25	$0.105_{\pm 1.37e-01}$	$47.484_{\pm 1.11e+01}$	$0.193_{\pm 1.65e-01}$	$34.182_{\pm 1.39e+01}$

value of the quantile or QC setting has a direct impact on the population diversity, providing an effective tool for achieving the level of exploitation vs. exploration preferred.

Structure Validation

In this second experimental step, the protein sequences generated by the best performing algorithm are validated. To this end the I-TASSER [37] prediction tool is used to generate their secondary and tertiary structures that will be compared to the structure of the targeted protein. For each sample, the 5 best generated sequences of the final population in each of the 30 individual runs are selected. This means a total of 300 I-TASSER runs for the 2 protein samples, each run taking around 2 days, which amounts to almost 2 years of CPU time. It is to be noted that the I-TASSER prediction itself is subject to erroneous results; hence, a 100% certainty can never be achieved unless the proteins are synthesized in a wet lab. In the following, the sequences and their I-TASSER predictions are analyzed in terms of primary and secondary structure in section "Primary and Secondary Structure Validation Results" and then tertiary structure in section "Tertiary Structure Validation Results".

Primary and Secondary Structure Validation Results

The goal in this section is to analyze how well the secondary structure of the reference protein is reproduced in the predicted model.

Table 7 shows a summary of the two proteins tested. Clearly, the generated sequences share very little resemblance with the original input sequence seen from a sequence identity of about 20% and 15%, respectively, with a very low deviation. Achieving low sequence identity by itself is not a challenging task unless a good structure match is obtained at the same time. The table shows this as the average percentage, μ, of positions in the secondary annotation of the I-TASSER predicted model that correctly matches those of the input annotation. Average percentage μ and standard deviation of the average percentage σ are given for each of the three structure types *H*, *E*, and *L*. As it can be seen, the helices are correctly predicted on more than 90% of the positions in both proteins. For the slightly bigger *1URR* sample which contrary to *1OAI* contains many extended sheets, the sheet match percentage is lower – slightly below 50%.

Table 7 Summary of secondary structure prediction match

Protein	μ_{Identity}	σ_{Identity}	μ_{Helix}	σ_{Helix}	μ_{Sheet}	σ_{Sheet}	μ_{Loop}	σ_{Loop}
1OAI	20.67	4.100	93.348	6.343	0	0	82.814	7.368
1URR	15.23	3.225	93.787	6.563	42.108	8.898	85.523	8.239

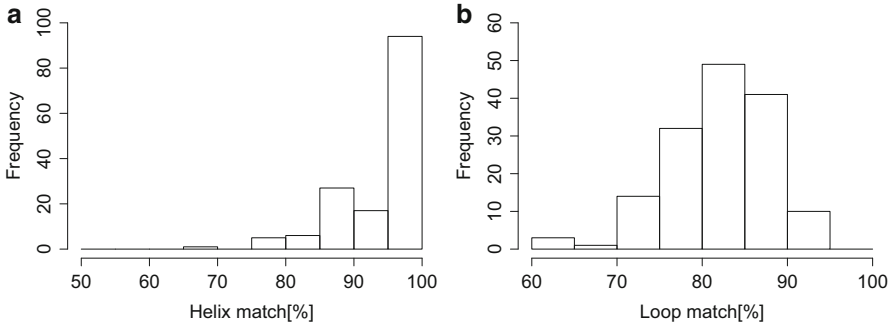

Fig. 7 Match histograms of *1OAI*. (**a**) *Helix* match histogram. (**b**) *Loop* match histogram.

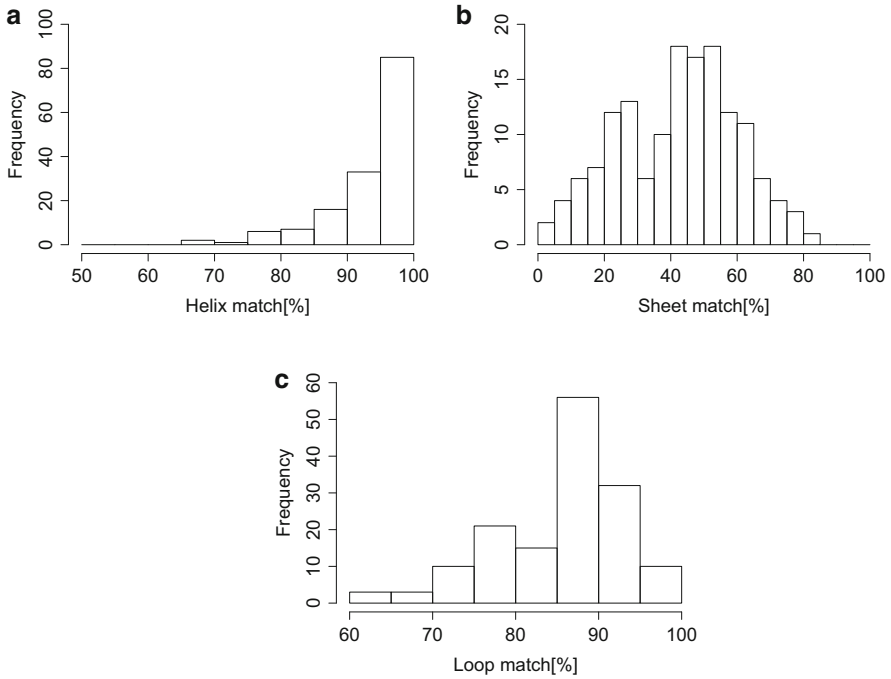

Fig. 8 Match histograms of *1URR*. (**a**) *Helix* match histogram. (**b**) *Sheet* match histogram. (**c**) *Loop* match histogram

Figures 7 and 8 illustrate the same data as histograms. Figures 7a and 8a clearly demonstrate that *helix* structures are very well matched in all 300 structure predictions. Almost all of the tested generated individuals have a match percentage of over 80%, and the majority is above 90% for both samples.

For *loop* segments presented in Figs. 7b and 8c, the majority is still above 80% but with a high spread. The statistics for *sheet* segments show that there is a limit to

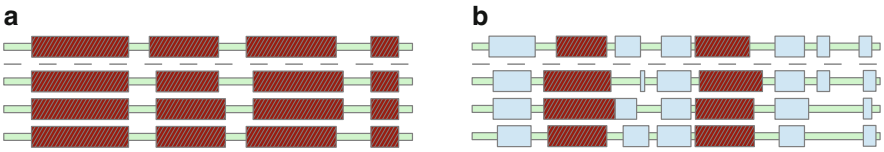

Fig. 9 Secondary structure of reference (on top) compared to three selected generated models. Darker sections are *helices*, lighter are *sheets*, and the rest represents *loop* structure. (**a**) *1OAI*. (**b**) *1URR*

the performance of an approach optimizing only an approximate secondary structure prediction. Considering that the *1URR* sample consists of six *sheet* segments across the whole of its length, then 42% can be considered as a rather good result. The lower success rate of predicting sheets is due to the fact that a sheet can only be observed in the secondary structure if the I-TASSER predicted structure actually did fold close enough to the reference tertiary structure, to allow the extended sheet to form. A *helix* is a much more local structure mostly independent of the global fold, hence easier to achieve in this analysis.

Figure 9 and Table 9 show the alignment of three of the best aligned individually generated sequences. This is to show specific examples of the results which have been averaged in Table 7, and the tendency remains the same: *helices* are very well defined with above 95% positions matched, *loops* slightly less with $\pm$90%, and $\pm$80% for samples *1OAI* and *1URR*, respectively. The *1OAI* sample is clearly an easier target due to its *helix*-only structure compared to the majority of *sheet* structures in the *1URR* sample. The other columns of the table will be discussed in the next section.

Tertiary Structure Validation Results

In the following the tertiary structure of the predicted proteins is validated by three-dimensional comparison.

The TM-Score detailed in [39] is a measure that is used to assess the similarity between two structures, with larger values indicating greater resemblance and 1.0 a maximum value for identical structures. According to Xu and Zhang [36], two proteins can be considered to be in the same fold if comparing them gives a TM-Score above 0.5. Though the average TM-Score is above 0.4 and close to 0.5 for the first sample, this is actually the case for $1{-}in{-}5$ for *1OAI* and $1{-}in{-}15$ for *1URR* as seen in Table 8. The table further shows the number N of predictions that had a TM-Score above 0.2, 0.4, 0.6, 0.7, and 0.8. The general results presented in section "Primary and Secondary Structure Validation Results" are confirmed here, and it is clear that the *sheet* structures of *1URR* are hard to match and that the approach is much more successful in predicting *helix* structures (see Table 9).

Table 8 Summary of tertiary structure prediction match

Protein	$\mu_{TM\text{-Score}}$	$\sigma_{TM\text{-Score}}$	$N_{TM>0.2}$	$N_{TM>0.4}$	$N_{TM>0.5}$	$N_{TM>0.6}$	$N_{TM>0.7}$	$N_{TM>0.8}$
1OAI	0.493	0.135	150	102	51	32	18	4
1URR	0.416	0.061	150	91	10	0	0	0

Table 9 Three selected generated models and their alignment scores with *1OAI* and *1URR* as reference

Nr.	Identity	N< $5A$	$\text{RMSD}_{N<5A}$	RMSD	GDT_{TS}	TM-Score	Helix	Sheet	Loop
1	13.6	58	1.21	1.760	92.797	0.8667	95.12	0	94.44
2	25.4	58	1.35	1.838	88.983	0.8350	95.12	0	88.89
3	18.6	56	1.84	2.722	88.136	0.8015	97.56	0	94.44

Nr.	Identity	N< $5A$	$\text{RMSD}_{N<5A}$	RMSD	GDT_{TS}	TM-Score	Helix	Sheet	Loop
1	19.6	73	2.85	7.484	50.258	0.5374	96	75.68	71.43
2	20.6	67	3.20	4.933	50.773	0.5027	100	81.08	80
3	17.5	74	2.94	9.059	48.711	0.5138	100	72.97	80

The last step in the tertiary validation consists in superposing the fully I-TASSER predicted tertiary structure model of one generated sequence with the target reference. This is illustrated in Figs. 10 and 11 where the first of the three individually generated sequences in Table 9 and Fig. 9 is used.

The models for *1OAI* are all very close to the reference seen from the high *helix* and *loop* match percentage, and in addition the first model for *1OAI* has a very low sequence identity and at the same time very high TM and GDT scores (see Table 9). The first model for *1URR* also has very high *helix* match percentage and good *loop* and *sheet* percentages. However, the TM and GDT scores are less satisfactory. This result is visible in Fig. 11 where the *helices* and *sheets* cannot be fully aligned with the reference and the fact that one *sheet* has been bound to the structure in the wrong location (at the top of the figure rather than at the bottom).

In Table 9 the second column shows sequence identity with gaps, the third shows the length of the longest continuous segment $N < 5A$ that can be fitted below a $5A$ threshold after super-positioning the two structures. The root-mean-square deviation (RMSD) measure is based on the pairwise distance between every residue position in the two tertiary structures, and the fourth column regards only those positions counted in column three, the fifth column regards the total of position. The global distance test (GDT) total score (TS) is a measure indicating the total average of the average percentage of residue positions that can be fitted below each of the thresholds $\{0.5A, 1.0A, 1.5A, \ldots 10.0A\}$. The final four columns are TM-Score and the percentage match of *helix*, *sheet*, and *loop* positions already discussed. Columns three to six were computed with the tertiary structure alignment tool LGA detailed in [38] with default global distance test (GDT) and longest continuous segment (LCS) analysis settings.

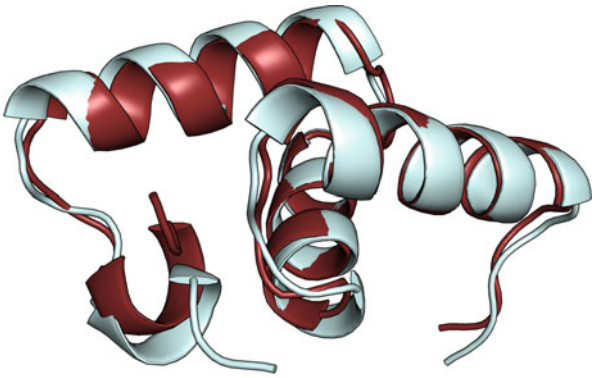

Fig. 10 Super-positioning of a predicted model (dark) with 1OAI reference (light)

Fig. 11 Super-positioning of a predicted model (dark) with 1URR reference (light)

Conclusion

In this chapter an evolutionary-based approach to find a large amount of protein sequences that may result in a given reference secondary and tertiary structure was presented. This problem, referred to as the inverse folding problem (IFP), has received a lot of attention in theoretical chemistry and biophysics over the last 30 years, mostly for its potential application in protein design. It is also of interest to study the extent of the sequence space that may produce similar tertiary structures and how far from the original reference sequence such solutions can be found.

By defining the task as finding highly diverse sequences with most similar secondary structures, an optimization problem was modeled to find many well-scoring sequences in a few hours, which is fast compared to state-of-the-art methods. To achieve high diversity, the requirement has been adapted as an additional objective and extending the problem through *multi-objectivization* to become multi-objective with diversity as objective (DAO). Combining the quantile constraint (QC) with the DAO approach allows to shift focus arbitrarily between diversity and fitness, and final results found significantly better than the standard GA with statistical significance. At the same time, the final diversity remains

significantly higher for all QC-settings except the DAO-QC25 which produces diversity comparable to the standard GA for the *1URR* sample. For the *1OAI* sample with increased QC setting, a clear increase in diversity is observed toward the end of the run, once very good fitness values have been found. In addition to the higher performance on diversity, the algorithm fitness convergence was observed as being generally faster and partially steeper toward the end of runs, than for the standard GA.

For further validation, the five best generated sequences of each independent run of the $DAO-QC25$ algorithm variant were selected systematically and their folded structure predicted by I-TASSER, an established structure prediction software. The 300 predicted tertiary structures were annotated by DSSP for secondary structure analysis of *helix*, *sheet*, and *loop* formations. As could be expected, the method works better for the sample with more defined *helical* secondary structure, and less well in *sheet* and *loop* regions, especially as the latter region is not expressed by the objective function. Indeed *sheet* formations require the tertiary structure to fold properly to be captured in secondary structure. Nevertheless the *1URR* sample *sheet* match percentage is slightly below 50% averaged over all generated predictions. In addition the majority of match percentages are above 80% for *loops* and above 90% for *helices* in both samples. Tertiary structure validation was done by comparing the predicted structures to their respective reference by tertiary structure super-position. For both samples meaningful predictions were generated with a $TM-Score$ above 0.5 observed $1-in-5$ for *1OAI* and $1-in-15$ for *1URR*. These results indicate that this approach is able to generate a massive amount of sequences, with a reasonable amount being likely to actually fold as expected. At the same time, the limits in terms of achieving larger formations of *sheets* are demonstrated.

Future and ongoing works will address the identification of those sequences that actually fold into the reference structure by designing new objectives and constraints and also addressing *loop* and *beta-sheet* regions. Independent of this, sequences found could already be used as starting points for other exact protein design methods and possibly generate successful designs with a very low sequence identity compared to the reference. Additional possible applications could be generating meaningful decoy sets for other studies or finding bridges in sequence space between known proteins of the same structural classes.

Cross-References

- ▸ Diversity and Equity Models
- ▸ Evolutionary Algorithms
- ▸ Genetic Algorithms
- ▸ Multi-objective Optimization

Acknowledgments Work was funded by the National Research Fund of Luxembourg (FNR) as part of the EVOPERF project at the University of Luxembourg with the AFR contract no. 1356145. Experiments were carried out using the HPC facility of the University of Luxembourg [31].

References

1. Alba E, Dorronsoro B (2005) The exploration/exploitation tradeoff in dynamic cellular genetic algorithms. IEEE Trans Evol Comput 9(2):126–142
2. Alberts B, Johnson A, Lewis J, Raff M, Roberts K, Walter P (2002) Molecular biology of the cell. Garland Science, New York
3. Bellows ML, Fung HK, Taylor MS, Floudas CA, Lopez de Victoria A, Morikis D (2010) New compstatin variants through two de novo protein design frameworks. Biophys J 98(10):2337–2346
4. Bellows ML, Taylor MS, Cole PA, Shen L, Siliciano RF, Fung HK, Floudas CA (2010) Discovery of entry inhibitors for HIV-1 via a new de novo protein design framework. Biophys J 99(10):3445–3453
5. Bowie JU, Lüthy R, Eisenberg D (1991) A method to identify protein sequences that fold into a known three-dimensional structure. Science (New York, N.Y.) 253(5016):164–170
6. Brooks BR, Bruccoleri RE, Olafson BD, States DJ, Swaminathan S, Karplus M (1983) Charmm – a program for macromolecular energy, minimization, and dynamics calculations. J Comput Chem 4(2):187–217
7. Chen W, Brühlmann F, Richins RD, Mulchandani A (1999) Engineering of improved microbes and enzymes for bioremediation. Curr Opin Biotechnol 10(2):137–141
8. De Jong AK (1975) Analysis of the behavior of a class of genetic adaptive systems. PhD thesis, University of Michigan, Ann Arbor. Dissertation Abstracts International 36(10):5140B, University Microfilms Number 76–9381
9. Deb K, Saha A (2010) Finding multiple solutions for multimodal optimization problems using a multi-objective evolutionary approach. In: Proceedings of the 12th annual conference on genetic and evolutionary computation. ACM, pp 447–454
10. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197
11. Drexler KE (1981) Molecular engineering: an approach to the development of general capabilities for molecular manipulation. Proc Natl Acad Sci 78(9):5275–5278
12. Fung HK, Floudas CA, Taylor MS, Zhang L, Morikis D (2008) Toward full-sequence de novo protein design with flexible templates for human beta-defensin-2. Biophys J 94(2):584–599
13. Goldberg DE, Richardson J (1987) Genetic algorithms with sharing for multimodal function optimization. In: Grefenstette JJ (ed) Genetic algorithms and their applications: proceedings of the second international conference on genetic algorithms. Lawrence Erlbaum, Hillsdale, pp 41–49
14. Gutte B, Däumigen M, Wittschieber E (1979) Design, synthesis and characterisation of a 34-residue polypeptide that interacts with nucleic acids. Nature 281(5733):650–655
15. Harbury PB, Plecs JJ, Tidor B, Alber T, Kim PS (1998) High-resolution protein design with backbone freedom. Science 282(5393):1462–1467
16. Isogai Y, Ota M, Fujisawa T, Izuno H, Mukai M, Nakamura H, Iizuka T, Nishikawa K (1999) Design and synthesis of a globin fold. Biochemistry 38(23):7431–7443
17. Jones DT (1994) De novo protein design using pairwise potentials and a genetic algorithm. Protein Sci 3:567–574
18. Kabsch W, Sander C (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers 22(12):2577–2637
19. Klein F, Mouquet H, Dosenovic P, Scheid JF, Scharf L, Nussenzweig CM (2013) Antibodies in HIV-1 vaccine development and therapy. Science (New York, N.Y.) 341(6151):1199–204

20. Klepeis JL, Floudas CA, Morikis D, Tsokos CG, Lambris JD (2004) Design of peptide analogues with improved activity using a novel de novo protein design approach. Ind Eng Chem Res 43(14):3817–3826
21. Kuhlman B, Baker D (2000) Native protein sequences are close to optimal for their structures. Proc Natl Acad Sci 97(19):10383–10388
22. Laredo JLJ, Nielsen SS, Danoy G, Bouvry P, Fernandes CM (2014) Cooperative selection: improving tournament selection via altruism. Accepted for publication in EvoCOP14 – 14th European conference on evolutionary computation in combinatorial optimisation
23. Mitra P, Shultis D, Brender JR, Czajka J, Marsh D, Gray F, Cierpicki T, Zhang Y (2013) An evolution-based approach to de novo protein design and case study on mycobacterium tuberculosis. PLoS Comput Biol 9(10):e1003298
24. Pabo C (1983) Molecular technology. Designing proteins and peptides. Nature 301(5897):200
25. Ponder JW, Richards FM (1987) Tertiary templates for proteins: use of packing criteria in the enumeration of allowed sequences for different structural classes. J Mol Biol 193(4):775–791
26. Rost B, Sander C (1994) Combining evolutionary information and neural networks to predict protein secondary structure. Proteins 19(1):55–72
27. Shimodaira H (1997) Dcga: a diversity control oriented genetic algorithm. In: ICTAI, pp 367–374
28. Smadbeck J, Peterson MB, Khoury GA, Taylor MS, Floudas CA (2013) Protein wisdom: a workbench for in silico de novo design of biomolecules. J Vis Exp n77:50476
29. Su A, Mayo SL (1997) Coupling backbone flexibility and amino acid sequence selection in protein design. Protein Sci 6(8):1701–1707
30. Toffolo A, Benini E (2003) Genetic diversity as an objective in multi-objective evolutionary algorithms. Evol Comput 11(2):151–167
31. Varrette S, Bouvry P, Cartiaux H, Georgatos F (2014) Management of an academic HPC cluster: the UL experience. In: Proceedings of the 2014 international conference on high performance computing & simulation (HPCS 2014), Bologna
32. Voigt CA, Mayo SL, Arnold FH, Wang Z-G (2001) Computational method to reduce the search space for directed protein evolution. Proc Natl Acad Sci USA 98(7):3778–3783
33. Wernisch L, Hery S, Wodak S (2000) Automatic protein design with all atom force-fields by exact and heuristic optimization. J Mol Biol 301(3):713–736
34. Wessing S, Preuss M, Rudolph G (2013) Niching by multiobjectivization with neighbor information: trade-offs and benefits. In: 2013 IEEE congress on evolutionary computation (CEC), pp 103–110
35. Wilcoxon F (1945) Individual comparisons by ranking methods. Biom Bull 1(6):80–83
36. Xu J, Zhang Y (2010) How significant is a protein structure similarity with tm-score = 0.5? Bioinformatics 26(7):889–895
37. Yang J, Yan R, Roy A, Xu D, Poisson J, Zhang Y (2015) The i-TASSER suite: protein structure and function prediction. Nat Methods 12(1):7–8
38. Zemla A (2003) LGA: a method for finding 3D similarities in protein structures. Nucleic Acids Res 31(13):3370–3374
39. Zhang Y, Skolnick J (2004) Scoring function for automated assessment of protein structure template quality. Proteins Struct Funct Bioinf 57(4):702–710