Journal of Cell and Molecular Research (2025) 16 (2), 97-112 DOI: 10.22067/jcmr.2024.87614.1092

Research Article

An Evolutionary Algorithm to Predict Super Secondary Structures of Proteins from Secondary Ones, A Case Study: *β*-LACTAMAZE Enzyme

Shima Amirsadri[1], Seyed Morteza Babamir[1*], Shahyar Arab[2]

[1] *Department of Computer, University of Kashan, Kashan, Iran*
[2] *Department of Biology, Tarbiat Modares University, Tehran, Iran*

Received 17 Apr 2024 Accepted 1 Dec 2024

Abstract

The protein's motifs (called super secondary structures) are dense three-dimensional structures of proteins consisting of several secondary structures in a specific geometric arrangement. The prediction of motifs is a matter of concern and has been studied. The previous studies dealt with motif prediction based on the polypeptide chain; however, the prediction of motifs based on the secondary structures leads to more accurate prediction. This study aims to address such a prediction. First, several secondary structures are constructed and then, based on the energy level and using a metaheuristic (evolutionary) algorithm called Imperialist Competitive Algorithm. (ICA) The protein's motifs are predicted. The advantage of our approach over existing approaches is that secondary structural data as input to our algorithm leads to a more accurate prediction that is closer to the real protein third than previous algorithms. We applied our method to predict super secondaries of the enzyme *β−LACTAMASE,* whose specification was obtained from the PDB file in Yasara. This enzyme is produced by bacteria and provides multi-resistance to antibiotics *β−LACTAMA*. Then we evaluated our prediction using Root-Mean-Square Deviation (RMSD). It shows the average distance between the two proteins structurally having the same alignment. Having determined the structural alignment of the two proteins, we determined the similarity of their 3D structures using RMSD. If the RMSD between two structures is less than 2, it denotes they are very similar. Accordingly, we used RMSD to show how much similarity exists between the motif obtained by our proposed algorithm for *β−LACTAMASE* and its native structure.

***Keywords*:** Motif prediction, Secondary structure, Imperialist Competitive Algorithm, Root-Mean-Square Deviation

Introduction

To perform their biological function, proteins fold into one or more specific spatial *conformations*. In fact, the alternative structures of a protein are called conformations of a protein, and the change of a conformation of a protein changes its function. Such structural changes may occur in five levels (MacCarthy et al., 2019).

(1) The primary structure. Proteins are linear polymers of amino acids, a class of organic compounds including an *α*-carbon to which an Amino group (NH2-), a carboxyl group (COOH-), a Hydrogen atom (H), and a variable side chain (R) are bound. The side chains have a great variety of chemical structures and properties that are unique for each amino acid. A peptide bond consists of an atom of carbon of a carbonyl group that directly binds to the atom Nitrogen of a secondary amine as follows (Kuhlman and Bradley, 20192):

$$NH_3^+—CHR_1—CO—NH—CHR_2—CO\_\ldots\_NH—CHR_{n-1}—CO—CHR_n—CO$$

In general, there are 20 standard amino acids, which form protein structures. In the case of mixing two or more amino acids, a *peptide bond* is formed; when a

* Corresponding author's e-mail address: *babamir@kashanu.ac.ir*

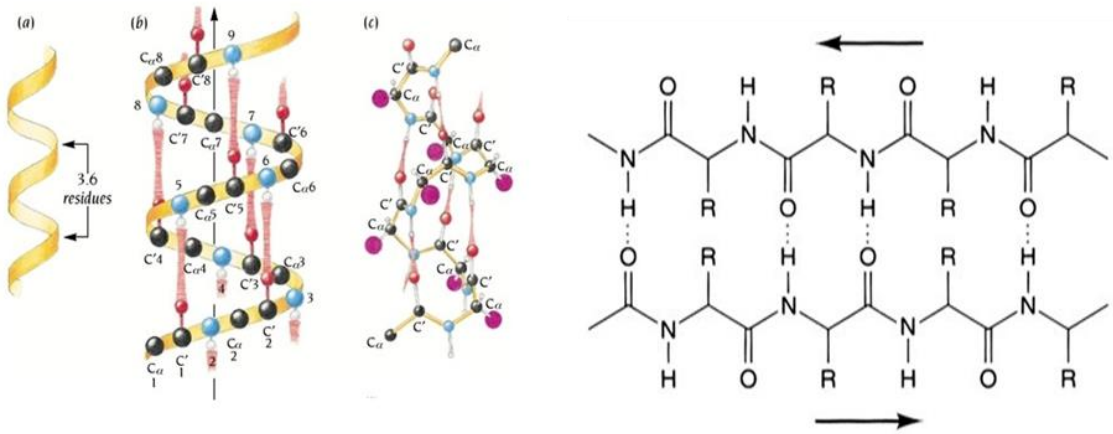
water molecule is eliminated, the remaining compound is called a *residue*.

(2) *The secondary structure*. It is formed when a polypeptide is folded. There are two common secondary structures in proteins, α-helix and β-sheet (Figures 1a and 1b) (Yang et al., 2018). It was unveiled by Pauling; he predicted that the α-helix is more regular and common than the β-sheet. One of the main differences between β-sheet and α-helix is that the amino acids are located far from each other in α-helix, but they are located close together in β-sheet; therefore, β-sheets tend to be tough, and therefore they have little flexibility. They consist of the β-strands connected laterally by at least two or three backbone hydrogen bonds, forming a generally twisted and pleated sheet (Yang et al., 2018). A helix contains some residues, and each residue contains several amino acids.

(3) *Motif*. Motifs are structural components comprising a few α-helices or β-strands that are frequently repeated within structures (Figure 2a) (Yang et al., 2018).

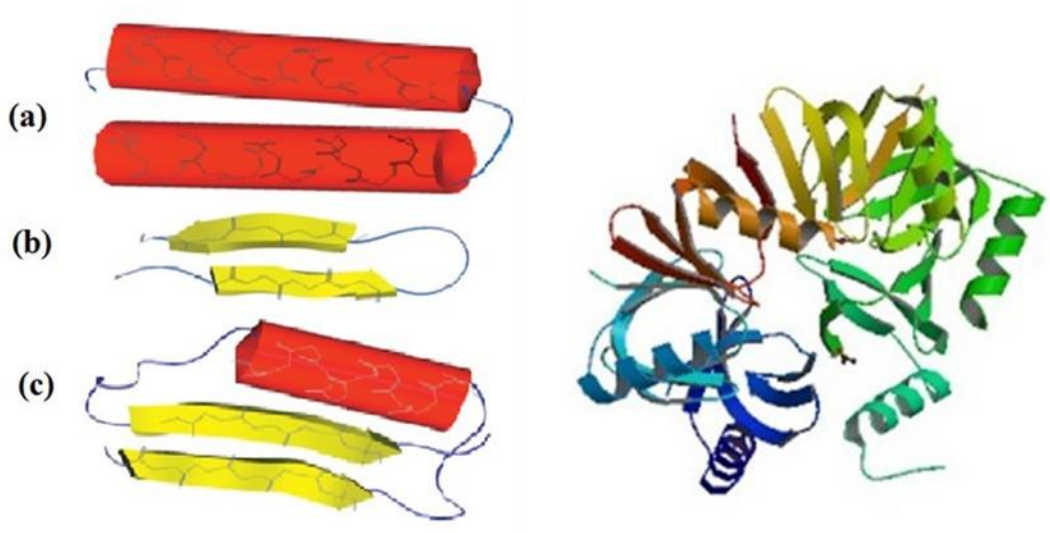
(4) *Tertiary structure*. The tertiary structure of a protein/a sub-unit of a protein is the arrangement of all its atoms in space, without regarding its relationship with neighboring molecules or sub-units (Yang et al., 2018).

(5) *Three–dimensional structure*. This protein structure indicates its function (Figure 2b). Therefore, its recognition of its primary structure provides additional information on the protein function. The detailed geometry of the biochemistry groups playing an essential role in protein function is determined by the correct folding of a protein sequence (Munoz, 2022). To determine the structures, experimental methods such as X-ray crystallography and NMR are time-consuming, costly, and sometimes impossible (Yang et al., 2018; Munoz, 2022; Saudagar and Tripathi, 2023). Therefore, due to the unbalanced increase of protein sequences concerning its known three-dimensional structures, several computational methods have been proposed to predict the structure (Huang et al., 2023).



(a) Pleated $\beta-$ *sheets* with Hydrogen bonds between the protein strands (b) $\alpha-$ *helix* from the main chain by taking all the atoms

Figure 1. β-sheet and α-helix (Yang et al., 2018)



(a) Motif: (a) *α−loop−α*, (b) *β−hairpin*, (c) *β−α−β* (b) The protein's third structure [Yang et al., 2018]

Figure 2. Protein's Motif and Third Structure (Yang et al., 2018)

Protein third structure prediction using amino acid sequences has remained an open problem after more than four decades because the prediction space contains a huge number of sequences. Accordingly, such prediction is considered an NP-hard (non-polynomial hard) problem in computational complexity theory (Guyeux et al., 2014). The computational cost of such a space exponentially grows with increasing protein size. Therefore, a suitable search algorithm can select the optimal structures in a reasonable time. Optimization algorithms are suitable candidates for such problems. To this end, evolutionary algorithms like Particle Swarm Optimization (PSO) (Yu, S., et al., 2022), Ant Colony Optimization (ACO) (Sekhar et al., 2015); Wang et al., 2020), and Bee Colony Optimization (BCO) (BU and Zhu, 2009; Li et al., 2015) have been used. In this paper, we used an evolutionary algorithm called Imperialist Competitive Algorithm (ICA) for the prediction because it has a much better convergence speed in comparison to other evolutionary algorithms (Kaveh and Bakhshpoori, 2019). ICA has been used to predict 3D protein structure in Khaji et al. (2016); in this paper, we use ICA to predict the supersecondary structure of proteins.

The tertiary and third structure protein prediction through its amino-acid sequences is a challenge, and some studies have dealt with it (Gao and Skolnick, 2021). Artificial Neural Network has been used to predict such motifs through amino-acid sequences (Kuhlman and Bradley, 2019). It addresses (1) the residues of the structure of type 2 that a sequence includes and (2) the type of motif that these residues include (e.g., *αα, ββ, αβ,* or *βα*), but doesn't give any information on the coordinates of atoms forming the motif. In contrast, our proposed method uses structural data of the residues that have the structure of type 2 and predicts the coordinates of atoms in the motif. In fact, the spatial structure of atoms is available when we have their coordinates. Protein third structure prediction using Amino-Acid sequences has three steps: prediction of (1) the secondary structure using the sequences, (2) motifs from the secondary structure elements, and (3) the third structure using the motifs. The more accurately each step is performed, the more accurately the third structure will be obtained. In our study, *the second step*, i.e., the prediction of the motif using the secondary structure, is carried out. Our algorithm: (1) receives spatial coordinates of atoms of the helices that are close to each other in their secondary structure, (2) constructs a population of structures by transforming the selected helices, and (3) based on the energy between residues of the transformed helices, produces optimal motif structures. The

produced optimal structures are the nearest ones to the native motif structure. While in a secondary structure, the coordinates of the atoms of two helices are seen as independent of each other; in the population, those of the atoms of each helix are considered regarding the other helix, as well as the energy between them.

The paper is continued as follows: The literature review is conducted in Section Related work and our proposed method is explained in Section Materials and Methods. Section Results considers applying our proposed method to the *β-LACTAMASE* enzyme case study. Section discusses deals with the analysis of the results and threats. Finally, the last section addresses conclusions and the future work.

Related work

In general, related studies in the field of protein structure prediction fall into three categories:

(1) Ab-initio. The ab initio prediction methods create possible structures (conformations) by changing the structural parameters of a protein and then deal with finding the structure with the lowest free energy. This approach is based on the 'thermodynamic hypothesis', which states that the native structure of a protein is the one that has minimum free energy. Ab-initio methods are the most difficult, but the most useful approaches. The proposed approach in the current study is based on the ab initio method because, according to our previous work (Arab et al., 2010), the best protein structure could be predicted. By analyzing *the energy* of structures of a protein, we can choose the best of them, which has the lowest energy. Genetic algorithm (Rashid et al., 2016), Monte Carlo (Rashid et al., 2019), and Molecular Dynamics are the ab initio methods that have been used to predict the protein structure.

(2) Threading. Threading methods compare a target sequence against a library of structural templates, producing a list of scores based on their similarity (Huang et al., 2023). The scores are then ranked, and the folder with the best score is considered.

(3) Comparative modeling. The Threading method is used when the similarity between the structure of a new sequence and those of previously identified sequences is less than 30%. Otherwise, comparative modeling is used. The similarity is measured through the percentage of identical residues at each position. The method exploits the fact that related proteins with similar sequences often have similar structures. For example, two sequences that have just 25% sequence identity usually have the same overall fold (Webb and Sali, 2014).

(4) Developed methods. Machine learning (Enireddy et al., 2022)) and evolutionary algorithms were developed based on the basic methods for protein structure prediction (Varela and Santo, 2022; Bouziane et al., 2015; Rayesha et al., 2023; Rashid et al., 2016; Lin et al., 2018) where in Varela and Santo (2022) a differential evolution algorithm, in Bouziane et al. (2015) an ensemble method, in Rayesha et al. (2023), a neural network, in (Rashid et al. (2016) the genetic algorithm, and in Lin et al. (2016) a simulated annealing based on the Ab-initio basic method were developed for the prediction. However, all of them deal with secondary structure prediction, while the current study presents an evolutionary algorithm for *super-secondary* (motif) structure prediction, which is a type of tertiary structure.

Moreover, to predict the tertiary structure of proteins using evolutionary algorithms, a few methods were proposed (Gao and Scolnick, 2021). In Sekhar et al. (2023) and Yousef et al. (2017), ACO and the Genetic algorithm were used for the prediction, whereas in Yousef et al. (2017), proteins with minimum free energy were focused on. The search algorithms were used in Lin et al. (2014) and Nabil and Sadek (2020), where in Lin et al. (2014) the Tabu-search algorithm and in Nabil and Sadek (2020), the scatter search algorithm based on torsion angles representation were used to predict the tertiary structure of proteins. These algorithms used the primary structure of protein to predict the tertiary structure, while we used the super-secondary structure. In Markuez-Chamorro et al. (2015), a survey of soft computing methods for the prediction of protein tertiary structures was presented.

Considering the related studies mentioned above, our method differs from them. Using a super secondary structure for the prediction of the tertiary structure is a *new method* leading to two improvements: (1) fewer computations and (2) more accurate prediction. The improvements occur because the difference between the structures tertiary and super-secondary of a protein is less than the difference between the structures tertiary and primary of a protein, which has been considered by the related studies.

Protein Data Bank (PDB)

The protein database (http://www.rcsb.org/pdb) contains information about the three-dimensional structure of atoms, and library information, i.e., the first, second, and third structures of proteins. In this database, by searching for a protein code, the protein information file is obtained and stored. Each record of this file contains the coordinates of an atom of a protein, which is indicated by keyword ATOM and includes: (1) serial number, (2) name, (3) the residue amino acid, (4) the residue number, (5) chain index, (6) coordinates X, Y, Z, (7) occupancy, and (8) temperature factor of atom. The following record describes the atom Nitrogen of Amino-Acid Glycine, where: (1) the residue number is 188, (2) the residue code is A, and (3) the spatial coordinates are $X = 29.353$, $Y = 66.696$, and $Z = 17.508$. We will use just this structural information to predict motifs.

$ATOM - 1281 - N - GLY - 188A - 29.353 - 66.969 - 17.508 - 1.00 - 28.84$

Materials and Methods

Our proposed method is based on an evolutionary optimization algorithm, called Imperialist Competitive Algorithm (ICA), which Subsection ICA briefly explains it, and next subsections in this section deal with our ICA-based method for motif prediction.

ICA

ICA, as a powerful optimization algorithm, is inspired by human social evolution. Like other evolutionary algorithms, ICA: (1) starts with a random initial population of countries (solutions), (2) considers some of the countries as imperialists and the rest as colonies, and (3) distributes colonies among imperialists based on their power (Kaveh and Bakhshpoori, 2019).

An imperialist will try to *assimilate* colonies into itself. The distance between an imperialist and a colony is shown as d. While moving toward its imperialist phase, a colony may reach a position better than its imperialist. In such a case, the position of the imperialist and the colony will be exchanged.

An *empire* consists of an imperialist and its colonies; therefore, the *total power* of an empire is calculated as the sum of the power of its imperialist and a percentage of the average power of its colonies. Each empire that cannot increase its power loses some of its colonies; such colonies are assimilated by other empires. If an imperialist loses the total of its colonies, the imperialist itself is regarded as a colony. Hence, the Empire's survival depends on its ability to assimilate colonies of other empires. The process continues until the most powerful empire(s) remain. In this case, ICA converges. In the next subsection, we explain the steps of our ICA-based method.

Representing population individuals, step 1

The first step is the representation of countries. We consider a likely motif as a country. Initial countries are considered the likely motifs whose first helix is the same and the second helix is located at a different limited distance from the first helix. For example, the distance between each two helices of the $\beta - LACTAMASE$ enzyme is limited to 10 Angstroms.

Generating the initial population of candidate solutions, step 2

(1) Since a country in ICA is specified by its properties and a motif plays a country role, we need properties of the protein we want to predict its super-secondaries. We extract properties of a protein (which were described in Subsection PDB) from http://www.rcsb.org/pdb by entering the protein code in the "Search" ribbon. From the ribbon "Download Files", the PDB file of the protein is saved.

(2) By entering the PDB file in the Yasara software (http://www.yasara.org), the structure of the protein is obtained, and the pairs of helices of this structure that are located close together are selected. Fig. 3 shows the structure of protein *β-LACTAMASE*, which we obtained using Yasara. This enzyme is produced by bacteria and provides multi-resistance to antibiotics *β −LACTAMA*. Ribbons *black, red, and green* show *helices*, *β-sheets*, and *loops* of the enzyme, respectively. Yasara enables us to select the pair of helices that are close to each other and obtain the result sooner. For instance, two selected helices of *β−LACTAMASE* include residue numbers 26-40 and 272-290 (two black ribbons in Fig. 3).

(3) From the helices data, which is available in the protein file, the first and last numbers of residues of selected helices are obtained using Yasara. Then, these two helices' data are separated from other data of the protein and saved in separate files. If there are more than two close helices in a protein, we obtain the first and last numbers of all of them and save their information in separate files. For *β-LACTAMASE*, there are 12 such files (six pairs of helices), for instance. Each pair of helices is regarded as a vector (member) of the initial population. As mentioned in Section Introduction, residues of a helix in a protein contain amino acids, and data of atoms for each of the helices are mentioned in the file of the protein. The data contain the name and number of the atoms, the name and number of amino acid residues of the atoms, and the spatial coordinate (x, y, z) of the atoms, respectively. To generate an initial population (200 members), the spatial coordinate of helices is used. Each member of the population consists of a couple of helices, which are obtained using the transformation of the secondary structure helices that are close to each other. To build the first helix of a candidate super secondary structure:

- the beginning of the first original helix is moved to the origin of coordinates,
- the moved helix is projected onto the plane *XY,*

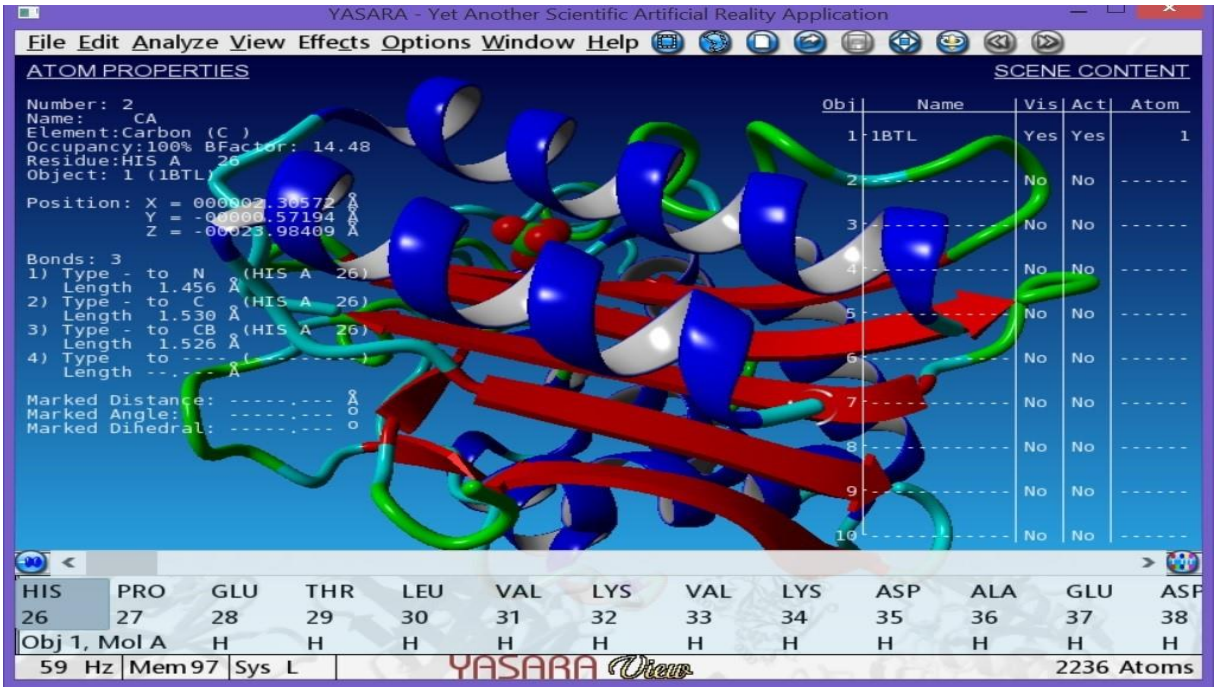


Figure 3. Structures *helices* (black), *β − sheets* (red), *loop* (green) in *β − LACTAMASE*

- The angle between the projected helix and the X axis, called q_1, is determined. Eq. 1 shows the projection of a vector on the X axis, where x denotes the coordinate of x of the vector, $|p|$ denotes the size of the vector, and α does the angle between the projected vector and the X axis.

$$cos(\alpha) = \frac{x}{|p|} \quad where \quad |p| = \sqrt{x^2 + y^2 + z^2} \tag{1}$$

- the projected helix is rotated around axis Z (Eq. 2; Burkowski, 2008) by the angle q_1 to move the helix to plane XY,

$$x' = x \times cos(q_1) - y \times sin(q_1) y' = x \times sin(q_1) + y \times cos(q_1) \tag{2}$$

- the angle between the rotated helix in in-plane XY and the X axis, called q_2, is determined, and
- the moved helix in plane XZ is rotated around axis Y (Eq. 3; Burkowski, 2008) by angle q_2 to move the helix on axis X.

$$z' = z \times cos(q_2) - x \times sin(q_2) x' = z \times sin(q_2) + x \times cos(q_2) \tag{3}$$

To build the second helix of a candidate super secondary structure:

- the beginning of the second helix is moved to the origin of coordinates,
- the moved helix is rotated by a random angle around axis X,
- the rotated helix is rotated by a random angle around axis Y,
- the beginning of the rotated helix is moved to a random point; the point is considered in a sphere whose center is the origin of coordinates and radius is 10 Angstroms. Therefore, the distance between the two obtained helices will be at most 10 Angstroms.

Determining the cost function, step 3

After creating the population in the previous step, imperialists and colonies should be determined. To this end, the cost of each member of the population is calculated using a cost. The less a country has, the more power it has. Therefore, countries with low-cost value are selected as imperialist. We define the cost function based on the member *energy*; according to our previous work (Arab et al., 2010), the lowest energy level lies in the native structure. Eq. 4 shows the cost function, which is one of the most important functions for calculating the energy of a protein. According to our previous work (Arab et al., 2010), it could identify the native structure of protein among other structures with high exactness.

$$E = \sum_{i,j} E(a_i, b_j) \quad where \quad E(a,b) = -k \ln(k(a,b) * S(a,b)) \tag{4}$$

Eq. 4 uses the estimation of the distance between each pair of atoms of the residues to calculate the energy of each member, where $E(a,b)$ denotes the energy between residues a and b and k=0.0019872041 kcal/mol/k is the Boltzmann constant. $k(a,b)$ is a parameter of the knowledge base, which is obtained from the analysis of known structures. $S(a,b)$ is the distance between residues a and b. According to past experiments, value $k(a,b)$ = 7 has led to suitable results in most proteins. Since any member structure has two helices, energy E is computed as the summation of the energy between each residue of the first helix and each residue of the second helix, which is considered as the total energy of the motif (Arab et al, 2010). Noted that energy $E(a_i,b_j)$ is regarded if the distance between C_α (carbon alpha) of a_i and that of b_j is less than or equal to a specific threshold (for instance, 7 Angstroms); in fact, there is no or little energy between two residues if the distance is more than the specified threshold.

Now, we will explain more about energy calculation. For each residue, the mean distance between the C_β (carbon beta) of each member and the atoms linked to the C_β is computed. This mean value is known as the new C_β coordinate. Then, the distance between the C_α of each amino acid in the first helix and that of the second helix is calculated. Afterward, the distance between C_β of the pair of amino acids is calculated as $S(a,b)$ in Eq. 4 if the distance is equal to or less than a threshold level (here, 7 Angstroms). The suitable k value for each pair of amino acids is further extracted from the table, which is associated with the threshold level

(i.e., 7; under normal conditions, in most proteins, the k value is equal to 7 Angstroms). The value of $E(a,b)$ in Eq. 4 is calculated for each pair of residues, and then, according to Eq. 4, the summation of all residue values for each member is considered as the total energy of the member (Arab et al., 2010). This way, the energy level (cost) is calculated for each member of the initial population.

Assimilation policy

Before dealing with the assimilation policy, the motif that is used by the policy is explained. In our proposed method, each motif (population member) is regarded as three successive helix vectors: (1) the first helix vector, (2) the distance vector between the two helices, and (3) the second helix. Upon transferring the first helix to the *X-axis*, the endpoint of the helix is transferred to the origin of the coordinate system, and the beginning of the second helix is held at a distance from the endpoint of the first helix. The distance vector is the one whose beginning point is the origin of the coordinate system, and its endpoint is the first atom of the second helix.

Now the policy is explained. As stated in Subsection ICA, the assimilation policy for an ICA is regarded as the movement of colonies toward their imperialist. In this stage, each imperialist tries to attract more colonies. The more colonies are attracted by an imperialist, the more power it will acquire. However, if some colony has a lower cost (higher power) of its imperialist, the role of the colony and its imperialist is swapped. In our proposed method, we suggest *two phases* to attract colonies by imperialists: 1) the spatial position of the second helix in all colonies is close to that of the second helix of their imperialist, and 2) the length of the distance vector between the two helices in all colonies is close to that of distance vector between the two helices in their imperialist. Through these two phases, a colony approaches its imperialist phase to approaching the colony to its native structure. Consequently, it is made to obtain a structure with lower cost (energy), and the algorithm converges with higher speed and exactness. Now, we explain each of the phases stated above as follows.

- Phase 1. This phase of assimilation is used to close the spatial position of the second helix of each colony to that of the second helix of its imperialist. If the angle between the first and second helices (indicated by H_1 and H_2, respectively) of a colony is α_1 and the angle between those of an imperialist is α_2, $\alpha_3=|\alpha_1|-|\alpha_2|$ denotes the difference between the two angles. This phase is done in five steps:

(1) Transmission of the beginning point of H_1 to the origin of coordinates. To this end (as stated in Sections Materials & Methods and Related work), while creating the initial population, the motifs are created so that H_1 is fixed in all of them. This is why the beginning point of H_1 in all structures (population members) is transferred to the origin of the coordinate system, and then H_1 is transferred to the *X-axis*.

(2) Obtaining the intersection point of H_2 and the X axis. Considering the beginning and end points of H_2, the slope of H_2, called m, is obtained from Eq. 5, where x and y are the coordinates of the endpoint and x_1 and y_1 are those of the beginning point of H_2.

$$y - y_1 = m(x - x_1) \text{ or } m = (y - y_1)/(x - x_1) \quad (5)$$

Now, by determining the amount of m and the coordinates of H_2, i.e., x_1 and y_1, the intersection of H_2 and the X axis is determined. To this end, the value of y in Eq. 5 is regarded as zero. In this way, the value of x at the intersection of H_2 and the X axis is obtained,

(3) Transferring H_1 and H_2 to the symmetry of the intersection point of H_2 and the *X-axis*. This is done because the direction of H_2 would pass the origin of the coordinate system. This makes the origin of both helices the same,

(4) Transmission of H_2 to the XY plane; this is done to complete step 5. To this end, the projection of H_2 on the Y Z plane is obtained. Then, H_2 is rotated around axis X. The angle of rotation is the angle between H_2 and axis Y. This leads H_2 to be moved to plane XY,

(5) Obtaining angles α_1, α_2, and α_3. One of the stages of the assimilation policy is to approach the spatial

location of H_2 in each colony to that of H_2 in its imperialist. To this end, we should obtain α_1, α_2, and α_3. To obtain α_1 and α_2, it is required to put H_1 and H_2 in the same plane; this was performed in step 4. To calculate angles α_1 and α_2, we use the inner product of two vectors (Eq. 6) and then angle α_3 is obtained ($|\alpha_3|=|\alpha_1|-|\alpha_2|$). In Eq. 6, a and b represent two vectors, and α shows the angle between the two vectors.

$$a.b = |a| \times |b| \times cos(\alpha) \text{ or } \alpha = cos^{-1}((a.b)/(|a| \times |b|)) \quad (6)$$

(6) Rotation of H_2 of each colony around the Z-axis by α_3. If $|\alpha_1|<|\alpha_2|$ then H_2 of the colony is rotated by α_3 so that the value of $|\alpha_1|$ increases and is closed to $|\alpha_2|$. If $|\alpha_1|>|\alpha_2|$ then H_2 of the colony is rotated by α_3 so that the value of $|\alpha_1|$ reduces and is closed to $|\alpha_2|$.

- Phase 2. This phase of the assimilation policy is used to close the distance between the two helices of each colony to the distance between the two helices of its imperialist. In the following, the steps of this phase are explained.

(1) Transferring the end of the first helix to the origin of coordinates,

(2) Obtaining the length of the distance vector of every residue of imperialist (indicated by d_imp) and its residues of colonies (indicated by d_col),

(3) Obtaining the length of step (indicated by l) using Eq. 7, which is suggested in our ICA,

$$l = \frac{max(d_{imp}, d_{col})}{2 \times min(d_{imp}, d_{col})} \quad (7)$$

(4) Decrease/increase of the length of the distance vector of the colony as step l (Eq. 8),

$$d'_{col} = \begin{cases} d_{col} + l & \text{if } d_{col} < d_{imp} \\ d_{col} - l & \text{if } d_{col} > d_{imp} \end{cases} \quad (8)$$

(5) Obtaining new coordinates (x',y') of the end point (x,y) of the d_{col} vector in each colony using Eq. 9. In fact, d'_{col} is a modified vector of $d_{col,}$ and α and β denote the angles between the vectors and axes X and Y, respectively.

$$cos(\alpha) = \frac{x}{d} = \frac{x'}{d'} \qquad sin(\beta) = \frac{y}{d} = \frac{y'}{d'} \quad \textit{where } d' = \quad (9)$$

In Eq. 9, d' is the Euclidean distance between two points d_{col} and d'_{col}. The beginning of the second helix of all colonies is moved to d' so that each d_{col} is close to its imperialist.

Imperialist competition

In ICA, each empire failing to increase its power will become a weak empire and lose its competitive capability; thus, it is removed from the competition. Weak empires will lose their colonies, and strong ones will take them. This leads to competition between strong empires to take the colonies of weak imperialists in iterations of the algorithm. Note that colonies of weak empires won't necessarily be possessed by the strongest empire, but stronger empires have a greater possibility for possessing. This algorithm will continue if only one imperialist remains. It is the selected motif protein structure with the lowest energy and highest similarity with its native structure.

Results

To show practically our proposed method, we apply it to predict the super secondary (motif) of enzyme *β-LACTAMASE* using its secondary structure.

Consider Fig. 3 showing two close helices (the black ribbons). As stated in Subsection Assimilation policy, the two selected close helices have residues with 26-40 (15 amino acids) and 272-290 (19 amino acids). According to ICA, first, we should create the initial population whose each member consists of the first and second helices. Six steps were stated in Subsection Assimilation policy to build the first helix, which we now apply.

Construction of the first helix of candidates

Step 1. We should transfer the beginning point of the first helix of *β-LACTAMASE* (Amino-Acids 26-40) to the origin. According to data in the PDB file, the beginning point of this vector is an atom with coordinates (x=2.610, y=1.454, z=10.018). To transfer the first helix vector to the origin, values

2.610, 1.454, and 10.018 are subtracted from the coordinates *x*, *y*, and *z* of the helix atoms, respectively. Therefore, the starting and ending points of coordinates of the new point of the first helix are (x=0, y=0, z=0), and those coordinates of the final atom of the first helix are *CD2* (14.724, 11.341, 11.112).

step 2. The coordinate of this atom is used for transferring the first helix to the *X-axis*. The projection of this vector with coordinates (x=14.724, y=11.341, z=11.112) on the *XY* plane is vector *p* with coordinates (14.724, 11.341, 0).

step 3. Based on the angle value between the projection vector and the *X-axis* (see Eq. 1), the first vector is rotated_around the *Z*-axis to transfer the vector to the *XZ* plane. Considering Eq. 1, $|p|$==18.585329 and =0.65632886, where *x* is coordinate *x* of *p*.

step 4. We rotate the first helix (vector) with coordinates (x=14.724, y=11.341, z=11.112) around the *Z*-axis by α; this is done to transfer the vector to the *XZ* plane. Note that values of *x*, *y*, and *z* are positive, and the rotation is done as $-\alpha$. According to Eq. 2 (rotation around the *Z* axis), the coordinate of the first helix in the *XZ* plane is:

$x' = x \times cos(-\alpha) - y \times sin(-\alpha) = 18.53886729$

$y' = x \times sin(-\alpha) + y \times cos(-\alpha) = 0,$

$z' = z = 11.112.$

step 5. We rotate the vector around the *Y* axis in the *XZ* plane, as the angle between the vector and the *X* axis; this is done to transfer the first helix to the *X* axis. Vector p' shows the transfer of the vector to the *XZ* plane with coordinate (18.585329, 0, 11.112). The angle between vector p' and axis *X* is =0.53886729 (*x* is coordinate *x* of p' and $|p'|$ is calculated similar to $|p|$).

Step 6. According to Eq. 2 (rotation around the *Y* axis):

$z' = z \times cos(\alpha) - x \times sin(\alpha) = 0,$

$x' = z \times sin(\alpha) + x \times cos(\alpha) = 21.653891,$

$y' = y = 0$

Therefore, the coordinates of the final atom of the first helix in axis *X* are (x=21.653891, y=0, z=0).

Construction of the second helix of candidates

At first, the second helix is rotated around the *X* and *Y* axes by two random angles, and then the beginning of the second helix is moved to a random point at no more than 10 Angstroms from the first helix. As described in Fig. 3, the second helix of the two selected close helices in *β-LACTAMASE* has amino-acid residues of 272-290. The rotation steps are:

step 1. Two random angles in the distance of (0,2Π) are selected as α_1=5.40737 and α_2=0.249119,

step 2. The random point *d* (3.95109, 5.59319, 9.84783) is selected for transferring the beginning of the second helix to the random point,

step 3. To rotate the second helix around the *X* and *Y* axes, the beginning point of this helix is transferred to the origin of the coordinate system. According to the data file of this protein, the beginning point of this vector is (10.047, 4.849, 9.671). These values are subtracted from atoms of 272-290, and the coordinates of the final atom of this helix are shown as p''=(- -21.841, -11.839, -18.084).

step 4. Atoms of 272-290 are rotated around the *X* axis as α_1; for example, for atom p'', we have:

$x' = x = -21.841,$

$y' = y \times cos(\alpha_1) - z \times sin(\alpha_1) = -21.4711,$

$z' = y \times sin(\alpha_1) + z \times cos(\alpha_1) = -2.48733$

step 5. Vector p'' is rotated around the *Y* axis as α_2:

$x'' = x' \times cos(\alpha_2) + z' \times sin(\alpha_2) = -2.78,$

$y'' = y' = -21.4711,$

$z'' = z' \times cos(\alpha_2) - x' \times sin(\alpha_2) = 2.97436$

step 6. For transferring the beginning point of the second helix to point *d*, the coordinates of point *d* are added to the atoms of the second helix. For example, the new coordinate of *p"* is (x=1.17109, y=-15.87791, z=12.82219) after transferring p'' to the new point.

Applying cost function

As stated in Subsection Assimilation Policy, imperialists are determined based on their cost, meaning that the population members with low cost are selected as imperialists. The cost is calculated using a cost function (Eq. 5), which is defined based on the energy between the two helices. Eq. 5 uses the estimation of the distance between the pair of atoms of the residues to calculate the energy in each structure. Now, we show the computation of energy between the two selected helices of *β-LACTAMASE* with residue numbers (26-40) and (272-290). Table 1 shows the first residue of the first helix (the left column) and the first residue of the second helix (the right column). Columns 2, 3, and 5 show the atom name, residue name, and residue number. Columns 6-8 show the coordinates *x*, *y*, and *z* of each atom. Since the helices have 15 (26-40) and 19 (272-290) residues, the energy value for 15 × 19=285 pairs of residues should be calculated, and their summation is regarded as the total energy of the population (structure). To calculate the energy between these two helices, the distance between their C_α is calculated. Coordinates of C_α in *HIS* (the residue of the first amino acid of the first helix) and *MET* (the residue of the first amino acid of the second helix) are $C_{\alpha 1}$=(2.907, 1.930, 8.674), $C_{\alpha 2}$=(8.681, 4.371, 9.886). Distance between these atoms is calculated using Euclidean distance (see *d* in Eq. 9), which is *d*(*HIS, MET*)=6.387<7. Since this distance is less than 7 (low distance), there is an exchange of energy between them. Now, to calculate energy between these two helices, the average of atom C_β and its following atoms for each helix should be obtained. In the residues of HIS, following atoms of C_β=(2.125, 3.225, 8.471), are:

CG= (2.090, 3.719, 7.037), *ND*1= (3.037, 4.237, 6.235),

*CD*2= (0.934, 3.631, 6.297) *CE*1= (2.496, 4.422, 5.047), *NE*2= (1.234, 4.057, 5.102)

In the MET residue, the following atoms of C_β=(8.224, 4.782, 11.305) are:

CG= (6.852, 4.223, 11.687),

SD= (6.787, 2.416, 11.894),

CE= (7.745, 2.059, 13.332).

To calculate the distance between *HIS* and *MET*, first, the average of the coordinates *x*, *y*, and *z* of C_β and its following atoms in *HIS* and the average for *MET* are calculated. The averages for *HIS* and *MET* are (1.986, 3.881, 6.364) and (7.402, 3.37, 12.054), respectively. Now, Euclidean distance between *HIS* and *MET* is calculated according to *d* in Eq. 9, which is *d*=7.872, i.e., *S*(*HIS, MET*)=7.872 and *k*(*HIS, MET*)=0.124928493938531. The value of the Boltzmann constant is 0.009872041. According to Eq. 4, we have:

$E(HIS,MET) = -kln(k(HIS,MET) * S(HIS,MET)) = E(HIS,MET) = -0.0019872041 * \ln(0.124928493938531 * 7.872) = 0.0000331895$

This way, the energy between HIS and the other 18 residues of the second helix is calculated. Then, the second residue is selected from the first helix, and the energy between the residue and each residue of the second helix is calculated. This is done for other residues of the first helix, and then the summation of these energies is calculated according to Eq. 4 (the calculated value is -1.147). Finally, the structural energy between residues 26-40 and 272-290 is calculated according to Eq. 10 where a_i and b_j denote a residue of the first and a residue of the second helix, respectively.

Eq. 10 shows the summation of energy between all pairs of residues where the first residue belongs to the first helix, and the second one belongs to the second helix. Based on Eq. 10, Table 2 shows values of the cost function value (energy) for the five structures (members) of the initial population.

$$E = = E(HIS,MET) + ... + E(HIS,ASP) + ... + E(HIS,TRP) + E(PRO,MET) + ... + E(PRO,ASP) + ... + E(LEU,TRP) = -1.323 \quad (10)$$

Table 1. Data of residue 26 of the first helix and residue 272 of the second helix

1 N HIS A 26 2.610 1.454 10.018 1.00 14.51	1 N MET A 272 10.047 4.849 9.671 1.00 8.04
2 CA HIS A 26 2.907 1.930 8.674 1.00 14.48	2 CA MET A 272 8.681 4.371 9.886 1.00 9.89
3 C HIS A 26 4.419 2.138 8.562 1.00 13.77	3 C MET A 272 7.714 4.931 8.843 1.00 10.27
4 O HIS A 26 5.021 2.717 9.466 1.00 11.71	4 O MET A 272 6.910 4.171 8.283 1.00 10.93
5 CB HIS A 26 2.125 3.225 8.471 1.00 17.01	5 CB MET A 272 8.224 4.782 11.305 1.00 11.17
6 CG HIS A 26 2.090 3.719 7.037 1.00 21.11	6 CG MET A 272 6.852 4.223 11.687 1.00 13.62
7 ND1 HIS A 26 3.037 4.237 6.235 1.00 22.46	7 SD MET A 272 6.787 2.416 11.894 1.00 19.17
8 CD2 HIS A 26 0.934 3.631 6.297 1.00 24.58	8 CE MET A 272 7.745 2.059 13.332 1.00 18.35
9 CE1 HIS A 26 2.496 4.422 5.047 1.00 26.87	
10 NE2 HIS A 26 1.234 4.057 5.102 1.00 26.43	

Table 2. The value of the cost function of 5 first five structures of the initial population for residues *HIS* and *MET* of *β-LACTAMASE*

Motif#	The value of the cost function after the 1^{st} stage of execution of the proposed algorithm	The value of the cost function after the 90^{th} stage of execution of the proposed algorithm
1	2.6783	0.4235
2	3.561	1.247
3	1.845	-0.039
4	3.721	1.267
5	6.521	1.368

Applying imperialist selection

Until now, we have applied the first three steps of ICA to *β−LACTAMASE*. The fourth step is the selection of imperialists and the allocation of colonies to them. As stated in Subsection Assimilation Policy, at first, 200 motifs were created as the initial population. Then, using the cost function, 30 structures having the minimum cost were selected as imperialists. By increasing or reducing the number of imperialists and determining the speed of convergence, we can adjust the number of imperialists to lead to the optimum solutions. In Section Discussion, we discuss the impact of the initial selection of imperialists, the number of executions of the algorithm, and the algorithm convergence on the optimality of the solutions. After selecting imperialists, the colonies of each imperialist are determined. To this end, by considering the cost of each imperialist, the normalized cost is obtained according to step 4 in Listing 1. The initial number of colonies of each imperialist is determined using $NC_n=round(p_n \times N_{col})$, where NC_n is the initial number of colonies of an empire, N_{col} is the number of colonies of the initial population, p_n is calculated according to step 4 in Listing 1, and function *round*() produces the nearest integer number of a real number.

Based on the value NC_n of each imperialist, an initial population of colonies is randomly allocated to the imperialist. Afterward, the imperialist competition begins and continues until one or a number of the most powerful imperialists remain; this is the convergence condition.

Applying assimilation policy

According to Listing 1, the assimilation policy is step 5, including two phases (see Subsection Assimilation Policy) where phase 1 contains 6 steps and phase 2 does 5 steps. We should apply each of the phases stated in Subsection Assimilation Policy for the two helices of *β- LACTAMASE* for residues 26-40 and 272-290. Since the details of the calculation of applying the assimilation policy are lengthy, we don't show them here.

Discussion

We evaluate the results of our method through two benchmarks, RMSD and convergence. Moreover, we analyze the impact of the revolution operator on the results.

Evaluation through RMSD

RMSD (Root-Mean-Square Deviation). It shows the average distance between atoms (generally main chain atoms) of two proteins structurally having the same alignment (Coutsias and Wester, 2019). After the structural alignment of the two proteins, the similarity of their 3D structures is determined using RMSD. Generally, if the RMSD between two structures is less than 2, it means the two structures are very similar. Thus, we use RMSD to show how much similarity exists between the motif obtained by our proposed algorithm for *β-LACTAMASE* and its native structure. If the difference is less than 2, the algorithm benefits from a high ability and precision for motif prediction. RMSD is calculated using Eq. 11, where v and w are two sets of n points with coordinates x, y, and z.

$$RMSD = \sqrt{\frac{1}{n}\sum_{i=1}^{n}||v_i - w_i||^2} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(v_{ix} - w_{ix})^2 + (v_{iy} - w_{iy})^2 + (v_{iz} - w_{iz})^2}$$

We show RMSD for the motif prediction of the $\beta - LACTAMAS$ enzyme. To this end, we selected 6 and 5 pairs of helices with a secondary structure of the *β-LACTAMASE*. Table 3 shows the RMSD results obtained by applying our ICA-based method for all pairs of helices of the secondary structure of the enzyme is less than 1.

To calculate RMSD, we considered native and predicted structures of the residues 26-40 and 272-290 of the first and second helices of β-LACTAMASE. Then we (1) considered the first helix invariant, (2) obtained the difference of atoms of the second helix in the predicted structure and those of the native structure, and (3) computed RMSD. For example, since the coordinates of atom N are (-14.9969, -1.8485, 4.38443) and (-13.8173, -2.6355, --3.2431) in the native and predicted structures, respectively, we have:

$$RMSD=\frac{1}{159}(|N_{Predicted} - N_{Native}|) + \ldots + (|OXT_{Predicted} - OXT_{Native}|)=\frac{1}{159}(1.8219+...) =0.738571$$

Threats

In this section, we consider two possible threats to our work: (1) the indeterminacy of the proposed algorithm because of being evolutionary and (2) considering just energy for decision on the motif structure.

Evolutionary algorithms, including ICA are non-deterministic, indicating they may produce different results in separate runs the algorithm even if the algorithm parameters don't vary. This is because the initial population is selected randomly. Moreover, while an evolutionary algorithm produces good results for a case study, it may not produce good results for other case studies. A way to resolve these two uncertainties is the consideration of the generalization of the algorithm. In future work, we address such considerations (the last section).

Motif parameter decision. In our proposed algorithm, we considered just the parameter energy for the decision on the motif structure. If we consider other chemical and structural parameters, the accuracy and the algorithm speed could be improved. Parameters like chemical shifts (CSs), which are derived from nuclear magnetic resonance (NMR) spectroscopy. Using NMR for the improvement of the prediction accuracy of super secondary structures is explicitly stated in MacCarthy et al. (2019). In addition to *α -helix*, using information of other secondary structures like topological and biological features can help to more

the accuracy of the method. We have previous experiments in using such features for the prediction of essential proteins (Elahi and Babamir, 2018).

Conclusions and future work

The approach proposed in this paper dealt with the motif prediction (super secondary structure of the protein) using the information of two helices with the *secondary structure*. We used the α-helix structure of the secondary structure for the prediction because it is more regular and common than the β-sheet structure of the secondary structure. Having selected the two closest helices in a protein, we created a population of pair helices through the rotation of the helices around the coordinate axes and the projection of one of the helices on the coordinate axes. Using an evolutionary algorithm called ICA, we tried to create the next generations of the initial population by evolving them. The population evolution of the helices caused the population members to draw nearer to the native structure in each generation, which was shown using the energy parameter. By including the number of members in the initial population, we will always have suitable solutions after generating and evolving the next generations in the algorithm. Through some figures, we showed that the selection of the number of imperialists in the proposed algorithm has a significant impact on the accuracy of the produced solution as well as the speed of the algorithm.

As stated in Section Discussion, the motif prediction may be generalized when an evolutionary algorithm is used. Through the generalization, one can feel sure of the achievement of results for other case studies when one or two case studies have generated good results. Frequent executions of an evolutionary algorithm and testing their results using Statistical tests can show the achievement. This may be considered in future work. As the second future work, we can: (1) select the best motif among the pair of residues that have the lowest energy level; (2) consider this motif as a helix; (3) obtain the best motif among this helix and others, and (4) generalize the algorithm to predict the best third structure that is closer to the native form. In fact, we can obtain the structural information of helices with the second structure using information of the Amino-Acid sequences, and then we can predict the best third structure of a protein (the most similar structure to the native one) using the proposed algorithm. As the third future work, in addition to energy, we may consider other secondary structural information, such as the amino-acid sequence structure, as well.

Table 3. Results of the proposed ICA-based method for 6 pairs of helices with the secondary structure of *β-LACTAMASE*. *Legends*: Eforcast: foretasted energy, Enative: native energy, Folding-amin1,2: amino acids of the 1st and 2nd helices

Iteration	RMSD	Eforecast	Enative	Folding-amin2	Folding-amin1
105	0.738571	-1.147	-1.323	272-290	26-40
95	0.91562	-1.412	-1.858	200-213	72-86
92	0.589088	-2.531	-2.872	118-129	72-86
90	0.825724	-2.982	-3.107	133-142	72-86
90	0.504832	-3.127	-3.247	145-154	72-86
195	0.96557	-1.184	-1.475	182-195	72-86

Conflict of interests
None.

References

Arab, S., Sadeghi, M., Eslahchi, C., Pezeshk, H. and Sheari, A. (2010) A pairwise residue contact area-based mean force potential for discrimination of native protein structure, BMC Bioinformatics, 11:1, 16.

Bouziane, H., Messabih, B. and Chouarfia, A. (2015) Effect of simple ensemble methods on protein secondary structure prediction, Soft Computing, 19:6, 1663-1678.

Bu, Y. and Zhu, Y. (2009) Artificial immune ant colony algorithm and its application, in 2009 IEEE International Conference on Intelligent Computing and Intelligent Systems. IEEE, 75-80.

Burkowski, F.J. (2008) Structural bioinformatics: an algorithmic approach. Chapman and Hall/CRC.

Coutsias, E.A. and Wester, M.J. (2019) RMSD and symmetry, Journal of Computational Chemistry, 40:15, 1496-1508.

Elahi, A. and Babamir, S.M. (2018) Identification of essential proteins based on a new combination of topological and biological features in weighted protein-protein interaction networks, IET Systems Biology, 12:6, 247-257.

Enireddy, V., Karthikeyan, C. and Babu, D.V. (2022) Onehotencoding and LSTM-based deep learning models for protein secondary structure prediction, Soft Computing, 26:8, 3825-3836.

Gao, M. and Skolnick, J. (2021) A general framework to learn tertiary structure for protein sequence characterization, Frontiers in Bioinformatics, 1, 1-12.

Guyeux, C., Cote, N.M.-L., Bahi, J.M. and Bienia, W. (2014) Is protein folding problem really a NP-complete one? First investigations, Journal of Bioinformatics and Computational Biology, 12:01, 1350017.

Huang, B. et al. (2023) Protein structure prediction: challenges, advances, and the shift of research paradigms, Genomics, Proteomics & Bioinformatics, 21:5, 913-925.

Kaveh, A. and Bakhshpoori, T. (2019) Chapter 6: imperialist competitive algorithm, in Metaheuristics: outlines, MATLAB codes and examples. Springer.

Khaji, E., Karami, M. and Garkani-Nejad, Z. (2016) 3D protein structure prediction using imperialist competitive algorithm and half sphere exposure prediction, Journal of Theoretical Biology, 391, 81-87.

Kuhlman, B. and Bradley, P. (2019) Advances in protein structure prediction and design, Nature Reviews Molecular Cell Biology, 20:11, 681-697.

Li, Y., Zhou, C. and Zheng, X. (2015) Artificial bee colony algorithm for the protein structure prediction based on the toy model, Fundamenta Informaticae, 136:3, 241-252.

Lin, J., Zhong, Y., Li, E., Lin, X. and Zhang, H. (2018) Multi-agent simulated annealing algorithm with parallel adaptive multiple sampling for protein structure prediction in AB off-lattice model, Applied Soft Computing, 62, 491-503.

Lin, X., Zhang, X. and Zhou, F. (2014) Protein structure prediction with local adjust tabu search algorithm, BMC Bioinformatics, 15:S15, S1.

MacCarthy, E., Perry, D. and Kc, D.B. (2019) Advances in protein super-secondary structure prediction and application to protein structure prediction, in Protein supersecondary structures: methods and protocols. Springer, 15-45.

Marquez-Chamorro, A.E. et al. (2015) Soft computing methods for the prediction of protein tertiary structures: a survey, Applied Soft Computing, 35, 398-410.

Muñoz, V. (ed.) (2022) Protein folding: methods and protocols. New York: Springer.

Nabil, B. and Sadek, B. (2020) Protein structure prediction in the HP model using scatter search algorithm, in The 4th International Symposium on Informatics and its Applications (ISIA). IEEE, 1-5.

Rashid, M. et al. (2016) An enhanced genetic algorithm for ab initio protein structure prediction,

IEEE Transactions on Evolutionary Computation, 20:4, 627-644.

Rashid, M. et al. (2019) Constructing effective energy functions for protein structure prediction through broadening attraction-basin and reverse monte carlo sampling, BMC Bioinformatics, 20:S3, 118.

Rayesha, S.M.S., Banu, W.A. and Priya, S. (2023) The prediction of protein structure using neural network, in International Conference on Data Management, Analytics and Innovation. Springer, 1021-1028.

Saudagar, P. and Tripathi, T. (eds.) (2023) Protein folding dynamics and stability: experimental and computational methods. Springer.

Sekhar, S.R.M., Matt, S.D. and Mahadevachar, V.K. (2023) Protein tertiary structure prediction by integrating ant colony optimization with path relinking and structure knowledge, International Journal of Information Technology, 15:3, 1399-1405.

Tantar, A.-A., Melab, N. and Talbi, E.-G. (2008) A grid-based genetic algorithm combined with an adaptive simulated annealing for protein structure prediction, Soft Computing, 12:12, 1185-1198.

Varela, D. and Santos, J. (2022) Protein structure prediction in an atomic model with differential evolution integrated with the crowding niching method, Natural Computing, 21:4, 537-551.

Wang, F., Xu, C., Jiang, S. and Xu, F. (2020) Application of improved intelligent ant colony algorithm in protein folding prediction, Journal of Algorithms & Computational Technology, 14, 1-7.

Webb, B. and Sali, A. (2014) Comparative protein structure modeling using MODELLER, Current Protocols in Bioinformatics, 47:1, 5.6.1-5.6.32.

Yang, L., Qing, Y., Liwei, W. and Jian, P. (2018) Learning structural motif representations for efficient protein structure search, Bioinformatics, 34:17, i773-i780.

Yousef, M., Abdelkader, T. and ElBahnasy, K. (2017) A hybrid model to predict proteins tertiary structure, in 2017 12th International Conference on Computer Engineering and Systems (ICCES). IEEE, 85-91.

Yu, S., Li, X., Tian, X. and Pang, M. (2022) Protein structure prediction based on particle swarm optimization and tabu search strategy, BMC Bioinformatics, 23, 537.