INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

WILEY

INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

Intl. Trans. in Op. Res. 0 (2022) 1–36
DOI: 10.1111/itor.13164

On the role of metaheuristic optimization in bioinformatics

Laura Calvet[a,b,*], Sergio Benito[a], Angel A. Juan[c] and Ferran Prados[d,e,f]

[a] *Department of Computer Science, Universitat Oberta de Catalunya, Barcelona 08018, Spain*
[b] *Department of Industrial Organization, Escola Universitària Salesiana de Sarrià (EUSS), Barcelona 08017, Spain*
[c] *Department of Applied Statistics and Operations Research, Universitat Politècnica de València, Alcoy 03801, Spain*
[d] *e-Health Center, Universitat Oberta de Catalunya, Barcelona 08018, Spain*
[e] *Centre for Medical Image Computing, Department of Medical Physics and Biomedical Engineering, University College London, London WC1V 6BH, United Kingdom*
[f] *Queen Square MS Centre, Department of Neuroinflammation, UCL Institute of Neurology, Faculty of Brain Sciences, University College London, London WC1B 5EH, United Kingdom*
E-mail: lcalvetl@uoc.edu [Calvet]; sbenitor@uoc.edu [Benito]; ajuanp@upv.es [Juan]; fpradosc@uoc.edu [Prados]

Received 27 April 2020; received in revised form 10 May 2022; accepted 23 May 2022

Abstract

Metaheuristic algorithms are employed to solve complex and large-scale optimization problems in many different fields, from transportation and smart cities to finance. This paper discusses how metaheuristic algorithms are being applied to solve different optimization problems in the area of bioinformatics. While the text provides references to many optimization problems in the area, it focuses on those that have attracted more interest from the optimization community. Among the problems analyzed, the paper discusses in more detail the molecular docking problem, the protein structure prediction, phylogenetic inference, and different string problems. In addition, references to other relevant optimization problems are also given, including those related to medical imaging or gene selection for classification. From the previous analysis, the paper generates insights on research opportunities for the Operations Research and Computer Science communities in the field of bioinformatics.

Keywords: metaheuristics; bioinformatics; combinatorial optimization

1. Introduction

According to Glover and Kochenberger (2006), metaheuristics are iterative processes that guide a set of subordinate heuristics in order to generate high-quality solutions for an optimization problem. When the optimization problem is NP-hard, metaheuristics are required to solve large-scale instances in reasonable computing times. As discussed in Gendreau and Potvin (2019), most metaheuristics have to keep a balance between diversification, that is, exploring the space of possible solutions, and intensification, that is, focusing the search in a given region of the solution space.

[*] Corresponding author.

Evolution in computer power and advances in methodological research have transformed these algorithms into one of the most popular approaches for solving NP-hard optimization problems, especially when dealing with large-scale instances. Hence, multiple applications of metaheuristics can be found in areas such as transportation and logistics (Gruler et al., 2018), telecommunication networks (Alvarez et al., 2018), bioinformatics (Sperschneider, 2008; Axelson-Fisk, 2010; Blum and Festa, 2016), quantitative finance (Soler-Dominguez et al., 2017), simulation–optimization (Gruler et al., 2017; Guimarans et al., 2018), etc. Bioinformatics has become an interdisciplinary knowledge area that combines computer science, life sciences, and data science in order to get insights from large biological datasets. There are a high number of applications in this field. Some of the most important applications are diagnosis of disease and disease risks, custom-tailored therapy, genetic counseling-carrier status, and drug discovery. Today, the landscape of bioinformatics is changing dramatically as a consequence of the increasing availability of data with growing quality, quantity, and variety. There is more quantitative data, more precise, and more comprehensive, as well as new ways to recombine data and a new set of tools for analysis and applications (Lesk, 2019). This data deluge has led to the proliferation of massive databases. Some examples are GenBank (https://www.ncbi.nlm.nih.gov/genbank/), which contains 93 million of protein sequences, and the Protein Data Bank (https://www.rcsb.org/), which holds roughly 147, 000 known structures.[1] For instance, databases of protein structures are critical in the research of diseases caused by protein folding, for example, Alzheimer, cystic fibrosis, cancer, and neurodegenerative disorders. Some experimental methods, such as X-ray crystallography or nuclear magnetic resonance, are costly to implement and time-consuming. In this context, the use of databases makes this research more efficient. There are multiple works applying metaheuristics in bioinformatics problems. Some examples are the studies of Blum and Festa (2016), Cohen (2004), Corne and Fogel (2003), Sperschneider (2008), or Axelson-Fisk (2010). In addition, there seems to be an increasing interest in applying metaheuristics to the analysis of motifs in DNA sequences, multiple sequence alignment of DNA sequences, and 3D protein structure prediction (PSP) (Fig. 1). All these applications can be modeled as NP-hard combinatorial optimization problems (COPs).

The main goal of this paper is to provide insights on how metaheuristic algorithms have been used in solving optimization problems in bioinformatics. In particular, we have focused on those problems that we consider especially relevant. Still, we have also included others in order to provide an “extended road map” of metaheuristics applications in the area. We consider that such a map can be very valuable for researchers and practitioners in the areas of bioinformatics and optimization. Of course, there are other ways to deal with most of these problems, for example, simulation-based methods, machine learning methods, fuzzy methods, classical optimization methods, and “black-box” commercial software. Nevertheless, it is our view that metaheuristics (and potential combinations of metaheuristics with simulation and machine learning) are one of the most powerful “white-box” scientific tools that both researchers and practitioners have in their hands to cope with bioinformatics optimization problems of large size.

The rest of the paper follows the structure described next. Section 2 considers some of the most popular optimization problems in the area of bioinformatics. Section 3 offers a quick review of metaheuristic algorithms. Sections 4 and 5 discuss the existing work on how metaheuristics have been applied to address the molecular docking problem and the PSP problem. Sections 6 and 7

[1] Data extracted from the websites on the 1 December 2019.

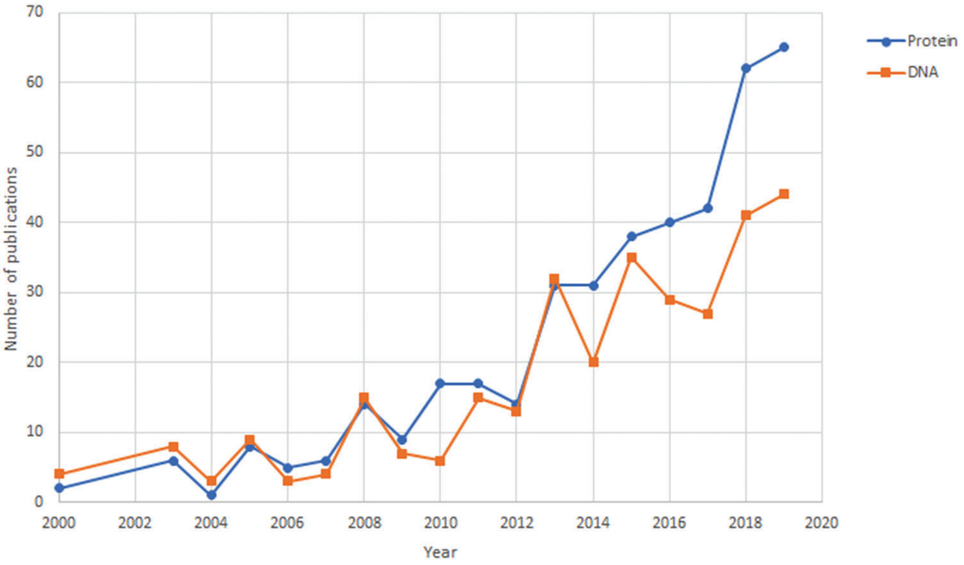

Fig. 1. Scopus-indexed publications applying metaheuristics to DNA problems (in particular, analysis of motifs and multiple sequence alignment) and 3D protein structure prediction for the period 2000–2019.

analyze metaheuristic approaches for dealing with problems in phylogenetics and string problems in bioinformatics. Section 8 discusses the use of metaheuristics in other bioinformatics problems. Section 9 elaborates on open challenges and emerging trends related to the use of metaheuristics in bioinformatics. Finally, Section 10 highlights the main results of this work. To enhance the readiness of the paper, a list of acronyms is provided in Table 1.

2. Optimization problems in bioinformatics

As studies in the field of bioinformatics progress, new research questions arise. A number of them may be modeled as optimization problems, where the goal is to identify the best solution among all the feasible ones. Optimization problems can be classified depending on the associated variables, which can be either continuous or discrete. The standard form of an optimization problem includes a single (or multiple) objective function to either maximize or minimize, equality and inequality constraints, and the variables' domain. In particular, many optimization problems are COPs, which aim at finding optimal solutions within a vast set of possible ones. Some examples of COPs are the following ones: the vehicle routing problem (Belloso et al., 2019), the flow-shop scheduling problem (Hatami et al., 2018), the facility location problem (Pagès-Bernaus et al., 2019), and the arc routing problem (González-Martín et al., 2012). In the area of bioinformatics, some of the most relevant COPs are as follows.

- Molecular docking problem (Section 4): Its main goal is to identify a minimum binding energy conformation between a receptor molecule and a small molecule (ligand). Solving this problem is important for drug discovery, among other applications.

Table 1
List of acronyms employed in this document

Acronym	Full form
ABC	Artificial bee colony
ACO	Ant colony optimization
AIS	Artificial immune systems
C-LCS	Constrained longest common subsequence
CMSA	Construct, merge, solve, and adapt
COP	Combinatorial optimization problem
CRW	Chaotic random walk
CSA	Cuckoo search algorithm
CSP	Closest string problem
CT	Computed tomography
CTMSP	Close to most string problem
DE	Differential evolution
EA	Evolutionary algorithm
FA	Firefly algorithm
FAP	Fragment assembly problem
FFMSP	Far from most string problem
FSP	Farthest string problem
GA	Genetic algorithm
GC-LCS	Generalized constrained longest common subsequence
GRASP	Greedy randomized adaptive search procedure
GSA	Gravitational search algorithm
IBEA	Indicator-based evolutionary algorithm
ICA	Imperialist competition algorithm
ILP	Integer linear programming
ILS	Iterated local search
LCS	Longest common subsequence
LNS	Large neighborhood search
MCSP	Minimum common string partition problem
MOEA/D	Multiobjective evolutionary algorithm based on decomposition
MR	Magnetic resonance
MSFBC	Most strings with few bad columns
NB	Naïve Bayes
NSGA	Nondominated sorting genetic algorithm
OLC	Overlap-layout-consensus
PBHS	Population-based harmony search
PSO	Particle swarm optimization
PSP	Protein structure prediction
RF-LCS	Repetition-free longest common subsequence
SA	Simulated annealing
SBH	Sequencing by hybridization
SS	Scatter search
SVM	Support vector machine
TFBS	Transcription factor binding sites
TS	Tabu search
UMCSP	Unbalanced minimum common string partition problem
VNS	Variable neighborhood search

- PSP (Section 5): It aims at predicting the minimum-energy structure of a protein by setting the angles that the amino acid forms. Determining the 3D structure of proteins is necessary to understand their functions at a molecular level.
- Phylogenetic trees (Section 6): Phylogenetics is the scientific study of the evolutionary history and relationships among individuals or groups of organisms such as species or populations. Inference methods are used to discover these relationships by evaluating observed heritable traits, such as DNA sequences. Phylogenetics has contributed to many scientific fields, for example, medicine, systematic biology, and epidemiology (Felix, 2015).
- String problems (Section 7):
  - The alignment problem aims at aligning biological sequences (generally proteins, DNA, or RNA) to determine their similarity or difference. This contributes to identifying functionally relevant DNA regions and spot mutations and evolutionary relationships.
  - DNA sequencing, which, given a DNA molecule, aims at determining the content and the order of the nucleotides. As an example of application, fast DNA sequencing technologies provide support to customized medical care.
  - Finding motifs has the goal to abstract the task of discovering short and conserved sites in genomic DNA. These patterns, motifs, play an essential role in recognizing transcription factor binding sites that help in learning the mechanisms for regulation of gene expression.
  - Consensus string problems aim to find a “consensus” string representing all the strings considering a set of criteria. Among others, some applications of these problems are related to the discovering of potential drug targets or primer design.
  - Longest common subsequence (LCS) problems aim at identifying the largest string that is a subsequence of every string in a set. Several variants include additional constraints.
  - Unbalanced minimum common string partition (UMCSP) problems, which rearrange a given string in order to obtain a specific one.
  - Most strings with few bad columns (MSFBC) problems aim at identifying outliers in a set of DNA sequences that come from a nonhomogeneous population.
  - Also, some optimization problems are related to genome rearrangements.

Finally, as discussed in Section 8, other COPs can be found in the following.

- Medical imaging: It involves the acquisition of *in vivo* data from inside our bodies. Here, the application of metaheuristic algorithms to improve standard image-based procedures seems to be far from reaching its full potential.
- Gene selection for classification: The selection of relevant genes for classification is an essential task in almost all gene expression studies. Researchers aim at finding the smallest set of genes that achieve a relatively good predictive performance.

3. Basics of metaheuristics

There is a wide range of methods that may be employed to tackle NP-hard COPs. The use of exact methods usually requires simplifying the model (objective functions, constraints, or variable domains) or solving only small-sized instances of these problems. Otherwise, the computational resources required (times and memory) tend to be exceedingly high. Heuristic and metaheuristic approaches overcome this issue, enabling us to solve large-sized instances employing a reasonable amount of computational resources. Heuristic methods are simple and fast procedures based on the COP tackled. Thus, they are relatively flexible but do not guarantee the optimality of the solutions obtained (Talbi, 2013). They are recommended for addressing more realistic and richer models. Regarding metaheuristic methods, they constitute a heterogeneous family of algorithms developed to address diverse COPs without having to deeply adapt them to each problem. Some of these methods are nature-inspired, have stochastic components, and have a number of parameters that must be fine-tuned (Boussaïd et al., 2013; Calvet et al., 2016). According to Feo and Resende (1995), the effectiveness of metaheuristic methods depends on factors such as their ability to adapt to a given instance and exploit the structure of the problem, while avoiding entrapment at local optima. During the last decades, the popularity of metaheuristics has grown and, nowadays, there is a wide range of applications in a number of fields, for instance: logistics and transportation (Onggo et al., 2019), telecommunication networks (Alvarez et al., 2018), bioinformatics (Ali and Hassanien, 2016), finance (Doering et al., 2019; Panadero et al., 2020), scheduling (Ferrer et al., 2016), etc. Talbi (2009) proposes the following criteria to classify metaheuristics: memory usage versus memory-less methods; iterative versus greedy; deterministic versus stochastic; and single solution based search versus population-based search (i.e., exploitation vs. exploration-oriented). In addition, metaheuristics may be designed to deal with a single objective or with multiple objectives. In the following, we mention some of the most common and cite some reference works: artificial bee colony (ABC) (Karaboga, 2005), ant colony optimization (ACO) (Dorigo, 1992), artificial immune systems (AIS) (Farmer et al., 1986), differential evolution (DE) (Storn and Price, 1997), genetic algorithms (GA) (Holland, 1962), greedy randomized adaptive search procedure (GRASP) (Feo and Resende, 1995; Ferone et al., 2019), iterated local search (ILS) (Martin et al., 1992), particle swarm optimization (PSO) (Eberhart and Kennedy, 1995), simulated annealing (SA) (Kirkpatrick, 1984), scatter search (SS) (Glover, 1977), tabu search (TS) (Glover, 1986), and variable neighborhood search (VNS) (Mladenovic, 1995). In addition, many hybrid approaches have been proposed in recent years providing good results in a number of fields. Among the most popular we have matheuristics (i.e., the combination of metaheuristics with mathematical programming) (Archetti and Speranza, 2014), simheuristics (i.e., the combination of metaheuristics with simulation) (Cabrera et al., 2014), learnheuristics (i.e., the combination of metaheuristics with machine learning) (Bayliss et al., 2020), etc.

4. Molecular docking

In molecular docking, the goal is to identify the best conformation, regarding root median square deviation and energy, between a ligand molecule and a macro-molecule, also called receptor. Since more than one criteria can be considered, the problem can be formulated as a multiobjective one. Typically, the relation between the receptor and the ligand molecule is represented as an objective function that includes several components: (i) the translation of the ligand across a 3D coordinate system, which is represented by a vector of coordinates (x, y, z); (ii) the orientation of the ligand, represented as a vector which also includes the angle slope (ω); and (iii) the flexibility parameters,

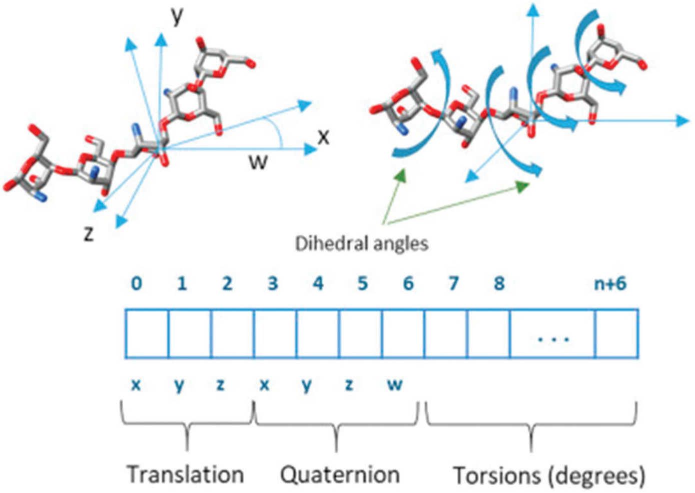

Fig. 2. Solution encoding. The coordinates of the center of rotation of the ligand are given by the first three values (translation). The ligand or macromolecule orientation is given by the next four values. The torsion angles in degrees are stored in the rest of the values (García-Godoy et al., 2019).

which model the degrees of torsion of the ligand and the receptor. This way, a solution to this problem can be represented as a vector with $n + 7$ variables, as depicted in Fig. 2.

Each solution to the molecular docking problem can be assessed using an energy-based objective function. Some of the most popular scoring functions are reviewed in Liu and Wang (2015). For example, the free binding energy function refers to the differences between the receptor and the ligand in both a bonded and an unbounded state. García-Godoy et al. (2015) analyze several optimization algorithms for the multiobjective version of the problem, in which the goal is to minimize the intermolecular and intramolecular energies. Among the algorithms considered in this work, the authors employ two versions of the nondominated sorting genetic algorithm II (NSGA-II) (Deb et al., 2002), a PSO, a DE algorithm, an evolutionary algorithm (EA), and the *S*-metric evolutionary algorithm. The authors analyze the performance of these algorithms by measuring several indicators, including their capability to find scattered points in the associated Pareto front. In a more recent work, Pérez-Serrano et al. (2018) test different metaheuristic algorithms like GRASP, GA, and SS. Likewise, García-Godoy et al. (2019) discuss how hybrid metaheuristics have inspired new solvers and local search methods, which can be effectively used to deal with docking problems. Korb et al. (2007) present the docking algorithm PLANTS (protein–ligand ANT system), which is based on the ACO metaheuristic and compare the results to a popular GA, confirming that ACO algorithms can contribute toward the state of the art for solving the practically relevant protein–ligand docking problems.

5. Protein structure prediction

Nuclear magnetic resonance and X-ray crystallography are the main experimental methods used for protein structure determination. They are time-consuming and expensive. To avoid these impediments, computational methods are applied, which take less time and are cheaper. The main computational methods can be classified into three categories: (i) homology or comparative modeling based in the similarity of the sequences; (ii) threading or fold recognition, used when there is no match to the sequence of some known structure; and (iii) the *ab initio* or *de novo* approach, where the 3D structure is predicted from the primary sequence alone, based on properties of amino acids. In real life, the latter approach is, computationally speaking, very expensive. Current energy functions have limited accuracy as the length of the sequence increases.

Our review focuses on the last method. Here, metaheuristics can help to predict the fold of proteins without homology of known structure, as well as to understand the folding process.

5.1. Computational model

The model on which the PSP is based tries to minimize the free energy of the protein with respect to all possible conformations, based on thermodynamical laws. A function that represents the free energy has been suggested by Li et al. (1996) and is defined in Equation (1):

$$E = \sum_{i<j} e(r_i r_j) \triangle (r_i - r_j), \tag{1}$$

where: r_i and r_j are the ith and jth amino acids of the sequence, while $\triangle(r_i - r_j)$ takes the value of 1 if amino acids r_i and r_j have a nonlocal bond, being 0 elseways. Depending of the type of contact between amino acids, the energy e_{HP}, e_{HH}, or e_{PP} corresponds to H-P, H-H, or P-P contacts.

Lattice models are the predominant class of simplified models, and are divided into HP lattice models and AB off-lattice models. These models were proposed by Dill (1985) and Stillinger et al. (1993), respectively, and classify the 20 amino acids according to their hydrophobic or hydrophilic properties. In the HP lattice model, the bond angle—that is, the angle between two bonds—is a 90° angle, while in the AB off-lattice model its folding angle can be any value between 0 and 360. In other words, in the AB off-lattice model any adjacent three amino acids are in the same plane. In addition, the AB off-lattice model considers the impact of nonlocal effects between nonadjacent amino acids. Hence, the AB off-lattice model is closer to real protein structures than the HP lattice model. For that reason, the PSP problem is addressed using the AB off-lattice model.

In the AB off-lattice model, the 20 different amino acids that can conform to a protein are simplified to 2 different monomers. These monomers represent the amino acids: A and B represent hydrophobic and polar amino acids, respectively. The primary structure of any protein can be simplified to a sequence of As and Bs. The monomers are connected by bonds, whose separation is also simplified to 1 unit. The angle formed by the bonds is represented by θ. A total of $n-2$ bend angles are needed for any protein structure formed by n-monomers addressed with the AB off-lattice model. Figure 3 represents a protein composed of nine monomers. The model is represented in 2D and 3D.

Figure 4 represents a 3D model as extension of a 2D model by adding the torsion angle to each bond. The 3D structure of a chain conformed by n monomers is specified by $2n-5$ angle parameters $(\theta_1, \theta_2, \ldots, \theta_{n-2}, \beta_1, \beta_2, \ldots, \beta_{n-3})$, where the θ_i represent bond angles $(\forall i \in \{1, 2, \ldots, n-2\})$, and the β_j represent the twist angles $(\forall j \in \{1, 2, \ldots, n-3\})$. The energy value (E) that the model

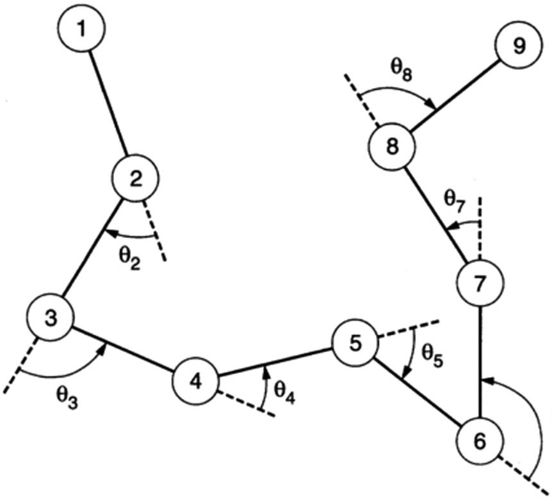

Fig. 3. Generic representation of a hypothetic 9-mer protein structure with its bended angles (Lin et al., 2014).

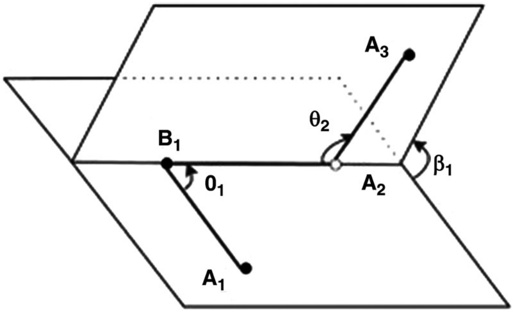

Fig. 4. AB off-lattice model representation (Li et al., 2015b).

intends to minimize is composed of the bending potential energy (E_1) of the peptide chain and by the gravitational potential energy (E_2) between the nearby amino acids, as shown in Equation (2):

$$E = \sum_{i=1}^{n-1} E_1(\theta_i) + \sum_{i=1}^{n-2} \sum_{j=i+2}^{n} E_2(r_{ij}, \xi_i, \xi_j). \tag{2}$$

Equation (3) shows that bending potential energy only takes into account the bond angles, while the gravitational potential energy considers the distance r between the two nonadjacent peptides and polarity, as shown in Equation (4):

$$E_1 = \sum_{i=1}^{n-1} \frac{1}{4}(1 - \cos\theta_i), \tag{3}$$

Table 2
Energy values for monomer combinations

i/j	A	B
A	1	−1/2
B	−1/2	+1/2

$$E_2 = \sum_{i=1}^{n-2} \sum_{j=i+2}^{n} 4[r_{ij}^{-12} - C(\xi_i, \xi_j) r_{ij}^{-6}]. \tag{4}$$

Here, r_{ij} indicates the separation between amino acids i and j in the protein. In Equation (5), ξ indicates the type of peptide and C indicates the polarity between monomers. The energy value for 'A' monomers is 1, while it is -1 for 'B':

$$C(\xi_i, \xi_j) = \frac{1}{8}(1 + \xi_i + \xi_j + 5\xi_i\xi_j). \tag{5}$$

Equation (5) takes the value 1 if $\xi_i = \xi_j = 1$, takes the value 0.5 if $\xi_i = \xi_j = -1$, and takes the value -0.5 otherwise. Given two amino acids i and j of two species ξ_i and ξ_j, the different values allowed of energy calculated in Equation (5) can be collected in Table 2.

Analyzing these values, we can deduce that correlations between AA pairs are stronger than BB pairs, and other pairs have a weaker repulsion. Therefore, the minimum protein energy E can be computed through optimal values of the bond and twist angles associated with the protein energy function, that is, $(\theta_1, \theta_2, \ldots, \theta_{n-2}\ , \beta_1, \beta_2, \ldots, \beta_{n-3})$.

5.2. State of the art

A list of recent works on PSP based on the 3D AB off-lattice is provided in Table 3 and presented in this subsection. Bošković and Brest (2020) suggest to divide the optimization process into two phases in order to improve the efficiency of the algorithm. According to the results, the proposed two-phase optimization mechanism improves the efficiency of the previous algorithm proposed by the authors in Bošković and Brest (2016). Jana et al. (2018a) present a DE algorithm with a fast convergence rate. For testing the algorithm, a selection of amino acid sequences frequently used in literature was performed. The proposed algorithm shows significant performance in terms of accuracy and convergence speed to obtain global optimum solution. In Jana et al. (2018c), the experimental results on standard benchmark functions show that ruggedness and deception measures are appropriately detected by the proposed chaotic random walk (CRW) algorithm. Li et al. (2015a) have demonstrated that balance-evolution ABC can outperform other state-of-the-art approaches from the literature with sequences shorter than 100 amino acids. Lin et al. (2014) introduce a combination of SA and TS algorithms in a method (SATS). Various strategies are adopted successfully for high-speed searching the optimal conformation. Experimental results show that some of the results obtained by the improved SATS are better than those reported in previous

Table 3
Recent articles on metaheuristics dealing with protein structure prediction

Article	Metaheuristic
Bošković and Brest (2020)	DE
Narloch and Dorn (2019)	DE
Jana et al. (2018a)	DE
Jana et al. (2018b)	PSO-DE
Jana et al. (2018c)	CRW
Narloch and Parpinelli (2017, 2016)	DE
Oliveira et al. (2017)	DE
Bošković and Brest (2016)	DE
Li et al. (2015a, 2015b)	Balance-evolution ABC
Lin and Zhang (2014)	Local adjust TS
Parpinelli et al. (2014)	Swarm intelligence
Sar and Acharyya (2014)	GA
Scalabrin et al. (2014)	Population-based harmony search
Zhou et al. (2014)	TS
Kalegari and Lopes (2013)	Parallel DE
Băutu and Luchian (2010)	PSO
Zhang et al. (2010)	GA

literature. Parpinelli et al. (2014) compare four different algorithms: PSO, ABC, gravitational search algorithm (GSA), and bat algorithm. The algorithms were evaluated using two criteria: quality of solutions and the processing time. The results show that the PSO algorithm presented the overall best balance. Also, both PSO and GSA displayed potential to evolve even better solutions, if more iterations were given. Sar and Acharyya (2014) have taken six variants of GA and compared their performances. The variant that uses elitist selection method with two points crossover outperforms other variants in minimizing energy. Scalabrin et al. (2014) present a new EA based on the standard harmony search strategy, called population-based harmony search (PBHS). To achieve multiple function evaluations at the same time, a parallelization method for the proposed PBHS was done. PBHS achieved significantly better quality of solutions and speed-ups than harmony search. Zhou et al. (2014) present an algorithm that combines PSO, GA, and TS algorithms. Experiments show that the proposed method outperforms single algorithms on the accuracy of calculating the protein sequence energy value especially for long protein sequences. However, the proposed algorithm needs more computation cost and more function evaluations. In Kalegari and Lopes (2013) three different implementations of the DE algorithm were developed, one sequential and two parallel. A better performance was shown in the parallel implementations than the sequential one. The sequence benchmark length used to perform the experiments was from 13 to 55 monomers. For most of the sequences where DE was implemented, good results were achieved compared with other works in the literature but not achieving optimal values. The growing interest in this topic has led scientists to create a biennial community named Critical Assessment of Techniques for Protein Structure Prediction (CASP) where participants build models of three-dimensional structures for amino acid sequences provided. In 2018, CASP13 experienced a noticeable progress in structure modeling without the need for using structural templates—historically *ab initio* modeling (Kryshtafovych et al., 2019). The AlphaFold 1 program (Torrisi et al., 2020) is a deep learning system that explores large databanks of related DNA sequences from different organisms and searches for changes at different residues that seem to be correlated—the residues may not be consecutive in the main chain. The observed correlations reveal that the residues may be close to each other physically, thus allowing for the estimation of a contact map. The second version of the program, AlphaFold 2, has won CASP14 in 2020 (Jumper et al., 2021). Basically, AlphaFold 1 combines local physics with a guide potential derived from pattern recognition. Researchers found that the program had a tendency to over-account for interactions between residues that were nearby. Hence, they modified it in the second version to further improve its performance.

5.2.1. Evolution-based metaheuristic algorithms

Gradient-based mathematical methods cannot solve the PSP problem due to the fact that energy function may not be differentiable. For this reason, researchers commonly use EA to address the PSP problem. DE is one of the most effective search strategies for complex problems, including the PSP (Narloch and Parpinelli, 2016, 2017; Oliveira et al., 2017). An *ab initio* method with a variation-adding problem-domain knowledge to enhance the search mechanism has been recently proposed by Narloch and Dorn (2019). As amino acids can assume different torsion angle values depending on their secondary structure, these occurrences consider information to reduce the search space while enhancing algorithms with better search capabilities. This allows for achieving better results –when compared with other approaches– in terms of energy and root mean square deviation.

5.2.2. Swarm intelligence based metaheuristic algorithms

The PSO metaheuristic is based on the principles of swarm intelligence, that is, it uses a set of potential solutions (called particle swarm) to address optimization problems. Particles navigate in the problem landscape communicating and collaborating with each other searching for high-quality solutions. Various off-lattice models have been addressed successfully with the real-valued PSO (Pérez-Hernández et al., 2009; Zhu et al., 2009). Despite the wide search space, PSO is endowed by the off-lattice models with freedom to explore it as well as feedback. Băutu and Luchian (2010) propose a PSO for the PSP using 2D lattice models.

5.2.3. Hybrid algorithms

Oftentimes, using a single optimization technique to solve a problem is not enough. In this way, different techniques are merged to get the benefits of each of them. During the last decade, several articles have been published with hybrid algorithms to tackle the PSP problem. For instance, Zhang et al. (2010) propose a hybrid algorithm where crossover and mutation operator of GA are improved by using TS. Zhou et al. (2014) introduce a hybrid optimization algorithm, which combines PSO, GA, and TS. In the proposed method each algorithm is improved itself using some improvement strategy. Another hybrid technique, named GAPSO and suggested by Lin and Zhang (2014), combines GA and PSO to predict the native conformation of proteins. In Jana et al. (2018b), an integrated framework of hybridization has been proposed. In this framework, called the hybrid PSO-DE algorithm, interleaving between improved versions of PSO and DE are performed for solving multimodal problems such as the PSP. The improved version of PSO is designed by

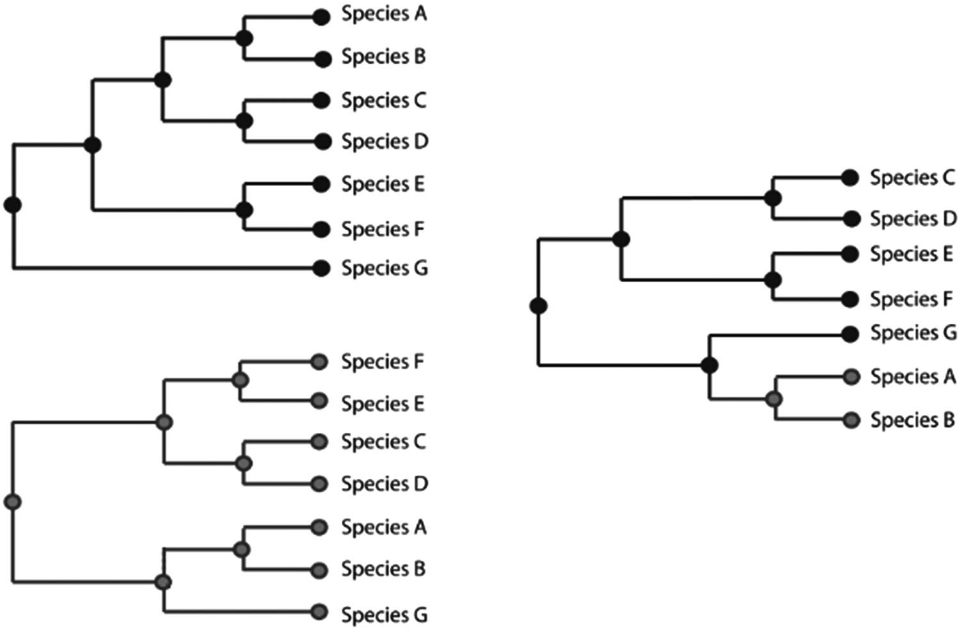

Fig. 5. Examples of phylogenetic trees.

incorporating adaptive polynomial mutation to the global best particle of PSO in order to increase the diversity in the swarm.

6. Phylogenetics

A phylogenetic tree (also called phylogeny) is a leaf-labeled tree representing the evolutionary relationships among various biological species. Figure 5, adapted from Villalobos-Cid et al. (2019), shows examples of phylogenetic trees, where a set of species (A through G) can be grouped in different ways. While the leaves represent the species being considered, the internal nodes constitute divergence events, and the length of the branches show the evolutionary time or distance.

Trees can be classified as rooted—if a common ancestor exists—or unrooted. Felsenstein (1978) presented a simple method based on recurrence relations to count the number of possible evolutionary trees. Given n species, the number of possible rooted, $r(n)$, and unrooted, $u(n)$, trees can be computed applying these equations:

$$r(n) = \frac{(2n-3)!}{(n-2)!\,2^{n-2}} \tag{6}$$

$$u(n) = \frac{(2n-5)!}{(n-3)!\,2^{n-3}}. \tag{7}$$

Table 4
Recent articles on metaheuristics dealing with phylogenetics

Article	Metaheuristic
Santander-Jiménez et al. (2022)	Shuffled frog-leaping algorithm
Villalobos-Cid et al. (2022)	Multimodal algorithm based on NSGA-II
Nayeem et al. (2020a)	Summation of normalized objectives-based GA
Nayeem et al. (2020b)	NSGA-II, NSGA-III, and MOEA/D
Santander-Jiménez et al. (2020)	Interalgorithm cooperation with elite island
Santander-Jiménez et al. (2019)	Shuffled frog-leaping algorithm
Villalobos-Cid et al. (2019)	Memetic algorithm
Zhang et al. (2019)	Parallel EA
Garnier et al. (2018)	GA, PSO, and SA
Santander-Jiménez et al. (2018)	Frog-leaping algorithm
Zambrano-Vega et al. (2016)	EA

There are several approaches to assemble phylogenetic trees (De Bruyn et al., 2014), which are classified in distance- and character-based methods. The former apply clustering techniques using a distance measure from the input sequences. Some examples are unweighted pair group method with arithmetic mean, weighted pair group method with arithmetic mean, neighbor-joining (NJ), and bio-NJ (BioNJ). Other distance-based methods return approximated solutions by optimizing a single criterion, such as the minimum evolution or least-squares error. In contrast, the character-based methods compare all sequences simultaneously considering one character/site at a time. This category includes the following methods: maximum parsimony (e.g., Hoang et al. 2018), maximum likelihood (Kozlov et al., 2019), and Bayesian inference (Dang and Kishino, 2019). One can see the phylogenetic inference as an optimization problem in which the goal is to find, for a given criterion, the best-fit tree among a number of topologies. Indeed, several metaheuristic-based methodologies have been proposed recently. Traditionally, phylogenetic trees were built based on comparisons among individuals of morphological and physiological characters. Molecular phylogenetics (Scornavacca et al., 2020) was born in the middle of the 20th century, but the emergence of high-throughput sequencing in the new century revolutionized this field, reshaping the methodological challenges. Having access to genome-scale data calls for more intensive algorithmic optimization approaches, which can consider the most widely used tree-building methods. The use of massive datasets requires from the study of issues such as data quality (bias, inconsistency, etc.) and model adequacy.

6.1. State of the art

A list of recent works on phylogenetics is provided in Table 4 and presented in this subsection. Zambrano-Vega et al. (2016) present a phylogenetic inference software called MO-Phylogenetics. This software combines the multiobjective optimization techniques of jMetalCpp (López-Camacho et al., 2014), the bioinformatics libraries of the framework BIO++ (Dutheil et al., 2006) and the optimized functions of the phylogenetic likelihood library (Flouri et al., 2015). In fact, jMetalCpp is a C++ based framework that includes well-known EAs such as NSGA-II and the multiobjective EA based on decomposition (MOEA/D) (Zhang and Li, 2007). Similarly, BIO++ includes methods to read, store, and manipulate phylogenetic trees and functions to reconstruct phylogenies from sequence data using maximum parsimony, maximum likelihood, and distance-based methods. Finally, the phylogenetic likelihood library offers some utilities for reducing the high memory requirements and allow the analysis of large datasets. The goal is to return a set of phylogenetic trees that represent trade-off solutions between parsimony and likelihood. A set of computational experiments based on three nucleotide datasets is described. This software is freely accessible from GitHub.

Santander-Jiménez et al. (2019) present the multiobjective shuffled frog-leaping algorithm, which combines parallel searches and swarm-based operators. Three different design alternatives are considered: a dominance-based approach, an indicator-based alternative, and an adaptive proposal incorporating both strategies. It is validated with instances containing protein data with divergent characteristics regarding the number of sequences and sequence length. The multiobjective quality metrics evaluated are set coverage and the indicator I_H. The results are compared against state-of-the-art algorithms such as the NSGA-II algorithm. Villalobos-Cid et al. (2019) put forward a memetic algorithm based on the NSGA-II algorithm and aims to maximize both parsimony and likelihood. The authors carried out computational experiments with four different crossover operators and three mutation operators that are also used in the local search strategies. The experiments allow them to compare different configurations of their algorithm and to validate the proposed approach by comparing their results with those from state-of-the-art optimization algorithms using several benchmark instances. The algorithms are implemented in R (R Core Team, 2021). The methodology described is enhanced in Villalobos-Cid et al. (2022). This work introduces an *ad hoc* multimodal operator that includes information about the decision space in order to increase the diversity of solutions. An optimization problem is labeled as multimodal when it has more than one Pareto set of solutions. Zhang et al. (2019) propose a parallelized multiobjective EA by deploying on the Spark open-source unified analytics engine. The algorithm uses consensus information in each subpopulation (to improve convergence) and has a membrane structure that limits the number of trees in each subpopulation (to eliminate the imbalance between parallel working nodes). Both parsimony and likelihood criteria are considered in the objective function. Computational experiments are based on three real nucleotide datasets. An interalgorithm cooperative approach is presented in Santander-Jiménez et al. (2020). It relies on three algorithms: the NSGA-II, the indicator-based EA (IBEA) (Zitzler and Künzli, 2004), and the MOEA/D. The cooperation among the algorithms is supervised by an elite island component. Experiments are based on five protein datasets with different number of sequences and sequence length. According to the results, this approach shows greater benefits, in terms of parsimony and likelihood, than stand-alone algorithms, standard island models, and other state-of-the-art algorithms.

Phylogenetic inference is becoming increasingly complex as a result of more realistic formulations and larger input datasets. Acknowledging this, Santander-Jiménez et al. (2022) explore the combination of multilevel parallelism and heterogeneous computing. In particular, they describe a parallel metaheuristic, which establishes joint exploitation of parallel tasks at the algorithm, iteration, and solution levels. The computational experiments are based on five real-world biological datasets and report accelerations up to 396× over the baseline metaheuristic and relevant energy savings in comparison to different parallel approaches. The baseline metaheuristic is a multiobjective frog-leaping optimization described in Santander-Jiménez et al. (2018).

An approach to construct a species tree from a set of estimated gene trees, considering several optimization criteria, is designed by Nayeem et al. (2020a). Gene trees may significantly differ from each other, which makes challenging the species tree estimation. The authors develop the summation of normalized objectives, and they use a GA to address the problem. Actually, their GA is an adaptation of the NSGA-II one. The experiments are based on challenging simulated datasets and compare the results of several popular methods. The multiple sequence alignment problem is studied as a previous task for estimating phylogenetic trees by Nayeem et al. (2020b). These authors state that the aforementioned task is usually carried out considering a single objective. This work shows that even high-quality alignments (according to popular measures) may fail to achieve acceptable accuracy when generating phylogenetic trees. In this context, they present an application-oriented multiobjective approach for computing MSAs for the purpose of phylogeny inference. Their approach relies on the multivariate linear regression and domain knowledge methods. The NSGA-II, the NSGA-III (Deb and Jain, 2013), and MOEA/D are implemented using an open-source framework called jMetalMSA (https://github.com/jMetal/jMetalMSA), while experiments are based on both simulated and biological datasets.

Also in the context of phylogenetics, Garnier et al. (2018) analyze a different problem. The aim is to build large-scale phylogenetic trees of plant species from completely sequenced chloroplast genomes. This is a challenging task because of a few core genes disturbing the phylogenetic information. Thus, it is necessary to identify these genes and remove them from the analysis. Computational experiments, based on a large set of chloroplasts, are carried out to assess and compare the performance of three metaheuristics: a GA, a PSO, and a SA. The effects of genes on tree topology and supports are assessed using Lasso tests.

From this subsection it can be concluded that several challenging optimization problems emerge from phylogenetics inference. There are many recent and diverse works. The researchers identify the following topics as highly relevant for future research: multiobjective, parallel computing, cooperation among algorithms, heterogeneous computing, energy savings, parameter fine-tuning (sensitivity analysis), deep learning, and use of prior knowledge, among others.

7. String problems

The DNA store most of the genetic instructions of living organisms and can be represented in the form of a string. For this reason, there is a number of string problems in bioinformatics. In addition, there are problems involving protein sequences. Blum and Festa (2016) study the following string problems in bioinformatics that can be naturally addressed with metaheuristics: the minimum common string partition (MCSP) problem, the LCS problem, the most strings with few bad columns problem, the consensus string problem, the alignment problem, the DNA sequencing problem, and the founder sequence reconstruction problem. This section briefly introduces some of the most relevant string problems and reviews recent work related to them.

7.1. Multiple sequence alignment problem

Comparing genomic sequences of individuals from different species means to determine their similarity or difference. This comparison may help to identify DNA regions with relevant

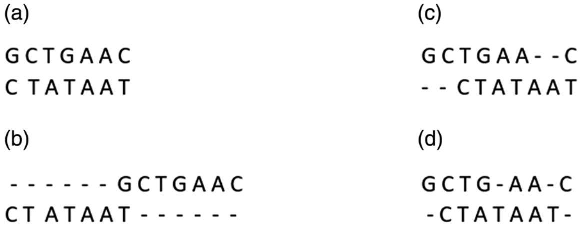

Fig. 6. Example of some possible alignments for the sequences GCTGAAC and GCTGAAC.

functionality, evolutionary relationships, and spot mutations. Given two (or more) sequences, a measure of similarity is designed/chosen, which constitutes a number related to the cardinality of subsets composed of corresponding elements/symbols. The correspondence must be order-preserving, and gaps may be considered. To decide which is the best option among all possible alignments (Fig. 6), a score reflecting the quality of each one is computed.

The pairwise alignment problem refers to two sequences, while the multiple alignment problem deals with a higher number of them. Multiple sequence alignments are more informative in terms of revealing patterns of conservation. Alignment problems can also be split in global or local: the global considers the entire length of the sequences, while the local discards regions of the sequences without similarity.

Solving the alignment problem is a primary phase for finding conserved motifs, performing phylogenetic analysis, predicting protein function, anticipating the secondary structure of protein or classifying proteins. However, this problem is a complex one, and its computational intractability was proven by Wang and Jiang (1994). Accordingly, many heuristics have been developed for addressing it, among others: MULTALIGN (Corpet, 1988), Clustal (Higgins and Sharp, 1988), ClustalW (Thomsom et al., 1994), T-Coffee (Notredame et al., 2000), CHAOS/DIALIGN (Brudno et al., 2003), MUSCLE (Edgar, 2004), ProbCons (Wallace et al., 2004), and Kalign (Lassmann and Sonnhammer, 2005). The use of metaheuristics is also extensive. For instance, Chowdhury and Garai (2017) review works proposing multiobjective GAs. To measure the performance of many GAs against ClustalW the benchmark database of reference alignments BAliBASE is used (Thompson et al., 1999). The authors conclude that GAs are widely used to face this problem and, once more, verify the no-free-lunch theorem in this field: there is no single algorithm that outperforms all the others considering all objective functions and all the benchmark instances.

A list of recent works on the sequence alignment problem is shown in Table 5. Most of them study a multiobjective version of the problem (Ortuno et al., 2013; Rubio-Largo et al., 2015; Zhu et al., 2015a; Rani and Ramyachitra, 2016; Rubio-Largo et al., 2016; Manikandan and Ramyachitra, 2017; Rubio-Largo et al., 2018a; Dabba et al., 2019). Common measures of the objective functions are BAliscore scores, structural information, nongaps percentage, and number of totally conserved columns.

Note that all the metaheuristics proposed are population based. Thus, a usually complex parameter fine-tuning process is required. This issue is tackled by Rubio-Largo et al. (2018b). The authors present a framework to increase both the accuracy and the conservation of the alignment obtained. This framework, for a given input dataset, runs the aligner with the best parameter configuration found for another dataset with similar biological characteristics. The high number of benchmark

Table 5
Recent articles on metaheuristics dealing with the sequence alignment problem

Article	Metaheuristic
Dabba et al. (2019)	Artificial fish swarm algorithm
Hussein et al. (2019)	Flower pollination algorithm
Chaabane (2018)	PSO and GA
Mohsen et al. (2018)	Harmony search algorithm
Rani and Ramyachitra (2018, 2016)	ABC and bacterial foraging optimization
Rubio-Largo et al. (2018a, 2015)	Memetic metaheuristic
Yadav (2018)	Biogeography-based optimization
Zemali and Boukra (2018b)	GSA
Manikandan and Ramyachitra (2017)	Bacterial foraging optimization and GA
Rubio-Largo et al. (2016)	ABC
Zhu et al. (2015b)	EA based on decomposition
Ortuno et al. (2013)	GA
Sievers et al. (2011)	GA

instances used is also remarkable, as well as the use of advanced statistical analysis to compare different algorithms' performance. The methods described in this subsection may provide optimal (or near-optimal) alignments, but an optimal alignment may not correspond to the biologically correct one. Thompson and Schulz (1999) proposed a scoring system in their program CLUSTALW, in which parameters such as gap penalties are adjusted according to the characteristics of input sequences (e.g., sequence divergence, length, and local hydropathy). Katoh et al. (2002) designed a method for rapid multiple sequence alignment based on fast Fourier transform named MAFFT.

7.2. DNA sequencing

DNA sequencing refers to the process of determining the content and the order of the nucleotides in a DNA molecule. This knowledge is relevant in many application fields. In medical diagnosis, for example, comparing healthy and mutated DNA sequences can help to diagnose diseases such as different cancers (Chmielecki and Meyerson, 2014). In addition, being able to sequence DNA in a rapid way boosts a faster and individualized medical care (Abate et al., 2013). In medical research, DNA sequencing can contribute to the detection of the genes associated with acquired or hereditary diseases. Yet another application can be found in forensic science, where DNA sequencing can constitute valid proof in trials. Since the first DNA sequences were obtained, in the early 1970s, constant advances in both sequencing technology and algorithmic development have made DNA sequencing increasingly faster (Myers, 2016). The process of DNA sequencing may be modeled as a COP. Since there are diverse sequencing technologies (with different features), there are different optimization models as well.

A widely used and relatively basic technology is shotgun sequencing. As explained by Parla et al. (2011), in shotgun sequencing large pieces of DNA are sheared into smaller fragments that are then sequenced. Then these smaller pieces must be realigned and arranged into longer contiguous sequences representative of the initial DNA fragment. Due to the arbitrariness of the process, to increase the probability of most of the original DNA fragment will be represented by overlapping fragments is necessary to use redundancy. Thus, generating a large number of sequences is typically required. Another example is DNA sequencing by hybridization (SBH), in which sets of oligonucleotides are hybridized under certain conditions that allow the detection of complementary sequences in the target nucleic acid. According to Drmanac et al. (2002), the sequence search parallelism of this strategy has enabled the creation of high-throughput, low-cost, miniaturized sequencing processes on arrays of DNA samples or probes. The use and progress of genome sequencing have advanced with the help of technological innovations that have improved the speed and quality and reduced the price, meaning that sequencing is feasible without large financial resources. At the very least, the price drop in DNA sequencing is as important as the efficiency, and for that reason, the National Human Genome Research Institute (NHGRI) has tracked the costs associated with DNA sequencing performed using this information as an important benchmark for assessing improvements in DNA sequencing technologies.

7.3. The fragment assembly problem

The DNA fragment assembly problem (FAP) constitutes a challenging problem where copies of DNA strands have to be sequenced and the resolution consists in assembling the pieces into the most plausible DNA sequence. It is usually linked to shotgun sequencing. The invention of gel sequencing techniques led to early fragment assembly methods, which followed the overlap-layout-consensus (OLC) paradigm (Kececioglu and Myers, 1995). This approach decomposes the FAP into three subproblems: (i) overlap, where all approximate overlaps between fragments are computed considering potential base call errors; (ii) layout, in which a credible ordering is set for joining overlapping fragments; and (iii) consensus, which is complex as a consequence of base call errors and unknown orientation. Most articles address the layout problem, which is the most challenging one due to the redundancy and the fact that the fragment can be in either orientation. A classical goal is to maximize the sum of overlap lengths between each fragment and the consequent one in the layout. In addition to the OLC paradigm, other popular strategies are greedy graph-based algorithms (Chikhi et al., 2016) and de Bruijn graphs, based on sequence k-mers, as the SPAdes program developed by Bankevich et al. (2012). For instance, Hughes et al. (2016) present a collection of GA variations: recentering–restarting GA, island model GA, and a GA that employs ring species. An interesting contribution is made by Vidal and Olivera (2018). This work develops a discrete firefly algorithm (FA) on a graphics processing unit (GPU) architecture aiming to speed up the search process for solving the FAP. More recently, Ali et al. (2020) design and implement variants of the PSO algorithm, which use heuristic information and local search.

A list of recent articles dealing with the fragment assembly problem is shown in Table 6. Most works have a single objective: the sum of overlap lengths. Other measures of performance reported sometimes are the computational time and the average number of fitness evaluations. The most common benchmarks are presented in Mallén-Fullerton et al. (2013).

7.3.1. Sequencing by hybridization

SBH was first presented by researchers in the early 1980s. This process consists of two phases. First, a biochemical experiment is carried out in a solvent, where the unknown, fluorescently labeled

Table 6
Recent articles on metaheuristics dealing with the fragment assembly problem

Article	Metaheuristic
Ali et al. (2020)	PSO
Allaoui et al. (2018)	Hybrid crow search algorithm
Vidal and Olivera (2018)	FA
Zemali and Boukra (2018a)	ABC
Ali et al. (2017)	Local search algorithm
Gheraibia et al. (2016)	Penguins search optimization
Huang et al. (2016)	Memetic gravitation search algorithm
Hughes et al. (2016)	Recentering–restarting GA
Ülker (2016)	Harmony search algorithm
Huang et al. (2015)	Memetic PSO
Indumathy et al. (2015)	Cuckoo search algorithm (CSA)
Rajagopal and Maheswari Sankareswaran (2015)	PSO

Table 7
Recent articles on metaheuristics dealing with DNA sequencing by hybridization

Article	Metaheuristic
Swaminathan et al. (2019)	Hypergraph-based GA
Kwarciak et al. (2016)	ACO
Caserta and Voß (2014)	Matheuristic
Kwarciak and Formanowicz (2014)	TS
Blazewicz et al. (2013)	Hyperheuristics

DNA sequence hybridizes to a DNA chip. The examined DNA sequence hybridizes only to those oligonucleotides that are complementary on the chip. The result of the experiment is a spectrum, which is a set of subfragments of the unknown DNA sequence being examined. In this first phase, two different types of errors can take place: (i) positive errors, which are the oligonucleotides in the spectrum but not in the subfragments of the studied sequence; and (ii) negative errors, which are represented by the oligonucleotides that are not in the spectrum despite being subfragments of the DNA sequence. The second phase is the reconstruction of the unknown DNA sequence using the elements in the spectrum. For instance, Caserta and Voß (2014) model the DNA SBH problem as an orienteering problem, which is a variation of the popular traveling salesman problem with profits. The authors present a matheuristic, which is validated using 400 benchmark instances. Kwarciak et al. (2016) explore the same problem. These authors develop a multilevel ACO and test two realistic multiplicity information models, which seem to enable their algorithm to outperform the existing ones. Finally, Swaminathan et al. (2019) present a hypergraph-based GA to address this problem.

A list of recent articles on DNA SBH is shown in Table 7. Performance measures commonly reported are related to similarity scores and computational times. Benchmark instances are described in Blazewicz et al. (2006). Typically, authors perform diverse computational experiments, considering different values for the percentage of both positive and negative errors.

Table 8
Recent articles on metaheuristics dealing with finding motifs in DNA

Article	Metaheuristic
Ge et al. (2019)	Random projection and PSO
Gohardani et al. (2019)	ICA
González-Álvarez et al. (2015)	DE, VNS, ABC, FA, GSA, NSGA-II, and SPEA2 (Zitzler et al., 2001)
Huan et al. (2015)	ACO
González-Álvarez et al. (2013)	ABC and GSA

7.4. Finding motifs in DNA

The process in which a gene is transcribed to form an RNA sequence is known as gene expression. The RNA sequence enables the production of the related protein sequence. This process begins when the transcription factor (a macromolecule) is bounded to a transcription factor binding site (TFBS), which is a short subsequence in the promoter region of the gene (Zare-Mirakabad et al., 2009). These TFBSs are characterized by highly similar domains with consensus patterns, so called motif. Motifs tend to be distributed in promoter sequences and have a length of 8–15 amino acids. The promoter is generally the base sequence with length 1000–2000 base pairs. The goal here is to recognize motifs. Finding motifs in DNA enables the uncovering of the underlying regulatory relationship and the understanding of the evolutionary mechanism associated with living organisms.

González-Álvarez et al. (2015) compare the performance of seven multiobjective metaheuristics for finding motifs. Gohardani et al. (2019) propose a multiobjective imperialist competition algorithm (ICA). The performance of the metaheuristics is assessed by many metrics, such as hypervolume and coverage relation. In these two articles, the goal is to maximize the length, support, and similarity of the motif simultaneously. Similarly, Ge et al. (2019) present a PSO and random projection-based algorithm. The nucleotide-level performance coefficient is selected to assess the performance of the metaheuristic. It depends on the number of nucleotide positions in both known and predicted sites. A list of recent articles on finding motifs in DNA is shown in Table 8, while benchmark instances are described in Tompa et al. (2005).

7.5. Consensus string problems

Consensus string problems aim to determine a consensus string for a given finite set of strings. Blum and Festa (2016) describe four different problems.

- The closest string problem (CSP): The consensus is a new string that minimizes the total distance from all the strings.
- The close to most strings problem (CTMSP): The consensus is a new string close to most of the strings.
- The farthest string problem (FSP): The consensus is a new string that maximizes the total distance from all the strings.

Table 9
Recent articles on metaheuristics dealing with consensus string problems

Article	Metaheuristic
Ferone et al. (2016)[b]	GRASP
Gallardo and Cotta (2015)[b]	GRASP-based memetic algorithm
Blum and Festa (2014)[b]	ACO
Della Croce and Garraffa (2014)[a]	Multistart
Pappalardo et al. (2014)[a]	SA
Ferone et al. (2013)[b]	GRASP and VNS
Mousavi et al. (2012)[b]	GRASP
Tanaka (2012)[a]	TS

[a] CSP.
[b] FFMSP.

- The far from most strings problem (FFMSP): The consensus is a new string far from most of the strings.

There are related problems, such as the center string problem, which is a restricted version of the CSP where the solution string must be taken from the given set of strings (Nicolas and Rivals, 2005). It is also worthwhile to mention that different metrics may be used (such as Hamming or Levenshtein distances), which implies various versions of the problem, each with its own distinct complexity. A list of recent articles on consensus string problems is shown in Table 9.

7.6. Longest common subsequence problems

Given a set $S = \{s_1, s_2, \dots, s_n\}$ of n strings over a finite alphabet, the LCS problem consists in finding the longest string t that is a subsequence of all the strings in S. It is considered that the string t is obtained by deleting characters of a string s. The LCS problem has applications in many fields, bioinformatics is one of them. It can be solved through dynamic programming, but this approach becomes impractical for large sizes of n. Other restricted versions have been tackled in the literature, being the most important ones.

- The repetition-free LCS (RF-LCS) problem: There are two input strings, s_1 and s_2, over a finite alphabet; the aim is to obtain the LCS of s_1 and s_2, taking into account that no letter may appear more than once in a valid solution.
- The constrained LCS (C-LCS) problem: There are three input strings, s_1, s_2, and s_c, over a finite alphabet; the aim is to obtain the LCS of s_1 and s_2 that contains s_c as a subsequence.
- The generalized constrained LCS (GC-LCS) problem: There are three input strings, s_1, s_2, and t, over a finite alphabet; the aim is to obtain the LCS of s_1 and s_2 that includes (or excludes) t as a subsequence.

A list of recent articles on LCS problems is shown in Table 10. Most algorithmic proposals for these problems have been evaluated using randomly generated instances or real data, but there are

Table 10
Recent articles on metaheuristics dealing with longest common subsequence problems

Article	Metaheuristic
Islam et al. (2019)[a]	Chemical reaction optimization
Blum and Blesa (2018a)[b]	Matheuristic algorithm
Blum and Blesa (2018b)[a]	Matheuristic algorithm
Blum and Blesa (2016)[b]	Construct, merge, solve, and adapt (CMSA) algorithm
Markvica et al. (2015)[a]	ACO
Blum et al. (2013)[b]	Beam-ACO
Castelli et al. (2013)[b]	GA
Mousavi and Tabataba (2012)[a]	Beam search
Tabataba and Mousavi (2012)[a]	Hyperheuristic
Blum (2010)[a]	Beam-ACO
Lozano and Blum (2010)[a]	VNS

[a] LCS.
[b] RF-LCS.

no benchmark instances. As far as we are aware, there is no metaheuristic approach that deals with the C-LCS problem or with the GC-LCS problem.

7.7. Unbalanced minimum common string partition problem

The MCSP is a special case of the UMCSP problem. Let s_1 and s_2 be two strings, both with the same length, over an alphabet such that each letter appears the same number of times in each string s_1, s_2. The MCSP problem is to find the smallest partition P of nonoverlapping substrings of s_1 such that P is a partition of nonoverlapping substrings of s_2 too. The following example from Blum and Festa (2016) illustrates the problem: Let there be two DNA sequences: $s_1 = AGACTG$ and $s_2 = ACTAGG$; clearly, s_1 and s_2 are related, since each letter appears the same number of times in both strings; a trivial valid solution would be $P = \{A, A, C, T, G, G\}$; the objective function value of this solution is $|P| = 6$; but the optimal solution, $P = \{ACT, AG, G\}$, has an objective function value of 3.

There is only one difference between the UMCSP and the MCSP problems: the first does not require the input strings to be related. It may be formally defined in this way: let there be two strings, s_1 and s_2, of lengths n_1 and n_2, respectively, over a finite alphabet. A solution is built by partitioning s_1, into a set P_1 of nonoverlapping substrings, and s_2, into a set P_2 of nonoverlapping substrings, such that (i) a subset $S_1 \subseteq P_1$ and a subset $S_2 \subseteq P_2$ exist, with $S_1 = S_2$; and (ii) no letter a is simultaneously present in a string $x \in P_1 \setminus S_1$ and in a string $y \in P_2 \setminus S_2$. Let S represent the largest subset $S_1 = S_2$ that meets the aforementioned conditions. The objective is to minimize $|S|$.

The MCSP problem was first studied by Goldstein et al. (2004). Blum et al. (2015) present the first integer linear programming (ILP) model for this problem, as well as an ILP-based heuristic. More efficient ILP models have been later proposed by Blum and Raidl (2016) and Ferdous and Rahman (2015). A list of recent articles on the MCSP problem is shown in Table 11. In contrast, the UMCSP has only been studied by Blum (2016), who proposed a CMSA algorithm.

Table 11
Recent articles on metaheuristics dealing with the minimum common string partition problem

Article	Metaheuristic
Blum (2020)	CMSA algorithm and reduced VNS
Ferdous and Rahman (2017, 2013)	MAX-MIN ant system
Blum et al. (2016)	CMSA algorithm
Blum et al. (2014)	Iterative probabilistic tree search

7.8. The most strings with few bad columns problem

The most strings with a few bad columns (MSFBC) problem was initially described by Boucher et al. (2013). It can be worded as follows: Let there be a set I of n input strings of length m over a finite alphabet, that is, $I = \{s_1, s_2, \dots, s_n\}$; let there be a parameter of the problem denoted by k; then, the aim is to create a subset $S \subseteq I$ of maximum size such that the strings in S differ in no more than k positions. A position j in which the strings from S differ is termed a bad column.

Boucher et al. (2013) prove that the MSFBC problem has no polynomial-time approximation unless NP has randomized polynomial-time algorithms. Later, Lizárraga et al. (2015) present an ILP model to address this problem, as well as a simple greedy heuristic and an extension, called the greedy-based pilot method. As expected, the authors find that using CPLEX to tackle the ILP model provides the optimal solution for small and medium instances in reasonable amounts of time, but the heuristic methods scale much better for large instances. Finally, Lizárraga et al. (2017) present a large neighborhood search (LNS) approach, which relies on the ILP solver CPLEX as a subroutine to get, at each iteration, the best possible neighbor in a large neighborhood of the current solution. According to the results, the LNS tends to outperform greedy strategies.

7.9. Genome rearrangement

The most common and most studied mutations operating on DNA sequences are local: they affect only a very small stretch on DNA sequence. These mutations include nucleotide substitutions (where one nucleotide is substituted for another), as well as nucleotide insertions and deletions. Most phylogenetic studies have been based on these types of mutations. Genome rearrangement is a different class of mutation affecting very large stretches of DNA sequence. A genome rearrangement occurs when a chromosome breaks at two or more locations (called the breakpoints), and the pieces are reassembled in a different order. This results in a DNA sequence that has essentially the same features as the original sequence, except that the order of these features has been modified. Although molecular biology gave birth to it, combinatorics of genome rearrangements is now a mathematical and algorithmic field with several studies that has found its own coherence (Pevzner, 2000; Fertin et al., 2009). For instance, Siqueira et al. (2020) analyze the reversal distance between genomes with duplicated genes when the orientation of genes is unknown. This work describes three approaches of metaheuristics using random maps, local search, and GAs. The comparison shows that the heuristic proposed based on local search and GA techniques tend to produce very good solutions in terms of average distance estimation error. In this paper, the authors plan to extend the heuristics by considering other genome rearrangement events as transposition, insertion, and deletion opening new challenges for readers.

8. Other combinatorial optimization problems

There are other COPs in bioinformatics that are interesting for both operational researchers and computer scientists. Since a detailed analysis would significantly increase the size of the manuscript, this section offers just a brief description and provides some relevant references for the interested reader.

8.1. Medical image analysis

Nowadays, medical imaging is the facto standard for acquiring *in vivo* visual information from inside our bodies. There are several ways to do it, among them: ultrasound, computed tomography (CT), magnetic resonance (MR), X-ray, or intraoperative cameras. The application of metaheuristic algorithms in medical imaging has been rather limited and mainly related to image segmentation. Recently, Natarajan and Kumarasamy (2019) use fuzzy logic with a spiking neuron model for segmenting brain tumors in MR images that optimized the weight and bias parameters applying a chicken behavior-based swarm intelligence metaheuristic. Another example is Rodrigues et al. (2017) who apply GAs over CT images to precisely delineate and detect the pericardium contour of the human heart. Image registration also has had some metaheuristic proposals, such as using memetic search optimization for improving multimodal registration through the optimal processing of the signal intensity relationship between image modalities (Hering et al., 2016; Bejinariu and Costin, 2018). Image registration is a classical optimization problem from medical imaging, and an essential step for diagnosis and prognosis. Hering et al. (2016) successfully validated their method using diffusion weighted images with important noise artifacts. This is a diffusion MR fitting method used to study the neuron spatial behavior that is composed of a high number of 3D volumes, which are acquired with different *b*-values (Zhang et al., 2012). It is key that all the 3D volumes are very well aligned for getting a clear representation of the neurite distribution in each voxel. Similarly, Bejinariu and Costin (2018) compare three different metaheuristic approaches (CSA, PSO, and multiswarm optimization) for registering two images of different modalities.

8.2. Gene selection for classifying

Selecting key genes for classification (e.g., to differentiate between people with and without a given disease) constitutes an essential task in the majority of gene expression studies (Li et al., 2002; Hua et al., 2004; Jirapech-Umpai and Aitken, 2005; Lee et al., 2005; Yeung et al., 2005). The objective of the researchers when they address this gene selection problem tends to be one of the following ones.

- Identifying key genes for future research, which involves selecting a set of genes related to the outcome of interest. These genes may perform similar functions and be highly correlated.
- Identifying small sets of genes that may be useful to diagnose diseases and conditions. This entails selecting the smallest set of genes that achieve a good predictive performance.

Metaheuristics have been largely used for gene selection for classifying (high-dimensional) microarray data, either considering just two classes or more. For instance, recently Pashaei et al. (2019) design a binary PSO and a binary black hole algorithm. Their framework works with the following classifiers: *k*-nearest neighbor, naïve Bayes (NB), and discriminant analysis. These authors test the aforementioned algorithms on different benchmarks and microarrays data with the following measures: convergence rate, accuracy, and number of selected genes. Similarly, Prasad et al. (2018) present a recursive PSO approach for gene selection that is combined with a linear support vector machine (SVM) with the aim of maximizing accuracy. An adaptive GA is proposed by Shukla et al. (2018). It employs an SVM and an NB classifier. The latter is used as a fitness function that allows to identify discriminating genes, as well as to maximize the accuracy of the classification process.

9. Open issues and trends

Despite the growing number of works addressing COPs in bioinformatics, there are many open challenges and some interesting trends that we discuss in this section. First, the technological development of the last decades has brought more powerful computers, making it possible to develop parallel and distributed implementations of metaheuristics. Boosted by this development, during recent years, experts on metaheuristics have presented more advanced and complex designs, such as *multiobjective approaches* or *hybrid metaheuristics*. Moreover, there is an increasing *amount of available data* in bioinformatics, as a consequence of new high-throughput technologies (microarray genomic data, protein and DNA sequences, image-based biomarkers, clinical test, bibliographic data, etc.). This poses *new challenges and problems*, as well as a need for knowledge discovery algorithms (Talbi, 2013). In the past, small benchmarks helped us to experiment with the concepts, and now it is possible to use evolved algorithms to explore large-scale and real-world data, which might show the true potential of these algorithms (Hughes et al., 2014).

Over the following years, metaheuristic algorithms are called to play a relevant role in *variable selection* within the context of artificial intelligence methods. For example, nowadays disease modeling is a vibrating research field within bioinformatics that requires a great amount of data—patient information from different sources and sample size: number of subjects and time points per subject. However, the biggest challenge for any disease modeling approach is that increasing the number of input variables will also raise the confounding factors that affect the performance and robustness of the proposed models. For this reason, there is a clear need for an optimal selection of the relevant information that will help our models to efficiently delineate the path of any disease progression.

10. Conclusions and future work

Bioinformatics constitutes an interdisciplinary and active field of science. The number of potential applications has grown during recent years. This has been caused by an increase of the quality, quantity, and variety of available data, which has led to new challenges and problems, as well as to the development of related methods and technologies. Many of the related problems may be modeled as NP-hard COPs involving large amounts of data. With the purpose of tackling these problems, the research community increasingly relies on metaheuristic approaches, which enable the solving of large-scale instances using a reduced number of computational resources, including computational time and memory.

Hence, metaheuristic algorithms constitute a powerful tool to solve large-scale optimization problems in the area of bioinformatics. Even when other approaches might also be available for many of these optimization problems, due to their flexibility metaheuristics are always an effective tool that can provide "high-quality" solutions in reasonably low computing times. Among the main advantages of using metaheuristics applied to bioinformatics we can highlight that optimization problems in this field are usually large scale and NP-hard, which impose severe limitations on the use of exact optimization methods. In addition, data provided by researchers and scientists inherently involve errors, and here is where extended metaheuristics methods, such as simheuristics (Chica et al., 2020) and learnheuristics (Bayliss et al., 2020) are more flexible than exact approaches. Several tasks in bioinformatics involve the optimization of different objectives, thereby making the application of metaheuristics more appropriate and natural.

Among the most popular COPs in bioinformatics, one can highlight the PSP problem and several string-related problems. The latter ones cover several optimization problems, such as finding motifs in DNA, alignment problems, etc. In addition, metaheuristics play an essential role in medical imaging and disease modeling (through variable selection, parameter fine-tuning, etc.).

Although computers are becoming increasingly powerful, time complexity will continue to be an issue in bioinformatics due to the huge amount and variety of data, and the need for relatively fast answers. Thus, there are several lines of future research related to metaheuristics and bioinformatics. Some of these seem particularly interesting in our eyes: (i) the development of more powerful algorithms relying on distributed and parallel paradigms, and the hybridization of different algorithms; (ii) the development of more robust algorithms, which take into account the uncertainty or stochasticity in bioinformatics (due to the nature of the data used or to errors caused by the technology employed to capture data); (iii) the deployment of multiobjective approaches to consider the diverse objectives (often conflicting ones) in most problems; and (iv) the design of frameworks for parameter fine-tuning to exploit instance-specific features in order to improve results.

References

Abate, A.R., Hung, T., Sperling, R.A., Mary, P., Rotem, A., Agresti, J.J., Weiner, M.A., Weitz, D.A., 2013. DNA sequence analysis with droplet-based microfluidics. *Lab on a Chip* 13, 24, 4864–4869.

Ali, A.B., Luque, G., Alba, E., 2020. An efficient discrete PSO coupled with a fast local search heuristic for the DNA fragment assembly problem. *Information Sciences* 512, 880–908.

Ali, A.B., Luque, G., Alba, E., Melkemi, K.E., 2017. An improved problem aware local search algorithm for the DNA fragment assembly problem. *Soft Computing* 21, 7, 1709–1720.

Ali, A.F., Hassanien, A.E., 2016. A survey of metaheuristics methods for bioinformatics applications. In *Applications of Intelligent Optimization in Biology and Medicine*. Springer, Berlin, pp. 23–46.

Allaoui, M., Ahiod, B., El Yafrani, M., 2018. A hybrid crow search algorithm for solving the DNA fragment assembly problem. *Expert Systems with Applications* 102, 44–56.

Alvarez, S., Ferone, D., Juan, A.A., Silva, D.G., de Armas, J., 2018. A 2-stage biased-randomized iterated local search for the uncapacitated single allocation p-hub median problem. *Transactions on Emerging Telecommunications Technologies* 29, 9, e3418.

Alvarez, S., Juan, A.A., de Armas, J., e Silva, D.G., Riera, D., 2018. Metaheuristics in telecommunication systems: network design, routing, and allocation problems. *IEEE Systems Journal* 12, 4, 3948–3957.

Archetti, C., Speranza, M.G., 2014. A survey on matheuristics for routing problems. *EURO Journal on Computational Optimization* 2, 4, 223–246.

Axelson-Fisk, M., 2010. Comparative gene finding. In *Comparative Gene Finding*. Springer, Berlin, pp. 157–180.

Bankevich, A., Nurk, S., Antipov, D., Gurevich, A.A., Dvorkin, M., Kulikov, A.S., Lesin, V.M., Nikolenko, S.I., Pham, S., Prjibelski, A.D., Pyshkin, A.V., Sirotkin, A.V., Vyahhi, N., Tesler, G., Alekseyev, M.A., Pevzner, P.A., 2012. SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. *Journal of Computational Biology* 19, 5, 455–477.

Băutu, A., Luchian, H., 2010. Protein structure prediction in lattice models with particle swarm optimization. International Conference on Swarm Intelligence. Springer, Berlin, pp. 512–519.

Bayliss, C., Juan, A.A., Currie, C.S., Panadero, J., 2020. A learnheuristic approach for the team orienteering problem with aerial drone motion constraints. *Applied Soft Computing* 92, 106280.

Bejinariu, S.I., Costin, H., 2018. A comparison of some nature-inspired optimization metaheuristics applied in biomedical image registration. *Methods of Information in Medicine* 57, 280–286.

Belloso, J., Juan, A.A., Faulin, J., 2019. An iterative biased-randomized heuristic for the fleet size and mix vehicle-routing problem with backhauls. *International Transactions in Operational Research* 26, 1, 289–301.

Blazewicz, J., Burke, E.K., Kendall, G., Mruczkiewicz, W., Oguz, C., Swiercz, A., 2013. A hyper-heuristic approach to sequencing by hybridization of DNA sequences. *Annals of Operations Research* 207, 1, 27–41.

Blazewicz, J., Oguz, C., Swiercz, A., Weglarz, J., 2006. DNA sequencing by hybridization via genetic search. *Operations Research* 54, 6, 1185–1192.

Blum, C., 2010. Beam-ACO for the longest common subsequence problem. Congress on Evolutionary Computation. IEEE, Piscataway, NJ, pp. 1–8.

Blum, C., 2016. Construct, merge, solve and adapt: application to unbalanced minimum common string partition. International Workshop on Hybrid Metaheuristics, Springer, Berlin, pp. 17–31.

Blum, C., 2020. Minimum common string partition: on solving large-scale problem instances. *International Transactions in Operational Research* 27, 91–111.

Blum, C., Blesa, M.J., 2016. Construct, merge, solve and adapt: application to the repetition-free longest common subsequence problem. In *Evolutionary Computation in Combinatorial Optimization*, Springer, Berlin, pp. 46–57.

Blum, C., Blesa, M.J., 2018a. A comprehensive comparison of metaheuristics for the repetition-free longest common subsequence problem. *Journal of Heuristics* 24, 3, 551–579.

Blum, C., Blesa, M.J., 2018b. Hybrid techniques based on solving reduced problem instances for a longest common subsequence problem. *Applied Soft Computing* 62, 15–28.

Blum, C., Blesa, M.J., Calvo, B., 2013. Beam-ACO for the repetition-free longest common subsequence problem. International Conference on Artificial Evolution. Springer, Berlin, pp. 79–90.

Blum, C., Festa, P., 2014. A hybrid ant colony optimization algorithm for the far from most string problem. European Conference on Evolutionary Computation in Combinatorial Optimization. Springer, Berlin, pp. 1–12.

Blum, C., Festa, P., 2016. *Metaheuristics for String Problems in Bio-informatics*. John Wiley & Sons, Hoboken, NJ.

Blum, C., Lozano, J.A., Davidson, P., 2015. Mathematical programming strategies for solving the minimum common string partition problem. *European Journal of Operational Research* 242, 3, 769–777.

Blum, C., Lozano, J.A., Davidson, P.P., 2014. Iterative probabilistic tree search for the minimum common string partition problem. International Workshop on Hybrid Metaheuristics. Springer, Berlin, pp. 145–154.

Blum, C., Pinacho, P., López-Ibáñez, M., Lozano, J.A., 2016. Construct, merge, solve & adapt: a new general algorithm for combinatorial optimization. *Computers & Operations Research* 68, 75–88.

Blum, C., Raidl, G.R., 2016. Computational performance evaluation of two integer linear programming models for the minimum common string partition problem. *Optimization Letters* 10, 1, 189–205.

Bošković, B., Brest, J., 2016. Differential evolution for protein folding optimization based on a three-dimensional AB off-lattice model. *Journal of Molecular Modeling* 22, 10, 252.

Bošković, B., Brest, J., 2020. Two-level protein folding optimization on a three-dimensional AB off-lattice model. *Swarm and Evolutionary Computation* 57, 100708.

Boucher, C., Landau, G.M., Levy, A., Pritchard, D., Weimann, O., 2013. On approximating string selection problems with outliers. *Theoretical Computer Science* 498, 107–114.

Boussaïd, I., Lepagnot, J., Siarry, P., 2013. A survey on optimization metaheuristics. *Information Sciences* 237, 82–117.

Brudno, M., Chapman, M., Göttgens, B., Batzoglou, S., Morgenstern, B., 2003. Fast and sensitive multiple alignment of large genomic sequences. *BMC Bioinformatics* 4, 1, 66.

Cabrera, G., Juan, A.A., Lázaro, D., Marquès, J.M., Proskurnia, I., 2014. A simulation-optimization approach to deploy internet services in large-scale systems with user-provided resources. *Simulation* 90, 6, 644–659.

Calvet, L., Juan, A.A., Serrat, C., Ries, J., 2016. A statistical learning based approach for parameter fine-tuning of metaheuristics. *Statistics and Operations Research Transactions* 1, 1, 201–224.

Caserta, M., Voß, S., 2014. A hybrid algorithm for the DNA sequencing problem. *Discrete Applied Mathematics* 163, 87–99.

Castelli, M., Beretta, S., Vanneschi, L., 2013. A hybrid genetic algorithm for the repetition free longest common subsequence problem. *Operations Research Letters* 41, 6, 644–649.

Chaabane, L., 2018. A hybrid solver for protein multiple sequence alignment problem. *Journal of Bioinformatics and Computational Biology* 16, 4, 1850015.

Chica, M., Juan, A.A., Bayliss, C., Cordón, O., Kelton, W.D., 2020. Why simheuristics? Benefits, limitations, and best practices when combining metaheuristics with simulation. *Statistics and Operations Research Transactions* 44, 2, 311–334.

Chikhi, R., Limasset, A., Medvedev, P., 2016. Compacting de Bruijn graphs from sequencing data quickly and in low memory. *Bioinformatics* 32, 12, i201–i208.

Chmielecki, J., Meyerson, M., 2014. DNA sequencing of cancer: what have we learned? *Annual Review of Medicine* 65, 63–79.

Chowdhury, B., Garai, G., 2017. A review on multiple sequence alignment from the perspective of genetic algorithm. *Genomics* 109, 5-6, 419–431.

Cohen, J., 2004. Bioinformatics: An introduction for computer scientists. *ACM Computing Surveys (CSUR)* 36, 2, 122–158.

Corne, D.W., Fogel, G.B., 2003. An introduction to bioinformatics for computer scientists. In *Evolutionary Computation in Bioinformatics*. Elsevier, Amsterdam, pp. 3–18.

Corpet, F., 1988. Multiple sequence alignment with hierarchical clustering. *Nucleic Acids Research* 16, 22, 10881–10890.

Dabba, A., Tari, A., Zouache, D., 2019. Multiobjective artificial fish swarm algorithm for multiple sequence alignment. *Information Systems and Operational Research* 58, 38–59.

Dang, T., Kishino, H., 2019. Stochastic variational inference for Bayesian phylogenetics: a case of CAT model. *Molecular Biology and Evolution* 36, 4, 825–833.

De Bruyn, A., Martin, D.P., Lefeuvre, P., 2014. Phylogenetic reconstruction methods: an overview. *Methods in Molecular Biology* 1115, 257–277.

Deb, K., Jain, H., 2013. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. *Transactions on Evolutionary Computation* 18, 4, 577–601.

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 2, 182–197.

Della Croce, F., Garraffa, M., 2014. The selective fixing algorithm for the closest string problem. *Computers & Operations Research* 41, 24–30.

Dill, K.A., 1985. Theory for the folding and stability of globular proteins. *Biochemistry* 24, 6, 1501–1509.

Doering, J., Kizys, R., Juan, A.A., Fito, A., Polat, O., 2019. Metaheuristics for rich portfolio optimisation and risk management: current state and future trends. *Operations Research Perspectives* 6, 100121.

Dorigo, M., 1992. Optimization, learning and natural algorithms. PhD thesis, Politecnico di Milano.

Drmanac, R., Drmanac, S., Chui, G., Diaz, R., Hou, A., Jin, H., Jin, P., Kwon, S., Lacy, S., Moeur, B., Shafto, J., Swanson, D., Ukrainczyk, T., Xu, C., Little, D., 2002. Sequencing by hybridization (SBH): advantages, achievements, and opportunities. In *Chip Technology*. Springer, Berlin, pp. 75–101.

Dutheil, J., Gaillard, S., Bazin, E., Glémin, S., Ranwez, V., Galtier, N., Belkhir, K., 2006. Bio++: a set of C++ libraries for sequence analysis, phylogenetics, molecular evolution and population genetics. *BMC Bioinformatics* 7, 1, 1–6.

Eberhart, R., Kennedy, J., 1995. Particle swarm optimization. Conference on Neural Networks, Vol. 4. IEEE, Piscataway, NJ, pp. 1942–1948.

Edgar, R.C., 2004. MUSCLE: Multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research* 32, 5, 1792–1797.

Farmer, J.D., Packard, N.H., Perelson, A.S., 1986. The immune system, adaptation, and machine learning. *Physica D: Nonlinear Phenomena* 22, 1-3, 187–204.

Felix, B., 2015. Phylogenetics: tracing the evolutionary legacy of organisms metastatic clones bioactive compounds and languages. *Journal of Phylogenetics & Evolutionary Biology* 3, 2, 1000.

Felsenstein, J., 1978. The number of evolutionary trees. *Systematic Biology* 27, 1, 27–33.

Feo, T.A., Resende, M.G., 1995. Greedy randomized adaptive search procedures. *Journal of Global Optimization* 6, 2, 109–133.

Ferdous, S., Rahman, M.S., 2013. Solving the minimum common string partition problem with the help of ants. International Conference in Swarm Intelligence. Springer, Berlin, pp. 306–313.

Ferdous, S., Rahman, M.S., 2015. An integer programming formulation of the minimum common string partition problem. *PLoS ONE* 10, 7.

Ferdous, S., Rahman, M.S., 2017. Solving the minimum common string partition problem with the help of ants. *Mathematics in Computer Science* 11, 2, 233–249.

Ferone, D., Festa, P., Resende, M.G., 2013. Hybrid metaheuristics for the far from most string problem. International Workshop on Hybrid Metaheuristics. Springer, Berlin, pp. 174–188.

Ferone, D., Festa, P., Resende, M.G., 2016. Hybridizations of GRASP with path relinking for the far from most string problem. *International Transactions in Operational Research* 23, 3, 481–506.

Ferone, D., Gruler, A., Festa, P., Juan, A.A., 2019. Enhancing and extending the classical GRASP framework with biased randomisation and simulation. *Journal of the Operational Research Society* 70, 8, 1362–1375.

Ferrer, A., Guimarans, D., Ramalhinho, H., Juan, A.A., 2016. A BRILS metaheuristic for non-smooth flow-shop problems with failure-risk costs. *Expert Systems with Applications* 44, 177–186.

Fertin, G., Labarre, A., Rusu, I., Vialette, S., Tannier, E., 2009. *Combinatorics of Genome Rearrangements*. MIT Press, Cambridge, MA.

Flouri, T., Izquierdo-Carrasco, F., Darriba, D., Aberer, A.J., Nguyen, L.T., Minh, B., Von Haeseler, A., Stamatakis, A., 2015. The phylogenetic likelihood library. *Systematic Biology* 64, 2, 356–362.

Gallardo, J.E., Cotta, C., 2015. A GRASP-based memetic algorithm with path relinking for the far from most string problem. *Engineering Applications of Artificial Intelligence* 41, 183–194.

García-Godoy, M.J., López-Camacho, E., García-Nieto, J., Del Ser, J., Nebro, A.J., Aldana-Montes, J.F., 2019. Bio-inspired optimization for the molecular docking problem: state of the art, recent results and perspectives. *Applied Soft Computing* 79, 30–45.

García-Godoy, M., López-Camacho, E., García-Nieto, J., Nebro, A., Aldana-Montes, J., 2015. Solving molecular docking problems with multi-objective metaheuristics. *Molecules* 20, 6, 10154–10183.

Garnier, R., Guyeux, C., Couchot, J.F., Salomon, M., Al-Nuaimi, B., AlKindy, B., 2018. Comparison of metaheuristics to measure gene effects on phylogenetic supports and topologies. *BMC Bioinformatics* 19, 7, 218.

Ge, H., Yu, J., Sun, L., Wang, Z., Yao, Y., 2019. Discovery of DNA motif utilising an integrated strategy based on random projection and particle swarm optimization. *Mathematical Problems in Engineering* 2019. https://doi.org/10.1155/2019/3854646

Gendreau, M., Potvin, J.Y., 2019. *Handbook of Metaheuristics* (3rd edn). Springer, Berlin.

Gheraibia, Y., Moussaoui, A., Kabir, S., Mazouzi, S., 2016. Pe-DFA: penguins search optimisation algorithm for DNA fragment assembly. *International Journal of Applied Metaheuristic Computing* 7, 2, 58–70.

Glover, F., 1977. Heuristics for integer programming using surrogate constraints. *Decision Sciences* 8, 1, 156–166.

Glover, F., 1986. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research* 13, 5, 533–549.

Glover, F.W., Kochenberger, G.A., 2006. *Handbook of Metaheuristics*, Vol. 57. Springer Science & Business Media, Berlin.

Gohardani, S.A., Bagherian, M., Vaziri, H., 2019. A multi-objective imperialist competitive algorithm (MOICA) for finding motifs in DNA sequences. *Mathematical Biosciences and Engineering* 16, 3, 1575.

Goldstein, A., Kolman, P., Zheng, J., 2004. Minimum common string partition problem: hardness and approximations. International Symposium on Algorithms and Computation. Springer, Berlin, pp. 484–495.

González-Álvarez, D.L., Vega-Rodríguez, M.A., Gómez-Pulido, J.A., Sánchez-Pérez, J.M., 2013. Comparing multiobjective swarm intelligence metaheuristics for DNA motif discovery. *Engineering Applications of Artificial Intelligence* 26, 1, 314–326.

González-Álvarez, D.L., Vega-Rodríguez, M.A., Rubio-Largo, Á., 2015. Multiobjective optimization algorithms for motif discovery in DNA sequences. *Genetic Programming and Evolvable Machines* 16, 2, 167–209.

González-Martín, S., Juan, A.A., Riera, D., Castellà, Q., Muñoz, R., Pérez, A., 2012. Development and assessment of the SHARP and RandSHARP algorithms for the arc routing problem. *AI Communications* 25, 2, 173–189.

Gruler, A., Panadero, J., de Armas, J., Moreno, J.A., Juan, A.A., 2018. Combining variable neighborhood search with simulation for the inventory routing problem with stochastic demands and stock-outs. *Computers & Industrial Engineering* 123, 278–288.

Gruler, A., Quintero-Araújo, C.L., Calvet, L., Juan, A.A., 2017. Waste collection under uncertainty: a simheuristic based on variable neighbourhood search. *European Journal of Industrial Engineering* 11, 2, 228–255.

Guimarans, D., Dominguez, O., Panadero, J., Juan, A.A., 2018. A simheuristic approach for the two-dimensional vehicle routing problem with stochastic travel times. *Simulation Modelling Practice and Theory* 89, 1–14.

Hatami, S., Calvet, L., Fernández-Viagas, V., Framinan, J.M., Juan, A.A., 2018. A simheuristic algorithm to set up starting times in the stochastic parallel flowshop problem. *Simulation Modelling Practice and Theory* 86, 55–71.

Hering, J., Wolf, I., Maier-Hein, K.H., 2016. Multi-objective memetic search for robust motion and distortion correction in diffusion MRI. *IEEE Transactions on Medical Imaging* 35, 10, 2280–2291.

Higgins, D.G., Sharp, P.M., 1988. CLUSTAL: a package for performing multiple sequence alignment on a microcomputer. *Gene* 73, 1, 237–244.

Hoang, D.T., Vinh, L.S., Flouri, T., Stamatakis, A., von Haeseler, A., Minh, B.Q., 2018. MPBoot: fast phylogenetic maximum parsimony tree inference and bootstrap approximation. *BMC Evolutionary Biology* 18, 1, 1–11.

Holland, J.H., 1962. The compact genetic algorithm. *Journal of the ACM* 1, 3.9, 297–314.

Hua, J., Xiong, Z., Lowey, J., Suh, E., Dougherty, E.R., 2004. Optimal number of features as a function of sample size for various classification rules. *Bioinformatics* 21, 8, 1509–1515.

Huan, H.X., Tuyet, D.T., Ha, D.T., Hung, N.T., 2015. An efficient ant colony algorithm for DNA motif finding. In *Knowledge and Systems Engineering*. Springer, Berlin, pp. 589–601.

Huang, K.W., Chen, J.L., Yang, C.S., Tsai, C.W., 2015. A memetic particle swarm optimization algorithm for solving the DNA fragment assembly problem. *Neural Computing and Applications* 26, 3, 495–506.

Huang, K.W., Chen, J.L., Yang, C.S., Tsai, C.W., 2016. A memetic gravitation search algorithm for solving DNA fragment assembly problems. *Journal of Intelligent & Fuzzy Systems* 30, 4, 2245–2255.

Hughes, J., Houghten, S., Ashlock, D., 2016. Restarting and recentering genetic algorithm variations for DNA fragment assembly: the necessity of a multi-strategy approach. *Biosystems* 150, 35–45.

Hughes, J., Houghten, S., Mallén-Fullerton, G.M., Ashlock, D., 2014. Recentering and restarting genetic algorithm variations for DNA fragment assembly. Conference on Computational Intelligence in Bioinformatics and Computational Biology. IEEE, Piscataway, NJ, pp. 1–8.

Hussein, A.M., Abdullah, R., AbdulRashid, N., 2019. Flower pollination algorithm with profile technique for multiple sequence alignment. Jordan International Joint Conference on Electrical Engineering and Information Technology. IEEE, Piscataway, NJ, pp. 571–576.

Indumathy, R., Maheswari, S.U., Subashini, G., 2015. Nature-inspired novel cuckoo search algorithm for genome sequence assembly. *Sadhana* 40, 1, 1–14.

Islam, M.R., Saifullah, C.K., Asha, Z.T., Ahamed, R., 2019. Chemical reaction optimization for solving longest common subsequence problem for multiple string. *Soft Computing* 23, 14, 5485–5509.

Jana, N.D., Das, S., Sil, J., 2018a. The Lévy distributed parameter adaptive differential evolution for protein structure prediction. In Jana, N.D., Das, S., Sil, J. (eds) *A Metaheuristic Approach to Protein Structure Prediction*. Springer, Berlin, pp. 151–167.

Jana, N.D., Das, S., Sil, J., 2018b. Hybrid metaheuristic approach for protein structure prediction. In Jana, N.D., Das, S., Sil, J. (eds) *A Metaheuristic Approach to Protein Structure Prediction*. Springer, Berlin, pp. 197–206.

Jana, N.D., Sil, J., Das, S., 2018c. Continuous fitness landscape analysis using a chaos-based random walk algorithm. *Soft Computing* 22, 3, 921–948.

Jirapech-Umpai, T., Aitken, S., 2005. Feature selection and classification for microarray data analysis: evolutionary methods for identifying predictive genes. *BMC Bioinformatics* 6, 1, 148.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S.A.A., Ballard, A.J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A.W., Kavukcuoglu, K., Kohli, P., Hassabis, D., 2021. Highly accurate protein structure prediction with AlphaFold. *Nature* 596, 7873, 583–589.

Kalegari, D.H., Lopes, H.S., 2013. An improved parallel differential evolution approach for protein structure prediction using both 2D and 3D off-lattice models. Symposium on Differential Evolution. IEEE, Piscataway, NJ, pp. 143–150.

Karaboga, D., 2005. An idea based on honey bee swarm for numerical optimization. Technical Report tr06, Erciyes University.

Katoh, K., Misawa, K., Kuma, K.i., Miyata, T., 2002. MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Research* 30, 14, 3059–3066.

Kececioglu, J.D., Myers, E.W., 1995. Combinatorial algorithms for DNA sequence assembly. *Algorithmica* 13, 1–2, 7.

Kirkpatrick, S., 1984. Optimization by simulated annealing: quantitative studies. *Journal of Statistical Physics* 34, 5-6, 975–986.

Korb, O., Stützle, T., Exner, T.E., 2007. An ant colony optimization approach to flexible protein–ligand docking. *Swarm Intelligence* 1, 2, 115–134.

Kozlov, A.M., Darriba, D., Flouri, T., Morel, B., Stamatakis, A., 2019. RAxML-NG: a fast, scalable and user-friendly tool for maximum likelihood phylogenetic inference. *Bioinformatics* 35, 21, 4453–4455.

Kryshtafovych, A., Schwede, T., Topf, M., Fidelis, K., Moult, J., 2019. Critical assessment of methods of protein structure prediction (CASP)—round XIII. *Proteins: Structure, Function, and Bioinformatics* 87, 12, 1011–1020.

Kwarciak, K., Formanowicz, P., 2014. Tabu search algorithm for DNA sequencing by hybridization with multiplicity information available. *Computers & Operations Research* 47, 1–10.

Kwarciak, K., Radom, M., Formanowicz, P., 2016. A multilevel ant colony optimization algorithm for classical and isothermic DNA sequencing by hybridization with multiplicity information available. *Computational Biology and Chemistry* 61, 109–120.

Lassmann, T., Sonnhammer, E.L., 2005. Kalign—an accurate and fast multiple sequence alignment algorithm. *BMC Bioinformatics* 6, 1, 298.

Lee, J.W., Lee, J.B., Park, M., Song, S.H., 2005. An extensive comparison of recent classification tools applied to microarray data. *Computational Statistics & Data Analysis* 48, 4, 869–885.

Lesk, A., 2019. *Introduction to Bioinformatics* (5th edn). Oxford University Press, Oxford.

Li, B., Chiong, R., Lin, M., 2015a. A balance-evolution artificial bee colony algorithm for protein structure optimization based on a three-dimensional AB off-lattice model. *Computational Biology and Chemistry* 54, 1–12.

Li, B., Lin, M., Liu, Q., Li, Y., Zhou, C., 2015b. Protein folding optimization based on 3D off-lattice model via an improved artificial bee colony algorithm. *Journal of Molecular Modeling* 21, 10, 261.

Li, H., Helling, R., Tang, C., Wingreen, N., 1996. Emergence of preferred structures in a simple model of protein folding. *Science* 273, 5275, 666–669.

Li, Y., Campbell, C., Tipping, M., 2002. Bayesian automatic relevance determination algorithms for classifying gene expression data. *Bioinformatics* 18, 10, 1332–1339.

Lin, X., Zhang, X., 2014. Protein folding structure optimization based on GAPSO algorithm in the off-lattice model. Conference on Bioinformatics and Biomedicine. IEEE, Piscataway, NJ, pp. 43–49.

Lin, X., Zhang, X., Zhou, F., 2014. Protein structure prediction with local adjust tabu search algorithm. In *BMC Bioinformatics*, Vol. 15, BioMed Central, London, p. S1.

Liu, J., Wang, R., 2015. Classification of current scoring functions. *Journal of Chemical Information and Modeling* 55, 3, 475–482.

Lizárraga, E., Blesa, M.J., Blum, C., Raidl, G.R., 2015. On solving the most strings with few bad columns problem: an ILP model and heuristics. 2015 International Symposium on Innovations in Intelligent SysTems and Applications. IEEE, Piscataway, NJ, pp. 1–8.

Lizárraga, E., Blesa, M.J., Blum, C., Raidl, G.R., 2017. Large neighborhood search for the most strings with few bad columns problem. *Soft Computing* 21, 17, 4901–4915.

López-Camacho, E., García Godoy, M.J., Nebro, A.J., Aldana-Montes, J.F., 2014. jMetalCpp: optimizing molecular docking problems with a C++ metaheuristic framework. *Bioinformatics* 30, 3, 437–438.

Lozano, M., Blum, C., 2010. A hybrid metaheuristic for the longest common subsequence problem. International Workshop on Hybrid Metaheuristics. Springer, Berlin, pp. 1–15.

Mallén-Fullerton, G.M., Hughes, J.A., Houghten, S., Fernández-Anaya, G., 2013. Benchmark datasets for the DNA fragment assembly problem. *International Journal of Bio-Inspired Computation* 5, 6, 384–394.

Manikandan, P., Ramyachitra, D., 2017. Bacterial foraging optimization-genetic algorithm for multiple sequence alignment with multi-objectives. *Scientific Reports* 7, 1, 8833.

Markvica, D., Schauer, C., Raidl, G.R., 2015. CPU versus GPU parallelization of an ant colony optimization for the longest common subsequence problem. International Conference on Computer Aided Systems Theory. Springer, Berlin, pp. 401–408.

Martin, O., Otto, S.W., Felten, E.W., 1992. Large-step Markov chains for the TSP incorporating local search heuristics. *Operations Research Letters* 11, 4, 219–224.

Mladenovic, N., 1995. A variable neighborhood algorithm—a new metaheuristic for combinatorial optimization. Papers Presented at Optimization Days, Vol. 12.

Mohsen, M.S., Abdullah, R., Omar, M.A., 2018. A hybrid-based harmony search algorithm for RNA multiple sequence alignment. *Life Science Journal* 15, 11.

Mousavi, S.R., Babaie, M., Montazerian, M., 2012. An improved heuristic for the far from most strings problem. *Journal of Heuristics* 18, 2, 239–262.

Mousavi, S.R., Tabataba, F., 2012. An improved algorithm for the longest common subsequence problem. *Computers & Operations Research* 39, 3, 512–520.

Myers, E.W. Jr, 2016. A history of DNA sequence assembly. *It-Information Technology* 58, 3, 126–132.

Narloch, P.H., Dorn, M., 2019. A knowledge based self-adaptive differential evolution algorithm for protein structure prediction. International Conference on Computational Science. Springer, Berlin, pp. 87–100.

Narloch, P.H., Parpinelli, R.S., 2016. Diversification strategies in differential evolution algorithm to solve the protein structure prediction problem. International Conference on Intelligent Systems Design and Applications. Springer, Berlin, pp. 125–134.

Narloch, P.H., Parpinelli, R.S., 2017. The protein structure prediction problem approached by a cascade differential evolution algorithm using ROSETTA. Brazilian Conference on Intelligent Systems (BRACIS). IEEE, Piscataway, NJ, pp. 294–299.

Natarajan, A., Kumarasamy, S., 2019. Efficient segmentation of brain tumor using FL-SNM with a metaheuristic approach to optimization. *Journal of Medical Systems* 43, 25.

Nayeem, M.A., Bayzid, M.S., Chakravarty, S., Rahman, M.S., Rahman, M.S., 2020a. A multi-objective metaheuristic approach for accurate species tree estimation. International Conference on Bioinformatics and Bioengineering. IEEE, Piscataway, NJ, pp. 79–84.

Nayeem, M.A., Bayzid, M.S., Rahman, A.H., Shahriyar, R., Rahman, M.S., 2020b. Multiobjective formulation of multiple sequence alignment for phylogeny inference. In *Transactions on Cybernetics*. IEEE, Piscataway, NJ, pp. 1–12.

Nicolas, F., Rivals, E., 2005. Hardness results for the center and median string problems under the weighted and unweighted edit distances. *Journal of Discrete Algorithms* 3, 2, 390–415.

Notredame, C., Higgins, D.G., Heringa, J., 2000. T-Coffee: a novel method for fast and accurate multiple sequence alignment. *Journal of Molecular Biology* 302, 1, 205–217.

Oliveira, M., Borguesan, B., Dorn, M., 2017. SADE-SPL: A self-adapting differential evolution algorithm with a loop structure pattern library for the PSP problem. Congress on Evolutionary Computation. IEEE, Piscataway, NJ, pp. 1095–1102.

Onggo, B.S., Panadero, J., Corlu, C.G., Juan, A.A., 2019. Agri-food supply chains with stochastic demands: a multiperiod inventory routing problem with perishable products. *Simulation Modelling Practice and Theory* 97, 101970.

Ortuno, F.M., Valenzuela, O., Rojas, F., Pomares, H., Florido, J.P., Urquiza, J.M., Rojas, I., 2013. Optimizing multiple sequence alignments using a genetic algorithm based on three objectives: structural information, non-gaps percentage and totally conserved columns. *Bioinformatics* 29, 17, 2112–2121.

Pagès-Bernaus, A., Ramalhinho, H., Juan, A.A., Calvet, L., 2019. Designing e-commerce supply chains: a stochastic facility–location approach. *International Transactions in Operational Research* 26, 2, 507–528.

Panadero, J., Doering, J., Kizys, R., Juan, A.A., Fito, A., 2020. A variable neighborhood search simheuristic for project portfolio selection under uncertainty. *Journal of Heuristics* 26, 353–375.

Pappalardo, E., Cantone, D., Pardalos, P.M., 2014. A combined greedy-walk heuristic and simulated annealing approach for the closest string problem. *Optimization Methods and Software* 29, 4, 673–702.

Parla, J., Kramer, M., McCombie, W.R., 2011. High-throughput sequencing. In Budowle, B., Schutzer, S.E., Breeze, R.G., Keim, P.S., Morse, S.A. (eds) *Microbial Forensics* (2nd edn). Academic Press, San Diego, CA, pp. 461–478.

Parpinelli, R.S., Benitiez, C.M., Cordeiro, J., Lopes, H.S., 2014. Performance analysis of swarm intelligence algorithms for the 3D-AB off-lattice protein folding problem. *Multiple-Valued Logic and Soft Computing* 22, 3, 267–286.

Pashaei, E., Pashaei, E., Aydin, N., 2019. Gene selection using hybrid binary black hole algorithm and modified binary particle swarm optimization. *Genomics* 111, 4, 669–686.

Pérez-Hernández, L.G., Rodríguez-Vázquez, K., Garduño-Juárez, R., 2009. Parallel particle swarm optimization applied to the protein folding problem. Annual Conference on Genetic and Evolutionary Computation, pp. 1791–1792. https://doi.org/10.1145/1569901.1570163

Pérez-Serrano, J., Imbernón, B., Cecilia, J.M., Ujaldon, M., 2018. Energy-based tuning of metaheuristics for molecular docking on multi-GPUs. *Concurrency and Computation: Practice and Experience* 30, 17, e4684.

Pevzner, P., 2000. *Computational molecular biology: An algorithmic approach*. MIT Press, Cambridge, MA.

Prasad, Y., Biswas, K., Hanmandlu, M., 2018. A recursive PSO scheme for gene selection in microarray data. *Applied Soft Computing* 71, 213–225.

R Core Team, 2021. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rajagopal, I., Maheswari Sankareswaran, U., 2015. An adaptive particle swarm optimization algorithm for solving DNA fragment assembly problem. *Current Bioinformatics* 10, 1, 97–105.

Rani, R.R., Ramyachitra, D., 2016. Multiple sequence alignment using multi-objective based bacterial foraging optimization algorithm. *Biosystems* 150, 177–189.

Rani, R.R., Ramyachitra, D., 2018. *A Hybridization of Artificial Bee Colony with Swarming Approach of Bacterial Foraging Optimization for Multiple Sequence Alignment*. Springer, Singapore. pp. 39–65.

Rodrigues, É., Rodrigues, L., Oliveira, L., Conci, A., Liatsis, P., 2017. Automated recognition of the pericardium contour on processed CT images using genetic algorithms. *Computers in Biology and Medicine* 87, 38–45.

Rubio-Largo, Á., Castelli, M., Vanneschi, L., Vega-Rodriguez, M.A., 2018a. A parallel multiobjective metaheuristic for multiple sequence alignment. *Journal of Computational Biology* 25, 9, 1009–1022.

Rubio-Largo, Á., Vanneschi, L., Castelli, M., Vega-Rodríguez, M.A., 2018b. Multiobjective characteristic-based framework for very-large multiple sequence alignment. *Applied Soft Computing* 69, 719–736.

Rubio-Largo, Á., Vega-Rodríguez, M.A., González-Álvarez, D.L., 2015. A hybrid multiobjective memetic metaheuristic for multiple sequence alignment. *IEEE Transactions on Evolutionary Computation* 20, 4, 499–514.

Rubio-Largo, Á., Vega-Rodríguez, M.A., González-Álvarez, D.L., 2016. Hybrid multiobjective artificial bee colony for multiple sequence alignment. *Applied Soft Computing* 41, 157–168.

Santander-Jiménez, S., Vega-Rodríguez, M.A., Sousa, L., 2018. Multiobjective frog-leaping optimization for the study of ancestral relationships in protein data. *IEEE Transactions on Evolutionary Computation* 22, 6, 879–893.

Santander-Jiménez, S., Vega-Rodríguez, M.A., Sousa, L., 2019. A multiobjective adaptive approach for the inference of evolutionary relationships in protein-based scenarios. *Information Sciences* 485, 281–300.

Santander-Jiménez, S., Vega-Rodríguez, M.A., Sousa, L., 2020. Inter-algorithm multiobjective cooperation for phylogenetic reconstruction on amino acid data. *IEEE Transactions on Cybernetics* 52, 3577–3591.

Santander-Jiménez, S., Vega-Rodríguez, M.A., Sousa, L., 2022. Exploiting multi-level parallel metaheuristics and heterogeneous computing to boost phylogenetics. *Future Generation Computer Systems* 127, 208–224.

Sar, E., Acharyya, S., 2014. Genetic algorithm variants in predicting protein structure. Conference on Communication and Signal Processing. IEEE, Piscataway, NJ, pp. 321–325.

Scalabrin, M.H., Parpinelli, R.S., Benítez, C.M., Lopes, H.S., 2014. Population-based harmony search using GPU applied to protein structure prediction. *International Journal of Computational Science and Engineering* 9, 1–2, 106–118.

Scornavacca, C., Delsuc, F., Galtier, N., 2020. *Phylogenetics in the Genomic Era*. https://hal.archives-ouvertes.fr/hal-02535070v3

Shukla, A.K., Singh, P., Vardhan, M., 2018. A hybrid gene selection method for microarray recognition. *Biocybernetics and Biomedical Engineering* 38, 4, 975–991.

Sievers, F., Wilm, A., Dineen, D., Gibson, T.J., Karplus, K., Li, W., Lopez, R., McWilliam, H., Remmert, M., Söding, J., Thompson, J.D., Higgins, D.G., 2011. Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Molecular Systems Biology* 7, 1, 539.

Siqueira, G., Brito, K.L., Dias, U., Dias, Z., 2020. Heuristics for reversal distance between genomes with duplicated genes. International Conference on Algorithms for Computational Biology. Springer, Berlin, pp. 29–40.

Soler-Dominguez, A., Juan, A.A., Kizys, R., 2017. A survey on financial applications of metaheuristics. *ACM Computing Surveys (CSUR)* 50, 1, 1–23.

Sperschneider, V., 2008. *Bioinformática: Paradigmas de resolución de problemas*. Springer Science & Business Media, Berlin.

Stillinger, F.H., Head-Gordon, T., Hirshfeld, C.L., 1993. Toy model for protein folding. *Physical Review E* 48, 2, 1469.

Storn, R., Price, K., 1997. Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* 11, 4, 341–359.

Swaminathan, V., Rajaram, G., Abhishek, V., Reddy, B.S., Kannan, K., 2019. A novel hypergraph-based genetic algorithm (HGGA) built on unimodular and anti-homomorphism properties for DNA sequencing by hybridization. *Interdisciplinary Sciences: Computational Life Sciences* 11, 397–411.

Tabataba, F.S., Mousavi, S.R., 2012. A hyper-heuristic for the longest common subsequence problem. *Computational Biology and Chemistry* 36, 42–54.

Talbi, E.G., 2009. *Metaheuristics: From Design to Implementation*, Vol. 74. John Wiley & Sons, Hoboken, NJ.

Talbi, E.G., 2013. *Hybrid Metaheuristics*, Vol. 166. Springer, Berlin.

Tanaka, S., 2012. A heuristic algorithm based on Lagrangian relaxation for the closest string problem. *Computers & Operations Research* 39, 3, 709–717.

Thompson, G.A., Schulz, A., 1999. Macromolecular trafficking in the phloem. *Trends in Plant Science* 4, 9, 354–360.

Thompson, J.D., Plewniak, F., Poch, O., 1999. BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs. *Bioinformatics* 15, 1, 87–88.

Thomsom, J., Higgins, D., Gibson, T., 1994. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice. *Nucleic Acids Research* 22, 4673–4680.

Tompa, M., Li, N., Bailey, T.L., Church, G.M., De Moor, B., Eskin, E., Favorov, A.V., Frith, M.C., Fu, Y., Kent, W.J., Makeev, V.J., Mironov, A.A., Stafford Noble, W., Pavesi, G., Pesole, G., Régnier, M., Simonis, N., Sinha, S., Thijs, G., van Helden, J., Vandenbogaert, M., Weng, Z., Workman, C., Ye, C., Zhu, Z., 2005. Assessing computational tools for the discovery of transcription factor binding sites. *Nature Biotechnology* 23, 1, 137–144.

Torrisi, M., Pollastri, G., Le, Q., 2020. Deep learning methods in protein structure prediction. *Computational and Structural Biotechnology Journal* 18, 1301–1310.

Ülker, E.D., 2016. Adaptation of harmony search algorithm for DNA fragment assembly problem. SAI Computing Conference. IEEE, Piscataway, NJ, pp. 135–138.

Vidal, P., Olivera, A.C., 2018. Solving the DNA fragment assembly problem with a parallel discrete firefly algorithm implemented on GPU. *Computer Science and Information Systems* 15, 2, 273–293.

Villalobos-Cid, M., Dorn, M., Ligabue-Braun, R., Inostroza-Ponta, M., 2019. A memetic algorithm based on an NSGA-II scheme for phylogenetic tree inference. In *Transactions on Evolutionary Computation*. IEEE, Piscataway, NJ, pp. 776–787.

Villalobos-Cid, M., Rivera, C., Kessi-Perez, E.I., Inostroza-Ponta, M., 2022. A multi-modal algorithm based on an NSGA-II scheme for phylogenetic tree inference. *Biosystems* 213, 104606.

Wallace, I.M., Orla, O., Higgins, D.G., 2004. Evaluation of iterative alignment algorithms for multiple alignment. *Bioinformatics* 21, 8, 1408–1414.

Wang, L., Jiang, T., 1994. On the complexity of multiple sequence alignment. *Journal of Computational Biology* 1, 4, 337–348.

Yadav, R.K., 2018. A novel bio-geography based approach for multiple sequence alignment. *IITM Journal of Management and IT* 9, 1, 58–66.

Yeung, K.Y., Bumgarner, R.E., Raftery, A.E., 2005. Bayesian model averaging: Development of an improved multi-class, gene selection and classification tool for microarray data. *Bioinformatics* 21, 10, 2394–2402.

Zambrano-Vega, C., Nebro, A.J., Aldana-Montes, J.F., 2016. Mo-phylogenetics: a phylogenetic inference software tool with multi-objective evolutionary metaheuristics. *Methods in Ecology and Evolution* 7, 7, 800–805.

Zare-Mirakabad, F., Ahrabian, H., Sadeghi, M., Hashemifar, S., Nowzari-Dalini, A., Goliaei, B., 2009. Genetic algorithm for dyad pattern finding in DNA sequences. *Genes & genetic systems* 84, 1, 81–93.

Zemali, E., Boukra, A., 2018a. CS-ABC: a cooperative system based on artificial bee colony to resolve the DNA fragment assembly problem. *International Journal of Data Mining and Bioinformatics* 21, 2, 145–168.

Zemali, E., Boukra, A., 2018b. EGSA: a new enhanced gravitational search algorithm to resolve multiple sequence alignment problem. *International Journal of Intelligent Engineering Informatics* 6, 1–2, 204–217.

Zhang, H., Schneider, T., Wheeler-Kingshott, C.A., Alexander, D.C., 2012. NODDI: practical in vivo neurite orientation dispersion and density imaging of the human brain. *NeuroImage* 61, 4, 1000–1016.

Zhang, Q., Li, H., 2007. MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *Transactions on Evolutionary Computation* 11, 6, 712–731.

Zhang, Q., Zhang, J., Zhong, Y., Ye, C., Min, X., 2019. Parallel MOEA based on consensus and membrane structure for inferring phylogenetic reconstruction. IEEE Access, pp. 6177–6189. https://doi.org/10.1109/ACCESS.2019.2959783

Zhang, X., Wang, T., Luo, H., Yang, J.Y., Deng, Y., Tang, J., Yang, M.Q., 2010. 3D protein structure prediction with genetic tabu search algorithm. *BMC Systems Biology* 4, 1, S6.

Zhou, C., Hou, C., Wei, X., Zhang, Q., 2014. Improved hybrid optimization algorithm for 3D protein structure prediction. *Journal of Molecular Modeling* 20, 7, 2289.

Zhu, D., Wu, Y., Wang, X., 2015a. A dynamic programming algorithm for a generalized LCS problem with multiple subsequence inclusion constraints. International Conference on Internet of Vehicles. Springer, Cham, pp. 439–446.

Zhu, H., He, Z., Jia, Y., 2015b. A novel approach to multiple sequence alignment using multiobjective evolutionary algorithm based on decomposition. *IEEE Journal of Biomedical and Health Informatics* 20, 2, 717–727.

Zhu, H., Pu, C., Lin, X., Gu, J., Zhang, S., Su, M., 2009. Protein structure prediction with EPSO in toy model. Conference on Intelligent Networks and Intelligent Systems. IEEE, Tianjin, China, pp. 673–676.

Zitzler, E., Künzli, S., 2004. Indicator-based selection in multiobjective search. Conference on Parallel Problem Solving from Nature. Springer, Berlin, pp. 832–842.

Zitzler, E., Laumanns, M., Thiele, L., 2001. SPEA2: improving the strength Pareto evolutionary algorithm. *TIK-report* 103, 1–21.