République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université A.MIRA-BEJAIA

Faculté des Sciences Exactes
Département Informatique

THÈSE

Présentée par

Ali DABBA

Pour l'obtention du grade de

DOCTEUR EN SCIENCES

Filière :Informatique

Option : Cloud Computing

Thème

Bio-Inspired Computation For Bio-Informatics Problems

Soutenue le : 03/01/2021

Devant le Jury composé de :

Nom et Prénom	Grade		
Pr. AMROUNKamal	M.C.A	Université de Bejaia	Président
Pr. TARI Abdelkamel	Professeur	Université de Bejaia	Rapporteur
Dr. MEFTALI Samy	M.C.HDR	Université de Lille 1	Examinateur
Pr. SEBAA Abderrazak	M.C.A	Université de Bejaia	Examinateur
Pr. KAZAR Okba	Professeur	Université de Biskra	Examinateur
Dr. BOUAMAMA Salim	M.C.A	Université de Sétif 1	Examinateur

Année Universitaire : 2019/2020

ABDERRAHMANE MIRA UNIVERSITY - BEJAIA
Faculty of Exact Sciences
Computer Science Department.

PhD of Science

Bio-Inspired Computation For Bio-Informatics Problems.

by Ali DABBA

Abstract

The field of bioinformatics opens up great opportunities to understand biological phenomena, which has attracted great interest from the scientific community in recent years. Consequently, there are many problems of bioinformatics, including multiple sequence alignment, protein structure prediction, construction of the phylogenetic tree and molecular docking, etc., which need the cooperation between biologists and computer scientists to be solved.

This work addresses two problems: multiple sequence alignment and gene selection using bio-inspired algorithms. Firstly, we developed a method to solve the multiple sequence alignment problem, called a multi-objective artificial fish swarm algorithm (MOAFS), using the behaviors of artificial fish swarm algorithms, Pareto-optimal set, and genetic operations. Secondly, we proposed an algorithm to solve the gene selection problem by using mutual information, moth flame optimization algorithm, and support vector machine with leave one out cross-validation (SVM-LOOCV). It called the Mutual Information Maximization-modified Moth Flame Algorithm (MIM-mMFA) that consists of two simple phases.

The thesis has processed a full test of the MOAFS on the BaliBASE 2.0 and BaliBASE 3.0 alignment benchmark datasets as well as the MIM-mMFA test on sixteen binary and multi-classes cancer gene expression datasets. Finally, we have given a deep insight into the performance of each algorithm. In addition, our proposed algorithms achieved competitive or better results than the well-established algorithms in the literature.

Keywords: Bio-informatics; Bio-inspired Algorithms ; Multiple Sequence Alignment ; Artificial Fish Swarm Algorithm ; Gene Selection Genes Expression ; Micro-array ; Cancer Classification ; Moth Flame Optimization Algorithm.

UNIVERSITÉ ABDERRAHMANE MIRA - BÉJAÏA
Faculté des sciences exactes
Département Informatique.

Doctorat en Sciences

Algorithmes Bio-inspirés Appliqués aux Problèmes Bio-Informatique

par Ali DABBA

Résume

Le domaine de la bio-informatique offre de grandes possibilités de comprendre les phénomènes biologiques, ce qui a suscité un grand intérêt de la part de la communauté scientifique ces dernières années. Par conséquent, il existe de nombreux problèmes de bio-informatique, y compris l'alignement de séquences multiples, la prédiction de la structure des protéines, la construction de l'arbre phylogénétique et l'amarrage moléculaire, etc. qui nécessitent la coopération entre biologistes et informaticiens pour être résolus.

Ce travail aborde deux problèmes: l'alignement de séquences multiples et la sélection de gènes à l'aide d'algorithmes bio-inspirés. Premièrement, nous avons développé une méthode pour résoudre le problème de l'alignement des séquences multiples, appelée algorithme d'essaim de poissons artificiels multi-objectifs (MOAFS), en utilisant les comportements des algorithmes d'essaim de poissons artificiels, l'ensemble Pareto-optimal, et les opérations génétiques.

Deuxièmement, nous avons proposé un algorithme pour résoudre le problème de sélection de gènes en utilisant l'information mutuelle, l'algorithme d'optimisation de flamme de papillon de nuit, et le Machine à vecteurs de support avec leave-one-out cross-validation (SVM-LOOCV). Il a appelé Mutual Information Maximization-modified Moth Flame Algorithm (MIM-mMFA) qui se compose en deux phases simples.

La thèse a traité un test complet du MOAFS sur les ensembles de données de référence d'alignement BaliBASE 2.0 et BaliBASE 3.0 ainsi que le test MIM-mMFA sur seize ensembles de données du cancer binaires et multi-classes. Enfin, nous avons donné un aperçu approfondi des performances de chaque algorithme. De plus, nos algorithmes proposés ont obtenu des résultats compétitifs ou meilleurs que les algorithmes bien établis dans la littérature.

Mots-clés: Bioinformatique ; Algorithmes bio-inspirés ; Alignement de séquences multiples ; Algorithme d'essaim de poissons artificiels ; Sélection des gènes ; Expression des gènes ; Puces à ADN ; Classification du cancer; Algorithme d'optimisation de la flamme papillon.

جـــامعـــة عبد الرحمـــان ميـــرة – بجاية
كليـــة العلـــوم الدقيـــقة
قســـم الإعـــلام الآلـــي

دكـتـــوراه علـــوم

تطبيق الخوارزميات المستوحاة من البيولوجيا على مشاكل المعلوماتية الحيوية

إعداد: دابـــة علـــي

ملخص

يوفر مجال المعلوماتية الحيوية فرصا كبيرة لفهم الظواهر البيولوجية ، التي جذبت اهتماما كبيرا من قبل المجتمع العلمي في السنوات الأخيرة. ومع ذلك ، هناك العديد من مشاكل المعلوماتية الحيوية ، بما في ذلك محاذاة التسلسل المتعدد ، والتنبؤ بهيكل البروتين ، وبناء الأشجار التطورية والالتحام الجزيئي ، الخ. و ليتم حلها يتطلب التعاون بين علماء الأحياء وعلماء الكمبيوتر .

يعالج هذا العمل مشكلتين هما: محاذاة التسلسل المتعدد وانتقاء الجينات باستخدام الخوارزميات المستوحاة من البيولوجيا. أولاً ، قمنا بتطوير طريقة لحل مشكلة محاذاة التسلسل المتعدد ، تسمى خوارزمية سرب الأسماك الاصطناعية متعددة الأغراض (MOAFS) ، وذلك باستخدام سلوكيات خوارزمية سرب الأسماك الاصطناعية ، مجموعة باريتو المثلى ، والعمليات الوراثية. ثانياً ، اقترحنا خوارزمية لحل مشكلة انتقاء الجينات باستخدام المعلومات المتبادلة ، وخوارزمية تحسين عثة اللهب (فراشة الليل) ، وتقنية التحقق من الصحة المتقاطع مع دعم جهاز المتجه (SVM-LOOCV). يطلق عليها تعظيم المعلومات المتبادلة و تعديل خوارزمية تحسين أداء عثة اللهب (MIM-mMFA) والتي تتكون من مرحلتين بسيطتين.

تناولت الأطروحة اختبارا كاملاً لـ MOAFS على مجموعات البيانات المرجعية للمحاذاة الخاصة بـ BaliBASE 2.0 و BaliBASE 3.0، كذلك اختبار MIM-mMFA على ستة عشر مجموعة من بيانات تعبير الجينات السرطانية ثنائية ومتعددة الطبقات. أخيرا ، قدمنا نظرة عميقة حول أداء كل خوارزمية. بالإضافة إلى ذلك ، حققت الخوارزميات المقترحة نتائج تنافسية أو أفضل من الخوارزميات الراسخة في هذا المجال.

الكلمات الرئيسية: المعلوماتية الحيوية ؛ الخوارزميات المستوحاة من البيولوجيا ؛ محاذاة التسلسل المتعدد ؛ خوارزمية سرب الأسماك الاصطناعية ؛ انتقاء الجينات ؛ تعبير الجينات ؛ رقائق الحمض النووي ؛ تصنيف السرطان ؛ خوارزمية تحسين عثة اللهب .

Acknowledgements

I would like to express my sincere gratitude to those who gave me assistance and support during my Ph.D. study.

My deepest gratitude goes to my supervisors Pr. Abdelkamel TARI has spent dedicated time and efforts to train my research skills, and provided constructive and challenging feedback to improve my research work, each in a unique way. This thesis would not have been possible without their encouragement, motivation, inspiration, and guidance.

I am grateful to Dr. Kamal AMROUN , Pr. Samy MEFTALI , Dr. Abderrazak SEBAA , Pr. Okba KAZAR , and Dr. Salim BOUAMAMA for agreeing to be members of the exam committee for this thesis, which took them time to read and review. Their suggestions will be taken into account and will surely significantly improve the final version.

I am grateful for Research center in Computer Science, Signal and Automatic Control of Lille (CRIStAL) laboratory for their greeted and support me over the past year.

Last but not least, I wish to thank my family for their love, encouragement and support.

Contents

List of Figures

List of Tables

Chapter 1

General Introduction

1.1 Introduction

This chapter introduces this thesis. It begins with the answer to the main question about what is Bioinformatics and why?. In addition, it presents the motivations, the research goals, then the major contributions and the thesis organization. Also, it presents the published and unpublished manuscripts that have been written during the thesis, which is presented in this chapter. Lastly, this chapter also presents the background that introduces the Bioinformatics field to a non-bioinformatician reader.

1.2 What is Bioinformatics? And Why?

Term Bioinformatics was invented by Paulien Hogeweg and Ben Hesper in 1970 as "the study of informatics processes in biotic systems" (Hogeweg, 2011).

Bioinformatics is the discipline dedicated to computational processing, storage, retrieving, analysis, and representation of biological data, usually at the molecular level. It is an interdisciplinary and combines computer science, statistics, mathematics, and engineering to

analyze and interpret biological data (Baldi, Brunak, and Bach, 2001). Therefore, Bioinformatics is a recent and fast-moving field, it is used to develop methods and software tools for understanding and knowledge of biological phenomena and processes.

The ultimate objective of this field is to develop new knowledge about the life sciences and to create a global perspective from which the unifying principles of biology can be derived (Altman, 2001). Three major goals of bioinformatics can be put forward as:

- ☞ In its simplest form, bioinformatics organizes data in such a manner as to facilitate researchers' access to existing information and suggests new entries as they are produced.
- ☞ To develop algorithms and mathematical models to explore the relationships among the members and assist in the analysis of a large biological dataset.
- ☞ Using these tools to analyze the data and accurately interpret the results in a biologically meaningful way.

In the early days of the "Genomic Revolution", the field of Bioinformatics was interested in developing and maintaining a database to store biological information, such as nucleotide sequences and amino acids. The development of this type of database requires not only design problems, but also the development of complex interfaces through which researchers can access existing data as well as provide new or revised data. Eventually, though, all this information needs to be aggregated to form a complete picture of normal cellular activities investigators can study, and how these activities evolve at different stages of the disease. As a result, the Bioinformatics field has evolved, and the most urgent task now is to analyze and interpret different types of data, for example, nucleotide and amino acid sequences, protein domains, and protein structures.

Bioinformatics has demonstrated to possess great potential for the early identification of diseases, identifying treatment and helping to make human life better. With the inspiration and knowledge of computer science, areas such as medicine, health care, and gene technology can develop from treating individual patients to treating the entire population.

All Bioinformatics problems have great importance in molecular biology, but they have another thing in common. Indeed, each of these problems is highly complex, which means that it is often impossible to solve them in an exact way. Each of these problems has been demonstrated to be NP-Complete. It is possible to calculate solutions for small instances using exact algorithms, however, other methods must be considered for larger instances. In fact, the high complexity of the exact algorithms often makes it impossible to use them with large biological data. Given the importance of the problems, it is necessary to be able to obtain good quality results for biologists to use them. Therefore, the Bioinformatics objective is to combine the skills available in each scientific field to help improve decision methods. For example, for the multiple sequence alignment problem, computer and mathematical solutions have been provided.

1.3 Motivation

Bioinformatics is the interaction of computer science and molecular biology, it can be considered as a branch of biology that uses computers to help answer biological questions. Through many years, the Bioinformatics field has been of great importance to the scientific community because it opens very rich perspectives for understanding biological phenomena.

There are several problems in this area, including the multiple sequence alignment, protein structure modeling, reconstruction of phylogenetic trees (phylogeny), and gene selection, etc. These problems require collaboration between biologists and computer scientists,

as the problems that need to be addressed often cause significant computational difficulties because they contain big data that requires huge memory space, which is what has generated an explosion of data. With this explosion of data, it is becoming increasingly difficult, if not impossible, to obtain optimal solutions for data-driven problems using standard algorithms. Therefore, intelligent approaches are needed to determine optimal solutions.

Consequently, our work focuses on Bioinformatics problems that are expressed as optimization problems. Besides, many algorithms have been applied to optimization problems, but none of them provide the optimal solution in all cases. However, Bio-inspired algorithms have proven to be highly effective in solving optimization problems.

In this thesis, we addressed two problems in the field of Bioinformatics through to use bio-inspired algorithms are a multiple sequence alignment and gene selection for microarray data classification.

1.3.1 An Analysis of Multiple Sequence Alignment

Multiple sequence alignment (MSA) represents a basic task for biological sequence analysis in molecular biology, computational biology, and Bioinformatics. In addition, the MSA is compared in order to perform phylogenetic tree reconstruction, protein secondary or tertiary structure prediction, and protein function prediction analysis, etc (Kemena and Notredame, 2009). The majority of these applications are based on the analysis of protein sequences and may be translated into DNA sequences as part of phylogenetic analysis.

The accurate alignment can have a good biological meaning, showing the relationships and homology between different sequences, and can also give useful information, which can be used to identify new members of protein families. The accuracy of the MSA is crucial because many Bioinformatics techniques and procedures depend on the results of MSA (Kemena and Notredame, 2009).

As the MSA problem has been studied for many decades, these studies have shown considerable progress in improving the accuracy, speed, and quality of multiple alignment tools. However, the most accurate alignment methods are unable to analyze the very large datasets. In addition, existing methods still suffer from the problem of high error rates and/or high computational cost.

However, to find the optimal multiple sequence alignment, it is necessary to design an efficient exploration approach that can explore a huge number of possible multiple sequence alignments. As well as, it is also necessary to use a powerful evaluation method to assess the suitability of these multiple sequence alignments.

1.3.2 An Analysis of Gene Selection

Gene selection, which is a branch of feature selection, is a difficult problem, particularly when the number of available genes (features) is large (Amaldi and Kann, 1998). In fact, for gene selection, there is a small number of samples available while each sample is described by a very large number of genes. Therefore, the task is challenging on many levels, including the complexity, the large search space, the dependence on the evaluation classifier, classifier interactions between genes (features), and so on.

In order to process these data, it is necessary to reduce the number of genes to propose a relevant gene subset and to construct a classifier that predicts the type of tumor that characterizes a cell sample. The number of genes is often reduced by a pre-processing step dedicated to removing noise and redundant genes (features).

In feature selection, the search space size increases exponentially with respect to the number of available features in the dataset (Guyon and Elisseeff, 2003). In practice, it cannot sweep a complete search because it scans the full area of the solution, which usually often

extremely time-consuming. Theoretically, the problem of optimal feature selection has been shown to be NP-hard (Cover and Van Campenhout, 1977).

Therefore, several techniques or methods may help in detecting diseases and cancer. Creating an effective method for extracting disease information is one of the major challenges in the classification of gene expression data as long as there is a massive amount of redundant data and noise.

1.3.3 Why Bio-inspired Algorithm?

The behavior of some insects or groups animals in nature like fish, birds, and bees, etc., has allowed attracting the attention of computer science researchers to the intelligent behavior of biological swarms that using only local knowledge and the interaction of individuals in such environments to solve real-world problems by simulating such biological behaviors, which are known as bio-inspired algorithms. Therefore, bio-inspired algorithms or swarm intelligence are considered as a subfield of artificial intelligence. Generally, the motivation for basing bio-inspired algorithms on nature is that the natural processes concerned are known to produce desirable results, such as finding an optimal value of some feature. Despite their effectiveness, methods modeled on nature have often been treated with suspicion.

The optimization process plays a very important role in solving complex mathematical problems, which is to find the optimal solution (s) to a given problem. However, the solutions are far from perfect, even for the fastest computers currently available, there are some problems that are often challenging due to their sheer size. Using traditional methods, it is difficult or impossible to make a thorough examination of all probabilities that would typically take longer than the universe has existed.

In this thesis, we use bio-inspired algorithms, which are considered motivational approaches to utilizing alternative models that are computationally efficient in the deterministic approach. Bioinformatics is one of the important educational fields of technology. In recent years, there are several bio-inspired algorithms developed and applied to various Bioinformatics problems. Bio-inspired algorithms are effective and efficient global search techniques and they are suitable algorithms to address Bioinformatics issues. We summarize the motivations to choose this kind of algorithms in the following points:

- ☞ The nature of bio-inspired algorithm depends on (i) based on simple concepts to be easy for implementation; (ii) no need to use data inclination; (iii) can find an optimal neighborhood solution; (iv) finally, it can be applied in different application fields.

- ☞ Generally, Bioinformatics problems have a huge search space, so we have to use a global search technique to avoid the problem of falling into the local optima. Bio-inspired algorithms are known for their ability to perform global research, which can effectively search huge spaces to find optimal or near-optimal solutions.

- ☞ Swarm intelligence is defined as the designing of intelligent algorithms that are stimulated by the behavior of different animal societies (Bonabeau et al., 1999). Consequently, we can be exploiting the advantages of swarm intelligence approaches such as adaptation, scalability, speed, autonomy, parallelism and fault tolerance to solve complex combinatorial optimization problems.

- ☞ Also, swarm intelligence has some important characteristics such as self-organization and working division. As animals and birds biological behavior, each member of the group is responsible for a specific task individually, and sometimes they work together to achieve a given task.

Ultimately, we can conclude that the nature-inspired algorithms such as bio-inspired algorithms are the next era power optimization algorithms that will play a key role in next-generation computing.

1.4 Goals

The major objective of this thesis is to investigate/improve the capability of bio-inspired algorithms to solve the Bioinformatics problem. In this research, we have been addressed two Bioinformatics problems, the first is the MSA problem that was solved using the artificial fish swarm (AFS) algorithm as a new iterative approach. The second is the gene selection in classification problems that have been resolved by applying the moth flame optimization (MFO) algorithm to reduce the number of genes and achieve similar or even better classification performance than using all the original genes.

Specifically, AFS and MFO algorithms have never been applied in the way explained in this thesis. To achieve these primary objectives and guide this research, we have been created a set of those objectives, which can be summarized as follows:

1.4.1 For MSA Problem

- ☞ To provide a brief overview of the concept of biological sequence alignment, with a focus on the MSA.
- ☞ Examine the effectiveness and efficiency of blending between the AFS and genetic operations in MSA, as they are adapted and create new operations such as center operation.
- ☞ How to maintain a trade-off between the exploration and the exploitation of the MSA problem's research space.
- ☞ In the original sequences, the gaps should be included in the way that they guarantee (i) increases the number of matching characters to the maximum and (ii) the gaps number inserted is minimized. Bearing in mind that the two requirements are conflicting, therefore, we have adopted on multi-objective optimization using the technique of Pareto-optimal set to be found an optimal or approximate alignment solution.
- ☞ In order to know MOAFS performance in relation to several factors, such as alignment accuracy, we compared MOAFS solutions with other MSA programs solutions that will be applied to the two benchmark databases: BAliBASE 2.0 and BAliBASE 3.0.

1.4.2 For Gene Selection Problem

- ☞ A short survey about the concept of feature selection and its process, with an emphasis on gene selection.
- ☞ Bio-inspired algorithms are particularly suitable for gene selection problem. The MFO has been successful for many optimization problems, but has never applied to gene selection.
- ☞ Gene selection has thousands of genes and dozens of samples that do not enable a good generalization ability for a classifier because the training needs lots of samples. Furthermore, data is generally sparse in high dimensional spaces and this leads to overfitting as not enough samples are often available. Consequently, How to explore the search space well using a new movement of the moth in the MFO algorithm.

- ☞ Microarray data is a hard challenge for researchers due to the high number of genes (features) but the small sample size. For this purpose, the ability of algorithms to select relevant genes and ignore non-relevant genes without allowing noise or redundancy hindering this process is assessed. The MFO algorithm is expected to select a small number of genes and achieve similar or better classification performance than all genes using a pre-filter procedure and a new fitness function.

- ☞ In order to test the performance of the MIM-mMFA proposed, such as classification accuracy, we compared the results MIM-mMFA (like the number of selected genes and classification accuracy) with other gene selection algorithms results that will be applied to binary class and multi-class gene expression microarray datasets.

1.5 Major Contributions

Biologically inspired computing has been given importance for its immense parallelism and simplicity in computation. For this reason, several bio-inspired algorithms have been invented and are used to address many complex problems in the real world. In this thesis, we focus on bio-inspired algorithms that have been solved certain Bioinformatics problems and have been applied for multiple sequence alignment and gene selection problems. Therefore, this thesis makes the following two major contributions:

The first proposed algorithm, which aims to address the multiple sequence alignment problem, presents a multi-objective artificial fish swarm algorithm (MOAFS) to solve multiple sequence alignment. The MOAFS uses, the behaviors of artificial fish swarm algorithms such as cooperation, decentralization, and parallelism to ensure a good trade-off between the exploration and the exploitation of the search space of the MSA problem.

To preserve the quality and consistency of alignment, two fitness functions have been used simultaneously by the MOAFS algorithm: (i) Weighted Sum of Pairs to determine horizontally similar regions and (ii) Similarity function to determine vertically similar regions between the sequences of an alignment.

After exploring space search, the Pareto-optimal set is obtained by the MOAFS, which performs the optimal multiple sequence alignments for both fitness functions. The performance of the MOAFS algorithm has been proved by comparing our algorithm with different progressive alignment methods, and other alignment methods based on evolutionary algorithms with single-objective and multi-objective. The experiment results conducted on BAliBASE 2.0 and BAliBASE 3.0 benchmarks confirm that the MOAFS algorithm provides a greater accurate statistical significance in terms of SP or CS scores.

The second proposed algorithm, which aims to address the gene selection problem, proposes a new extension of the MFO algorithm called the modified Moth Flame Algorithm (mMFA), the mMFA is combined with Mutual Information Maximization (MIM) to solve gene selection in microarray data classification.

Our approach Called Mutual Information Maximization - modified Moth Flame Algorithm (MIM-mMFA), the MIM based pre-filtering technique is used to measure the relevance and the redundancy of the genes, and the mMFA is used to evolve gene subsets and evaluated by the fitness function, which uses a Support Vector Machine (SVM) with Leave One Out Cross Validation (LOOCV) classifier and the number of selected genes.

In order to test the performance of the proposed MIM-mMFA algorithm, we compared the MIM-mMFA algorithm with other recently published algorithms in the literature. The experiment results which have been conducted on sixteen benchmark datasets either binary-class phenotypes or multi-class phenotypes, confirm that MIM-mMFA algorithm provides a greater classification accuracy.

The studies conducted during this work have produced papers published in Information Systems and Operational Research (INFOR) journals (Dabba, Tari, and Zouache, 2019). Moreover, there are still papers under review for journals and conferences.

1.6 Thesis Structure

The list below shows the organization of the chapters that make up this thesis. In addition, a brief description of the topics dealt with in each chapter is given.

Chapter 1 shows the motivation, objectives, contributions, and structure of this thesis. In addition, a list of the published and unpublished manuscripts that have been written during the course of the thesis is presented in this chapter. Moreover, the rest of this chapter introduces some basic concepts in the Bioinformatics field.

Chapter 2 covers the necessary theoretical background of the thesis relating to bio-inspired algorithms, and the two problems of multiple sequence alignment and gene selection. This chapter introduces the fundamentals of bio-inspired algorithms and a brief definition of the main algorithms that have been used in this thesis. It then focuses on a detailed introduction about the multiple sequence alignment (definition, evaluation, objective function, and classification of its algorithms). In addition, it reviews typical related work in feature selection (definition, feature selection process, and its conventional methods). Finally, it reviews a brief for multi-objective optimization.

Chapter 3 introduces a new MOAFS algorithm to solve the multiple sequence alignment problem, which updates and adapts genetic operations at the AFS algorithm, in addition, to adapt new techniques such as representation, initialization, Pareto-optimal set, and two fitness functions. It then examines the MOAFS performance against progressive and iterative algorithms that exist in the literature. Finally, the results are then presented and analyzed.

Chapter 4 is dedicated to the description of a novel MIM-mMFA algorithm for gene selection, which aims to maximize the classification accuracy and minimize the number of genes. First, it gives an introduction to MIM pre-selection (preparation and normalization) and the large section of this chapter is devoted to gives a detail description of mMFA. Finally, The performance of the MIM-mMFA algorithm is examined on microarray datasets and the summary of an extensive experimental evaluation is given.

Chapter 5 summaries the overall findings of this thesis, and provides insights into our future research directions that emanate from it.

1.7 Basic Concepts in Molecular Biology

1.7.1 Cell

The cell is the origin of biological creation. Thus, all living organisms on Earth are made up of cells. The cell is the structural and functional unit of all living organisms and is sometimes called the "building block of life". It contains the genetic information of the individual and is able to live independently when they are in a favorable environment (Kaiser et al., 2007).

There are two types of cells, eukaryotic, which contain a nucleus, can be unicellular or multicellular, heterotroph or autotroph, and prokaryotic, which do not, are heterotrophic single-celled organisms.

1.7.2 DNA

Deoxyribonucleic acid (DNA) is a complete set of instructions for manufacturing an organism. It includes the master blueprint for all cellular structures and activities during the lifetime of the cell or organism. Hence, examining the DNA of organisms is a fundamental issue

in biological and clinical research. From a chemical point of view, the basic components of DNA are nucleotides. Each nucleotide contains a phosphate group, a sugar group and a nitrogen base. The four types of nitrogen bases are adenine (A), thymine (T), guanine (G) and cytosine (C). Adenine and guanine are double-ring molecules known as purine, while other bases are single-ring molecules known as pyrimidine (Vuong et al., 2008).

Moreover, DNA is a linear, double-helix structure, and is composed of two intertwined chains made up of hydrogen bonds between pyrimidine and purine bases (nucleotides). These base pairs A-T and C-G, when are piled over one another via hydrophobic interactions will constitute what is known as a "Chromosome".

Therefore, an essential property of DNA is that it can replicate, or make copies of itself. Each strand of DNA in the double helix can serve as a model to duplicate the base sequence (Watson, Crick, et al., 1953). This is essential when cells divide because each new cell must have an exact copy of the DNA present in the old cell.

1.7.3 RNA

Unlike DNA, Ribonucleic acid (RNA) molecule consists of a single-stranded sequence of nucleotide units. RNA is manufactured utilizing one of the DNA strands as a model and except that it replaces thymine with uracil (U) in its chemical structure. The main function of RNA is to copy DNA information and take it out of the nucleus for use wherever it is needed. There are three main types of RNA:

- ☞ messenger RNA (mRNA) is a template for protein synthesis.
- ☞ transfer RNA (tRNA) is utilized as an adapter molecule between the mRNA and the amino acids in the process of protein-synthesizing.
- ☞ ribosomal RNA (rRNA) is a structural component of a large complex of proteins and RNA known as the ribosome. The ribosome is in charge for binding to messenger RNA and directing the protein-synthesis.

Mostly, RNA is used to transmit information from the DNA into proteins.

1.7.4 Proteins

Proteins play a vital role in almost all biological processes such as the building of cells. Proteins (also known as polypeptides) are large organic molecules made up of many amino acids linked by peptide bonds arranged in a linear polypeptide chain and folded into a spherical form (Setubal, Meidanis, and Setubal-Meidanis, 1997).

An amino acid is an organic compound that consists a carboxyl (-COOH) functional groups and an amine (-NH2), as well as a side chain (designated as R) that is specific to each amino acid (Lehninger et al., 2005). Amino acids are essential components to build blocks of polypeptides and proteins (see Figure 1.1). There are about 20 different amino acids (see Table 1.1), each one is characterized by its own chemical properties.

1.7.5 Genes and Gene Expression

Each DNA molecule contains many genes. A gene is the physical, fundamental and functional unit of heredity. Genes are parts of DNA usually located on chromosomes that contain instructions for protein production. Genes make up the genetic codes, or sequence of nucleotide bases in nucleic acids, each gene contains the information required to build specific proteins needed in an organism (Myers et al., 2007).

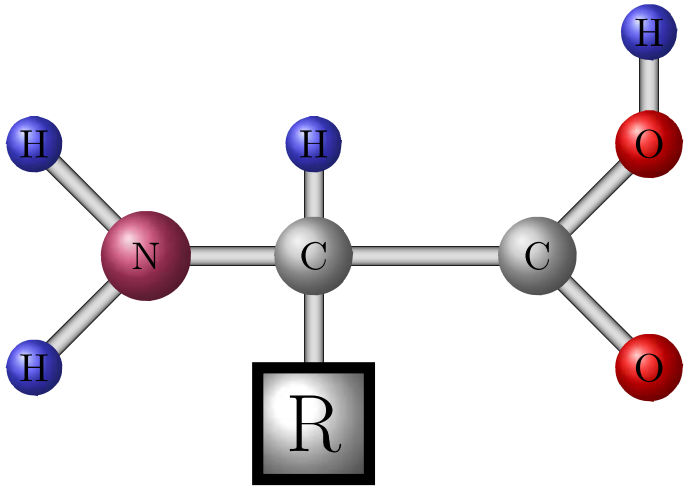


FIGURE 1.1: The general structure of an amino acid.

Gene expression is known in biology by the name of the central doctrine of molecular biology . Gene expression is the process by which the hereditary information of a gene, the sequence of DNA base pairs, is transformed into a functional gene product, such as a protein or RNA (Crick, 1958; Crick, 1970). In other words, gene expression describes how proteins are produced from DNA. Therefore, the basic idea is that DNA is transcribed into RNA, which is then translated into proteins (DNA $\rightarrow$ RNA $\rightarrow$ protein). This synthesis involves a set of biochemical processes that constitute the central dogma of molecular biology. The journey from gene to protein divided into two main steps illustrated in Figure 1.2

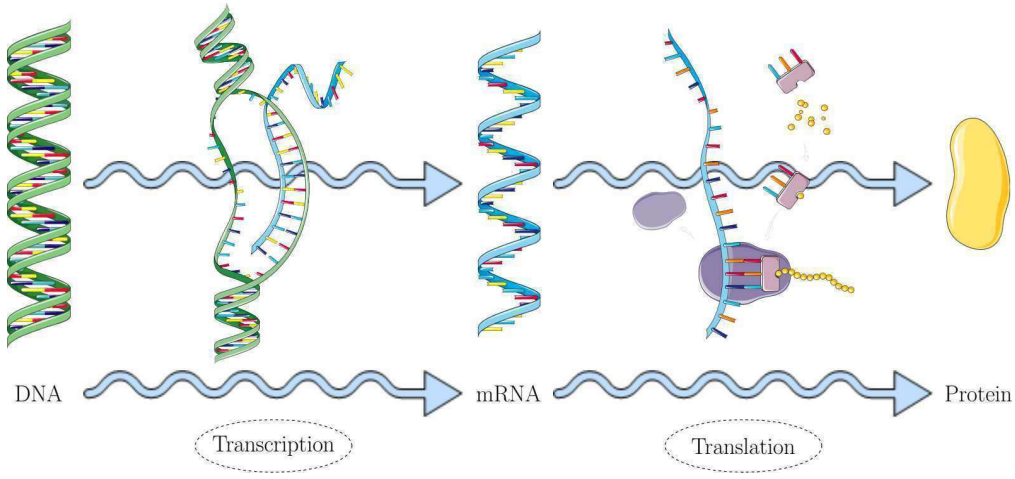


FIGURE 1.2: Central dogma of molecular biology: DNA $\rightarrow$ RNA $\rightarrow$ protein

☞ **Transcription –** is the first part of the central dogma of molecular biology: DNA $\rightarrow$ RNA. It is the transfer of genetic instructions in DNA to messenger RNA (mRNA). During transcription, the DNA is copied into the so-called messenger RNA (mRNA) by RNA polymerase, which occurs place in the cell nucleus.

TABLE 1.1: Amino acids with their abbreviations and codons.

Amino Acid	**Abbreviation 3-Lettres**	**Abbreviation 1-Lettre**	**Codon(s)**
Alanine	Ala	A	GCA, GCC, GCG, GCT
Arginine	Arg	R	CGA, CGC, CGG, CGT, AGA, AGG
Aspartic acid	Asp	D	GAC, GAT
Asparagine	Asn	N	AAC, AAT
Cysteine	Cys	C	TGC, TGT
Glutamic acid	Glu	E	GAA, GAG
Glutamine	Gln	Q	CAA, CAG
Glycine	Gly	G	GGA, GGC, GGG, GGT
Histidine	His	H	CAC, CAT
Isoleucine	Ile	I	ATA, ATC, ATT
Leucine	Leu	L	CTA, CTC, CTG, CTT, TTA, TTG
Lysine	Lys	K	AAA, AAG
Methionine (Start)	Met	M	ATG
Phenylalanine	Phe	F	TTC, TTT
Proline	Pro	P	CCA, CCC, CCG, CCT
Serine	Ser	S	TCA, TCC, TCG, TCT, AGC, AGT
Threonine	Thr	T	ACT, ACC, ACG, ACT
Tryptophan	Trp	W	TGG
Tyrosine	Tyr	Y	TAC, TAT
Valine	Val	V	GTA, GTC, GTG, GTT
STOP	-	-	TAG, TAA, TGA

☞ **Translation –** is the last part of the central dogma of molecular biology: RNA $\rightarrow$ Protein. It reads the genetic code in mRNA and makes a protein. Therefore, the mRNA formed in transcription out of the nucleus is translated by ribosomes (the cell's protein synthesis factory) into the sequence of amino acids outside the cell nucleus in the cytoplasm, with the assistance of transfer RNA (tRNA).

In this process, each three-base stretch of mRNA (triplet) is known as a codon, it always codes for an amino acid. Combinatorially, there are $(cardA)^3 = 64$ possible codons. We now know that the code is redundant, i.e. the same amino acid can be coded by more than one codon. Of these 64 codons, 61 represent amino acids and three are stop signals to stop the translation step (stop signals are TAA, TGA and TAG codons). The beginning of the coding region is regularly indicated by codons that also encode amino acids, like ATG for methionine (see Table 1.1 for more details).

This brief description of gene expression or the central dogma of molecular biology as a gene transcribed to RNA, RNA translated to protein is by no means complete and its details are always the subject of constant research.

1.8 Conclusion

By solving the Bioinformatics problem, biologists are helping to answer basic research questions, for example, a gene is suspected to encode whether an organism is immune to a specific virus and biologists comparing the genes of different organisms using a multiple sequence alignment to prove or disprove such a conjecture.

In this chapter, we have reviewed the motivations, the research goals, then outline the major contributions and the thesis structure. Ultimately, it presents some basic Concepts in Molecular Biology.

Chapter 2

Bio-inspired Algorithms In Bioinformatics Problems

2.1 Introduction

In this chapter, we aim at delivering some basic mathematical and methodological backgrounds on three major concepts present in this thesis, namely bio-inspired algorithm, multiple sequence alignment, and feature or gene selection.

In recent years, the field of bio-inspired computing represents the prominence in different studies of computer science, mathematics, and biology. All these bio-inspired algorithms try to replicate the way biological organisms and sub-organism entities (such as neurons and bacteria) operate to achieve a high level of efficiency, even if sometimes the actual optimal solution is not obtained. The bio-inspired algorithms distinguished by group collective intelligence, which is usually greater than the sum of individual intelligence. In this section, we have mentioned, but not limited to, three algorithms that we have used in our works are genetic algorithm, artificial fish swarm algorithm and moth flam optimization algorithm.

In addition, we going to focus on two Bioinformatics problems: the first is the alignment of multiple sequences, which is considered to be a fundamental problem in the Bioinformatics field. Therefore, we present a brief definition of MAS and concentrate on the measurements that determine the quality of the alignment, and objective functions that are based on quantifying the distance between sequences. Finally, we present a classification of MSA

algorithms. The second problem is gene or feature selection that consists in extracting the whole dataset in order to identify the relevant variables, with respect to the problem at hand. Feature selection can be achieved either through genes (features) selection subsets, i.e., by searching the best group of genes (features) among those available.

This chapter is organized as follows. Section 2.2 provides a general introduction to bio-inspired algorithms and a concept for some of its algorithms. Section 2.3 concerns with multiple sequence alignment and the classification for MSA algorithms. Section 2.4 involves with gene or feature selection and their methods. Section 2.5 provides a multi-objective optimization problem and introduces the concept of Pareto-optimal set. Finally, in this Section 2.6, we provide a conclusion of this chapter.

2.2 Bio-Inspired Algorithms

Nature is a great and immense source of inspiration to solve difficult and complex problems in computer science. Nowadays, most new algorithms are inspired by nature, called Nature-inspired algorithms. The real beauty of nature-inspired algorithms lies in the fact that it receives its sole inspiration from nature. By far the majority of the nature-inspired algorithms are based on some successful characteristics of the biological system.

Therefore, most of the algorithms inspired by nature are bio-inspired, for short biologically-inspired. Hence, the bio-inspired algorithms are a subset of the nature-inspired algorithms.

bio-inspired computing represents the umbrella of various studies of computer science, biology and mathematics in the last years. Basically, it is very closely linked to artificial intelligence and can be associated with machine learning, as well. Bio-inspired computing concentrates on the designs and developments of computer algorithms and models based on living phenomena and biological mechanisms. Hence, bio-inspired computing optimization algorithms are an emerging approach based on the principles and inspiration of nature's biological evolution to develop powerful new competing technologies. In the following, we present only the bio-inspired algorithms used in the design and development of our works.

2.2.1 Genetic Algorithm

Genetic algorithms (GAs) are optimization methods that are inspired by Charles Darwin's theory of natural evolution "survival of the fittest", proposed by Holland in 1975 (Holland, 1973). They are classic evolutionary algorithms based on randomness, which are probably the first algorithmic models developed to simulate genetic systems (Goldberg and Holland, 1988). The principal constituents of natural evolution (called genetic operators) that are part of a GA are inheriting, mutation, crossover, and selection. The process of optimization begins by initializing a population of solution (chromosome generally encoded by a standard representation in the form of a bit vector) that is randomly initialized or created by applying some heuristic. Genetic operators are applied at predetermined rates.

The genetic algorithm modifies a population of individual solutions several times. At each step, the GA randomly selects individuals from the current population to be parents and uses them to generate the children. The new individuals are added to the population, and the replacement procedure selects the solutions that will survive to the next generation to maintain the prescribed population size. Over successive generations, the population "evolves" toward an optimal solution. Therefore, for each chromosome, we evaluate fitness using an appropriate fitness function and suitable for the problem. According to this basis, the best chromosomes are selected into the population. In contrast to analytical optimization methods such as gradient based optimization, GAs are less likely to be trapped in local optima. However, they tend in many cases to be computationally expensive.

The genetic algorithm uses five phases:

1) **Initial population:** Create a random population of potential solutions composing of individuals. The population of individuals is preserved within search space. Each individual is a solution in the research space for a given problem and is coded as a vector of finite length (chromosome-like) of components.

2) **Fitness Score –** is the function that needs to be optimized. This is the value that indicates the ability of an individual to "compete". As the Genetic Algorithm continues through its cycle, the individual having optimal fitness score (or near-optimal) is sought.

3) **Selection Operator –** is the process of selecting two parents that will be used for recombination, which will result in two newly created genotypes. The idea is to give preference to individuals with high fitness scores and to determine who should reproduce.

4) **Crossover Operator –** is a genetic operator that combines (mates) two chromosomes (parents) to produce a new chromosome (offspring) and crossover sites are chosen randomly. This represents mating between individuals. The idea behind crossover is that the new chromosome may be better than both of the parents if it has the best characteristics from each of the parents.

5) **Mutation Operator –** The key idea is to insert random genes in offspring to maintain the diversity in population to avoid premature convergence. It is applied random changes to individual parents to form children.

The basic logic of GA is illustrated by Algorithm 2.1.

Algorithm 2.1 Genetic Algorithm

Initialize population with random candidate solutions
Evaluate each candidate

1: **repeat**
2: Select parents from population;
3: Crossover (Recombine) pairs of parents;
4: Perform the Mutation on new children;
5: Evaluate new candidates;
6: Select individuals for the next generation;
7: **until** (Termination condition is satisfied)

2.2.2 Artificial Fish Swarm Algorithm

Artificial fish swarm algorithm (AFSA) (Li, 2002) is a bio-inspired meta-heuristic optimization method introduced by Xiao Lei in 2002. This algorithm simulates the collective behavior of fish in the search for the most food regions. Each fish carries a candidate solution in the research space, the concentration of food in a region represents the value of the objective function of an artificial fish. To use the AFSA algorithm, many parameters should be defined such as the visual and the step.

☞ **Visual –** the parameter of $Visual_i$ of Artificial Fish (AF_i) is represented by the radius of a circle of the artificial fish candidate. This parameter is very important in the determination of nearest neighbors for AF_i and value of $Visual_i \in]0,1[$.

☞ **Step –** represents a jump of movement of artificial fish from the current position to a new position. Generally, an AF_i located in a region less concentration of food (value of the objective function) moves to another AF_j located in a food concentration region is richer. But the attractiveness between AF_i and AF_j decreases when the distance between the two fish increases. Moreover, attractiveness is relatively related by the environmental factor where the set of artificial fish finds. More obstacles are found between the artificial fish swarm while less attractive among fish swarms.

Figure 2.1 shows how AF achieves external perception by its vision. X is the current state of an AF, Visual is the visual distance, and X_v is the visual position at some moment. If the state at the visual position is better than the current one, it advances one step in this direction, and arrives the X_{next} state; otherwise, continues an inspection tour in the vision.

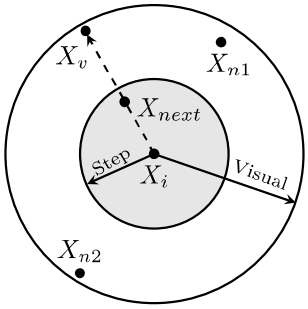


FIGURE 2.1: Vision concepts of Artificial Fish

Let $X = (x_1, x_2, ..., x_n)$ and $X_v = (x_1^v, x_2^v, ..., x_n^v)$, then this process can be expressed as follows:

$$x_v = x_i + Visual.rand(),\ i \in (0, n] \tag{2.1}$$

$$X_{next} = X + \frac{X_v - X}{||X_v - X||}.Step.rand() \tag{2.2}$$

where $rand()$ produces random numbers between $zero$ and 1, $Step$ is the step length, and x_i is the optimizing variable, n represents the number of variables.

The travel of the solution search space by this algorithm is made by the movements of fish bases used in the process of discovery of the most food regions that are: Praying, Swarming, and Following, Moving, Leaping and Evaluating.

1) **Prey –** fish perceive the concentration of food in water to determine movement through vision or sense, and then select the trend.

   Let X_i be the AF current state and select a state X_j randomly in its visual distance, Y is the food concentration (objective function value), the greater $Visual$ is, the more easily the AF finds the global extreme value and converges.

$$X_i = X_j + Visual.rand() \tag{2.3}$$

   If $Y_i < Y_j$ in the maximization problem, it goes forward a step in this direction;

$$X_i^{(t+1)} = X_i^{(t)} + \frac{X_j - X_i^{(t)}}{\left|\left|X_j - X_i^{(t)}\right|\right|}.Step.rand() \tag{2.4}$$

Otherwise, select a state X_j randomly again and judge whether it satisfies the forward condition. If it cannot satisfy after try_{number} times, it moves randomly one step. When the try_{number} is small in AF_Prey, the AF can randomly swim, which makes it flee from the local extreme value field.

$$X_i^{(t+1)} = X_i^{(t)} + Visual.rand() \tag{2.5}$$

2) **Swarm** – fish come together in groups naturally in the moving process, which is a kind of lifestyle in order to guarantee the existence of the colony and avoid dangers.

Let X_i be the AF current state, X_c be the center position and n_f be the number of its companions in the current neighborhood $(d_{ij} < Visual)$, n is total fish number and δ is the crowd factor $(0 < \delta < 1)$. If $Y_c > Y_i$ and $\frac{n_f}{n} < \delta$, which means that the companion center has more food (higher fitness function value) and is not very crowded, it goes forward a step to the companion center;

$$X_i^{(t+1)} = X_i^{(t)} + \frac{X_c - X_i^{(t)}}{\left|\left|X_c - X_i^{(t)}\right|\right|}.Step.rand() \tag{2.6}$$

Otherwise, executes the preying behavior. The crowd factor limits the scale of swarms, and more AF only cluster at the best area, which ensures that AF move to optimum in a wide field.

3) **Follow** – in the moving process of the swarm, when several or a single fish find food, the neighborhood partners will be trained and reach the food quickly.

Let X_i be the AF current state, and it explores the companion X_j in the neighborhood $(d_{ij} < Visual)$, which has the greatest Y_j. If $Y_j > Y_i$ and $\frac{n_f}{n} < \delta$, which means that the companion X_j state has higher food concentration (higher fitness function value) and the surrounding is not very crowded, it goes forward a step to the companion X_j, the $X_i^{(t+1)}$ is calculated by Eq. 2.4.

Otherwise, executes the preying behavior.

4) **Move** – fish randomly swim in water; in fact, they are seeking food or companions in larger ranges. Chooses a state at random in the vision, and then moves to that state by Eq. 2.5, in fact, it is the preying behavior i.e. the default AF behavior.

5) **Leap** – fish stop somewhere in the water, the behavior result of each AF will gradually be the same, the difference of objective values (food concentration, FC) become lower in certain iterations, it may fall within the local extreme to change the parameters randomly to the still states for leaping out current state.

If the lens function is almost the same or if the difference in lens functions is less than a proportion during the given iterations $(m - n)$, choose a few fish at random from the entire fish swarm, and set parameters randomly to the selected AF. The β is a parameter or a function that can makes some fish have other abnormal actions (values), ϵ is a smaller constant.

$$\begin{aligned} &If(BestFC(m) - BestFC(n)) < \epsilon \\ &X_{some}^{(t+1)} = X_{some}^{(t)} + \beta.Visual.rand() \end{aligned} \tag{2.7}$$

2.2.3 Moth Flame Optimization Algorithm

Moth Flame Optimization Algorithm (MFOA) is a bio-inspired algorithm that was computerized and developed by (Mirjalili, 2015). This algorithm inspired by the navigation method of moth in nature called transverse orientation.

In this method, moth flies in a straight curve by following the moonlight and always maintains a fixed angle with the moon. This mechanism works correctly with the moon because is very far, but it fails if the flame is nearer. In other words, the moth cannot keep the same angle with near flame, it gradually decreases, leading to a spiral path (Frank, Rich, and Longcore, 2006).

MFOA is based on stochastic population optimization, the moths can flay in different dimensions $(1-D, 2-D, 3-D, ...)$, where they considered as candidate solutions, the flames are the best positions of moths that obtain so far. The moths and flames can be described by two matrices $(n \otimes d)$ where n number of moths and d is the number of variables (dimension)(Eq. 2.8).

$$M = \begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & \cdots & m_{1,d} \\ m_{2,1} & m_{2,2} & \cdots & \cdots & m_{2,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ m_{n,1} & m_{n,2} & \cdots & \cdots & m_{n,d} \end{bmatrix} \; ; \; F = \begin{bmatrix} f_{1,1} & f_{1,2} & \cdots & \cdots & f_{1,d} \\ f_{2,1} & f_{2,2} & \cdots & \cdots & f_{2,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ f_{n,1} & f_{n,2} & \cdots & \cdots & f_{n,d} \end{bmatrix} \tag{2.8}$$

The fitness value is calculated for each moth and stored in the OM array as represented below (Eq. 2.9):

$$OM = \begin{bmatrix} OM_1 \\ OM_2 \\ \vdots \\ OM_n \end{bmatrix} \tag{2.9}$$

In the following steps, we mention the main phases of MFOA:

1) Each moth M_i uses the following Eq. 2.10 to update its next position with respect to the flame F_j.

$$M_i = S(M_i, F_j) \tag{2.10}$$

M_i: represents the i^{th} moth , F_j : represents the j^{th} flame, S: is the spiral function.

2) Each moth to fly around the flame. For guaranteed this motion, the author proposed the spiral motion a considered as a logarithmic spiral is given by Eq. 2.11

$$S(M_i, F_j) = D_i \exp(b * t).cos(2\pi t) + F_j \tag{2.11}$$

where, D_i is the absolute distance between the i^{th} moth and the j^{th} flame given by Eq. 2.12:

$$D_i = |F_j - M_i| \tag{2.12}$$

b : constant for defining the shape of the logarithmic. t : represents spiral random number in $[r, +1]$ where r is linearly decreased from -1 to -2 over the course of iteration. Note that r is named as the convergence constant.

3) Update the number of flames using adaptive reduction is given by Eq. 2.13

$$N_{flame} = round(N - l * \frac{N-1}{T}) \tag{2.13}$$

Where l is the number of iteration, T: represents the maximum number of iteration and N is the maximum number of flames.

4) If the termination criterion is satisfied, the process should be stopped. After termination, the best moth is returned as the best-obtained approximation of the optimum.

After all, Algorithm 2.2 gives the pseudo-code of the Moth Flame Optimization Algorithm.

Algorithm 2.2 Moth Flame Optimization Algorithm

InitialSolutions(M); ▷ generate initial solutions
OM = FitnessFunction(M); ▷ calculate the objective function values

while (T(M) == false) **do**
    Update n flame using Eq. 2.13;
    OM ← FitnessFunction(M);;
    if (iteration = 1) **then**
        F ← sort(M);
        OF ← sort(OM);
    else
        F ← sort(M_{t-1}, M_t);
        OF ← sort(M_{t-1}, M_t);
    end if
    for (i ← 1: n) **do**
        for (j ← 1: d) **do**
            Update r and t
            Calculate D using Eq. 2.12 with respect to the corresponding moth
            Update $M(i, j)$ using Eqs. 2.10 and 2.11 with respect to the corresponding moth
        end for
    end for
end while

2.3 Multiple Sequence Alignment

2.3.1 Definition

Sequence alignment and specifically multiple sequence alignment (MSA) is one of the major research subjects within the Bioinformatics field (Notredame, 2002). Multiple sequence alignment depends on the number of sequences and their length. MSA is the typical alignment of three or more protein or DNA sequences, in search of maximal similarity among them (Chellapilla and Fogel, 1999). Then, using some measures identifies the regions of similar sequences and produces homogeneous positions in columns. By analyzing the similarities and differences of either protein or DNA sequences, it is possible to infer structural, functional or evolutionary relationships between the sequences being studied (Baxevanis and Ouellette, 2004). It is also necessary to align sequences in the first stage of the process of creating phylogenetic trees

Let $\sum = \{\alpha_1, \alpha_2, .., \alpha_n\}$ be an alphabet without the character $'-'$ of size m containing the characters that may constitute any given sequence, and $\sum^{'} = \sum \cup \{'-'\}$.

DNA sequences, $\sum = \{A, C, G, T\}$. In contrast, for the protein sequences, $\sum$ constitutes mainly of the twenty amino acids.

A sequence can be denoted as $S = \alpha_1\alpha_2...\alpha_l$ where l is the length of the sequence. In addition, let $S_1, .., S_k$ be K sequences on $\sum$ with lengths $n_1, .., n_k$, respectively.

An alignment of k sequences $S_1, S_2, .., S_k$ is the set of sequences $S_1^{'}, S_2^{'}, .., S_k^{'}$ where S_1 is transformed to $S_1^{'}$, S_2 is transformed to $S_2^{'}$,... and S_k is transformed to $S_k^{'}$ by inserting gaps in the original sequences in certain positions allowing the new produced sequences to share more similarity.

Generally, a multiple sequence alignment is represented by a matrix $A(KL)$ (see Figure 2.2), and respects the following properties:

$$Max\{n_1, .., n_k\} \leq L \leq \sum_{i=1}^{K} n_i \tag{2.14}$$

- $A[i][j] \in \sum^{'} \forall 1 \leq i \leq k; 1 \leq j \leq l$
- The i^{th} line A_i without a gap is equal to S_i.
- A has no column which only contains gaps.

S1=	G	K	G	D	P	K	K			
S2=	M	Q	D	R	R	P	M	N		
S3=	M	K	K	K	H	P	D	F		
S4=	M	H	I	K	K	P	L	N	A	F

S1=	G	K	G	D	P	K	K	-	-	-	-
S2=	M	Q	D	R	-	-	R	P	-	M	N
S3=	M	K	K	-	K	-	H	P	-	D	F
S4=	M	H	I	K	K	P	L	-	N	A	F

FIGURE 2.2: Alignment with gaps

In biological perspective, the gaps represent residue symbols that are deleted from the sequences during the course of evolution. In general, when a gap is inserted in the sequence, a penalty is applied to the alignment. The gap penalty is determined by the location of the insertion.

2.3.2 Evaluate Multiple Sequence Alignment

There are many different methods for evaluating a multiple sequence alignment (Chia and Bundschuh, 2006). In order to perform a comprehensive evaluation of the algorithms and the quality of the results, there are two popular quantitative methods for measuring such conformity are Sum-of-Pairs score (SP) and the Column Score (CS).

Sum-of-Pairs score (SP) – is the ratio of all residue pairs in the core blocks of the reference alignment that are also correctly aligned in the test alignment which is used to determine the extent to which the programs succeed in aligning. Also, It is widely used in applications such as finding conserved regions. In SP score, we assume all sequences equally relate to each other, then all pairs of sequences are assigned the same weight. The SP score is defined as below:

Suppose we have a test alignment A of size NM, and a reference alignment R of size NMr, where N is the number of sequences, and M, Mr are the number of columns in the test and reference alignment respectively, and with column j from A represented as $A_{1j}, A_{2j}, .., A_{Nj}$, for each pair of residues A_{ij} and A_{kj} we define $p_{ikj} = 1$ if residues A_{ij}

and A_{kj} are aligned with each other in the reference alignment, otherwise $p_{ikj} = 0$. Then the score for the j^{th} column S_j is given by:

$$S_j = \sum_{i=1,\, i\neq k}^{N} \sum_{k=1}^{N} P_{ikj} \tag{2.15}$$

In addition, the same applies to S_{rj} is the score S_j for the j_{th} column in the reference alignment, the SP score for the test alignment is given by:

$$SP(A) = \frac{\sum_{j=1}^{M} S_j}{\sum_{j=1}^{Mr} S_{rj}} \tag{2.16}$$

Column score (CS) – is the percentage of columns in the core blocks of the reference alignment that are also correctly aligned in the test alignment i.e. it tests the ability of the programs to align ALL of the sequences correctly. So, it is most popular in many alignment benchmark tests.

Based on the same definition of a test alignment A and a reference alignment R described above, the score $C_j = 1$ if all the residues in the column are aligned in the reference alignment, otherwise $C_j = 0$.

$$SP(A) = \frac{\sum_{j=1}^{M} C_j}{M} \tag{2.17}$$

Moreover, Both scoring systems have been implemented with great success in the BaliBASE program referred to as *Bali_score* (Thompson, Plewniak, and Poch, 1999), which takes as input a reference alignment and a test alignment in MSF format. In addition, we have both SP and CS range from 1.0 for perfect agreement to 0.0 for no agreement. Therefore, in this thesis, we will use the *Bali_score* to estimate the quality and accuracy of the test alignment in our experiment.

2.3.3 Objective Functions to Multiple Sequence Alignment

In order to find the best alignment, two important factors must be identified: the MSA optimization algorithm and the objective function. Thus, all MSA algorithms require an objective function to determine a quantitative measure and which alignment is the best. Generally, an alignment's quality relies on its score such that alignments with higher scores have better quality. On the other hand, the purpose of an MSA algorithm is to utilize a score function and find the MSA that maximizes this score. Several alignment methods are based on a scoring system that made up of character substitution scores as well as penalties for gaps. Therefore, many methods have been developed to quantify multiple sequence alignment, among them the following.

Sum-Of-Pairs (SOP) Objective Function – is one of the most popular quantitative measurements for multiple sequence alignment (MSA). it is an extension of the typical pairwise scoring method to an MSA. Here, a pairwise matched residue gets a positive score, a mismatch gets a negative score, and a gap or space gets a negative score. Formally, the SOP score is defined as (Eq. 2.18):

$$SOP = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} f(S_i, S_j) \tag{2.18}$$

where n is the total number of sequences in the alignment, and $f(S_i, S_j)$ is the score of the pairwise alignment between sequences S_i and S_j. This score can be calculated by a measuring of similarity or distance.

$$f(S_i, S_j) = \sum_{p=1}^{l} Score(a_p, b_p) - \sum GAP \tag{2.19}$$

where l is the length of the aligned sequences, $Score(a_p, b_p)$ is determined by a substitution matrix, and $\sum GAP$ is the total penalty score due to inserting gap, so we can be obtained by the gap penalty model which is explained below.

Generally, for DNA sequences, the simple match/mismatch cost scheme is often used, and for protein sequences, the PAM matrix (Dayhoff, Schwartz, and Orcutt, 1978) and BLOSUM matrix (Henikoff and Henikoff, 1992) are the most widely used.

Gap – Gaps can be seen as artificial insertions into sequences to move similar segments of sequences into alignment. In any given alignment, inserting gaps will lower the score of this alignment, this is due to what is called a "gap penalty". On the other hand, From a biological point of view, indel is a very rare process between homologous sequences in natural evolution and thus gap must be penalized in the scoring. So, an opening gap is penalized more than extending gaps (Chowdhury and Garai, 2017).

The gap is divided into two types: gap open penalty (GOP) and gap extension penalty (GEP), so the affine penalty formula is given as follows (Eq. 2.20):

$$\sum GAP = N_{GOP} * GOP + N_{GEP} * GEP \tag{2.20}$$

where N_{GOP} is the number of the GOP, N_{GEP} the number of the GEP, and $GOP > GEP$.

2.3.4 Classification Multiple Sequence Alignment Algorithms

Even though there have been many algorithms and software programs implemented to produce multiple sequence alignments of protein and DNA sequences, but finding the optimal solution for MSA is an NP-Complete problem as demonstrated by Wang and Jiang (Wang and Jiang, 1994). MSA is a complicated problem either because of its high complexity cost or the lack of appropriate evaluation function to assess the quality of multiple sequence alignment (Wang and Jiang, 1994). Therefor, MSA algorithms can be classified into exact algorithms and heuristic algorithms. The heuristic algorithms, of course, are the most more numerous, and they can also be subdivided into two categories: progressive and iterative algorithms (Notredame, 2002).

Exact Algorithms

This approach is easy to implement and produces high-quality alignment, but the running time is NP-Complete. Generally, exact methods are high-quality heuristics that find multiple alignments very close to the optimal. They are known as dynamic programming based solutions. The dynamic programming, mathematically guaranteeing optimal alignment, is good at finding the optimal alignment for two sequences. The dynamic programming, mostly, requires too much time and space; to be practical, it is useful to apply it to small sequences of medium length. The time complexity of this algorithm is given by $O(n^m 2^m)$ to construct an alignment of m sequence of length n (Waterman, Smith, and Beyer, 1976). The main idea of these algorithms is to exclude part of the multidimensional space that does not contribute to the solution, therefore, thus saving less computational time and memory.

In the literature, several methods have been proposed to optimize the alignment by running dynamic programming alignment on all sequences simultaneously. Divide-and-Conquer Alignment (DCA) proposed by Jens Stoye (Stoye, 1998) for MSA, which utilizes a "divide-and-conquer" strategy. DCA algorithm cuts the sequences in subsets of segments that are small enough to be fed to MSA. The critical issue is to cut the sequences at the right points so that the produced alignments remain as close as possible to optimal. The performance of DCA depends on how it divides the sequences. Hence, DCA does not guarantee to find an optimal solution. DCA manages to align up to 20 – 30 closely related sequences.

Progressive Algorithms

Methods based on a progressive approach are known to be faster (Zhou and Hansen, 2004) and give satisfactory results. These algorithms are the most widely used, since they can align multiple sequences in little time and with little memory. The authors in (Feng and Doolittle, 1987) proposed one of the first progressive alignment methods. The main idea behind any progressive alignment approach is building a guide tree out of a set of sequences and then aligning those sequences according to the order proposed by the tree.

Usually, all progressive algorithms are based on the same principle: Firstly, we start by aligning sub-groups of sequences (a pair of sequences) and then try to added one by one so that never more than two sequences (or multiple alignments) are simultaneously aligned until all sequences are aligned into a single consensus sequence. Thus, the progressive methods ensure that never more than two sequences or profiles are aligned simultaneously. Lastly, the algorithm stops when all sequences have been grouped together to form the complete multiple alignment. At each step of these algorithms, we can be used different strategies, even if the general principle is the same for all.

The major disadvantage of this kind of methods is to stop at the local minima. In this case, if an error is made at the beginning of the alignment, it will transfer to the final alignment (MSA final), i.e., this is due to heavy dependence on the accuracy of the initial pairwise alignments.

Several algorithms have been proposed can be grouped in this category, such as ClustalW (Thompson, Higgins, and Gibson, 1994), MultAlign (Corpet, 1988), T-COFFEE (Notredame, Higgins, and Heringa, 2000), Treealign (Hein, 1989), POA (Lee, Grasso, and Sharlow, 2002), and MAFFT (Katoh et al., 2002).

Iterative Algorithms

Unlike exact algorithms, the iterative approaches are simple, fast and efficient, that are aimed to improve multiple alignment methods. This approach can be used to increase the result of existing software with any objective function. It can also be incorporated into a progressive alignment strategy to establish alignments from the beginning, then repeatedly refine this alignment through a series of iterations until to give better results (Wallace et al., 2006) i.e., no more improvements can be made. The algorithm behind iterative alignment methods can be summarized by the following steps (Yasin, 2016):

1) An initial alignment is calculated.

2) One sequence is taken from the alignment and re-aligned to the profile of the remaining sequences. Only cases where the score is being optimized are considered, this means that the overall score is increased or stays the same.

3) Step 2 is repeated by choosing another sequence and re-aligning it to the profile of the remaining aligned sequences until the alignment does not change.

Iterative methods are able to produce excellent alignments, however, it require more computational resources than progressive alignment methods. Also, the iterative approaches can be divided into two types: iterative and iterative plus stochastic (Naznin, Sarker, and Essam, 2011).

In the literature, we find several approaches that take the advantage of performing an iterative refinement, in order to obtain more precise alignments, among them there are MUltiple Sequence Comparison by Log-Expectation (MUSCLE) (Edgar, 2004), FAlign (Chakrabarti et al., 2004), SAGA (Notredame and Higgins, 1996), MSA-GA (Gondro and Kinghorn, 2007), hidden Markov model (Mount, 2009), PROBabilistic CONSistency-based multiple sequence alignment (ProbCons) (Do et al., 2005).

2.4 Feature (Gene) Selection

In this section, keep in mind that the gene represents a feature, hence, everything that will be said in the feature selection applies to gene selection. On the other hand, Gene selection is a branch of feature selection.

2.4.1 Definition

Feature selection is also called variable selection or attribute selection. It is a process to select a small subset of the relevant features from the original ones by eliminating irrelevant, redundant, or noisy features. Eliminating irrelevant features will not influence learning performance. Generally, feature selection aims to better-learning performance i.e. higher learning accuracy, lower computational cost, and better model interpret-ability (Dash and Liu, 1997).

In general, the feature selection problem can be defined by (see Figure 2.3):

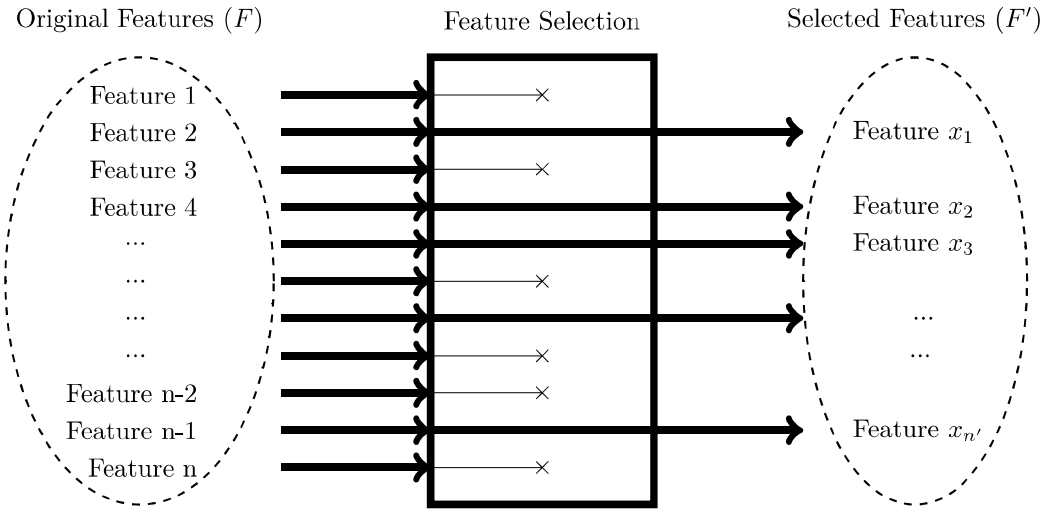


FIGURE 2.3: Feature Selection, when ($F' \subseteq F$) and ($1 \leq n' < n$).

Let $F = \{f_1, f_2, .., f_n\}$ represents a non-empty feature subset of cardinality $N \in \mathbb{N}$. Consider a criterion $\mathcal{J} : F' \subseteq F \mapsto \mathbb{R}$ function scoring the quality (e.g., discriminatory ability) of feature subsets.

Let us assume that the highest value of $\mathcal{J}$ is obtained for the best feature subset. The goal of the selection is to find a subset $F'(F' \subseteq F)$ of size $N'(N' \leq N)$ such that :

$$\mathcal{J}(F') = \max_{Z \subseteq F} \mathcal{J}(Z) \tag{2.21}$$

where $|Z| = N'$ and N' is either a number pre-defined by the user or controlled by a sub-set creation method.

In addition, feature selection is a challenging task due mainly to a large search area, wherever the full range of probable solutions is 2^n for a dataset with n attributes (Dash and Liu, 1997; Guyon and Elisseeff, 2003). The search of optimal feature subset for the criterion $\mathcal{J}$, is then an NP-Complete problem (Davies and Russell, 1994; Cotta and Moscato, 2003).

2.4.2 General Feature Selection Process

Although there are several feature selection methods, the authors in (Dash and Liu, 1997) given a general idea to a typical feature selection process. This technique is illustrated in Figure 2.4 in which there are five basic stages.

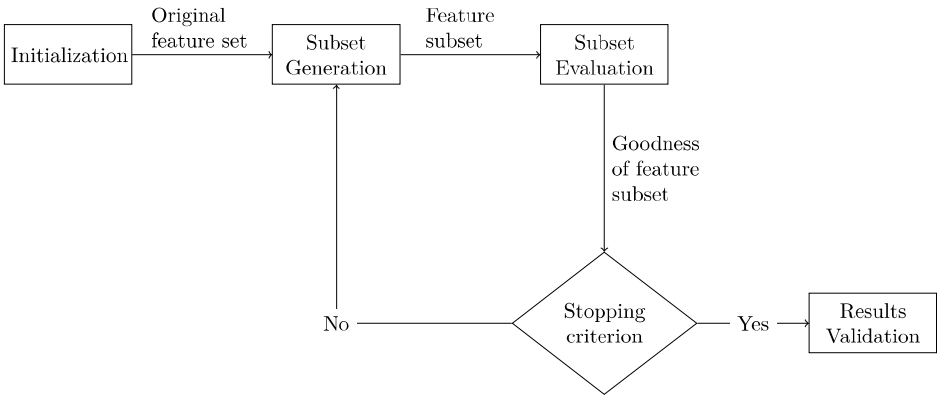


FIGURE 2.4: General feature selection process (Dash and Liu, 1997).

1) A feature selection algorithm begins with an initialization procedure. This procedure is the initial stage of a feature selection algorithm, and it is prepared by all the original features in the problem.

2) Generation procedure is used to create a subset of features that is relevant to the target concept and determines the exploration of the search space. It can begin without features, all features, or a random subset of features.

3) An evaluation function tries to measure the discriminating ability of a feature or a feature subset to give the quality of the candidate subsets. An optimal subset is always relative to a certain evaluation function.

4) The stopping criterion decides when to stop the process. Stopping criterion can be based on the evaluation function or the search procedure. The stopping criteria most commonly used are: (i) When the search completes (ii) When some given bound (minimum number of features or a maximum number of iterations) is reached.

5) A validation process to verify if the required objective is achieved. The validation procedure is not a portion of the feature selection process itself, but a feature selection algorithm should be validated. In addition, the test set technique will validate a feature subset.

2.4.3 Feature Selection Methods

Usually, feature selection methods are categorized into three categories (Kohavi and John, 1997; Guyon and Elisseeff, 2003): filter, wrapper and embedded methods.

Filter Methods

The filter methods allow gene (feature) selection independently of the classification model. In this approach, the selected genes are often independent of the classification model.

In the filter method (see Algorithm 2.3), the basic idea is to evaluate each attribute (univariate case) to assign a relevance score. Firstly, we use this score to determine the ranking of attributes. Ultimately, we only select the best-ranked attributes, i.e. the most relevant ones (Zhang and Rajapakse, 2009). It should be underlined that there are also some multivariate methods that assign scores to a group of attributes.

The advantage of this approach is that it can be used when working with a very large number of attributes because it is of reasonable complexity. Therefore, filter methods are extremely easy to implement and fast to run. Taking into account the distributed data, filters can be divided into two classes: parametric and non-parametric (Hameed et al., 2018a).

Algorithm 2.3 Filter Method (from (Liu and Yu, 2005))

Input:
$F(F_0, F_1, .., F_{n-1})$ ▷ data set X including n feature
S_0 ▷ initial subset
δ ▷ a stopping criterion
Output: S_{best} ▷ An optimal subset

1: $S_{best} \leftarrow S_0$ ▷ Initialize S_{best}
2: $\mu_{best} \leftarrow eval(S_0, F, J)$ ▷ evaluate S_0 by an independent measure J
3: **while** (δ *is not satisfied*) **do**
4: $S \leftarrow generate(F)$ ▷ generate a subset for evaluation
5: $\mu = eval(S, F, J)$
6: **if** (μ *is better than* μ_{best}) **then**
7: $\mu_{best} \leftarrow \mu$
8: $S_{best} \leftarrow S$
9: **end if**
10: **end while**
11: **return** S_{best}

Wrapper Methods

In the wrapper methods (see Algorithm 2.4), the selection of gene (feature) subsets is performed in interaction with a classifier. The objective is to find a gene (feature) subset that achieves the best predictive performance for a particular learning model. The interest of these methods is that the chosen subset is perfectly adapted to the classifier. On the other hand, there is a risk of over-fitting.

Moreover, these wrapper approaches are significantly more costly in the computational time since a classifier must evaluate a subset each time. Their computational complexity is dependent on the complexity of the used learning model (Deng et al., 2019).

Algorithm 2.4 Wrapper Method (from (Liu and Yu, 2005))

Input:
$F(F_0, F_1, .., F_{n-1})$ ▷ data set X including n feature
S_0 ▷ initial subset
δ ▷ a stopping criterion
Output: S_{best} ▷ An optimal subset

1: $S_{best} \leftarrow S_0$ ▷ Initialize S_{best}
2: $\mu_{best} \leftarrow classification\_accuracy(S_0, F, A)$ ▷ apply mining algorithm A to S_0
3: **while** (δ *is not satisfied*) **do**
4: $S \leftarrow generate(F)$ ▷ generate a subset for evaluation
5: $\mu = classification\_accuracy(S, F, A)$ ▷ gets classification accuracy of S
6: **if** (μ *is better than* μ_{best}) **then**
7: $\mu_{best} \leftarrow \mu$
8: $S_{best} \leftarrow S$
9: **end if**
10: **end while**
11: **return** S_{best}

Embedded Methods

Embedded approaches are close to wrapper approaches because they combine the exploration process with a learning algorithm (Duval, Hao, and Hernandez Hernandez, 2009). The search for an optimal subset is performed to a specific learning algorithm, but they are characterized by deeper interaction between gene selection and classifier construction. This leads to a combination of the advantages of filter and wrapper based methods with a reduced computational time (Chandrashekar and Sahin, 2014; Xiao et al., 2008).

The advantage of this category is that the search process is guided by interesting information provided by the classifier, which makes these methods more efficient than wrapper methods.

2.5 Multi-Objective Optimization Problem

The Multi-objective Optimization Problem (MOP) (also called multi-criteria optimization) can be defined (in words) as the problem of finding. Many real problems involve multiple performance measures or objectives, which must be optimized simultaneously. The multi-objective optimization works by seeking to optimize the component of a vector function in objective value. Unlike single-objective optimization, the solution to a multi-objective optimization problem is not one unique solution, but it is a set of solutions known as the Pareto-optimal set (Coello, Dhaenens, and Jourdan, 2009). Each point of the set is optimal in the sense that no improvement can be obtained in a component of the objective vector that does not lead to degradation in at least one of the remaining components. Given a set of possible solutions, a candidate is considered Pareto-optimal if there are no other solutions in the set of solutions that can dominate any of the candidate solutions. In other words, the candidate solution would be a non-dominated solution.

More precisely, MOPs are those problems where the goal is to optimize k objective functions simultaneously. This may involve the maximization of all k functions, the minimization of all k functions or a combination of maximization and minimization of these k functions. Indeed, we can convert the maximization into minimization by multiplying the objective function by (-1) and vice versa.

A MOP global minimum (or maximum) problem can be defined in the following by Eq. 2.22 (Btissam and Abounacer, 2017):

$$\begin{cases} Maximize\ (orMinimize)\ f(x) = (f_1(x), f_2(x), .., f_k(x)) \\ with \begin{cases} x = (x_1, x_2, .., x_n) \\ g_i(x) \leq 0\ with\ i = 1, .., I \\ h_j(x) = 0\ with\ j = 1, .., J \end{cases} \end{cases} \tag{2.22}$$

where

- x is a n-dimensional decision variable vector $x = (x_1, x_2, .., x_n)$ from some universe Ω.
- The objective function vector $F : F(x) = \{f_1(x), f_2(x), .., f_k(x)\}$ with f_i the objectives or decision criteria, and k is the number of objectives.
- $g_i(x) \geq 0$ and $h_j(x) = 0$ represent constraints that must be full filled while maximizing(or minimizing) $F(x)$ and contains all possible x that can be used to satisfy an evaluation of $F(x)$.

When there are several objective functions, the notation of "optimum" changes, because, in MOPs, the aim is to find compromises (or "trade-off") rather than a single solution as in global optimization.

The concept of the optimum most commonly adopted in multi-objective optimization is the one proposed by Francis Ysidro Edgeworth and later generalized by Vilfredo Pareto (Schumpeter, 1949). It is defined as:

Pareto Optimality – a solution $x^* \in \Omega$ is Pareto-optimal if and only if there is not another point $x \in \Omega$ that satisfies Eq. 2.23 and there exists at least one $i \in I$ that satisfies Eq. 2.24.

$$\forall i \in \{1, 2, .., k\}\ (f_i(x) \leq f_i(x^*)) \tag{2.23}$$

$$\wedge\, \exists i \in \{1, 2, .., k\}\ (f_i(x) < f_i(x^*)) \tag{2.24}$$

Pareto dominance – a vector $u = (u_1, u_2, .., u_n)$ is said to dominate $v = (v_1, v_2, .., v_n)$, denoted $u \prec v$, if and only if u is partially less than v, i.e. (see Eq. 2.25),

$$\begin{aligned} \exists i \in \{1, 2, .., k\}\ v_i \leq u_i \\ \wedge \exists i \in I\ v_i < u_i \end{aligned} \tag{2.25}$$

Pareto Optimal Set – the set of all the Pareto optimal solutions for multi-objective problems is called Pareto-optimal Set, also known as Pareto-optimal. For given an MOP, $F(x)$, the Pareto-optimal set, P^*, is defined as (Eq. 2.26):

$$P^* = \{x \in \Omega \setminus \neg \exists x' \in \Omega F(x) \prec F(x')\} \tag{2.26}$$

2.6 Conclusion

For the last decade, Bioinformatics has become an essential interdisciplinary scientific field that has been developed to identify and analyze various components of cells such as gene and protein function, etc. Therefore, it is a critical need to use techniques to process complex biological data. Among the famous techniques applied in this field are bio-inspired algorithms.

In this chapter, we reviewed the main concepts of bio-inspired algorithms and some of their algorithms, the multiple sequence alignment and the classification of their algorithms, feature selection and their methods, and multi-objective optimization with a brief definition of Pareto-optimal.

Chapter 3

MOAFS Algorithm For MSA

3.1 Introduction

Multiple sequence alignment of DNA or protein is considered one of the most widely used techniques in sequence analysis, especially for the most difficult Bioinformatics problems (Bacon and Anderson, 1986). Multiple Sequence Alignment (MSA) is a crucial and significant task in molecular biology (Feng and Doolittle, 1987). MSA provides to the biologists a means to analyze DNA or protein sequences in terms of determining subsequently their degree of homology or divergence. In the construction of phylogenetic trees, MSA has been used to describe the motif into families' proteins (Wang and Jiang, 1994). This allows for the prediction of the structural and functional aspects.

Biologists need specific tools for the multiple sequence alignment of the protein family to study gene evolution in various organizations (Taylor and Thornton, 1984). MSA is a complex problem due to the high complexity cost or the lack of appropriate evaluation function to assess the quality of multiple sequence alignment (Wang and Jiang, 1994). The optimal realization of MSA requires a high computing time and a lot of storage space to store the massive number of sequences. The MSA has been shown that is an NP-Complete problem by (Wang and Jiang, 1994). So the resolution of MSA by exact method seems a mission difficult if not impossible.

Several multiple sequence alignment methods have been proposed and can be classified under three main categories: exact, progressive and iterative alignments (see Section 2.3.4).

Generally, the proposed methods in the literature are heuristics. These heuristic methods try to approach the optimal alignment without actually achieving it. However, the optimized mathematical alignment is not necessarily the optimal biological alignment. This is due to the nature of biological data (Chao and Zhang, 2008). The possibility to have different solutions gives another difficult to the MSA problem. To build an MSA, different approaches (or methods) are proposed in the literature. Though these methods are sometimes associated, concatenated, and/or hybridized to build one single new method (Edgar, 2004).

In the literature, we find several approaches that take the advantage of performing an iterative refinement in order to obtain more precise alignments, among them there are MUltiple Sequence Comparison by Log-Expectation (MUSCLE) (Edgar, 2004), Multiple Alignment using Fast Fourier Transform (MAFFT) (Katoh et al., 2002), PROBabilistic CONSistency-based multiple sequence alignment (ProbCons) (Do et al., 2005), MSAProbs (Liu, Schmidt, and Maskell, 2010), and MUMMALS (Pei and Grishin, 2006).

In the stochastic approach, we can cite the simulated annealing (Huo and Stojkovic, 2007), genetic algorithms (Silva et al., 2010), hidden Markov model (Mount, 2009) and Gibbs Sampling (Lawrence et al., 1993).

In addition, several bio-inspired algorithms have been proposed in the literature to solve the MSA problem such as: SAGA (Notredame and Higgins, 1996), MSA-GA (Gondro and Kinghorn, 2007), RBT-GA (Taheri and Zomaya, 2009), VDGA (Naznin, Sarker, and Essam, 2011), GAPAM (Naznin, Sarker, and Essam, 2012), MOMSA-W (Zhu, He, and Jia, 2016), HMOABC (Rubio-Largo, Vega-Rodríguez, and González-Álvarez, 2016), EGSA (Zemali and Boukra, 2018), MO-SAStrE (Ortuño et al., 2013), Heuristics for multi-objective multiple sequence alignment (Abbasi, Paquete, and Pereira, 2016), ABC-BFO (Rani and Ramyachitra, 2018), IWO-GA (Gao et al., 2018) and MSA-CRO (Wadud et al., 2018).

The previous methods have many weaknesses, among them,

- ☞ They do not provide the necessary diversity in the input parameters to maintain a trade-off between the exploration and the exploitation of the search space of the MSA problem.
- ☞ Also, its principle is to align all the sequences given, i.e. even if one or more sequences are not very similar or remain sequences that are not yet aligned, they should be aligned with the first aligned sequences. If these dissimilar sequences are aligned with others, the similarity value used to compare the sequences will be decreased.
- ☞ A few algorithms use only one criterion in their objective function, the improvement of one objective may deteriorate one or more other objectives. It is impossible to optimize a single objective to achieve all objectives at the same time.

In proposing a new algorithm to solve the MSA problem, these and other weaknesses must be taken into account.

Initially, the AFSA works based on the population and stochastic search. This algorithm is one of the best approaches of the Swarm Intelligence method with considerable advantages such as high convergence speed, good robustness, global search capability, parameter tolerance, flexibility, error tolerance, great accuracy and it is also proved to be insensitive to initial values (Neshat et al., 2012).

This chapter proposes a new algorithm to solve the MSA problem called Multi-Objective Artificial Fish Swarm (MOAFS). The MOAFS is a new method dedicated to multiple sequence alignment that uses a multi-objective optimization based on the artificial fish swarm algorithm. To evaluate the possible candidate multiple sequence alignment, we used two objective functions: Weighted Sum of Pairs (WSP) and Similarity. To explore the search space of the MSA problem, we proposed three procedures inspired by three artificial fish swarm behaviors.

These procedures have been implemented based on genetic crossover and mutation operators, we have newly introduced. Particularly in the following operator, the fish swarms swim around the center fish and a new consensus sequence is defined to adapt this operation to the MSA problem. We performed experiments on BAliBASE 2.0 and BAliBASE 3.0 (Bahr et al., 2001) to demonstrate the effectiveness of MOAFS. The MOAFS algorithm has shown better performance than other algorithms recently published in the literature. MOAFS achieved the highest accuracy in terms of SP or CS scores. This proves that the MOAFS algorithm is a promising approach to solve the MSA problem.

Ultimately, we have addressed some and other weaknesses above-mentioned, and the power of our approach represents as follows:

1) The initial population is separated to ensure a balance between exploration and exploitation of MSA problem search space.

2) We used Pareto-optimal solutions, which based on a selection of non-dominated solutions. They include all solutions that have no other solution exists which can optimize one of the objective functions without aggravating some other objectives.

3) We use WSP and Similarity functions to determine vertically and horizontally similar regions, respectively. This technique gives the balance between alignment sequences and similar regions that is one of the key strengths of our algorithm, and to our knowledge, which one has not been used by other algorithms.

4) And other powers are inherited directly from the original artificial fish swarm algorithm (AFSA) (Li, 2002) and genetic operators.

The remainder of this chapter is organized as follows: Section 3.2 is devoted to the description of our algorithm MOAFS. Section 3.3 reports the obtained findings from the application of the proposed approach to BAliBASE 2.0 and BAliBASE 3.0. Finally, the conclusion of this chapter in Section 3.4.

3.2 Multi-Objective Artificial Fish Swarm Algorithm for MSA Problem

The proposed MOAFS algorithm uses the multi-objective concept to solve the MSA problem that has been used to find a non-dominated optimal solution for MSA. Also, it uses AFSA, and fish behaviors (Praying, Swarming, and Following). Thus, candidate solutions to the MSA problem are represented by artificial fish (AF).

The principle of our proposal is to use an initial population to initialize all artificial fish, each one is a solution to the MSA problem, and an archive that contains all non-dominated individuals. After the creation of the archive, each individual chooses its behavior. The proposed method operations are population initialization, population evolution, the distance between two alignments, genetic operators and MOAFS behaviors (Swarm, Follow and Prey). The explanation of these operations will be described in detail below.

3.2.1 Representation of Candidate Solutions

In evolutionary algorithms, the alignments coded using the classical representation: the standard alphabet for the amino acids and the symbol $'-'$ for the gaps (Figure 2.2). However, this representation can lead to more complex and inefficient operators. For this reason, we propose a codification: each artificial fish (AF_i) is represented by a matrix with two fulfilled conditions:

1) The amino acids are encoded by their positions in the sequence to which they belong;

2) The gaps are encoded by the position of the last amino acid in the sequence where they belong, but with a negative value.

In all optimization phases, our proposal uses coded alignments (coded individuals) and at the last phase of the optimization, the coded alignments are returned (decoded) to the standard alignment representation. An example of a coded alignment is illustrated in Figure 3.1.

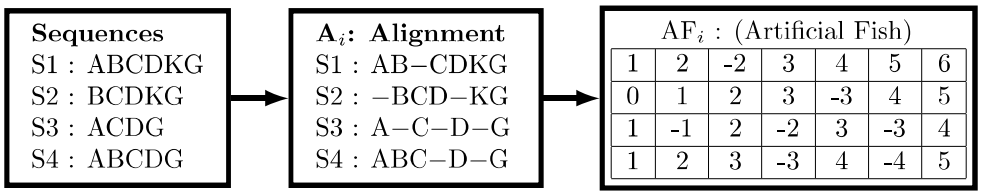


FIGURE 3.1: Alignment coding.

3.2.2 Population Initialization

For a rapid convergence towards the best alignment, and at the same time, we avoid the premature convergence of the MOAFS algorithm. Therefore, we have been proposed an intelligent initialization method that combined the following two strategies: the first insert the gaps randomly in the sequences so that they are all of the same lengths. The second one based on the right or left offset of the gaps in a solution, these initial solutions are generated by progressive alignment methods such as ClustalW (Thompson, Higgins, and Gibson, 1994) or MUSCLE (Edgar, 2004).

In our work, we have 60% of initial population alignments generated by ClustalW or MUSCLE, while the remaining 40% are randomly generated. In summation, we have initialized the population by two strategies, so as to ensure a trade-off between exploitation and exploration, it is the main objective of this separation. Our initialization method remains in the continuous spectrum between these two extremes.

3.2.3 Objective Function

In order to find the best alignment, the MOAFS is generating the Pareto-optimal set solutions, which simultaneously maximizes the Weighted Sum of Pairs (WSP) and Similarity functions. In both objective functions, we used two substitution matrices PAM (Dayhoff, Schwartz, and Orcutt, 1978) and Blosum (Henikoff and Henikoff, 1992) to calculate the cost of each residue pair.

Weighted Sum of Pairs

The Weighted Sum of Pairs Score (WSP) is one of the most popular scoring functions. It is a variant of the sum of pairs score that gives a weight to each sequence depending on its importance in the alignment. This weight depends on the relationships between the sequences. The WSP can be formulated as follows (Eq. 3.1):

$$WPS = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w_i * w_j * score(S_i, S_j) \tag{3.1}$$

where n is the number of sequences; the weights w_i and w_j of sequences i and j calculated by the phylogenetic tree (Thompson, Higgins, and Gibson, 1994). Finlay, $f(S_i, S_j)$ is the score of the pairwise alignment between sequences S_i and S_j, calculated by Eq. 2.19.

Similarity Score

Firstly, we generate a position weight matrix (Stormo et al., 1982) from the alignment found by MOAFS. Next, the dominance value (dv) of the overhead amino acid or nucleotide column is established as follows (Eq. 3.2):

$$dv(j) = max_k\{g(k,j)\} \qquad j = 1,2,..,l \tag{3.2}$$

where $g(k,j)$ is the score of amino acid k on column j in the position weight matrix despite the presence of gaps. l is the length of sequence alignment, and $dv(j)$ is the dominance value of the dominant on amino acid on column j.

The similarity objective function of alignment A is represented by the average of the dominance values of all columns in the position weight matrix. It can be formulated as follows (Eq. 3.3):

$$Similarity(A) = \frac{\sum_{j=1}^{l} dv(j)}{l} \tag{3.3}$$

The best alignment is the one with a very high probability, i.e. the value of similarity is closer to 1.

3.2.4 Distance Between two Alignments

The distance between two different alignments is the number of mismatches for the gap positions among these alignments. We assume that we have:

- ☞ two alignments A and B,
- ☞ $P(A_i)$: is the set of gap positions of sequence i in alignment A,
- ☞ $P(B_i)$: is the set of gap positions of sequence i in alignment B.

The distance between two alignments A and B gives by the following formula (Eq. 3.4):

$$Distance(A,B) = 1 - \frac{\sum_{i=1}^{n} |P(A_i) \bigcap P(B_i)|}{\sum_{i=1}^{n} |P(A_i) \bigcup P(B_i)|} \tag{3.4}$$

where n is the total number of sequences.

3.2.5 Genetic Operators

Mutation Procedure

We distinguish two types of mutation: simple mutation and Block Shuffling.

Simple Mutation – the mutation operator only changes the gaps, in order to maintain the order of the amino acids. Firstly, a random set of closed gap spaces is moved to another random position in the same sequence. Then, columns with full gaps, if any, are removed. In this mutation, it is highlighted two important aspects, namely: new variants of alignments that have not yet been taken into account can be introduced. On the other hand, columns containing only gaps can be eliminated, thus reducing the number of gaps. A particular example of the simple mutation operation is shown in Figure 3.2.

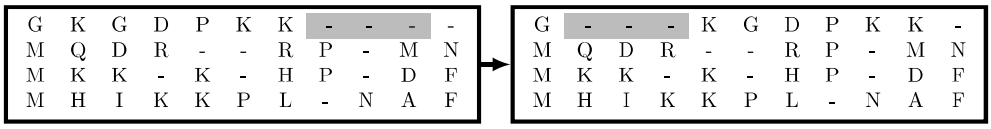


FIGURE 3.2: Simple mutation procedure.

BlockShuffling operator – is based on block definition, and it moves aligned blocks to the left or right; a block is selected in each alignment from a random point in a sequence. The purpose of the BlockShuffling operator is to move blocks of amino acids or gaps. We used two different approaches are:

BlockMove – moves whole blocks, either to the left or to the right (Figure 3.3).

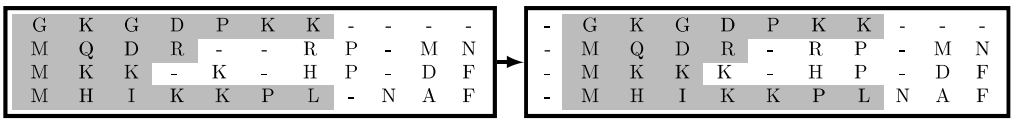


FIGURE 3.3: BlockMove works.

BlockSplitHor – the blocks are divided into two parts, upper and lower, and moves only one randomly chosen part (Figure 3.4).

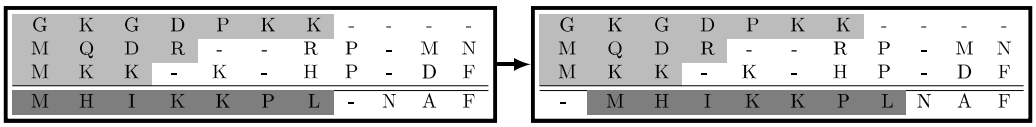


FIGURE 3.4: BlockSplitHor works.

Crossover Procedure

We used two kinds of crossovers.

Single Point Crossover – is considered as a simple crossover. Firstly, the algorithm randomly selects a column from one of the parents and divides it into two blocks. The same positions selected in this column are also found in the second parent, but not necessarily in the same column. Finally, selected blocks are crossed between these two parents. In order to match blocks of both parents, these undefined positions are filled by gaps. In this way, it can be ensured that the obtained children do not modify their sequences. The complete operation is illustrated and explained graphically in Figure 3.5.

Multiple Point Crossovers – each parent is divided into three parts. Then, these parts are exchanged between the parents and then merged together to generate two new individuals. However, the best individual will be considered a new child. The crossover is implemented in two steps as described below.

Step 1 – in order to effectively cut the first part, we choose a random parent and vertically split its first 25% columns. The other parent is also divided using the mechanism introduced in the single point crossover, as shown in Figure 3.5.

Step 2 – we now have two parts of each parent in Step 1. To create another part, we follow the same procedure in Step 1, but considering the last 25% of the columns (as shown in Figure 3.6). This gives us three parts for each parent. To complete the crossover, exchange

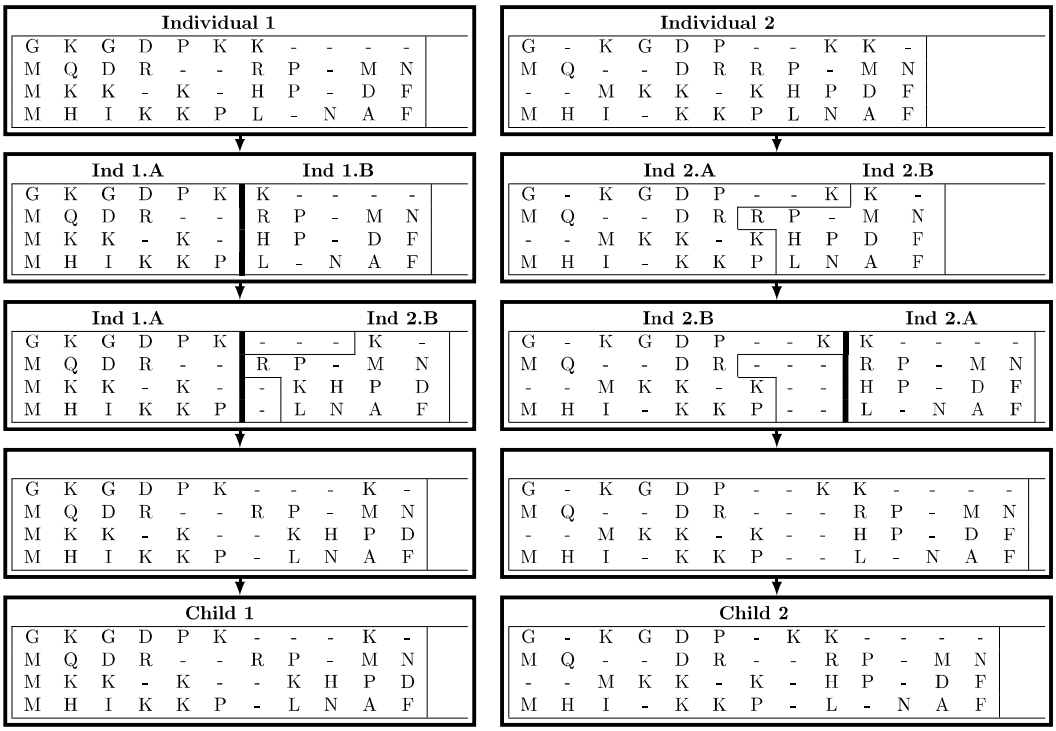


FIGURE 3.5: Single Point Crossover.

the intermediate parts between the parents, then all three parts are merged together to produce two new individuals as shown in Figure 3.6.

3.2.6 Function Center Artificial Fish (Consensus)

In molecular biology and Bioinformatics, the consensus sequence is the most common nucleotide sequence or peptide sequence at each position of a sequence alignment. In our work, we have inspired the idea of consensus to get a center individual of all individuals who represent the Artificial Fish. We have noted:

- ☞ each alignment represents as follows $A_k : (S_1, S_2, S_3, ..., S_n)$, where S_i is the sequence of alignment, n is the total number of sequences.
- ☞ each sequence of this alignment represents by S_i : $c_{i1}\ c_{i2}\ c_{i3}\ ...\ c_{il}$, where i is the sequence number, and l is the length of the alignment.

In order to define the center individual (consensus alignment), we pass through two steps: First step, each character c_{ij} of sequence S_i in the A_c consensus alignment is the most frequent character in the j^{th} column of all the i^{th} sequences in different A_k alignments. In the second step, each consensus sequence obtained is corrected by correcter process. Both steps are shown in Figure 3.7.

3.2.7 Three Behaviors of MOAFS

AF_Swarm

In the moving process, the fish naturally gather in groups, which is a kind of lifestyle in order to guarantee the existence of the colony and avoid dangers.

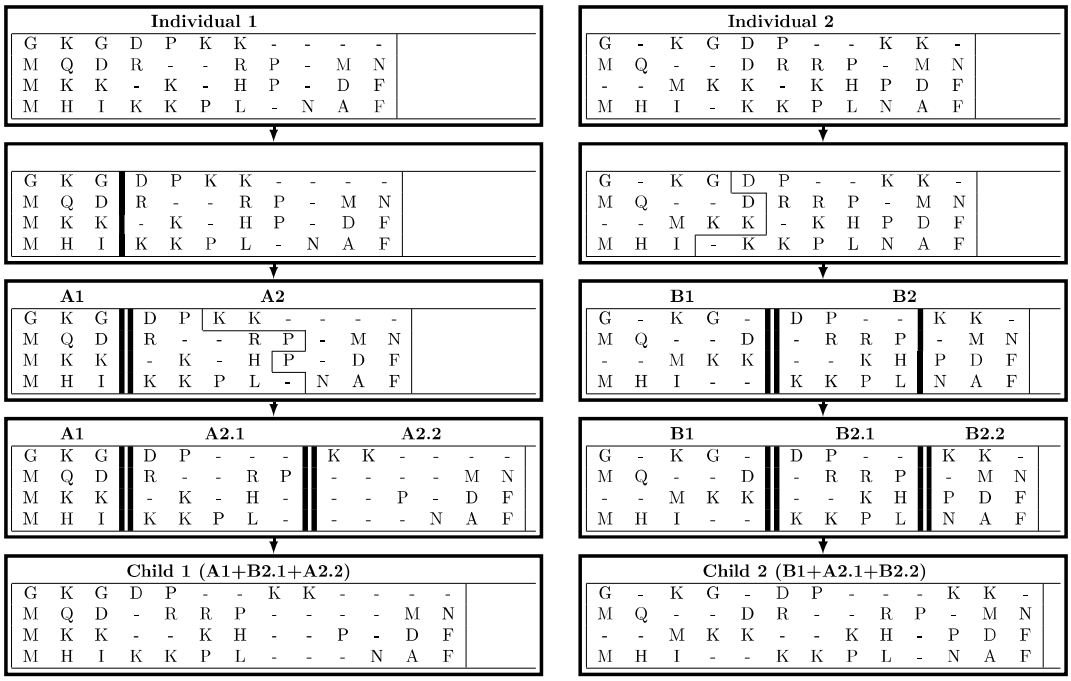


FIGURE 3.6: Multiple Point Crossovers.

In AF_Swarm behavior, The center artificial fish (AF_c) is determined by the consensus sequence method (see Section 3.2.6).

Let AF_i is the current artificial fish, AF_c be the center artificial fish, nf be the number of its companions in the current neighborhood $(Distance(AF_i, AF_c) < Visual_i)$, and n is the population size.

If $[(AF_c \prec AF_i)\ and\ (\frac{nf}{n} < \delta)]$ which means that the companion center has more food $(AF_c\ dominates\ AF_i)$ and is not very crowded, the AF moves forward a step to the companion center by a crossover operation (single or multiple) i.e. between AF_i and AF_c; (see Eq. 3.5)

$$AF_i^{(t+1)} = Crossover(AF_i, AF_c) \tag{3.5}$$

Otherwise, executes the AF_Prey behavior. The crowd factor limits the size of the swarms, and AF focuses only on the optimal area, which guarantees that AF move to optimum in a broad field. The general scheme of AF_Swarm described in Algorithm 3.1

AF_Follow

In the moving process of the fish swarm, when one or more Fish find food, the neighboring partners of the artificial fish will trail and quickly reach the food.

Let AF_i be the current AF, and AF_b is the best one who dominates all neighboring fish of AF_i that verify $(AF_b \prec AF_j) \quad \forall j\ (Distance(AF_i, AF_j) < Visual_i)$.

If $[(AF_b \prec AF_i)\ and\ (\frac{nf}{n} < \delta)]$, which means that the companion AF_b has higher food concentration $(AF_b\ dominates\ AF_i)$ and the surrounding is not very crowded, the AF_i goes forward a step to the companion AF_b, by a crossover operation (single or multiple) i.e. between AF_i and AF_b (see Eq. 3.6).

$$AF_i^{(t+1)} = Crossover(AF_i, AF_b) \tag{3.6}$$

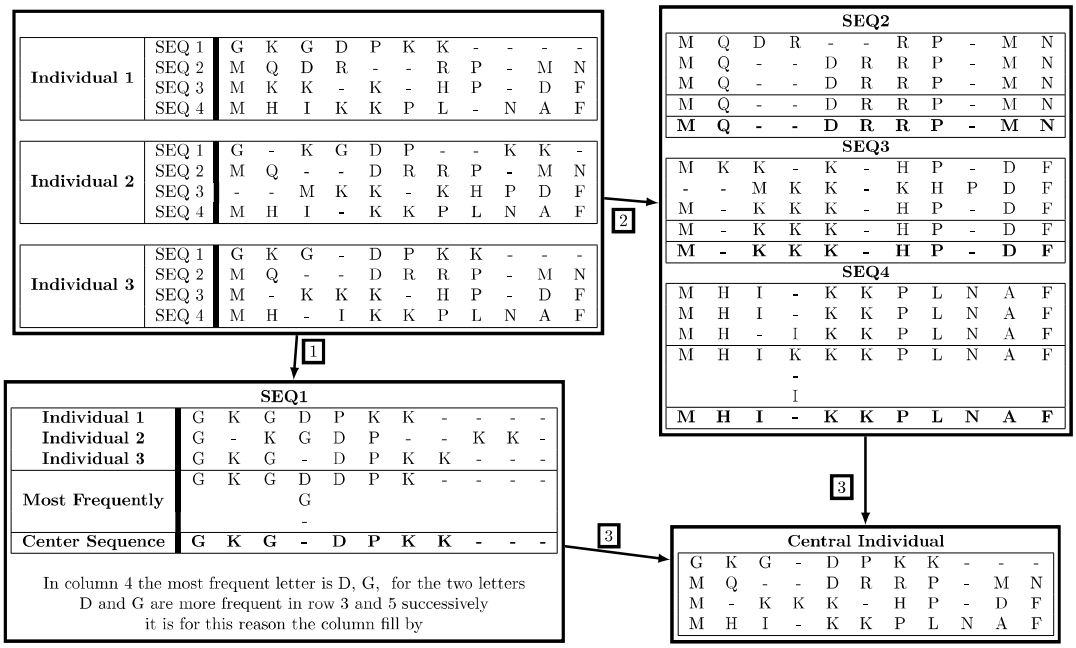


FIGURE 3.7: Center individual.

Otherwise, executes the AF_Prey behavior. The general scheme of AF_Follow described in Algorithm 3.2

AF_Prey

This is a fundamental biological behavior that leads to food; usually, the fish see the concentration of food in water to determine the movement by a vision and then chooses the trend.

In this operation, let AF_i be the AF current and select the AF_j randomly by a mutation operation (simple or Block Shuffling) in its visual distance $(Distance(AF_i, AF_j) < Visual_i)$ (see Eq. 3.7)

$$AF_j = Mutation(AF_i) \tag{3.7}$$

Algorithm 3.1 Pseudo-code of the $AF\_Swarm\ (AF_i)$

Input:

n of integer ; ▷ Population size number
AF_c of Artificial Fish; ▷ Central Artificial fish
nf of integer;

1: $AF_c \leftarrow Center(P)$ ▷ P: Population
2: $Dist \leftarrow Distance(AF_i, AF_c)$
3: $nf \leftarrow Calcul_{nf}(AF_i, Dist)$
4: **if** $[(AF_c \prec AF_i)\ and\ (\frac{nf}{n} < \delta)]$ **then**
5: $AF_i \leftarrow Crossover(AF_i, AF_c)$
6: **else**
7: $AF\_Prey(AF_i)$
8: **end if**

Algorithm 3.2 Pseudo-code of the $AF\_Follow\ (AF_i)$

Input:

n of integer ; ▷ Population size number
AF_b of Artificial Fish; ▷ Best Artificial fish
nf of integer;
$Dist$ of integer;
$Visual_i$ of integer; ▷ Visual distance

1: $AF_b \leftarrow Best_{Neighbour}(AF_i, Visual_i)$
2: $Dist \leftarrow Distance(AF_i, AF_b)$
3: $nf \leftarrow Calcul_{nf}(AF_i, Dist)$
4: **if** $[(AF_b \prec AF_i)\ and\ (\frac{nf}{n} < \delta)]$ **then**
5: $AF_i \leftarrow Crossover(AF_i, AF_b)$
6: **else**
7: $AF\_Prey(AF_i)$
8: **end if**

If $(AF_j \prec AF_i)$ i.e. $(AF_j\ dominates\ AF_i)$, it moves one step in this direction; the AF_i moves to the best neighbor, by a crossover operation (simple or multiple)(see Eq. 3.8).

$$AF_i^{(t+1)} = Crossover(AF_i, AF_j) \tag{3.8}$$

Otherwise, select a random AF_j again and decide whether it satisfies the forward condition. If it cannot satisfy after try_{number} times, it moves to a randomly chosen neighbor. The general scheme of AF_Prey described in Algorithm 3.3

Algorithm 3.3 Pseudo-code of the $AF\_Prey\ (AF_i)$

Input:

AF_j of Artificial Fish;
try_{number} of integer; ▷ Initialized by N

1: **repeat**
2: $AF_j \leftarrow Mutation(AF_i)$
3: **if** $(AF_j \prec AF_i)$ **then**
4: $AF_i \leftarrow Crossover(AF_i, AF_j)$
5: **return** $Exit()$;
6: **else**
7: $try_{number} \leftarrow try_{number} - 1$
8: **end if**
9: **until** $(try_{number} = 0)$
10: $AF_i \leftarrow Mutation(AF_i)$

3.2.8 MOAFS Approach

This section presents the proposed MOAFS algorithm for solving the MSA problem; the basic steps of the proposed algorithm are as follows:

Step 1 *Initialization of population* – the generation of the alignments population is based on the random and specific method (right or left offset of multiple sequence alignment obtained by ClustalW and MUSCLE) of each individual of the population.

Step 2 *Evaluation fitness* – the most important task is to evaluate the individual by two fitness functions. In general, the objective of multiple sequence alignment is to maintain or improve compliance with reference alignment. The first fitness function of MOAFS is the Weighted Sum of Pairs (WSP), and the second function is "Similarity", both of which perform a similarity measuring among all sequences. After the evaluation operation, the non-dominated individuals of the population are stored in the archive population.

Step 3 *Behavior selection* – each individual of the population chooses one "Swarm" or "Follow" operations, to investigate the best solution in the search space.

- ☞ If the AF chooses AF_Follow operation, it should choose a neighbor artificial fish that dominates it and whose environment is not very crowded, thus, the AF makes a crossover operation with the latter (AF advances a Step to this latter) and chooses the only child who has dominated among all the children. Otherwise, it chooses the AF_Prey operation.

- ☞ If the AF chooses AF_Swarm operation, it chooses the center individual of the population. Then, it sees which individual dominant and which environment have more food, hence, the individual makes a crossover operation with the center individual (AF advances a Step to the center individual) and chooses the only child who has dominated among all the children. Otherwise, it chooses the AF_Prey operation.

- ☞ In the AF_Prey operation, the AF randomly selects a neighbor from its visual distance by the mutation operation. If this latter dominated AF, the crossover operation will be performed between the two and chooses the only child who has dominated among all the children. Otherwise, if the condition is not satisfied after try_{number} times the individual makes a mutation operation.

Step 4 *Update the archive population*

Step 5 *Stopping criteria* – if a maximum number of iteration has been done, the process should be stopped and extract the best alignment from the archive population. Otherwise, return to Step 3.

The general scheme of the MOAFS approach is described in Algorithm 3.4.

3.3 Results and Discussion

3.3.1 Datasets

In order to evaluate the biological alignment quality produced by the MOAFS algorithm, we have used classic BAliBASE versions 2.0 and 3.0 (Bahr et al., 2001). BAliBASE Benchmark is a database that has been developed to evaluate and compare multiple sequence alignment programs containing high-quality MSA.

BAliBASE 2.0 (Bahr et al., 2001) contains 141 sets of multiple protein alignments, divided into five categories, called references. Each of these references corresponds to a different class of problems. For example, the problem of the orphan sequence $(Reference\ 2)$, which has no similarity with the other sequences. Also, the problem of sequences of very different sizes, or requiring very large gaps (length greater than 100). $Reference\ 1$ contains alignments of equidistant sequences of similar length; $Reference\ 2$ contains alignments of a family (closely related sequences with $identity > 25\%$) and three orphan sequences with 20% identity; $Reference\ 3$ consists of four families at maximum with $< 25\%$ identity between two sequences of different families; And $references\ 4$ and 5 contain sequences with large extensions $N/C - terminal$ or internal insertions.

BAliBASE 3.0 (Thompson et al., 2005) contains six sets of multiple protein alignments, each with different characteristics. $RV11$ contains 38 equidistant families with a sequence

Algorithm 3.4 Pseudo-code of the MOAFS Algorithm

Input:

Pop_Size of integer ; ▷ Population Size
$Arch\_Size$ of integer ; ▷ Archive Size
$P[Pop\_Size]$ of Artificial Fish; ▷ Population
$A[Arch\_Size]$ of Artificial Fish; ▷ Archive
Max_it of integer ; ▷ Maximum iteration
i, t of integer ;

Output:

AF_{best} of Artificial Fish

1: **for** $(i \leftarrow 1;\ i \leq 0.6 * Pop\_Size;\ i++)$ **do**
2: $Random\_Clustal\_Muscl(P[i]);$
3: **end for**
4: **for** $(i \leftarrow 0.6 * Pop\_Size;\ i < Pop\_Size;\ i++)$ **do**
5: $Random\_initialization(P[i]);$
6: **end for**
7: Evaluate individuals of P according to two fitness;
8: Create the Archive set;
9: $t \leftarrow 0;$
10: **while** $(t \leq Max\_it)$ **do**
11: **for** $(i \leftarrow 1;\ i \leq Pop\_Size;\ i++)$ **do**
12: **if** $(rand() > 0.5)$ **then**
13: $AF\_Swarm(P[i]);$
14: **else**
15: $AF\_Follow(P[i]);$
16: **end if**
17: Evaluate individuals of P according to two fitness;
18: $Update(A);$
19: **end for**
20: $t \leftarrow t + 1;$
21: **end while**
22: $AF_{best} \leftarrow Choose\_Best(A);$ ▷ Choose the best individual from Archive set
23: **return** AF_{best}

identity $< 20\%$, while $RV12$ contains 44 equidistant families with a sequence identity between 20% and 40%. These two sequences lack sequences with large internal insertions (> 35 residues). $RV20$ contains 41 families with $> 40\%$ similarity and an orphan sequence that shares $< 20\%$ similarity with the rest of the family. $RV30$ contains 30 families that contain subfamilies with $> 40\%$ similarity but $< 20\%$ similarity between subfamilies. $RV40$ contains sequences with large extensions $N/C - terminal$ and is the largest set of 49 alignments, While $RV50$ contains sequences with large internal inserts and is the smallest with 16 alignments. $RV40$ and $RV50$ contain sequences that share a 20% similarity with at least one other sequence in the set. Overall, there are 218 alignments in BAliBASE 3.0.

In order to assess the accuracy of multiple sequence alignment programs, the alignment produced by the program in each BALiBASE test case is compared to the reference alignment. Two scores are used to evaluate the alignment (Edgar and Sjoelander, 2002) (For more see Section 2.3.2):

TABLE 3.1: Average SP score on the 18 test sets from BAliBASE 2.0.

		MOAFS	HMOABC	MO-SAStrE	MSA-GA w/p	VDGA_D3	VDGA_D4	VDGA_D2	GAPAM	RBT-GA	SAGA	MSA-GA
Rer1	1ped	**0.883**	0.732	0.716	0.482	0.451	0.443	0.498	n/a	0.687	n/a	0.501
	1uky	**0.643**	0.559	0.403	0.459	0.464	0.416	0.402	n/a	0.405	n/a	0.443
	2myr	**0.602**	0.466	0.544	0.590	0.282	0.347	0.317	n/a	0.302	n/a	0.212
	kimase	**0.836**	0.783	0.808	0.545	0.548	0.531	0.487	n/a	0.488	n/a	0.295
Rer2	1lvl	**0.923**	0.912	0.825	0.819	0.816	0.803	0.781	0.567	n/a	0.726	n/a
	1pamA	0.853	0.880	**0.913**	0.863	0.853	0.857	0.860	0.660	0.758	0.623	0.755
	1ubi	0.867	**0.925**	0.911	0.778	0.794	0.732	0.767	0.795	n/a	0.492	n/a
	1wit	**0.914**	0.855	0.417	0.815	0.774	0.875	0.851	0.825	n/a	0.694	n/a
	2hsdA	**0.918**	0.898	0.855	0.829	0.742	0.856	0.796	0.745	0.768	0.498	0.761
	2pia	**0.894**	0.893	0.879	0.850	0.839	0.847	0.826	0.730	n/a	0.763	n/a
	3grs	**0.885**	0.863	0.864	0.751	0.781	0.717	0.746	0.755	n/a	0.282	n/a
	4enl	0.903	**0.930**	0.912	0.889	0.899	0.890	0.896	0.812	n/a	0.739	n/a
Ref3	1ajsA	0.535	0.539	**0.586**	0.453	0.408	0.383	0.311	0.180	n/a	0.186	n/a
	1ubi	**0.751**	0.651	0.590	0.414	0.410	0.398	0.386	0.310	n/a	0.585	n/a
	1uky	0.656	**0.692**	0.673	0.481	0.526	0.469	0.468	0.350	n/a	0.269	n/a
	4enl	0.819	0.832	0.862	**0.866**	**0.866**	0.836	0.800	0.680	n/a	0.672	n/a
	kimase	0.868	0.874	**0.918**	0.890	0.887	0.870	0.825	n/a	0.619	n/a	0.580
Ref4	kimase1	**0.937**	0.928	0.865	0.542	0.478	0.330	0.384	n/a	0.635	n/a	0.710
	Overall	**0.816**	0.790	0.780	0.671	0.657	0.644	0.633	0.617	0.583	0.544	0.532

Best results is highlighted.

1) *SP* is the ratio of the number of correctly aligned pairs of positions in the test (predicted) alignment to the number of aligned pairs in the reference (structurally informed) alignment.

2) *CS* is the ratio of the number of correctly aligned columns in the test alignment to the number of aligned columns in the reference alignment.

Both SP and CS range from 1.0 for perfect agreement to 0.0 for no agreement.

3.3.2 Parameters Setting

In the proposed MOAFS algorithm, the population size (Artificial Fish Number) is 50, the maximum number of iteration is 1000. The size of the archive is 100. Each individual chooses randomly the distance and the segment between 0 and 1. The parameter *GOP* (gap opening penalty) and *GEP* (gap extending penalty) are 10 and 1, respectively.

3.3.3 Experimental Results and Analysis

In this section, we use two comparisons to see further efficiency of the proposed (MOAFS) algorithm that has been proved better by comparing it with various multiple sequence alignment methods.

MOAFS applied on BALiBASE 2.0

Firstly, we compare MOAFS with well-known evolutionary algorithms published in the literature applied to BALiBASE 2.0, which presents in Table 3.1 and 3.2, such as MO-SAStrE (Ortuño et al., 2013), GAPAM (Naznin, Sarker, and Essam, 2012), MSA-GA (Gondro and Kinghorn, 2007), RBT-GA (Taheri and Zomaya, 2009), SAGA (Notredame and Higgins, 1996), VDGA (Naznin, Sarker, and Essam, 2011), HMOABC (Rubio-Largo, Vega-Rodríguez, and González-Álvarez, 2016), MOMSA-W (Zhu, He, and Jia, 2016) and IMSA (Cutello et al., 2010).

In Table 3.1, we compare the quality of alignments (SP) obtained by the MOAFS algorithm and the aforementioned algorithms published in the literature. As we can see, from these eighteen alignments, the MOAFS achieved a total of eleven more accurate alignments against the other algorithms; whereas, HMOABC obtains the best results in only three and

TABLE 3.2: Alignment accuracies of MOAFS, MOMSA-W and IMSA on the BAliBASE 2.0.

	Ref1 (82)		Ref2 (23)		Ref3 (12)		Ref4 (12)		Ref5 (12)		Overall (141)	
	SP	CS	SP	CS	SP	CS	SP	CS	SP	CS	SP	CS
MOMSA-W	0.844	0.771	**0.925**	**0.557**	0.766	0.488	0.871	0.617	**0.936**	**0.802**	0.861 ± 0.181	0.702 ± 0.305
IMSA	0.834	0.653	0.921	0.413	**0.786**	0.362	0.730	0.319	0.730	0.319	0.821 ± 0.090	0.463 ± 0.069
MOAFS	**0.891**	**0.825**	0.916	0.532	0.778	**0.494**	**0.884**	**0.628**	0.923	0.770	$\mathbf{0.872 \pm 0.133}$	$\mathbf{0.710 \pm 0.293}$

Best results is highlighted.

MO-SAStrE is the same thing. Specifically, VDGA Decomp_4 and VDGA Decomp_3 are better than MOAFS in one case.

Moreover, Figure 3.8 presents an illustrative view of this comparison in which we can see the good performance of MOAFS in terms of minimum, average or maximum values of SP score. Furthermore, our swarm approach (MOASF) presents a strong advantage over both HMOABC and MO-SAStrE.

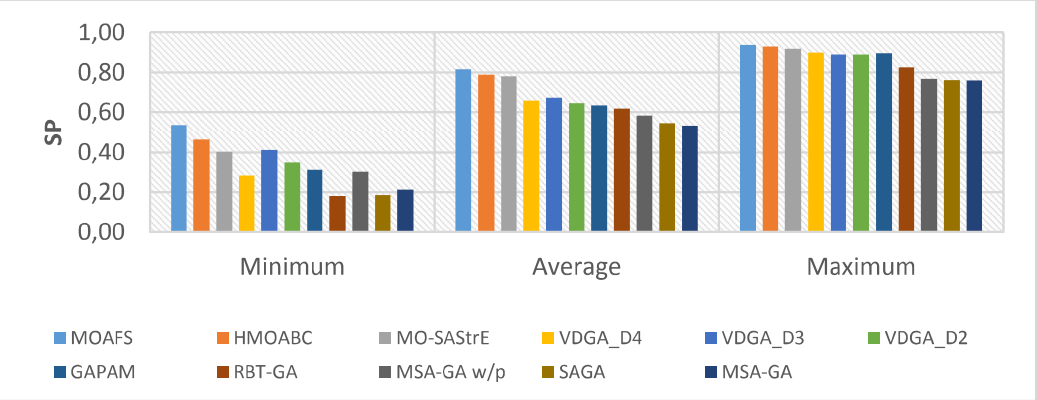


FIGURE 3.8: Comparison among MOAFS and other approaches published in the literature in terms of (Min, Max, average) SP-score test sets from BAliBASE 2.0.

In addition, Table 3.2 compares the results obtained by MOMSA-W, IMSA and MOAFS algorithms on all BALiBASE 2.0 alignments. In this table, we have highlighted the best results for each family in both measures (SP and CS). As we can see, MOAFS obtains the highest results in 2 out of 5 families in terms of SP, and in 3 out of 5 families in terms of CS. Furthermore, we focus on the last column of Table 3.2. (Overall), the best and second-best approaches are MOAFS and MOMSA-W, respectively; if we compare their overall results on SP and CS, we find a difference of more than $1.1 \pm 0.26\%$ in terms of SP and of more than $0.8 \pm 1.4\%$ in terms of CS.

Accordingly, in this first comparison, we can conclude that MOAFS can give better results in terms of alignment accuracy relative to both SP and CS, efficiency and robustness than other algorithms applied on the BAliBASE 2.0 benchmark.

MOAFS applied on BALiBASE 3.0

For completeness, MOAFS has been compared with the state-of-the-art alignment algorithms also on the BALiBASE 3.0 benchmark, such as HMOABC (Rubio-Largo, Vega-Rodríguez, and González-Álvarez, 2016), Clustal W (Thompson, Higgins, and Gibson, 1994), Clustal Ω (Sievers et al., 2011), DIALIGN-TX (Subramanian, Kaufmann, and Morgenstern, 2008),

TABLE 3.3: Average SP and TC scores on each BAliBASE3.0 subset.

	RV11		RV12		RV20		RV30		RV40		RV50		Overall	
	SP	CS	SP	CS	SP	CS	SP	CS	SP	CS	SP	CS	SP	CS
MOAFS	**0,7788**	**0,5935**	*0,9476*	*0,8719*	0,9202	*0,5008*	**0,8875**	**0,6463**	0,9246	*0,6538*	0,8995	**0,6345**	**0,893**	**0,6501**
HMOABC	*0,7473*	*0,5762*	**0,9506**	**0,8868**	**0,9355**	**0,5127**	*0,8868*	*0,6343*	**0,9384**	**0,6549**	0,8919	*0,6103*	*0,8918*	*0,6459*
Clustal W	0,5006	0,2299	0,8649	0,717	0,852	0,2216	0,725	0,2759	0,7894	0,3982	0,7424	0,3116	0,7457	0,359
Clustal W	0,5901	0,3622	0,906	0,7938	0,9116	0,4529	0,8624	0,5791	0,901	0,5826	0,862	0,5374	0,8389	0,5513
DIALIGN-TX	0,5047	0,2681	0,8821	0,7569	0,8781	0,3078	0,7614	0,389	0,834	0,4517	0,8215	0,4705	0,7803	0,4407
FSA	0,5027	0,2723	0,9238	0,8222	0,865	0,1899	0,6896	0,2629	0,8609	0,478	0,7895	0,4206	0,7719	0,4076
FSA-maxsn	0,6188	0,3658	0,9365	0,8479	0,901	0,3434	0,8137	0,4826	0,9161	0,5846	0,8719	0,5657	0,843	0,5317
Kalign2	0,6053	0,3687	0,9121	0,7934	0,9008	0,3625	0,8126	0,4799	0,8833	0,5078	0,8201	0,4396	0,8224	0,492
MAFFT-EINSi	0,66	0,4402	0,9361	0,839	0,9264	0,4515	0,8612	0,592	0,9143	0,5751	0,8991	0,5985	0,8662	0,5827
MAFFT-GINSi	0,6071	0,3466	0,927	0,8252	0,905	0,3913	0,8531	0,5322	0,8864	0,5156	0,8839	0,5499	0,8438	0,5268
MAFFT-LINSi	0,6712	0,45	0,9363	0,8423	0,9262	0,4574	0,8555	0,5733	0,9191	0,6009	*0,8999*	0,5661	0,868	0,5816
MSAProbs	0,6818	0,444	0,9463	0,8703	0,9283	0,4694	0,8646	0,6115	0,9232	0,6104	**0,9076**	0,6101	0,8753	0,6026
MUMMALS	0,6694	0,4197	0,943	0,8447	0,9062	0,4313	0,8479	0,4981	0,8714	0,4884	0,8791	0,5329	0,8528	0,5359
MUSCLE	0,6826	0,4409	0,9447	0,8619	*0,9284*	0,4794	0,8755	0,6227	*0,9254*	0,6039	0,8944	0,5931	0,8752	0,6003
ProbCons	0,6697	0,4196	0,9412	0,8605	0,9168	0,4116	0,8453	0,5473	0,9003	0,5361	0,8941	0,5789	0,8612	0,559
PRANK	0,4618	0,2162	0,8377	0,6208	0,8014	0,1242	0,5784	0,0638	0,7477	0,3422	0,6735	0,2099	0,6834	0,2628
ProbAlign	0,695	0,4569	0,9464	0,8669	0,9259	0,4444	0,853	0,5695	0,9221	0,6123	0,8886	0,5552	0,8718	0,5842
T-Coffee	0,6572	0,4141	0,9447	0,8593	0,9157	0,406	0,8369	0,4776	0,8964	0,5538	0,8949	0,5911	0,8576	0,5503

Best and second best results are highlighted.

FSA and FSA-maxsn (Bradley et al., 2009), Kalign2 (Lassmann, Frings, and Sonnhammer, 2008), MAFFT (Katoh et al., 2002) (L-INS-i, E-INS-i, and G-INS-i), MSAProbs (Liu, Schmidt, and Maskell, 2010), MUMMALS (Pei and Grishin, 2006), MUSCLE (Edgar, 2004), ProbCons (Do et al., 2005), PRANK (Löytynoja and Goldman, 2005), ProbAlign (Roshan and Livesay, 2006) and T-Coffee (Notredame, Higgins, and Heringa, 2000).

In Table 3.3, we compare the results obtained with approaches cited in (Rubio-Largo, Vega-Rodríguez, and González-Álvarez, 2016) and the MOAFS algorithm. As we can see, MOAFS obtains the highest results in 2 out of the 6 families in terms of SP, and in 3 out of the 6 families in terms of CS. In the overall comparison, our method gives good results compared to other methods. When we compare the two best approaches MOAFS and HMOABC. If we compare their overall results on SP, we find a difference greater than $0.13 \pm 0.56\%$, and in fact the same for CS, we find a difference greater than $0.43 \pm 1.01\%$.

Based on the above analysis, we can also conclude from this kind of comparison, that MOAFS can possibly provide comparable alignments in terms of quality (SP and CS) with other algorithms applied on the BAliBASE 3.0 benchmark.

3.4 Conclusion

In this chapter, we have presented a novel multi-objective swarm intelligence algorithm for the multiple sequence alignment problem. Our proposed method includes new techniques such as the center individual inspired by consensus, two specific operators of Block Shuffling, mutation and crossover operations. All these techniques have been successfully adapted to the Artificial Fish Swarm Algorithm.

We have used two well-known metrics to evaluate the accuracy of multiple sequence alignment methods: the sum-of-pairs score (SP) and the column score (CS). The results obtained with experiments conducted on the widely used BAliBASE 2.0 and BAliBASE 3.0 datasets (benchmarks) show that our proposed algorithm (MOAFS) is better than the following algorithms (MOSAStrE, HMOABC, SAGA, IMSA, ...) in terms of alignment accuracy i.e. the average score of the two accuracy measures SP and CS are better.

The experiments demonstrate that the most favorable feature of our proposed MOAFS resides in its ability to generate, for any MSA instance, more than one single sub-optimal alignment. This ability is due to the behavior of the cooperation, decentralization, and parallelism used by the MOAFS. The obtained results are consistent and encouraging, which will help biologists to evaluate and select the biologically relevant multiple sequence alignment.

In addition, MOAFS can also be applied directly to other types of sequential data sets or it can be extended to address other issues not yet tackled.

Chapter 4

Gene Selection Based On MIM-mMFA For Cancer Classification

4.1 Introduction

DNA Microarray is a modern biological research technology for analyzing gene expression. This technology (DNA Microarray) has the possibility to measure the expression levels of thousands of genes during important biological processes and gives the ability to diagnose cancer based on gene expression (Jeffrey, Lønning, and Hillner, 2005; Chang, Hilsenbeck, and Fuqua, 2005). Given the very high number of genes, it is useful to select a limited number of genes relevant to the classification of tissue samples. The analyzing expression profiles of multiple genes provides an opportunity to illuminate some aspects of functional genomics.

DNA microarray studies are used to discover which specific genes are important for the development of a disease. They are used to analyze gene expression data associated with a specific diagnosis. Computational analysis and computer science can help researchers assemble a group of gene signatures for a given disease (Buturović, 2005; Wang et al., 2004). However, gene expression data obtained from microarray are usually contained a very large number of genes and a few numbers of samples. To obtain a good classification accuracy, it is important to select the most pertinent genes that are needed and sufficient to describe the target concept. In addition, gene selection is also a procedure for reducing the input dimension, resulting in a much lower computational burden on the classifier (Mudaliar et al., 2015).

Many gene selection methods have been proposed and can be arranged into three main categories: filter, wrapper and embedded methods (Mundra and Rajapakse, 2010; Du et al., 2014) (see Section 2.4.3).

The challenge of high-dimensional data analysis is to extract information about the disease from a huge amount of redundant data and noise. It is important to select a small subset of microarray data for accurate classification. All this motivates researchers to search for possible solutions and to propose various algorithms. In order to obtain the best accuracy in gene selection problem and to achieve this goal, various evolutionary and bio-inspired algorithms have been applied.

In the literature, we find several approaches that solve this issue, among them, the GA/SVM has been proposed as a hybrid approach between the Genetic Algorithm (GA) and Support Vector Machine (SVM) for the classification of high dimensional microarray data. This approach uses the Fuzzy logic based pre-filtering technique and GA with an SVM classifier as fitness (Huerta, Duval, and Hao, 2006). Gene Selection Programming (GSP) proposed in (Alanni et al., 2019), is based on Gene Expression Programming (GEP) method, and the main goal is to select relevant genes for efficient cancer classification. The work reported in (Ghosh et al., 2019) proposed two simple models-ensemble of filter rankings along with GA.

Moreover, PCC-BPSO/GA was proposed as a combination between Pearson's Correlation Coefficient (PCC) and Binary Particle Swarm Optimization (BPSO) or Genetic Algorithm (GA) (Hameed et al., 2018b), Multi-objective version of Bat algorithm for binary feature selection (Dashtban, Balafar, and Suravajhala, 2018) and Genetic Bee Colony (GBC) algorithm (Alshamlan, Badr, and Alohali, 2015) were efficaciously used in high dimensional datasets for selection and classification. A modified Artificial Bee Colony Algorithm (mABC) (Moosa et al., 2016) was proposed for the gene selection problem, to select a minimum number of genes for predictive accuracy. Another proposition, a modified Support Vector Machine (SVM) was proposed to select the minimum possible genes (Ghaddar and Naoum-Sawaya, 2018). In (Wang et al., 2013) authors have proposed a Chi-square-statistic-based Top Scoring Genes (Chi-TSG or TSG), it starts with the top two genes and sequentially adds an additional gene into the candidate gene set to perform an informative gene selection. Another study (Aziz et al., 2017) was proposed an approach for Artificial Neural Networks (ANNs) classification of high-dimensional microarray data.

In another method called MIM-AGA, a hybrid feature selection algorithm was proposed to combine the Mutual Information Maximization (MIM) and the Adaptive Genetic Algorithm (AGA) (Lu et al., 2017). Genetic Bee Colony (GBC) proposed a new hybrid gene selection method that combines the use of a Genetic Algorithm (GA) along with the Artificial Bee Colony (ABC) algorithm. The objective of this combination is to integrate the advantages of both algorithms (Alshamlan, Badr, and Alohali, 2015). (Chen, Zhang, and Gutman, 2016) proposed an approach called Kernel-Based Clustering method for Gene Selection (KBCGS). This approach is based on a clustering method and uses adaptive distances that change at each iteration step. It searches the best weights of genes and optimizes the clustering objective function. This algorithm has been modified by (Liu et al., 2018) who used double RBF-kernels with weighted analysis to implement DKBCGS (Double-kernel KBCGS).

Moreover, many bio-inspired algorithms have been proposed in the literature in recent years. Moth Flame Optimization Algorithm (MFOA) was one of the most recent bio-inspired methods that quickly became well-known for its superior performance in solving many optimization problems. However, to our knowledge, this is the first attempt at applying the MFOA-based as a gene selection and classification of DNA Microarray Data. MFOA was generally used for its guaranty transition between exploration and exploitation, faster convergence and better time complexity.

In this chapter, we propose a new algorithm called Mutual Information Maximization - modified Moth Flame Algorithm (MIM-mMFA). MIM-mMFA is proposed to investigate and improve the performance of gene selection. For this purpose, we propose a hybrid model

that uses several techniques: Mutual information maximization (MIM),MFOA combined with a Support Vector Machine (SVM) with Leave One Out Cross Validation (LOOCV), and the number of selected genes. Our approach has three particular features. First, for high dimensional data, we use a pre-processing MIM to handle this difficulty. Second, mMFA uses three procedures, one for the presentation of individual, another for moth movement and the last is a new fitness function (an SVM with LOOCV classifier). Finally, the main objective of mMFA is to select the best gene subset from among the predictive gene subsets, which is evaluated with the SVM classifier to provide high classification accuracy. For the performance of our proposal, we apply an SVM classifier to the best subset of genes selected by the MIM-mMFA. In order to test the accuracy of the proposed methodology, we used sixteen binary class and multi-class gene expression microarray datasets and compared our results with those obtained with other recently published algorithms. Experimental results showed that the MIM-mMFA achieves better performance of classification accuracy with a competitive average of the number of selected genes for solving the gene selection problem in both binary class and multi-class.

The remainder of the chapter is organized as follows: The next Section4.2 describes briefly the proposed method. Section 4.3 presented experimental results and discussion. Finally, the conclusion of this chapter is given in Section 4.4.

4.2 The Proposed Algorithm

In this section, we propose a new MIM-mMFA algorithm for predictive gene selection for cancer classification. This work is based on a hybrid approach between mutual information maximization and a bio-inspired algorithm, which is Moth-Flame Optimization Algorithm (MFOA), the principle of our proposed algorithm consists of two stages. Firstly, preprocessing begins by normalization of microarray data with the Min-Max method can guarantee a stable convergence of weights and biases (Jain, Nandakumar, and Ross, 2005). We also used the statistical technique of maximized the mutual information (MIM) to measure the relevance and the redundancy of the selected genes, in order to reduce the high number of genes by eliminating genes redundancy.

Secondly, the modified Moth Flame Optimization Algorithm (mMFA) was applied to the dataset provided by MIM. The basic idea is to integrate SVM with LOOCV classifier as the fitness function in the mMFA (to evaluate all population individuals). The objective of this integration is to select a high accuracy genes subset that contains a smaller number of genes. Finally, the better gene subset, obtained by MIM-mMFA, will be evaluated with the SVM classifier. The general scheme of MIM-mMFA approach is shown in Figure 4.1.

4.2.1 Pre-selection (Preparation-Normalization)

This stage consists of two phases: the first normalizes the dataset and the second is based on a tool to reduce the number of genes. The following explains in detail these two operations.

Normalization

One of the most essential steps in the pre-selection of genes expression data is the normalization. Normalization is indispensable in the earliest stage of microarray data analysis because we have a lot of reasons, among them the starting quantity is not the same in gene expression data, differences in labeling or detection efficiency between the fluorescent dyes used and systematic biases in the measured expression levels. Normalization is the process of removing the units of measurement for data, reducing the training error and allowing us to easily compare data.

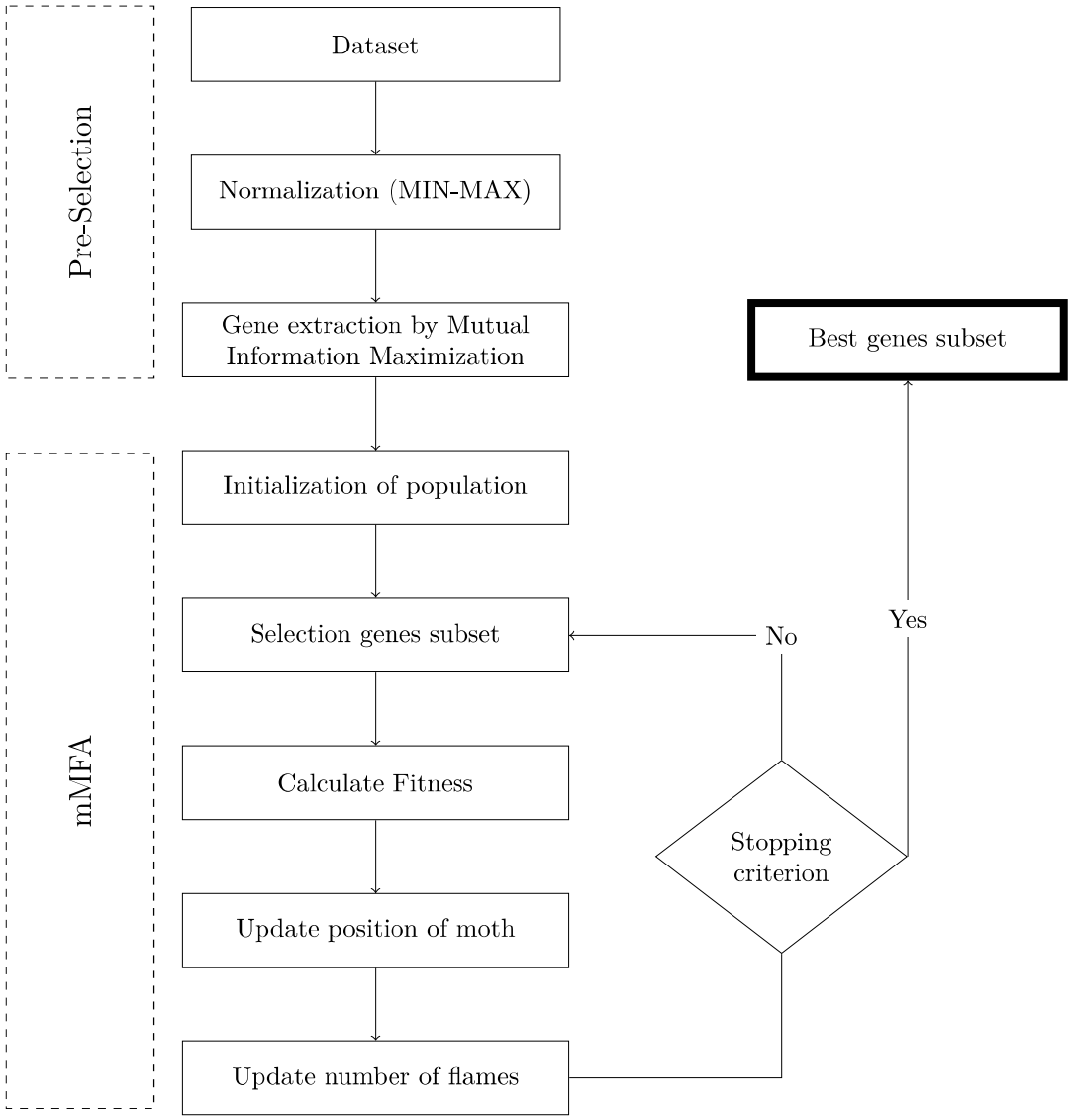


FIGURE 4.1: The main phases of MIM-mMFA.

Some of the more common ways to normalize data using statistical models, including standardization, are to transform data using a z-score or t-score (Crawford and Howell, 1998), feature scaling is re-scaling data to obtain all values in an interval $[a, b]$. In this study, we used a feature scaling to normalize each gene value between $[-1, 1]$ as shown in the following Eq. 4.1.

$$X_{new} = (b - a)\frac{X - X_{min}}{(X_{max} - X_{min})} + a \tag{4.1}$$

where X is the original value, X_{min} is the minimum original value, X_{max} is the maximum original value, and X_{new} is the normalized expression level.

Why Do We Use Mutual Information?

In wrapper methods, the large number of genes causes great computational complexities when searching for significant genes, it is difficult and impractical for a classifier to be accurately trained. On the other hand, we must use a pre-selection method to select a subset of genes containing a smaller number of informative genes to reduce the complexity. This method based on certain filtering criteria.

In the literature, several methods can be used to pre-process data. These include maximum entropy, or Kullback-Leibler divergence (Burnham and Anderson, 2001), Euclidean Distance (Hu et al., 2006), Student's t-test (Press et al., 1992), are very popular parametric filter methods. Also, these are Information Gain (IG) (Wang, 2005), Bhattacharyya coefficient (Guorong, Peiqi, and Minhui, 1996), BW ratio (Dudoit, Fridlyand, and Speed, 2002), another statistical test that can be used as a filter is the Wilcoxon rank-sum test (Wilcoxon, 1992), Kruskal-Wallis test is an extended to Wilcoxon's test (Breslow, 1970) etc.

In this study, we used maximizing the mutual information (Torkkola, 2003) that can be explained as an information potential induced by samples of data in different classes.

Our choice (MI) respected the three components of the information force:

1) Samples within the same class attract each other,

2) All samples regardless of class attract each other, and

3) Samples of different classes repel each other.

Mutual Information Maximization

Mutual Information (MI) of two random variables is a measure of the amount of information between the two variables (Venkateswara et al., 2015). We use these methods to reduce the size of the initial problem, in other words, to eliminating gene redundancy. In the context of the feature selection, it provides a metric to quantify the relevance of a feature subset respecting to the output vector, noted to C.

Formally, the MI is given by Eq. 4.2:

$$I(X,Y) = \sum_{x \in S} \sum_{x \in T} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)} \tag{4.2}$$

where $p(x)$ and $p(y)$ are probability density functions of variable x, y, respectively. $p(x,y)$ is the joint probability density.

Consider expression data there are n samples and each one has m genes, the data can be represented by the matrix T of dimension NM. Let g_{ij} denote the expression level of the j^{th} gene in the i^{th} sample.

Denote $G = \{G_1, G_2, .., G_m\}$, a vector of vectors where m is the number of genes and each gene $G_j = \{g_{1j}, g_{2j}, .., g_{nj}\}$ is a vector of gene expressions, for the j^{th} gene where n is the sample size.

Let $S = \{S_1, S_2, .., S_N\}$ be the class labels for the N samples, where S_k takes one of the values in the set of all possible class $C = \{C_1, C_2, .., C_M\}$. Where $I(G_x, C)$ represents the value of mutual information between an individual gene G_x that belongs to G and class C. For example, the output vector of Prostate Tumors $C = \{c_1, c_2\}$, where c_1 and c_2 denote the Normal and Tumor classes.

Suppose that:

- ☞ M: represents the number of genes in the dataset
- ☞ α: represents the number of genes with genes expression profile t in class c.

☞ β: represents the number of genes with genes expression profile t not in class c.

☞ δ: represents the number of genes without genes expression profile t in class c.

$I(t,c)$: is mutual information of t of class c, it is calculated as follows (Eq. 4.3).

$$I(t,c) = \log \frac{p(t \setminus c)}{p(t)} = \log \frac{p(t,c)}{p(t)p(c)} \approx \log \frac{\alpha M}{(\alpha+\beta)(\alpha+\delta)} \tag{4.3}$$

During the application of Eq.4.3, if the gene expression profile t is irrelevant to the class c, $I(t,c) = 0$. The maximum mutual information can be expressed as (Eq. 4.4):

$$MaxMI(t) = \sum_{i=1}^{k} p(C_i \setminus t) \log \frac{p(C_i \setminus t)}{p(C_i)} \tag{4.4}$$

where k represents the number of classes in the dataset. The Mutual Information Maximization (MIM) method is extremely easy to implement and fast to run. The principle of the MIM method is each score $MaxMI(g_j)$ of gene g_j $(j = 1, ..., m)$ is calculated independently of all other genes in the same class by Eq. 4.4. Afterward, the genes are ranked in descending order, and the most ordered are selected to obtain a new subset called the MIM dataset, as shown in Figure 4.2. The objective of maximizing mutual information is to find genes that are highly dependent on all other genes in the same class.

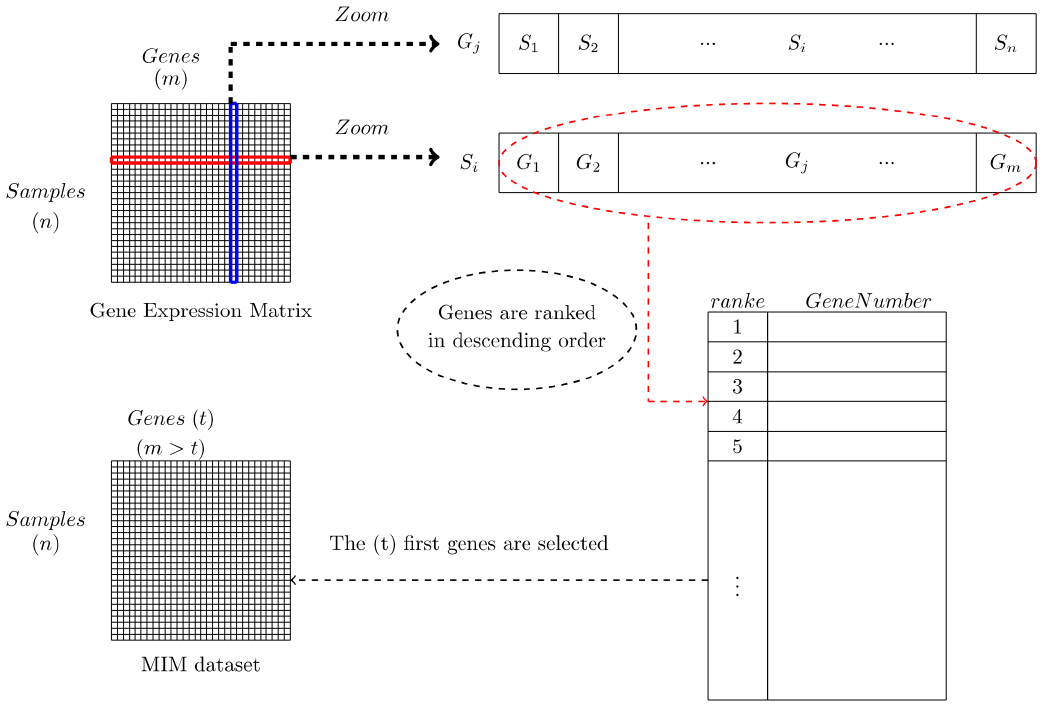


FIGURE 4.2: A gene expression matrix and Pre-filter of MIM.

4.2.2 Modified MFO Algorithm for Genes Expression Selection

Moth Flame Optimization Algorithm (MFOA) has been proposed by (Mirjalili, 2015) that is a bio-inspired algorithm (see Section 2.2.3).

In the following steps, we highlight the main phases of our modification on MFOA to solve the gene selection problem. This new proposal, we called it the modified Moth Flame Optimization Algorithm (mMFA).

Step 1 *Representation of candidate solutions and initialization of population* – In our work, the moths can fly in two-dimension ($2D$), considered as a candidate solution, the flames are the best position of moths that obtain so far, these both represented by two matrices ($n \otimes d$), see Eq.4.5, each matrix has n number of moths and d number of possible positions, in our proposition, the moth (individual) represents the subset of genes, and the number of positions represents the maximum number of genes in an individual. Also, each position has two coordinates one for the x-axis and another for the y-axis. In this study, we called the subset of genes by moth or individual.

$$M = \begin{bmatrix} (m^x_{1,1}, m^y_{1,1}) & (m^x_{1,2}, m^y_{1,2}) & \cdots & \cdots & (m^x_{1,d}, m^y_{1,d}) \\ (m^x_{2,1}, m^y_{2,1}) & (m^x_{2,2}, m^y_{2,2}) & \cdots & \cdots & (m^x_{2,d}, m^y_{2,d}) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ (m^x_{n,1}, m^y_{n,1}) & (m^x_{n,2}, m^y_{n,2}) & \cdots & \cdots & (m^x_{n,d}, m^y_{n,d}) \end{bmatrix}$$

$$F = \begin{bmatrix} (f^x_{1,1}, f^y_{1,1}) & (f^x_{1,2}, f^y_{1,2}) & \cdots & \cdots & (f^x_{1,d}, f^y_{1,d}) \\ (f^x_{2,1}, f^y_{2,1}) & (f^x_{2,2}, f^y_{2,2}) & \cdots & \cdots & (f^x_{2,d}, f^y_{2,d}) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ (f^x_{n,1}, f^y_{n,1}) & (f^x_{n,2}, f^y_{n,2}) & \cdots & \cdots & (f^x_{n,d}, f^y_{n,d}) \end{bmatrix} \tag{4.5}$$

The first step in the mMFA is to initialize a population of two kinds of categories, sets of moths and flames. In fact, no existing initialization strategies have been proposed specifically for the gene selection problem. Our proposition starts with a random initialization of the moth sets population, each position moth presented by two coordinates (defined by $(m^x_{i,j}, m^y_{i,j})$ for the i^{th} moth and the j^{th} position). The initialization of flame sets will be done in the first iteration, the positions of flames equal to the position of moths (flames are the flags which dropped by moth during the search process and move around that position and update accordingly). In other words, we put the matrix F equal to the matrix M.

Step 2 *Gene subset selection and fitness calculation for each moth (individual)* – In the mMFA algorithm, the individuals are represented by continued values. In this step, we used the *Gene_select* function to determine which genes are selected in this individual (subset). Our algorithm is applied to gene selection problems, the length of the individual is d, where d is the number of genes selected by MIM. Each individual in the population is encoded using a vector of d real numbers. We used a *sigmoid* function, for each distance $D_{i,j}$ between the j^{th} position in the i^{th} moth, $sigmoid(D_{i,j})$, is usually in the interval $[0,1]$ (Eq. 4.7). In order to determine whether the gene will be selected or not, a threshold $0 < \vartheta < 1$ is needed to compare with the value of *sigmoid* in the distance vector. The threshold ϑ is given by function $random(0,1)$. If $sigmoid(D_{i,j}) > \vartheta$, then the j^{th} gene is selected in individual i. Otherwise, the gene is not selected. The formula of the equation and the pseudo-code of *Gene_select* function are given by Eq.4.6 and Algorithm 4.1, respectively.

$$GenSel(D_{i,j}) = \begin{cases} true \ if sigmoid(D_{i,j}) > 0 \\ false \qquad\qquad Otherwise \end{cases} \tag{4.6}$$

and

$$sigmoid(x) = \frac{1}{(1+e^{-x})} \ \wedge \ sigmoid(x) \in [0,1] \tag{4.7}$$

After, starting the evaluation of each individual by the fitness function, this function is used as an evaluator to select the best gene subsets. Here, the fitness of each individual gives

Algorithm 4.1 Genes selection pseudo-code

```
1: function Gene_select(i, NbrGenes, ϑ)        ▷ Where i: number of the individual
2:     SG[NbrGenes] array of Boolean
3:     for (j ← 1 to NbrGenes) do
4:         D ← sqrt((m^x_{i,j})^2 + (m^y_{i,j})^2)
5:         if sigmoid(D) > ϑ then
6:             SG[j] ← true
7:         else
8:             SG[j] ← false
9:         end if
10:    end for
11:    return SG
12: end function
```

the quality of a subset of genes. This quality depends on two measures: the smallest subset of genes and the highest accuracy. In our fitness function, we used the SVM classifier and the number of selected genes. We define the fitness value of each individual (moth i) as follows (Eq 4.8):

$$Fitness_i = \begin{cases} w_1 * Acc_{SVM\_with\_LOOCV}(Moth_i) + w_2 * \frac{t_g - s_g}{t_g} \\ With \quad w_1 + w_2 = 1 \end{cases} \tag{4.8}$$

where $Acc_{SVM\_with\_LOOCV}$ is the accuracy of SVM with LOOCV classifier, t_g is the number of total genes in the dataset, s_g is the number of selected genes in the individual (moth i), w_1 and w_2 are the coefficients of each part of fitness. The first part of the fitness represents the accuracy SVM with LOOCV classifier and the second part represents the percentage of genes that are not selected. In our fitness, w_1 and w_2 give the balance between accuracy and the number of selected genes. When the value of accuracy is more important than the number of selected genes, w_1 takes the biggest value.

Why do we use SVM with LOOCV?

Furthermore, we apply in our fitness the SVM with leave-one-out cross-validation (LOOCV) (Ng et al., 1997), in order to evaluate the performance of each individual. The predictive accuracy of the gene subset is calculated by an SVM with LOOCV classifier. The best gene subset is who has a higher SVM with LOOCV classification accuracy. LOOCV is a special case of k-fold cross-validation, where k is chosen as the total number of examples. In LOOCV, we get test error estimates with lower bias and higher variance, because each training set contains $N - 1$ examples, which means that we use almost all training set in each iteration. This also leads to higher variance, as there is a lot of overlap between training sets. Therefore, test error estimates are highly correlated, which means that the mean value of the test error estimate will have a higher variance. We conclude, from the above, that LOOCV is very suitable for our problem because it has the ability to prevent the "over-fitting" problem (Ng et al., 1997).

Step 3 *Update moth position* – In our study, we used three techniques to define the next moth position or moth movement (displacement) operation: i) Cartesian coordinates, ii) polar coordinates and iii) Archimedes spiral function. However, we utilized the following conditions in our movement:

- ☞ The initial point of the spiral should start from the moth.
- ☞ The final point of the spiral should be the position of the flame.

☞ The fluctuation of the range of spiral should not exceed from the search space.

Firstly, the Archimedes spiral is a curve described by a point moving uniformly on a rotation line itself around a point which can also be expressed as a simple polar equation. It is represented by the Eq.4.9:

$$\rho = a.\theta \tag{4.9}$$

where a indicates the distance of the i^{th} moth for the j^{th} flame, i.e., the Euclidean distance between the i^{th} moth and the j^{th} flame divided by $2k\pi$, and θ is a random angle in $[0, 2k\pi]$.

We define by the Eq.4.9, of the Archimedes spiral, the distance between the next position of the i^{th} moth and the j^{th} flame ($\theta = 0$ is the closest position of the flame and $\theta = 2k\pi$ is the furthest) (see Figure 4.3).

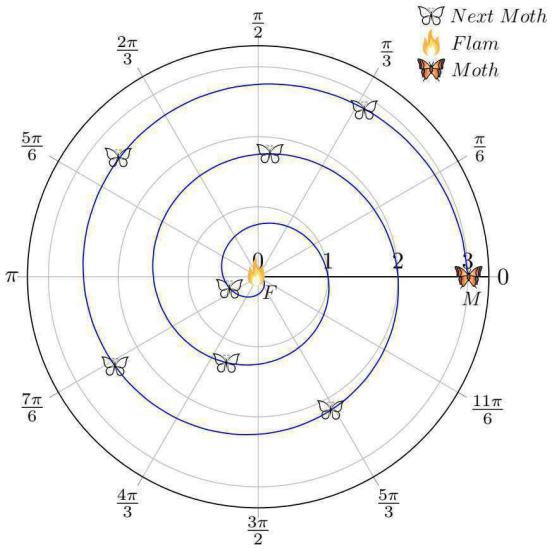


FIGURE 4.3: Logarithmic Archimedes spiral, space around a flame, and some next possible moth positions (when k=3).

The next moth coordinates are calculated by the following Eq.4.10:

$$\begin{cases} X = \rho\cos(\theta + \alpha) + X_F \\ Y = \rho\sin(\theta + \alpha) + Y_F \end{cases} \tag{4.10}$$

where ρ is the distance between the next position of i^{th} moth and the j^{th} flame given by the Archimedes spiral function. α is the angle between the line (FM) and the x-axis in the landmarks R. θ is the angle given by the Archimedes spiral function.

Proof – This section shows how to calculate the next moth position by a geometric method.

We assume that we have three landmarks $R = (xOy)$, $R' = (x'Fy')$ and $R'' = (x''Fy'')$ (see Figure 4.4)

All coordinates moths or flames have two-dimensions ($2D$) presented by two matrices M and F, are considered in the same orthonormal landmark R (x-axis and y-axis).

All moths and flames have each one two-dimensions ($2D$) coordinates as shown by two matrices M and F, we described in Step 1 of Section 4.2.2. These $2D$ coordinates are all expressed in the same orthonormal landmark R.

We express the i^{th} moth (M) and the j^{th} Flame (F) by the notations (X_M, Y_M) and (X_F, Y_F), respectively.

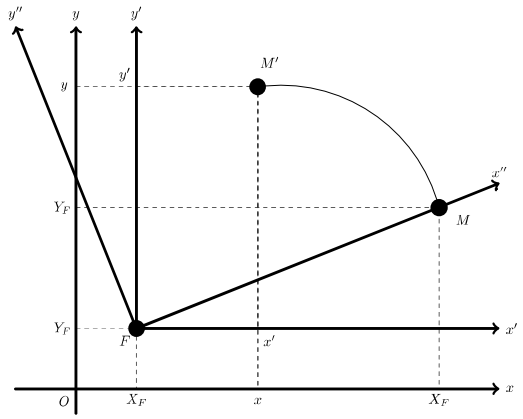


FIGURE 4.4: Three landmarks represented in the same space.

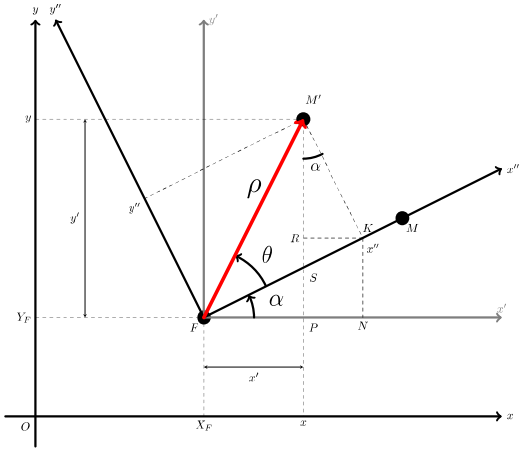


FIGURE 4.5: Algebraic representation of the next moth in three landmarks.

Let M' the next position of the i^{th} moth (M) for the j^{th} flame (F) in the plan. M' has respectively the following coordinates in the different landmarks R, R' and R'' :

We have :

in the landmark R : $\overrightarrow{OF} = X_F \overrightarrow{i} + Y_F \overrightarrow{j}$ and $\overrightarrow{OM'} = x' \overrightarrow{i} + y' \overrightarrow{j}$

and in the landmark R' : $\overrightarrow{FM'} = x' \overrightarrow{i} + y' \overrightarrow{j}$

According to the Chasles' Relation (Bell, 2014) we have:

$\overrightarrow{OM'} = \overrightarrow{OF} + \overrightarrow{FM'} = X_F \overrightarrow{i} + Y_F \overrightarrow{j} + x' \overrightarrow{i} + y' \overrightarrow{j}$

$\overrightarrow{OM'} = (X_F + x') \overrightarrow{i} + (Y_F + y') \overrightarrow{j}$

by the uniqueness of the coordinates of a vector in a landmark, we have Eq.4.11 (see Figure 4.5):

$$\begin{cases} x = X_F + x' \\ y = Y_F + y' \end{cases} \tag{4.11}$$

The rotation by (α) angle of landmark R' gives the landmark R'', we also can calculate α by Eq.4.12:

$$\tan\alpha = \frac{Y_F - Y_M}{X_F - X_M} \tag{4.12}$$

According to that rotation, from Figure 4.5, we have : $x' = FN - FP$ and $y' = M'R + RP$

From two triangles we have :

$FNK :\Rightarrow \quad (FN = x'' \cos\alpha)$ AND $(NK = RP = x'' \sin\alpha)$

$M'RK :\Rightarrow \quad (M'R = y'' \cos\alpha)$ AND $(RK = PN = y'' \sin\alpha)$

From all these equalities, we conclude Eq.4.13:

$$\begin{cases} x' = x'' \cos\alpha - y'' \sin\alpha \\ y' = x'' \sin\alpha + y'' \cos\alpha \end{cases} \tag{4.13}$$

And from the polar coordinates given by the Archimedes spiral function we have (see Figure 4.3):

$$\begin{cases} x'' = \rho \cos\theta \\ y'' = \rho \sin\theta \end{cases} \tag{4.14}$$

From Eq.4.11, 4.13 and 4.14 we deduce coordinates of M' in the landmark R

$$\begin{cases} x = \rho\cos\theta\cos\alpha - \rho\sin\theta\sin\alpha + X_F \\ y = \rho\cos\theta\sin\alpha + \rho\sin\theta\cos\alpha + Y_F \end{cases} \quad OR \quad \begin{cases} x = \rho\cos(\theta+\alpha) + X_F \\ y = \rho\sin(\theta+\alpha) + Y_F \end{cases} . \tag{4.15}$$

Step 4: *Update number of flames* – In order to maintain the exploitation of the best promotional solutions of our proposition, it is preferable to reduce the number of flames in each iteration because the moth can update its position among S different possible positions in the search space. To resolve this issue, several adaptive mechanisms are proposed in the literature to reduce gradually the number of flames over the course of iterations, in our work we used the following formula proposed by (Mirjalili, 2015).

$$Nbr\_Flame = round(Max_{Nbr\_F} - C_{it} * \frac{Max_{Nbr\_F} - 1}{Max_{it}}) \tag{4.16}$$

Where C_{it} is the current number of iterations, Max_{it} represents the maximum number of iterations and Max_{Nbr_F} is the maximum number of flames. When we use this formula to update the number of flames, in the final stage of the iteration, each moth updates its position only for the best flame. However, the gradual decrement in the number of flames to ensure a good trade-off between the exploration and the exploitation of the search space of genes expression selection problem.

Step 5: *Stopping Criterion is satisfied* – If a maximum number of iteration is reached, the iterative process stops for extract and evaluate the best subset of genes. Otherwise, return to Step 2.

The pseudo-code of the mMFA algorithm is described in Algorithm 4.2.

4.3 Results and Discussion

Our algorithm is an iterative method described by two phases. During the first phase, the microarray expression data was normalized and reduced by the Min-Max and MIM methods, respectively. The second phase is based on a bio-inspired algorithm called mMFA, which we have modified to achieve an optimal gene subset and used the SVM with LOOCV as the fitness function to confirm that the whole dataset is used in the training and testing phases. Our MIM-mMFA runs multiple times to obtain the most discriminating genes in each dataset. In order to assess the experimental results, the performance of the proposed MIM-mMFA genes selection and classification is applied to binary and multi classes high-dimensional microarray cancer datasets. For the cancer classification of each gene selection approach, we have used two comparison parameters : classification accuracy and the number of predicted genes.

Accuracy is one of the evaluation criteria for model classification. Classification accuracy is the overall rightness of the classifier, and it is calculated as the sum of the true correct cancer classifications divided by the total number of classifications (the proportion of correct results among the total number of examined cases). The classification accuracy is computed according to Eq. 4.17.

$$Classification\_Accuracy = \frac{CC}{N} 100 \tag{4.17}$$

Algorithm 4.2 Pseudo-code of the mMFA

for $(i \leftarrow 1$ **to** $Pop\_Size)$ **do**
  $Random\_initialization(P[i])$
end for
$Iteration \leftarrow 1$
repeat
  Nbr_Flames is calculated using (Eq. 4.16)
  for $(i \leftarrow 1$ **to** $Pop\_Size)$ **do**
    $SG_M[i] \leftarrow selection - genes(P[i])$
    $FM[i] \leftarrow$Evaluate $PM[i]$ to fitness function using (Eq. 4.8)
  end for
  if $Iteration = 1$ **then**
    $PF, FF, Subset\_F \leftarrow sort(PM, FM, Subset\_M);$
  else
    $PF, FF, Subset\_F \leftarrow sort([PF, PM], [FF, FM], [Subset\_F, Subset\_M]);$
  end if
  for $(i \leftarrow 1$ **to** $Pop\_Size)$ **do**
    for $(j \leftarrow 1$ **to** $Nbr\_Flames)$ **do**
      Calculate distance (ρ) between the i^{th} moth and the j^{th} flame using (Eq. 4.9)
      Update the moth position using (Eq. 4.15)
    end for
  end for
  $Subset_{best} \leftarrow PF[0], FF[0], Subset\_F[0]$
  $Iteration \leftarrow Iteration + 1$
until $(Iteration > Max\_iteration)$
return $Subset_{best}$

where N is the total number of the instances in the initial microarray dataset and CC refers to correct classified instances. Moreover, we have compared our algorithm with the well-known and recent algorithms in the literature that used the same datasets and discussed the results in the next subsections.

4.3.1 Dataset

There are two types of microarray datasets presented in the literature: binary class and multi-class datasets. In this study, we have used sixteen (16) gene expression datasets for different types of diseases. The binary-class microarray datasets are CNS, Colon, Leukemia1, Breast, Ovarian, DLBCL, and Prostate_Tumor. We also tested in this experiment other multi-class microarray datasets: Leukemia2, Lung_cancer, Lymphoma, MLL, SRBCT, Brain_Tumors1, Brain_Tumors1, 9_Tumors, and 11_Tumors. Table 4.1 displays in details of different used microarray datasets.

In Binary class, the first is an CNS (Central Nervous System) (Zhu, Ong, and Dash, 2007) was obtained from 60 samples composing of 7129 genes, which include 39 from Medulloblastoma Survivors (MS) and 21 from Treatment Failures (TF). The colon cancer contains 6000 genes with 62 samples (40 samples are tumor and 22 are normal) originally analyzed by (Alon et al., 1999). The Leukemia1 cancer dataset (Golub et al., 1999) was obtained from Acute Lymphoblastic Leukemia1 (ALL), and Acute Myeloid Leukemia1 (AML). Among them, 47 samples are from ALL and 25 samples are from AML. This dataset including 7129 genes. The Breast (Zhu, Ong, and Dash, 2007) was obtained from non-relapse and relapse. The total samples of this dataset are 97 distributed 51 non-relapse and 46 relapse samples.

TABLE 4.1: Summary of gene expression datasets.

Dataset Name	Samples	Features	Classes	Notes	Source
CNS	60	7129	2 (Binary class)	'MS': 39, 'TF': 21	(Zhu, Ong, and Dash, 2007)
Colon	62	2000	2 (Binary class)	'Tumor': 40, 'Normal': 22	(Alon et al., 1999)
Leukemia1	72	7129	2 (Binary class)	'ALL': 47, 'AML': 25	(Golub et al., 1999)
Breast	97	24481	2 (Binary class)	'non-relapse': 51, 'relapse': 46)	(Zhu, Ong, and Dash, 2007)
Ovarian	253	15154	2 (Binary class)	'Cancer': 162, 'Normal': 91	(Petricoin III et al., 2002)
DLBCL	77	5469	2 (Binary class)	'DLBCL': 57, 'FL': 19	(Shipp et al., 2002)
Prostate_Tumor	102	10509	2 (Binary class)	'Normal': 52, 'Tumor': 50	(Singh et al., 2002)
Lymphoma	66	4026	3 (Multi class)	'DLBCL': 46, 'CLL': 11, 'FL': 9	(Alizadeh et al., 2000)
MLL	72	12582	3 (Multi class)	'AML': 28, 'ALL': 24, 'MLL': 20	(Zhu, Ong, and Dash, 2007)
Leukemia2	72	7129	3 (Multi class)	'B-cell': 38, 'AML': 25, 'T-cell': 9	(Armstrong et al., 2001)
SRBCT	83	2308	4 (Multi-class)	'EWS': 29, 'RMS': 25, 'NB': 18, 'BL': 11	(Khan et al., 2001)
Brain_Tumors2	50	10367	4 (Multi-class)	'1': 15, '2': 14, '3': 13, '4': 7	(Nutt et al., 2003)
Brain_Tumors1	90	5920	5 (Multi-class)	'1': 59, '2': 10, '3': 10, '4': 6, '5': 4	(Pomeroy et al., 2002)
Lung_cancer	203	12600	5 (Multi-class)	'A': 139, 'SQ': 21, 'P': 20, 'N': 17, 'SM': 6	(Bhattacharjee et al., 2001)
9_Tumors	60	5726	9 (Multi-class)	{'1': 8, '2': 8, '3': 8, '4': 8, '5': 7, '6': 6, '7': 6, '8': 6, '9': 2}	(Staunton et al., 2001)
11_Tumors	174	12533	11 (Multi class)	{'1': 27, '2': 26, '3': 25, '4': 23, '5': 14, '6': 14, '7': 12, '8': 11, '9': 8, '10': 7, '11': 6}	(Su et al., 2001)

The Ovarian dataset (Petricoin III et al., 2002) consists of 253 (162 are Cancer and 91 are Normal) Microarray experiments (samples) with 15154 genes. DLBCL dataset (Shipp et al., 2002) has 5469 genes and 77 samples, divided into two classes, 58 from Diffuse Large B-Cell Lymphomas (DLBCL) and 19 from Follicular Lymphomas (FL). Prostate_Tumor dataset (Singh et al., 2002) was obtained from 102 samples (52 normal and 50 tumor) where each sample has 10509 genes.

The number of classes in Multi-class microarray datasets is 3, 4, 5, 9, or 11 and the number of genes varies from 2308 to 12600. We have divided all multi-class datasets we have worked on in four groups.

The first group contains Lymphoma, MLL and Leukemia2 datasets each one has three classes. The number of genes is 4026, 12582 and 7129, respectively. The Lymphoma (Alizadeh et al., 2000) was obtained 66 from samples divided into 46 from the Diffuse Large B-Cell Lymphoma (DLBCL), 11 from Chronic Lymphocytic Leukemia (CLL) and 9 from Follicular Lymphoma (FL). MLL (Zhu, Ong, and Dash, 2007) contains 28 samples from Acute Myeloid Leukemia (AML), 24 from Acute Lymphoblastic Leukemia (ALL), and 20 Mixed-Lineage Leukemia (MLL). The Leukemia2 (Armstrong et al., 2001) obtains after the modification of Leukemia1, Class ALL is divided into two classes, namely B-cell ALL and T-cell ALL. The 47 samples of ALL divided into 38 from B-cell ALL and 9 from T-cell ALL.

The Small Round Blue-Cell Tumor (SRBCT) microarray dataset (Khan et al., 2001), there are four classes : 29 samples are from Ewing's Sarcoma (EWS), 25 from RhabdoMyoSarcoma (RMS), 18 from NeuroBlastoma (NB) and 11 from Burkitt's Lymphoma (BL). The dataset has 83 samples, and the number of genes is 10 367. Also, Brain_Tumors2 (Nutt et al., 2003) contains 90 samples divided by 4 classes : classic anaplastic oligodendrogliomas, nonclassic glioblastomas, classic glioblastomas, and nonclassic anaplastic oligodendrogliomas. Each sample has 10367 genes.

The third category has Brain_Tumors1 (Pomeroy et al., 2002) dataset contains 90 samples of 5 classes: AT/RT, normal cerebellum, PNET, malignant glioma and medulloblastoma. The number of genes is 5920. Also, Lung_cancer Carcinomas dataset (Bhattacharjee et al., 2001) contains 203 samples distributed in 5 classes, 139 samples are from Adenocarcinomas (A),

21 from Squamous cell lung carcinomas (SQ), 20 from Pulmonary carcinoids (P), 17 from Normal (N) and 6 from Small-cell lung carcinomas (SM). The number of genes is 12 600.

The last category each dataset has 9 or 11 classes. 9_Tumors (Staunton et al., 2001) contains 60 samples distribute in 9 classes: CNS, breast, NSCLC, melanoma, ovary, leukemia1, renal, prostate and colon. Each sample has 5726 genes. Also, 11_Tumors (Su et al., 2001) obtains 12 533 genes in each sample. There are 174 samples of gene expression distributed in 11 various human tumor types: ovary, breast, kidney, liver, prostate, pancreas, adeno lung, squamous lung, bladder/ureter, gastro-esophagus and colorectal.

4.3.2 Parameters Settings

The MIM-mMFA parameters used in our experiments, the number of population or the number of gene subset, with a value of 50. Another important parameter is the number of moth or number of genes selected after the pre-filter operation. A value of 100 is used for this parameter. For the parameter stopping criterion or a maximum number of iterations, with a value of 30. In this study, to perform our experiments, the number of runs is 10 times on each dataset.

4.3.3 Experimental Results and Analysis

In this section, we present and analyze the results obtained by our proposed algorithm (MIM-mMFA). In order to prove the high-performance of MIM-mMFA, it should be compared with various gene selection methods. For this purpose, we have done comparisons with two types of recently published algorithms.

Table 4.2, Figure 4.6a and 4.6b show the accuracy and the number of genes selected by our algorithm (MIM-mMFA) in both binary and multi classes of sixteen datasets.

From the Figure 4.6a we can see, that MIM-mMFA can obtain 100% (Best, worst and average) accuracy with zero standard deviation (S.D) for the leukemia1, DLBCL, Colon, Prostate_Tumor from binary class and 11_Tumors, 9_Tumors, Brain_Tumors1, Brain_Tumors2, leukemia2, Lung_cancer, Lymphoma, MLL from multi class.

We can see that MIM-mMFA did not obtain 100% accuracy for four datasets, but three CNS, Ovarian and SRBCT datasets are more than 98% average accuracy, and we have the only Breast that has an average accuracy of 86.80%.

Also, from Figure 4.6b, can see, for the leukemia1 and Lymphoma the number of selected genes is inferior of nine (9), we can find that MIM-mMFA achieves 100% accuracy with smaller selected genes. For the datasets DLBCL, Prostate_Tumor, Brain_Tumors1and Brain_Tumors2, the MIM-mMFA can provide 100% accuracy with the number of selected genes between 20 and 10.

As well, For the gene expression datasets 9_Tumors, Lung_cancer and MLL, the MIM-mMFA can provide 100% accuracy with the number of selected genes between 30 and 40.

For the dataset Colon and 11_Tumors, the MIM-mMFA can provide 100% average accuracy with 26.3 and 41.9 selected genes, respectively. Finally, for all datasets, our proposal did not give the 100% in average accuracy, CNS, Ovarian, SRBCT and Breast, the MIM-mMFA provides 99.83%, 98.18%, 99.40% and 86.80% average accuracy with 24.70, 35.90, 27.30 and 25.90 average selected genes, respectively.

MIM-mMFA applied on binary class

Comparison of experimental results obtained by MIM-mMFA with other methods for binary class datasets.

In our comparison, firstly, we compare MIM-mMFA with well-known gene selection algorithms published in the literature, applied to the binary class which presents in Table 4.3,

TABLE 4.2: Experimental results by MIM-mMFA on all datasets.

Class	Dataset	Accuracy				# Genes			
		Best	Worst	Avg.	S.D.	Best	Worst	Avg.	S.D.
Binary Class	Leukemia	100,00	100,00	100,00	0,00	6,00	9,00	7,50	0,97
	DLBCL	100,00	100,00	100,00	0,00	11,00	18,00	14,70	2,00
	CNS	100,00	98,33	99,83	0,53	13,00	31,00	24,70	6,09
	Colon	100,00	100,00	100,00	0,00	20,00	31,00	26,30	3,56
	Ovarian	98,42	98,02	98,18	0,20	26,00	40,00	35,90	4,70
	Breast	91,75	83,51	86,80	3,10	11,00	45,00	25,90	9,60
	Prostate Tumor	100,00	100,00	100,00	0,00	14,00	23,00	18,60	2,63
Multi-Class	11_Tumors	100,00	100,00	100,00	0,00	31,00	54,00	41,90	6,21
	9_Tumors	100,00	100,00	100,00	0,00	22,00	43,00	31,50	7,06
	Brain Tumors1	100,00	100,00	100,00	0,00	11,00	21,00	17,00	2,91
	Brain Tumors2	100,00	100,00	100,00	0,00	8,00	15,00	11,93	2,20
	Leukemia2	100,00	100,00	100,00	0,00	15,00	20,00	18,70	1,49
	Lung_cancer	100,00	100,00	100,00	0,00	20,00	46,00	35,30	8,71
	SRBCT	100,00	98,80	99,40	0,63	23,00	30,00	27,30	2,31
	Lymphoma	100,00	100,00	100,00	0,00	4,00	8,00	6,50	1,35
	MLL	100,00	100,00	100,00	0,00	19,00	43,00	33,00	8,11

such as PCC-BPSO and PCC-GA (Hameed et al., 2018b), MOBBA_LS (Dashtban, Balafar, and Suravajhala, 2018), (GBC) (Alshamlan, Badr, and Alohali, 2015), (ICA-ABC) (Aziz et al., 2017), MIM-AGA (Lu et al., 2017), EPSO (Mohamad et al., 2013), mABC (Moosa et al., 2016) and A multi-objective binary differential evolution method (MOBDE) (Ma, Li, and Wang, 2016).

Table 4.3 shows the experimental results of the MIM-mMFA algorithm and other existing methods in terms of the best, worst, average and standard deviation (S.D.) of the number of genes selected and the classification accuracy. As can be seen in Table 4.3, for Leukemia1, all algorithms can obtain 100% except MOBBA_LS, ICA-ABC and MIM-AGA. But our algorithm obtained a marginally large amount of genes than EPSO, mABC and MOBDE in the best-obtained results for Leukemia1 (5.9 more). On the other hand, for the MOBBA_LS, GBC and MIM-AGA methods selected 3, 5 and 7 genes and achieved 97.1%, 96.43% and 97.68% classification accuracy, respectively. In contrast, the MIM-mMFA algorithm selects 7.5 genes and achieves 100% classification accuracy.

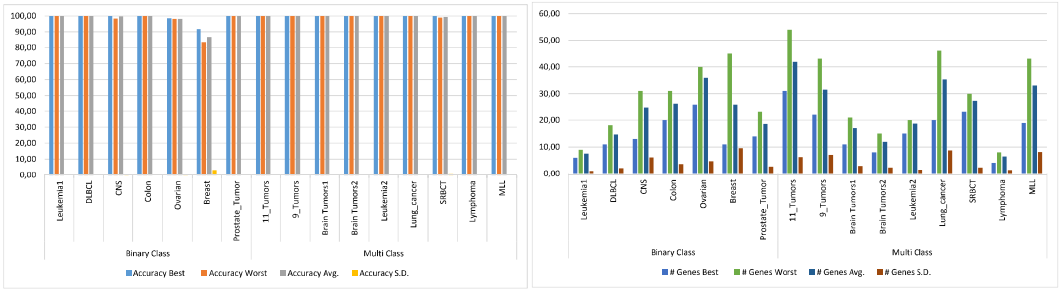
For DLBCL, all algorithms can obtain 100% accuracy with zero (0) standard deviation. But on the average number genes selected is bigger in our algorithm. For Prostate_Tumor, our method has achieved the highest accuracy (100%) like mABC. But our method achieves a slightly higher number of genes than mABC. For CNS, our method has achieved the highest accuracy, the MIM-mMFA is better than all methods with 100%, 98.33% and 99.83% at the best, worst and average accuracy, respectively.

For Colon, the ICA-ABC method selected 12 genes and achieved 90.22% average accuracy. In contrast, the MIM-mMFA selects 26.3 genes and achieves 100% classification

TABLE 4.3: Comparison of experimental results obtained by MIM-mMFA with other methods for binary class datasets.

Algorithms		Dataset	Leukemia1	DLBCL	Prostate Tumor	CNS	Colon	Breast	Ovarian
MIM-mMFA	Accuracy	Best	**100,00**	**100,00**	**100,00**	**100,00**	**100,00**	91,75	98,42
		Worst	**100,00**	**100,00**	**100,00**	**98,33**	**100,00**	83,51	98,02
		Avg.	**100,00**	**100,00**	**100,00**	**99,83**	**100,00**	86,80	98,18
		S.D.	**0,00**	**0,00**	**0,00**	**0,53**	**0,00**	3,10	0,20
	# Genes	Best	6,00	11,00	14,00	**13,00**	20,00	**11,00**	26,00
		Worst	9,00	18,00	23,00	**31,00**	31,00	**45,00**	40,00
		Avg.	7,50	14,70	18,60	**24,70**	26,30	**25,90**	35,90
		S.D.	0,97	2,00	2,63	**6,09**	3,56	**9,60**	4,70
PCC-BPSO	Accuracy	**Best**	**100,00**	-	97,06	98,33	91,94	90,72	**100,00**
	# Genes	Best	18,00	-	33,00	39,00	25,00	41,00	**17,00**
PCC-GA	Accuracy	**Best**	**100,00**	-	96,08	98,33	91,94	88,66	**100,00**
	# Genes	Best	35,00	-	26,00	48,00	29,00	38,00	22,00
MOBBA_LS	Accuracy	Best	97,10	-	94,10	-	-	-	-
	# Genes	Best	3,00	-	6,00	-	-	-	-
GBC	Accuracy	Best	**100,00**	-	-	-	98,38	-	-
		worst	93,05	-	-	-	91,93	-	-
		Avg	96,43	-	-	-	94,62	-	-
	# Genes	Best	5,00	-	-	-	20,00	-	-
ICA-ABC	Accuracy	Best	98,21	-	97,88	-	97,34	-	-
		worst	55,76	-	77,81	-	82,34	-	-
		Avg	83,22	-	82,34	-	90,22	-	-
	# Genes	Best	12,00	-	20,00	-	**12,00**	-	-
MIM-AGA	Accuracy	**Best**	97,68	-	97,69	-	89,09	**95,21**	-
	# Genes	Best	7,00	-	93,00	-	19,00	216,00	-
EPSO	Accuracy	Best	**100,00**	100,00	99,02	-	-	-	-
		Avg.	**100,00**	100,00	97,84	-	-	-	-
		S.D.	**0,00**	0,00	0,62	-	-	-	-
	# Genes	Best	**2,00**	**3,00**	**5,00**	-	-	-	-
		Avg.	**3,20**	4,70	**6,60**	-	-	-	-
		S.D.	**0,63**	0,82	2,17	-	-	-	-
mABC	Accuracy	Best	**100,00**	**100,00**	**100,00**	-	-	-	-
		Avg.	**100,00**	**100,00**	**100,00**	-	-	-	-
		S.D.	0,00	0,00	0,00	-	-	-	-
	# Genes	Best	4,00	**3,00**	**5,00**	-	-	-	-
		Avg.	5,67	**4,05**	10,73	-	-	-	-
		S.D.	0,73	**0,78**	3,15	-	-	-	-
MOBDE	*Accuracy	Average	**100,00**	**100,00**	98,63	-	-	-	-
		S.D.	0,00	0,00	0,83	-	-	-	-
	# Genes	Average	5,90	5,60	10,90	-	-	-	-
		S.D.	1,29	1,83	4,65	-	-	-	-

Best results are highlighted.



(A) Accuracy

(B) Number of genes selected

FIGURE 4.6: The accuracy and number of selected gene obtained by MIM-mMFA

accuracy, he MIM-mMFA is better than all methods.

For Breast, the MIM-AGA method selected 216 genes and achieved 95.21% average accuracy. But, the MIM-mMFA algorithm selects 11 genes and achieves 91.75% classification accuracy. The percentage of genes selected by the MIM-AGA is 0.88%, but our method selects 0.04%.

For the Ovarian dataset, the PCC-BPSO and PCC-GA can provide the best solution of 100% with the number of genes selected being 17 and 22 respectively. The MIM-mMFA can provide the best accuracy of 98.48% and a number of genes selected is 26.

Therefore, in this comparison, we can conclude that MIM-mMFA can perform better accuracy than other algorithms.

MIM-mMFA applied on multi-class

Secondly, we compare MIM-mMFA with some other recent algorithms in the literature, applied to the multi-class which shown in Table 4.4. Also, we are chosen: PCC-BPSO and PCC-GA (Hameed et al., 2018b), EPSO (Mohamad et al., 2013), mABC (Moosa et al., 2016) and MOBDE (Ma, Li, and Wang, 2016).

As shown in Table 4.4, the MIM-mMFO algorithm provides the highest (100%) average classification accuracy with zero (0) standard deviation in 8 out of the 9 datasets and obtains the highest (100%) best classification accuracy in all datasets.

For 11_Tumors and 9_Tumors, the MIM-mMFA has achieved the highest (100%) accuracy, also, the mABC has the same accuracy of our algorithm in 11_Tumors, but in 9_Tumors, the mABC has 100% and 99.50% on best and average classification accuracy, respectively. For both datasets (11_Tumors and 9_Tumors), the MOBD method has selected 27.50 and 20.70 genes and achieved 97.19% and 92.67% average accuracy, respectively. In contrast, the MIM-mMFA algorithm selects more genes, but it achieves 100% classification accuracy.

For Brain_Tumors1, Brain_Tumors2, and Lung_cancer, our method has achieved the highest (100%) accuracy like mABC. The EPSO method selected 7.5, 6 and 8.30 genes and achieved 92.11%, 92.40% and 95.67% average classification accuracy, respectively. In contrast, the MIM-mMFA algorithm selects more genes, but it achieves 100% classification accuracy.

For Leukemia2 all algorithms can obtain 100% accuracy with zero (0) standard deviation. But on the average number genes selected is bigger in our algorithm. For the SRBCT dataset, our algorithm, PCC-BPSO, MOBDE, mABC and PCC-GA can provide the highest (100%) accuracy with the number of the genes selected were 23, 19, 5.40, 5.59 and 20 respectively.

TABLE 4.4: Comparison of experimental results obtained by MIM-mMFA with other methods for multi-class datasets.

Algorithms		Dataset	11_Tumors	9_Tumors	Brain_Tumors1	Brain_Tumors2	Leukemia2	Lung_cancer	SRBCT	Lymphoma	MLL
MIM-mMFA	Accuracy	Best	**100,00**	**100,00**	**100,00**	**100,00**	**100,00**	**100,00**	**100,00**	**100,00**	**100,00**
		Worst	**100,00**	**100,00**	**100,00**	**100,00**	**100,00**	**100,00**	98,80	**100,00**	**100,00**
		Avg.	**100,00**	**100,00**	**100,00**	**100,00**	**100,00**	**100,00**	99,40	**100,00**	**100,00**
		S.D.	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	0,63	**0,00**	**0,00**
	# Genes	Best	31,00	22,00	11,00	8,00	15,00	20,00	23,00	**4,00**	**19,00**
		Worst	54,00	43,00	21,00	15,00	20,00	46,00	30,00	**8,00**	43,00
		Avg.	41,90	31,50	17,00	11,93	18,70	35,30	27,30	**6,50**	**33,00**
		S.D.	6,21	7,06	2,91	2,20	1,49	8,71	2,31	**1,35**	8,11
MOBDE	Accuracy	Average	97,19	92,67	97,67	99,00	**100,00**	99,12	**100,00**	-	-
		S.D.	1,42	2,62	1,22	1,70	**0,00**	15,50	**0,00**	-	-
	# Genes	Average	**27,50**	20,70	11,50	7,50	**6,00**	0,65	**5,40**	-	-
		S.D.	**6,24**	6,83	2,17	1,72	**1,41**	4,50	**1,17**	-	-
EPSO	Accuracy	Best	96,55	96,55	93,33	94,00	**100,00**	96,06	100,00	-	-
		Avg	95,40	95,40	92,11	92,40	**100,00**	95,67	99,64	-	-
		S.D.	0,61	0,61	0,82	1,27	**0,00**	0,31	0,58	-	-
	# Genes	Best	243,00	243,00	**8,00**	**4,00**	**4,00**	**7,00**	7,00	-	-
		Avg	237,70	237,70	**7,50**	**6,00**	6,80	**8,30**	14,90	-	-
		S.D.	9,66	9,66	**2,51**	**1,83**	2,20	**2,11**	13,03	-	-
mABC	Accuracy	Best	100,00	100,00	**100,00**	**100,00**	**100,00**	**100,00**	**100,00**	-	-
		Avg	99,50	98,65	**100,00**	**100,00**	**100,00**	**100,00**	**100,00**	-	-
		S.D.	-	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	-	-
	# Genes	Best	42,00	30,00	12,00	7,00	**4,00**	14,00	**5,00**	-	-
		Avg	47,27	34,73	16,87	10,52	6,29	23,31	5,59	-	-
		S.D.	7,79	5,64	2,85	1,72	0,98	5,14	0,51	-	-
PCC-BPSO	Accuracy	Best	-	-	-	-	-	97,04	**100,00**	**100,00**	**100,00**
	# Genes	Best	-	-	-	-	-	40,00	19,00	30,00	40,00
PCC-GA	Accuracy	Best	-	-	-	-	-	97,54	**100,00**	**100,00**	**100,00**
	# Genes	Best	-	-	-	-	-	42,00	20,00	39,00	22,00

Best results are highlighted.

For Lymphoma and MLL, our method has achieved the highest (100 %) accuracy like PCC-BPSO and PCC-GA. And for both datasets, the MIM-mMFA selects the same number of genes as PCC-BPSO and PCC-GA.

Based on the above analysis, we can conclude that, when only considering the accuracy of the classification, the MIM-mMFA algorithm is better than all other algorithms.

4.4 Conclusion

In this chapter, we have presented a new bio-inspired algorithm for gene selection problems. Our approach consists of two stages. The first begins with a normalization of the dataset by Min-Max statistic method to guarantee a stable convergence of weight and biases. Then, in order to reduce the initial size of the input dataset and to handle with the inaccurate nature of the expression levels, we used a Mutual Information Maximization (MIM) based pre-processing technique.

In the second stage, the mMFA introduced a new technique to obtain the next moth, a specific operator of gene selection and uses an SVM with LOOCV based fitness function to evaluate the selected gene subsets accuracy. The overall goal of this study is to select a smaller number of genes and achieve similar or better classification accuracy than using all genes. The tests of MIM-mMFA on sixteen datasets show that our algorithm is better than all other compared algorithms to the classification accuracy and presents competitive results with the number of genes.

Chapter 5

Conclusions And Future Work

5.1 General Conclusions

This thesis aimed to contribute to the resolution of Bioinformatics problems by using the development of bio-inspired algorithms. It focuses on two important problems: multiple sequence alignment and gene selection.

The first goal of this thesis is to focus on solving the multiple sequence alignment problem that is a very fundamental activity in sequence analysis, which is inherently difficult. In addition, The MSA problem is an NP-Complete, therefore, it cannot process a large quantity of data and solve the problem in a practical time period.

In order to solve this problem, we introduced an algorithm called multi-objective artificial fish swarm (MOAFS) algorithm for the MSA problem to reach an optimal or semi-optimal alignment of the original large sequences. The MOAFS starts with an efficient representation of candidate solutions, then it combines two initialization operations to explore search space of the MSA problem. In addition, to explore more search space, we have been proposed some procedures inspired by artificial fish swarm behaviors such as Swarm, Follow and Prey, that used some operations such as mutation, crossover and center fish (consensus). In order to ensure a good trade-off between the exploration and the exploitation of the search space of the MSA problem, the MOAFS uses and adapts AFSA behaviors to update the population of solutions (such as the initialization with two methods and various operations). Accordingly, All candidate solutions are evaluated by two fitness functions: Weighted Sum of Pairs and Similarity to determine horizontally and vertically similar regions, respectively. Next, the Pareto-optimal set is obtained by the MOAFS, which performs the optimal multiple sequence alignments for both fitness functions.

In order to ensure comparison quality, this thesis chooses to make a full experiment on BaliBASE 2.0 and 3.0 benchmarks, so we can the quality alignment of each test case. The MOAFS performance has given better quality compared with different algorithm methods. Therefore, the better performance of MOAFS has biologically more meaningful.

The last goal of this thesis is focused to solve gene selection in classification problems. Thus, the gene selection problem consists in extracting the entire dataset with in order to identify the relevant variables in relation to the problem at hand, which is an NP-Complete (2^N possible subsets for N features).

Hence, we have introduced a hybrid model that used many techniques: Mutual information maximization (MIM), Moth Flame optimization (MFO), Support Vector Machine

(SVM), and Leave One Out Cross Validation (LOOCV). We have suggested the Mutual Information Maximization-modified Moth Flame Algorithm called MIM-mMFA to solve gene selection problem, in order to reduce the gene number and achieve similar or even better classification performance than that obtained using all the original genes. The first stage of this approach is the Mutual Information Maximization (MIM) based pre-filtering technique that used to measure the relevance and the redundancy of genes, the second stage is the modified Moth Flame Algorithm (mMFA) that used to evolve gene subsets and evaluated by the fitness function, which uses a Support Vector Machine (SVM) with Leave One Out Cross Validation (LOOCV) classifier and the number of selected genes.

The MIM-mMFA was examined and compared to existing methods of classification problems of varying difficulty by using binary class and multi-class gene expression microarray datasets. The results clearly showed that the MIM-mMFA can be used effectively to select genes and reduce dimensions in classification.

This thesis has provided a detailed understanding of the bio-inspired algorithms and their application to Bioinformatics problems. In addition, two new methods have been developed in an efficient and effective manner. Ultimately, this thesis has been largely succeeded in achieving its overall goals.

5.2 Future Work

With the current deluge of data, computational methods have become indispensable to biological investigations. Bioinformatics was originally developed for the biological sequence analysis, which now encompasses a wide range of fields, including genomics, structural biology, and gene expression studies. In addition, the main aims of Bioinformatics are to understand and organize the information associated with biological molecules on a large scale. As a result, Bioinformatics has not only provided greater depth to biological investigations, but also added the dimension of breadth.

Even though the current methods in bio-inspired algorithms are very helpful in identifying patterns and functions of proteins, genes, etc., the final results are far from being perfect. However, there are some general problems that should be needed to address by the researchers in the future in order to fully exploit the potential of bio-inspired algorithms in Bioinformatics.

As the first step in this process of investigation, this thesis will require additional efforts and years of training and work. Nonetheless, our short-term goals focus on improving and complementing the work initiated in this thesis, namely:

- ☞ Our algorithms (MOAFS and MIM-mMFA) should be applied to some reference databases, and they are also necessary to obtain good results using other Benchmarks (like Homstrad and Oxbench for MSA). In other words, to say that these are effective algorithms in all cases, i.e. The MOAFS and MIM-mMFA are generally efficient. The current research should progress in this direction also. Future work will be in this direction.

- ☞ Inspire new algorithms based on nature in general animal and human life, in particular, to address optimization problems in a broad sense and in a special sense Bioinformatics problems.

- ☞ Employing our algorithms (MOAFS and MIM-mMFA) for other Bioinformatics problems such as protein secondary and tertiary structure prediction, protein-ligand docking, promoter identification and the reconstruction of evolutionary trees, etc.

- ☞ Hybridization between bio-inspired algorithms to solve problems of multiple sequence alignment, gene selection, and others.

Bibliography

Abbasi, Maryam, Luís Paquete, and Francisco B Pereira (2016). "Heuristics for multiobjective multiple sequence alignment". In: *Biomedical engineering online* 15.1, p. 70.

Alanni, Russul et al. (2019). "A novel gene selection algorithm for cancer classification using microarray datasets". In: *BMC medical genomics* 12.1, p. 10.

Alizadeh, Ash A et al. (2000). "Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling". In: *Nature* 403.6769, p. 503.

Alon, Uri et al. (1999). "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays". In: *Proceedings of the National Academy of Sciences* 96.12, pp. 6745–6750.

Alshamlan, Hala M, Ghada H Badr, and Yousef A Alohali (2015). "Genetic Bee Colony (GBC) algorithm: A new gene selection method for microarray cancer classification". In: *Computational biology and chemistry* 56, pp. 49–60.

Altman, Russ B (2001). "Challenges for intelligent systems in biology". In: *IEEE Intelligent Systems* 16.6, pp. 14–18.

Amaldi, Edoardo and Viggo Kann (1998). "On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems". In: *Theoretical Computer Science* 209.1-2, pp. 237–260.

Armstrong, Scott A et al. (2001). "MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia". In: *Nature genetics* 30.1, p. 41.

Aziz, Rabia et al. (2017). "Artificial neural network classification of microarray data using new hybrid gene selection method". In: *International Journal of Data Mining and Bioinformatics* 17.1, pp. 42–65.

Bacon, David J and Wayne F Anderson (1986). "Multiple sequence alignment". In: *Journal of molecular biology* 191.2, pp. 153–161.

Bahr, Anne et al. (2001). "BAliBASE (Benchmark Alignment dataBASE): enhancements for repeats, transmembrane sequences and circular permutations". In: *Nucleic Acids Research* 29.1, pp. 323–326.

Baldi, Pierre, Søren Brunak, and Francis Bach (2001). *Bioinformatics: the machine learning approach*. MIT press.

Baxevanis, Andreas D and BF Francis Ouellette (2004). *Bioinformatics: a practical guide to the analysis of genes and proteins*. Vol. 43. John Wiley & Sons.

Bell, Eric Temple (2014). *Men of mathematics*. Simon and Schuster.

Bhattacharjee, Arindam et al. (2001). "Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses". In: *Proceedings of the National Academy of Sciences* 98.24, pp. 13790–13795.

Bonabeau, Eric et al. (1999). *Swarm intelligence: from natural to artificial systems*. 1. Oxford university press.

Bradley, Robert K et al. (2009). "Fast statistical alignment". In: *PLoS computational biology* 5.5, e1000392.

Breslow, Norman (1970). "A generalized Kruskal-Wallis test for comparing K samples subject to unequal patterns of censorship". In: *Biometrika* 57.3, pp. 579–594.

Btissam, DKHISSI and Rachida Abounacer (2017). "Multi-Objective examination Timetabling Problem: Modeling and resolution using a based ε-constraint method". In: *IJCSNS* 17.4, p. 192.

Burnham, Kenneth P and David R Anderson (2001). "Kullback-Leibler information as a basis for strong inference in ecological studies". In: *Wildlife research* 28.2, pp. 111–119.

Buturović, Ljubomir J (2005). "PCP: a program for supervised classification of gene expression profiles". In: *Bioinformatics* 22.2, pp. 245–247.

Chakrabarti, Saikat et al. (2004). "Improvement of alignment accuracy utilizing sequentially conserved motifs". In: *BMC bioinformatics* 5.1, p. 167.

Chandrashekar, Girish and Ferat Sahin (2014). "A survey on feature selection methods". In: *Computers & Electrical Engineering* 40.1, pp. 16–28.

Chang, Jenny C, Susan G Hilsenbeck, and Suzanne AW Fuqua (2005). "The promise of microarrays in the management and treatment of breast cancer". In: *Breast Cancer Research* 7.3, p. 100.

Chao, Kun-Mao and Louxin Zhang (2008). *Sequence comparison: theory and methods*. Vol. 7. Springer Science & Business Media.

Chellapilla, Kumar and Gary B Fogel (1999). "Multiple sequence alignment using evolutionary programming". In: *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*. Vol. 1. IEEE, pp. 445–452.

Chen, Huihui, Yusen Zhang, and Ivan Gutman (2016). "A kernel-based clustering method for gene selection with gene expression data". In: *Journal of Biomedical Informatics* 62, pp. 12–20.

Chia, Nicholas and Ralf Bundschuh (2006). "A practical approach to significance assessment in alignment with gaps". In: *Journal of Computational Biology* 13.2, pp. 429–441.

Chowdhury, Biswanath and Gautam Garai (2017). "A review on multiple sequence alignment from the perspective of genetic algorithm". In: *Genomics* 109.5-6, pp. 419–431.

Coello, Carlos Coello, Clarisse Dhaenens, and Laetitia Jourdan (2009). *Advances in multi-objective nature inspired computing*. Vol. 272. Springer.

Corpet, Florence (1988). "Multiple sequence alignment with hierarchical clustering". In: *Nucleic acids research* 16.22, pp. 10881–10890.

Cotta, Carlos and Pablo Moscato (2003). "The k-Feature Set problem is W [2]-complete". In: *Journal of Computer and System Sciences* 67.4, pp. 686–690.

Cover, Thomas M and Jan M Van Campenhout (1977). "On the possible orderings in the measurement selection problem". In: *IEEE transactions on systems, man, and cybernetics* 7.9, pp. 657–661.

Crawford, John R and David C Howell (1998). "Comparing an individual's test score against norms derived from small samples". In: *The Clinical Neuropsychologist* 12.4, pp. 482–486.

Crick, FHC (1958). *The Biological Replication of Macromolecules, XIIth Symposium of the Society for Experimental Biology*.

Crick, Francis (1970). "Central dogma of molecular biology". In: *Nature* 227.5258, p. 561.

Cutello, Vincenzo et al. (2010). "Protein multiple sequence alignment by hybrid bio-inspired algorithms". In: *Nucleic acids research* 39.6, pp. 1980–1992.

Dabba, Ali, Abdelkamel Tari, and Djaafar Zouache (2019). "Multiobjective artificial fish swarm algorithm for multiple sequence alignment". In: *INFOR: Information Systems and Operational Research*, pp. 1–22.

Dash, Manoranjan and Huan Liu (1997). "Feature selection for classification". In: *Intelligent data analysis* 1.1-4, pp. 131–156.

Dashtban, M, Mohammadali Balafar, and Prashanth Suravajhala (2018). "Gene selection for tumor classification using a novel bio-inspired multi-objective approach". In: *Genomics* 110.1, pp. 10–17.

Davies, Scott and Stuart Russell (1994). "NP-completeness of searches for smallest possible feature sets". In: *AAAI Symposium on Intelligent Relevance*. AAAI Press, pp. 37–39.

Dayhoff, MO, RM Schwartz, and BC Orcutt (1978). "22 A Model of Evolutionary Change in Proteins". In: *Atlas of protein sequence and structure*. Vol. 5. National Biomedical Research Foundation Silver Spring, MD, pp. 345–352.

Deng, Xuelian et al. (2019). "Feature selection for text classification: A review". In: *Multimedia Tools and Applications* 78.3, pp. 3797–3816.

Do, Chuong B et al. (2005). "ProbCons: Probabilistic consistency-based multiple sequence alignment". In: *Genome research* 15.2, pp. 330–340.

Du, Dajun et al. (2014). "A novel forward gene selection algorithm for microarray data". In: *Neurocomputing* 133, pp. 446–458.

Dudoit, Sandrine, Jane Fridlyand, and Terence P Speed (2002). "Comparison of discrimination methods for the classification of tumors using gene expression data". In: *Journal of the American statistical association* 97.457, pp. 77–87.

Duval, Béatrice, Jin-Kao Hao, and Jose Crispin Hernandez Hernandez (2009). "A memetic algorithm for gene selection and molecular classification of cancer". In: *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*. ACM, pp. 201–208.

Edgar, Robert C (2004). "MUSCLE: multiple sequence alignment with high accuracy and high throughput". In: *Nucleic acids research* 32.5, pp. 1792–1797.

Edgar, Robert C and Kimmen Sjoelander (2002). "Simultaneous sequence alignment and tree construction using hidden Markov models". In: *Biocomputing 2003*. World Scientific, pp. 180–191.

Feng, Da-Fei and Russell F Doolittle (1987). "Progressive sequence alignment as a prerequisitetto correct phylogenetic trees". In: *Journal of molecular evolution* 25.4, pp. 351–360.

Frank, Kenneth D, C Rich, and T Longcore (2006). "Effects of artificial night lighting on moths". In: *Ecological consequences of artificial night lighting*, pp. 305–344.

Gao, Chong et al. (2018). "Hybrid Invasive Weed Optimization and GA for Multiple Sequence Alignment". In: *International Conference on Bio-Inspired Computing: Theories and Applications*. Springer, pp. 72–82.

Ghaddar, Bissan and Joe Naoum-Sawaya (2018). "High dimensional data classification and feature selection using support vector machines". In: *European Journal of Operational Research* 265.3, pp. 993–1004.

Ghosh, Manosij et al. (2019). "Genetic algorithm based cancerous gene identification from microarray data using ensemble of filter methods". In: *Medical & Biological Engineering & Computing* 57.1, pp. 159–176.

Goldberg, David E and John H Holland (1988). "Genetic algorithms and machine learning". In: *Machine learning* 3.2, pp. 95–99.

Golub, Todd R et al. (1999). "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring". In: *science* 286.5439, pp. 531–537.

Gondro, Cedric and Brian P Kinghorn (2007). "A simple genetic algorithm for multiple sequence alignment". In: *Genetics and Molecular Research* 6.4, pp. 964–982.

Guorong, Xuan, Chai Peiqi, and Wu Minhui (1996). "Bhattacharyya distance feature selection". In: *Proceedings of 13th International Conference on Pattern Recognition*. Vol. 2. IEEE, pp. 195–199.

Guyon, Isabelle and André Elisseeff (2003). "An introduction to variable and feature selection". In: *Journal of machine learning research* 3.Mar, pp. 1157–1182.

Hameed, Shilan S et al. (2018a). "Filter-Wrapper Combination and Embedded Feature Selection for Gene Expression Data." In: *International Journal of Advances in Soft Computing & Its Applications* 10.1.

Hameed, Shilan S et al. (2018b). "Gene Selection and Classification in Microarray Datasets using a Hybrid Approach of PCC-BPSO/GA with Multi Classifiers." In: *JCS* 14.6, pp. 868–880.

Hein, Jotun (1989). "A new method that simultaneously aligns and reconstructs ancestral sequences for any number of homologous sequences, when the phylogeny is given." In: *Molecular Biology and Evolution* 6.6, pp. 649–668.

Henikoff, Steven and Jorja G Henikoff (1992). "Amino acid substitution matrices from protein blocks". In: *Proceedings of the National Academy of Sciences* 89.22, pp. 10915–10919.

Hogeweg, Paulien (2011). "The roots of bioinformatics in theoretical biology". In: *PLoS computational biology* 7.3, e1002021.

Holland, John H (1973). "Genetic algorithms and the optimal allocation of trials". In: *SIAM Journal on Computing* 2.2, pp. 88–105.

Hu, Hong et al. (2006). "Combined gene selection methods for microarray data analysis". In: *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*. Springer, pp. 976–983.

Huerta, Edmundo Bonilla, Béatrice Duval, and Jin-Kao Hao (2006). "A hybrid GA/SVM approach for gene selection and classification of microarray data". In: *Workshops on Applications of Evolutionary Computation*. Springer, pp. 34–44.

Huo, Hongwei and Vojislav Stojkovic (2007). "A simulated annealing algorithm for multiple sequence alignment with guaranteed accuracy". In: *Natural Computation, 2007. ICNC 2007. Third International Conference on*. Vol. 2. IEEE, pp. 270–274.

Jain, Anil, Karthik Nandakumar, and Arun Ross (2005). "Score normalization in multimodal biometric systems". In: *Pattern recognition* 38.12, pp. 2270–2285.

Jeffrey, Stefanie S, Per Eystein Lønning, and Bruce E Hillner (2005). "Genomics-based prognosis and therapeutic prediction in breast cancer". In: *Journal of the National Comprehensive Cancer Network* 3.3, pp. 291–300.

Kaiser, Chris A et al. (2007). *Molecular cell biology*. WH Freeman.

Katoh, Kazutaka et al. (2002). "MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform". In: *Nucleic acids research* 30.14, pp. 3059–3066.

Kemena, Carsten and Cedric Notredame (2009). "Upcoming challenges for multiple sequence alignment methods in the high-throughput era". In: *Bioinformatics* 25.19, pp. 2455–2465.

Khan, Javed et al. (2001). "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks". In: *Nature medicine* 7.6, p. 673.

Kohavi, Ron and George H John (1997). "Wrappers for feature subset selection". In: *Artificial intelligence* 97.1-2, pp. 273–324.

Lassmann, Timo, Oliver Frings, and Erik LL Sonnhammer (2008). "Kalign2: high-performance multiple alignment of protein and nucleotide sequences allowing external features". In: *Nucleic acids research* 37.3, pp. 858–865.

Lawrence, Charles E et al. (1993). "Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment". In: *science* 262.5131, pp. 208–214.

Lee, Christopher, Catherine Grasso, and Mark F Sharlow (2002). "Multiple sequence alignment using partial order graphs". In: *Bioinformatics* 18.3, pp. 452–464.

Lehninger, Albert L et al. (2005). *Lehninger principles of biochemistry*. Macmillan.

Li, Xiao-lei (2002). "An optimizing method based on autonomous animats: fish-swarm algorithm". In: *Systems Engineering-Theory & Practice* 22.11, pp. 32–38.

Liu, Huan and Lei Yu (2005). "Toward integrating feature selection algorithms for classification and clustering". In: *IEEE Transactions on Knowledge & Data Engineering* 4, pp. 491–502.

Liu, Shenghui et al. (2018). “Feature selection of gene expression data for Cancer classification using double RBF-kernels”. In: *BMC bioinformatics* 19.1, p. 396.

Liu, Yongchao, Bertil Schmidt, and Douglas L Maskell (2010). “MSAProbs: multiple sequence alignment based on pair hidden Markov models and partition function posterior probabilities”. In: *Bioinformatics* 26.16, pp. 1958–1964.

Löytynoja, Ari and Nick Goldman (2005). “An algorithm for progressive multiple alignment of sequences with insertions”. In: *Proceedings of the National academy of sciences of the United States of America* 102.30, pp. 10557–10562.

Lu, Huijuan et al. (2017). “A hybrid feature selection algorithm for gene expression data classification”. In: *Neurocomputing* 256, pp. 56–62.

Ma, Shijing, Xiangtao Li, and Yunhe Wang (2016). “Classification of Gene Expression Data Using Multiobjective Differential Evolution”. In: *Energies* 9.12, p. 1061.

Mirjalili, Seyedali (2015). “Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm”. In: *Knowledge-Based Systems* 89, pp. 228–249.

Mohamad, Mohd Saberi et al. (2013). “An enhancement of binary particle swarm optimization for gene selection in classifying cancer classes”. In: *Algorithms for Molecular Biology* 8.1, p. 15.

Moosa, Johra Muhammad et al. (2016). “Gene selection for cancer classification with the help of bees”. In: *BMC medical genomics* 9.2, p. 47.

Mount, David W (2009). “Using hidden Markov models to align multiple sequences”. In: *Cold Spring Harbor Protocols* 2009.7, pdb–top41.

Mudaliar, Pallavi U et al. (2015). “A Fast Clustering Based Feature Subset Selection Algorithm for High Dimensional Data”. In: *International journal of emerging trend in engineering and basic science* 2.1, pp. 494–499.

Mundra, Piyushkumar A and Jagath C Rajapakse (2010). “Gene and sample selection for cancer classification with support vectors based t-statistic”. In: *Neurocomputing* 73.13-15, pp. 2353–2362.

Myers, Amanda J et al. (2007). “A survey of genetic human cortical gene expression”. In: *Nature genetics* 39.12, p. 1494.

Naznin, Farhana, Ruhul Sarker, and Daryl Essam (2011). “Vertical decomposition with genetic algorithm for multiple sequence alignment”. In: *BMC bioinformatics* 12.1, p. 353.

— (2012). “Progressive alignment method using genetic algorithm for multiple sequence alignment”. In: *IEEE Transactions on Evolutionary Computation* 16.5, pp. 615–631.

Neshat, Mehdi et al. (2012). “A review of artificial fish swarm optimization methods and applications.” In: *International Journal on Smart Sensing & Intelligent Systems* 5.1.

Ng, Andrew Y et al. (1997). “Preventing" overfitting" of cross-validation data”. In: *ICML*. Vol. 97, pp. 245–253.

Notredame, Cédric (2002). “Recent progress in multiple sequence alignment: a survey”. In: *Pharmacogenomics* 3.1, pp. 131–144.

Notredame, Cédric and Desmond G Higgins (1996). “SAGA: sequence alignment by genetic algorithm”. In: *Nucleic acids research* 24.8, pp. 1515–1524.

Notredame, Cédric, Desmond G Higgins, and Jaap Heringa (2000). “T-Coffee: A novel method for fast and accurate multiple sequence alignment”. In: *Journal of molecular biology* 302.1, pp. 205–217.

Nutt, Catherine L et al. (2003). “Gene expression-based classification of malignant gliomas correlates better with survival than histological classification”. In: *Cancer research* 63.7, pp. 1602–1607.

Ortuño, Francisco M et al. (2013). “Optimizing multiple sequence alignments using a genetic algorithm based on three objectives: structural information, non-gaps percentage and totally conserved columns”. In: *Bioinformatics* 29.17, pp. 2112–2121.

Pei, Jimin and Nick V Grishin (2006). "MUMMALS: multiple sequence alignment improved by using hidden Markov models with local structural information". In: *Nucleic Acids Research* 34.16, pp. 4364–4374.

Petricoin III, Emanuel F et al. (2002). "Use of proteomic patterns in serum to identify ovarian cancer". In: *The lancet* 359.9306, pp. 572–577.

Pomeroy, Scott L et al. (2002). "Prediction of central nervous system embryonal tumour outcome based on gene expression". In: *Nature* 415.6870, p. 436.

Press, WH et al. (1992). "Numerical Recipes, Cambridge Univ". In: *Press Cambridge MA, USA* 55, p. 62.

Rani, R Ranjani and D Ramyachitra (2018). "A Hybridization of Artificial Bee Colony with Swarming Approach of Bacterial Foraging Optimization for Multiple Sequence Alignment". In: *Soft Computing for Biological Systems*. Springer, pp. 39–65.

Roshan, Usman and Dennis R Livesay (2006). "Probalign: multiple sequence alignment using partition function posterior probabilities". In: *Bioinformatics* 22.22, pp. 2715–2721.

Rubio-Largo, Álvaro, Miguel A Vega-Rodríguez, and David L González-Álvarez (2016). "Hybrid multiobjective artificial bee colony for multiple sequence alignment". In: *Applied Soft Computing* 41, pp. 157–168.

Schumpeter, Joseph A (1949). "Vilfredo Pareto (1848-1923)". In: *The Quarterly Journal of Economics*, pp. 147–173.

Setubal, Joao Carlos, Joao Meidanis, and Setubal-Meidanis (1997). *Introduction to computational molecular biology*. 04; QH506, S4. PWS Pub. Boston.

Shipp, Margaret A et al. (2002). "Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning". In: *Nature medicine* 8.1, p. 68.

Sievers, Fabian et al. (2011). "Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega". In: *Molecular systems biology* 7.1, p. 539.

Silva, Fernando José Mateus et al. (2010). "An evolutionary approach for performing multiple sequence alignment". In: *Evolutionary Computation (CEC), 2010 IEEE Congress on*. IEEE, pp. 1–7.

Singh, Dinesh et al. (2002). "Gene expression correlates of clinical prostate cancer behavior". In: *Cancer cell* 1.2, pp. 203–209.

Staunton, Jane E et al. (2001). "Chemosensitivity prediction by transcriptional profiling". In: *Proceedings of the National Academy of Sciences* 98.19, pp. 10787–10792.

Stormo, Gary D et al. (1982). "Use of the 'Perceptron'algorithm to distinguish translational initiation sites in E. coli". In: *Nucleic acids research* 10.9, pp. 2997–3011.

Stoye, Jens (1998). "Multiple sequence alignment with the divide-and-conquer method". In: *Gene* 211.2, GC45–GC56.

Su, Andrew I et al. (2001). "Molecular classification of human carcinomas by use of gene expression signatures". In: *Cancer research* 61.20, pp. 7388–7393.

Subramanian, Amarendran R, Michael Kaufmann, and Burkhard Morgenstern (2008). "DIALIGN-TX: greedy and progressive approaches for segment-based multiple sequence alignment". In: *Algorithms for Molecular Biology* 3.1, p. 6.

Taheri, Javid and Albert Y Zomaya (2009). "RBT-GA: a novel metaheuristic for solving the multiple sequence alignment problem". In: *Bmc Genomics* 10.1, S10.

Taylor, William R and Janet M Thornton (1984). "Recognition of super-secondary structure in proteins". In: *Journal of molecular biology* 173.4, pp. 487–514.

Thompson, Julie D, Desmond G Higgins, and Toby J Gibson (1994). "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice". In: *Nucleic acids research* 22.22, pp. 4673–4680.

Thompson, Julie D, Frédéric Plewniak, and Olivier Poch (1999). "A comprehensive comparison of multiple sequence alignment programs". In: *Nucleic acids research* 27.13, pp. 2682–2690.

Thompson, Julie D et al. (2005). "BAliBASE 3.0: latest developments of the multiple sequence alignment benchmark". In: *Proteins: Structure, Function, and Bioinformatics* 61.1, pp. 127–136.

Torkkola, Kari (2003). "Feature extraction by non-parametric mutual information maximization". In: *Journal of machine learning research* 3.Mar, pp. 1415–1438.

Venkateswara, Hemanth et al. (2015). "Efficient approximate solutions to mutual information based global feature selection". In: *2015 IEEE International Conference on Data Mining*. IEEE, pp. 1009–1014.

Vuong, Phu et al. (2008). "Analysis of YfgL and YaeT interactions through bioinformatics, mutagenesis, and biochemistry". In: *Journal of Bacteriology* 190.5, pp. 1507–1517.

Wadud, Md Shams et al. (2018). "Multiple Sequence Alignment Using Chemical Reaction Optimization Algorithm". In: *International Conference on Intelligent Systems Design and Applications*. Springer, pp. 1065–1074.

Wallace, Iain M et al. (2006). "M-Coffee: combining multiple sequence alignment methods with T-Coffee". In: *Nucleic acids research* 34.6, pp. 1692–1699.

Wang, Haiyan et al. (2013). "TSG: a new algorithm for binary and multi-class cancer classification and informative genes selection". In: *BMC medical genomics* 6.1, S3.

Wang, Lusheng and Tao Jiang (1994). "On the complexity of multiple sequence alignment". In: *Journal of computational biology* 1.4, pp. 337–348.

Wang, Yuhang et al. (2004). "HykGene: a hybrid approach for selecting marker genes for phenotype classification using microarray gene expression data". In: *Bioinformatics* 21.8, pp. 1530–1537.

Wang, Zhenyu (2005). "Neuro-fuzzy modeling for microarray cancer gene expression data". In: *First year transfer report, University of Oxford*.

Waterman, Michael S, Temple F Smith, and William A Beyer (1976). "Some biological sequence metrics". In: *Advances in Mathematics* 20.3, pp. 367–387.

Watson, James D, Francis HC Crick, et al. (1953). "Molecular structure of nucleic acids". In: *Nature* 171.4356, pp. 737–738.

Wilcoxon, Frank (1992). "Individual comparisons by ranking methods". In: *Breakthroughs in Statistics*. Springer, pp. 196–202.

Xiao, Zhongzhe et al. (2008). "ESFS: A new embedded feature selection method based on SFS". In:

Yasin, Layal (2016). "Multiple Sequence Alignment Using External Sources Of Information". PhD thesis. Georg-August University.

Zemali, Elamine and Abdelmadjid Boukra (2018). "EGSA: a new enhanced gravitational search algorithm to resolve multiple sequence alignment problem". In: *International Journal of Intelligent Engineering Informatics* 6.1-2, pp. 204–217.

Zhang, Yan-Qing and Jagath Chandana Rajapakse (2009). *Machine learning in bioinformatics*. Vol. 4. Wiley Online Library.

Zhou, Rong and Eric A Hansen (2004). "K-Group A* for multiple sequence alignment with quasi-natural gap costs". In: *Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on*. IEEE, pp. 688–695.

Zhu, Huazheng, Zhongshi He, and Yuanyuan Jia (2016). "A novel approach to multiple sequence alignment using multiobjective evolutionary algorithm based on decomposition". In: *IEEE journal of biomedical and health informatics* 20.2, pp. 717–727.

Zhu, Zexuan, Yew-Soon Ong, and Manoranjan Dash (2007). "Markov blanket-embedded genetic algorithm for gene selection". In: *Pattern Recognition* 40.11, pp. 3236–3248.

Abstract

The field of bioinformatics opens up great opportunities to understand biological phenomena, which has attracted great interest from the scientific community in recent years. Consequently, there are many problems of bioinformatics, including multiple sequence alignment, protein structure prediction, construction of the phylogenetic tree and molecular docking, etc., which need the cooperation between biologists and computer scientists to be solved. This work addresses two problems: multiple sequence alignment and gene selection using bio-inspired algorithms. Firstly, we developed a method to solve the multiple sequence alignment problem, called a multi-objective artificial fish swarm algorithm (MOAFS), using the behaviors of artificial fish swarm algorithms, Pareto optimal set, and genetic operations. Secondly, we proposed an algorithm to solve the gene selection problem by using mutual information, moth flame optimization algorithm, and support vector machine with leave one out cross-validation (SVMLOOCV). It called the Mutual Information Maximization-modified Moth Flame Algorithm (MIM-mMFA) that consists of two simple phases. The thesis has processed a full test of the MOAFS on the BaliBASE 2.0 and BaliBASE 3.0 alignment benchmark datasets as well as the MIM-mMFA test on sixteen binary and multi-classes cancer gene expression datasets. Finally, we have given a deep insight into the performance of each algorithm. In addition, our proposed algorithms achieved competitive or better results than the well-established algorithms in the literature.

Keywords: Bio-informatics; Bio-inspired Algorithms ; Multiple Sequence Alignment ; Artificial Fish Swarm Algorithm ; Gene Selection Genes Expression ; Microarray ; Cancer Classification ; Moth Flame Optimization Algorithm.

Résume

Le domaine de la bio-informatique offre de grandes possibilités de comprendre les phénomènes biologiques, ce qui a suscité un grand intérêt de la part de la communauté scientifique ces dernières années. Par conséquent, il existe de nombreux problèmes de bio-informatique, y compris l'alignement de séquences multiples, la prédiction de la structure des protéines, la construction de l'arbre phylogénétique et l'amarrage moléculaire, etc. qui nécessitent la coopération entre biologistes et informaticiens pour être résolus. Ce travail aborde deux problèmes: l'alignement de séquences multiples et la sélection de gènes à l'aide d'algorithmes bio-inspirés. Premièrement, nous avons développé une méthode pour résoudre le problème de l'alignement des séquences multiples, appelée algorithme d'essaim de poissons artificiels multi-objectifs (MOAFS), en utilisant les comportements des algorithmes d'essaim de poissons artificiels, l'ensemble Pareto-optimal, et les opérations génétiques. Deuxièmement, nous avons proposé un algorithme pour résoudre le problème de sélection de gènes en utilisant l'information mutuelle, l'algorithme d'optimisation de flamme de papillon de nuit, et le Machine à vecteurs de support avec leave-one-out cross-validation (SVM-LOOCV). Il a appelé Mutual Information Maximization-modified Moth Flame Algorithm (MIM-mMFA) qui se compose en deux phases simples. La thèse a traité un test complet du MOAFS sur les ensembles de données de référence d'alignement BaliBASE 2.0 et BaliBASE 3.0 ainsi que le test MIM-mMFA sur seize ensembles de données du cancer binaires et multi-classes. Enfin, nous avons donné un aperçu approfondi des performances de chaque algorithme. De plus, nos algorithmes proposés ont obtenu des résultats compétitifs ou meilleurs que les algorithmes bien établis dans la littérature.

Mots-clés: Bioinformatique ; Algorithmes bio-inspirés ; Alignement de séquences multiples ; Algorithme d'essaim de poissons artificiels ; Sélection des gènes ; Expression des gènes ; Puces à ADN ; Classification du cancer; Algorithme d'optimisation de la flamme papillon.

ملخص

يوفر مجال المعلوماتية الحيوية فرصا كبيرة لفهم الظواهر البيولوجية ، التي جذبت اهتماما كبيرا من قبل المجتمع العلمي في السنوات الأخيرة. ومع ذلك ، هناك العديد من مشاكل المعلوماتية الحيوية ، بما في ذلك محاذاة التسلسل المتعدد ، والتنبؤ بهيكل البروتين ، وبناء الأشجار التطورية والالتحام الجزيئي ، الخ. و ليتم حلها يتطلب التعاون بين علماء الأحياء وعلماء الكمبيوتر.

يعالج هذا العمل مشكلتين هما: محاذاة التسلسل المتعدد وانتقاء الجينات باستخدام الخوارزميات المستوحاة من البيولوجيا. أولاً ، قمنا بتطوير طريقة لحل مشكلة محاذاة التسلسل المتعدد ، تسمى خوارزمية سرب الأسماك الاصطناعية متعددة الأغراض (MOAFS) ، وذلك باستخدام سلوكيات خوارزمية سرب الأسماك الاصطناعية ، مجموعة باريتو المثلى ، والعمليات الوراثية. ثانياً ، اقترحنا خوارزمية لحل مشكلة انتقاء الجينات باستخدام المعلومات المتبادلة ، وخوارزمية تحسين عثة اللهب (فراشة الليل) ، وتقنية التحقق من الصحة المتقاطع مع دعم جهاز المتجه (SVM-LOOCV). يطلق عليها تعظيم المعلومات المتبادلة و تعديل خوارزمية تحسين أداء عثة اللهب (MIM-mMFA) والتي تتكون من مرحلتين بسيطتين.

تناولت الأطروحة اختبارا كاملاً لـ MOAFS على مجموعات البيانات المرجعية للمحاذاة الخاصة بـ BaliBASE 2.0 و BaliBASE 3.0، كذلك اختبار MIM-mMFA على ستة عشر مجموعة من بيانات تعبير الجينات السرطانية ثنائية ومتعددة الطبقات. أخيرا ، قدمنا نظرة عميقة حول أداء كل خوارزمية. بالإضافة إلى ذلك ، حققت الخوارزميات المقترحة نتائج تنافسية أو أفضل من الخوارزميات الراسخة في هذا المجال.

الكلمات الرئيسية: المعلوماتية الحيوية ؛ الخوارزميات المستوحاة من البيولوجيا ؛ محاذاة التسلسل المتعدد ؛ خوارزمية سرب الأسماك الاصطناعية ؛ انتقاء الجينات ؛ تعبير الجينات ؛ رقائق الحمض النووي ؛ تصنيف السرطان ؛ خوارزمية تحسين عثة اللهب.