International Journal of Cognitive Informatics and Natural Intelligence
Volume 15 • Issue 4 • October-December 2021

Bio-Inspired Data Mining for Optimizing GPCR Function Identification

Safia Bekhouche, Laboratoire de Recherche en Informatique (LRI), Badji Mokhtar University, Algeria
Yamina Mohamed Ben Ali, Laboratoire de Recherche en Informatique (LRI), Badji Mokhtar University, Algeria

ABSTRACT

GPCRs are the largest family of cell surface receptors; many of them remain orphans. The GPCR functions prediction represents a very important bioinformatics task. It consists in assigning to the protein the corresponding functional class. This classification step requires a good protein representation method and a robust classification algorithm. However, the complexity of this task could be increased because of the great number of GPCRs features in most databases, which produce combinatorial explosion. In order to reduce complexity and optimize classification, the authors propose to use bio-inspired metaheuristics for both the feature selection and the choice of the best couple (feature extraction strategy [FES], data mining algorithm [DMA]). The authors propose to use the BAT algorithm for extracting the pertinent features and the genetic algorithm to choose the best couple. They compared the results they obtained with two existing algorithms. Experimental results indicate the efficiency of the proposed system.

KEYWORDS

BAT Algorithm, Classification, Data Mining Algorithms, Feature Extraction Strategies, Feature Selection, Function Prediction, Genetic Algorithm, GPCR

1. INTRODUCTION

The identification of G-protein coupled receptors (GPCRs) function is an area of current interest in pharmaceutical and biological research. Of the approximately 500 clinically marketed drugs, greater than 30% are modulators of GPCR function, making GPCRs the most successful of any target class in terms of drug discovery (Drews 2000).

Intense efforts have been devoted to identifying new GPCR functions for orphans. However, for many GPCRs, such efforts have failed to yield reliable results.

At this stage several questions have been asked: what are the necessary steps for good protein function identification? What is the adequate protein representation method (PRM) that can be used to extract features and construct numerical attribute vectors? Which Data Mining Algorithm (DMA) that should be selected to make an accurate classification? How to avoid the combinatorial explosion of classification algorithms due to the complex nature of protein data?

DOI: 10.4018/IJCINI.20211001.oa40

Although many GPCR function prediction approaches have been proposed, a great number of GPCR are still orphan. The previous common methodology is sequence similarity searching in protein databases which is mainly based on pairwise sequence alignment such as

BLAST (Zhang et al., 2012). But it is difficult to identify GPCR successfully because there are no significant shared sequence similarities. However, two proteins can have very different sequences and perform a similar function, or have very similar sequences and perform different functions (Nemati et al., 2009). To solve this problem, some statistical and machine learning approaches have been developed (Secker et al., 2007).

There are three major problems in the task of computational protein function prediction with classification algorithms, which are the choice of the classification algorithm and the choice of the PRM, also the selection of relevant attributes to avoid the combinatorial explosion problem. Those are open problems, even in any classification problem as there are many choices and it is not clear which one is the best.

Generally, there are several strategies to extract attributes from a protein sequence, and the choice of the PRM might be as important as the choice of the DMA, contrary to few works (King et al., 2001) that are often overlooked the used feature extraction strategy and more focused on which classification algorithm to use. Other researchers have developed a hybrid feature extraction strategy (Rehman & Khan, 2011) that can exploit both pseudo-amino-acid composition strategy (PseAAC) and multiscale energy representation, while some authors (Secker et al., 2010; Naveed & Khan, 2012) have made a comparison of the predictive accuracies of few PRM in protein classification.

The transformation of the protein chain can give an enormous numerical attribute vector, the size and the components of this later, strongly influences the predictive accuracy and the error rate of the classification. To improve these rates it's strictly necessary to eliminate noises "redundancies or useless information" present in the examples to be classified. Furthermore, datasets with hundreds and thousands of attributes may cause the curse of dimensionality and combinatorial explosion problems (Chen et al., 2014).

One of the most feasible techniques to cope with this problem is feature selection (FS) (Sayes et al., 2007; Bagherzadeh-Khiabani et al., 2016) to optimize the classification model and improve the performance measurements. This technique is widely used in different fields to improve results such as: protein function prediction (Nemati et al., 2009) and it is mostly used in big data and data mining (Li & Liu, 2017; Tupe & Wakchaure, 2017).

Several researchers aim to optimize the GPCR classification, either by selecting of attributes and classifiers (Secker et al., 2010), or by using the bio-inspired meta-heuristic (Naveed & Khan 2012; Holden & Freitas 2008; Holden & Freitas 2009; Nemati et al., 2009). In this work, we will use two different bio-inspired methods that show their efficiency in several areas (Nayyar et al., 2018; Nayyar & Nguyen, 2018), one is a meta-heuristic optimization approach: the Bat Algorithm introduced by Yang (Yang 2010) and used in (Yang & Gandomi 2012) for global engineering optimization, (Gandomi et al. 2013) for constrained optimization tasks, (Cai et al. 2019) for large scale optimization and in (Guo et al. 2019) for solving global function optimization problem. The second is an evolutionary approach which is the genetic algorithm has been used for a large number of modeling applications in chemical and biological fields (Pedersen & Moult 1996, Judson 1997), moreover Judson developed a GA for protein structure prediction (Judson 2008) and in (Santos et al. 2019) authors propose a methodology using a multi-objective GA for feature selection to predict protein function.

In this paper, the authors address three problems: the feature extraction strategy (FES) selection, the data mining algorithm (DMA) selection and the feature selection. We used two different bio-inspired algorithms which are: Genetic Algorithm (GA) for both (FES, DMA) selection and BAT algorithm for FS.

The remainder of this paper is organized as follows: Section 2 presents a brief literature review concerning an overview of the protein function prediction and the five different PRM employed in this work. Section 3 presents the bio-inspired framework to optimize GPCR classification model.

Section 4 presents the experimental setup for the experiments, the computational results and their discussion and finally, in Section 5 we state our conclusions and future works.

2. LITERATURE REVIEW

2.1. Overview of The Protein Function Prediction Problem

The discovery of the protein function is a major line of research in genomics, since biological processes are activated by these molecules (Alberts et al., 2002; Duc et al., 2017) For example, hemoglobin is a protein that carries oxygen through the blood. Knowledge of the protein sequence is also important in determining pathogenic abnormalities. The modification of an amino acid can in certain cases have harmful consequences. Therefore, Knowledge of protein functions is very important in biomedical sciences, not only for a better understanding of cell biology in general, but also because many diseases are caused by or at least associated with defects in protein functions (Silla & Freitas, 2011). Hence, an effective method for the prediction of protein functions can potentially contribute to generate new biological knowledge that can lead to a better treatment and diagnosis of diseases, design of more effective medical drugs.

Although many GPCR prediction approaches have been proposed during past two decades, in many cases, conventional bioinformatics techniques, such as pairwise sequence alignment or by comparing sequences to motifs are undoubtedly valid, they cannot be optimal when it comes to identifying GPCR. First, the sequence of a GPCR super-family varies in length (between 290 and 1,200 amino acids), which means that many of the subfamilies cannot be aligned effectively. Sophisticated computation is therefore a more efficient approach to the problem of the GPCR classification, using techniques based on data mining and machine learning.

The classification methods used for identifying the GPCRs function are many and varied. We can distinguish two essential approaches: methods based on hierarchical classification with their two types (tree or Directed Acyclic Graph (DAG)) (Secker et al. 2007; Costa & al. 2008; Freitas et al. 2007; Nakano et al. 2017; Secker et al. 2010 ; Silla & Freitas 2011) and methods based on the standard classification (flat classification) such as the C4.5 algorithm used in (Huang et al. 2004), HMM (Munoz et al., 2017) and SVM classifier (Kumari et al. 2010; Shrivastava et al. 2010).

Online tools have been developed as well. For instance, GPCRPred (Bhasin & Raghava, 2004) based on SVM method for prediction of two levels for GPCRs families and sub-families. On the basis of the dipeptide composition from the primary amino acid sequence, PRED-GPCR (Papasaikas et al., 2004) provides a complement to the existing pattern database analysis servers and potentially a computational tool for GPCR family classification (Zekri et al. 2011), GPCRTree is an online hierarchical classifications webserver (Davies et al., 2008). It is the first server to implement an alignment-independent representation of protein sequences and is also the first to classify sequences using a hierarchical classification, PCA-GPCR (Peng et al., 2010), and the most best web service is GPCR-MPredictor (Naveed and Khan 2012) that based on evolutionary approach and predicted GPCR at five level.

An additional category of methods consisting on the bio-inspired approaches is used in molecular biology field. In (Secker et al., 2009) authors propose an Artificial Immune System (AIS) that solves the problem of clustering to find the optimal grouping of amino acids for the protein type. This method achieves an accuracy of 72.75% at the third level, this result marks a good improvement compared to the top-down approach (Secker et al., 2007), which provided a predictive accuracy of 70.46%. (Gu and Ding 2009) use a Swarm Intelligence approach that they operate, more precisely, the Binary particle swarm optimization algorithm (BPSO) which has better performance optimization on discrete binary variables (98.02%) than the PSO. Furthermore, There is Holden and Freitas's works (Holden & Freitas, 2006) using a bio-inspired approach for hierarchical classification, specifically, the hybrid algorithm PSO/ACO (Particle Swarm Optimization / Ant Colony Optimization) which provided an accuracy of 89.64%. Also in (Correa et al., 2007) authors use the Discrete Particle Swarm Optimization (DPSO)

for the task of attributes selection and for application to complex protein functional classification data set. The results show that the selection of attributes provides better performance than if we used all the set of attributes.

2.2. Protein Representation Methods

In this section, we will describe the techniques used and evaluated in this work for protein sequence representations.

2.2.1. The Amino Acid Composition (AAC) strategy

This method has been widely used in this field (Gao et al. 2013; Kumar et al. 2015; Kumari et al. 2010; Secker et al. 2009; Shrivastava et al. 2010), It provides numerical vectors of 20 components, with each reflecting the occurrence frequency for the 20 amino acids as follow: $P = \{A_1, A_2, A_3, \ldots, A_n\}$, it can be expressed by: $V = \{ f_1, f_2, f_3, \ldots, f_{20}\}$ such as: P is the amino acids sequence and V is the frequencies vector calculating using Eq. (1).

$$f\left(Ai\right) = \frac{N\left(Ai\right)}{N} \tag{1}$$

Where N represents the sequence length and $N(A_i)$ is the total number of amino acid A_i present in the sequence. This technique is very simple and easy to implement but all the sequence-order information is lost.

2.2.2. Dipeptide Composition (DC)

The dipeptide components (pair of amino acids: *AA, AC, .., AY, CA, CC,..,CY...YA,...YY*) are interesting parameters for protein representation that can preserve the order of each two consecutive amino acids. They have been obtained using Eq. (2). (Khan et al., 2017; Kumar et al., 2017; Li et al., 2010)

$$f\left(a_i a_j\right) = \frac{N\left(a_i a_j\right)}{\sum_{i=1}^{20}\sum_{j=1}^{20} N\left(a_i a_j\right)} \tag{2}$$

2.2.3. The Pseudo-Amino Acid Composition (PseAAC)

This method was developed by (Chou 2001) to formulate an amino acid sequence of arbitrary length, such as a digital vector. A protein sequence with length L amino acid residues $(R_1, R_2, R_3, \ldots, R_L)$ where R_1 represents the residue at sequence position 1, R_2 represents the residue at position 2 and so on, may be denoted as a $(20 + \lambda)$-dimensional vector, defined by $20 + \lambda$ discrete numbers. (Cheng et al., 2017; Khan et al., 2015; Kumar et al., 2015). In our case the PseAAC method provides an attribute vector of 50 dimensions. It has been rapidly and widely used in nearly all the areas of computational proteomics (Liu et al., 2017).

2.2.4. The Amphiphilic Pseudo-Amino Acid Composition (Am-PseAAC)

This method try to save as well as the sequence order information using the different amphiphilic features corresponding to different hydrophobic and hydrophilic order patterns (Chou 2005). $Am\text{-}PseAAC = \{A_1; \ldots; A_{20}; A_{21}; \ldots A_{20+\lambda}; A_{20+\lambda+1}; \ldots\ldots A_{20+2\lambda}\}$ Where: The first 20 elements are the occurrence frequencies of the 20 amino acids. 2λ discrete numbers reflect the amphiphilic sequence correlation

along a protein chain. According to the λ value, in this case the Am-PseAAC method provides an attribute vector of 80 dimensions. (Chou and Shen, 2006; Rehman et al., 2013; Khan et al., 2010).

2.2.5. The Local Descriptors (LD)

This method is known as global protein sequence descriptors, were widely used in (Cai et al., 2003; Cui et al., 2007; Davies et al., 2008; Secker et al., 2010; Silla & Freitas, 2011; Tong & Tammi, 2008), The GPCR sequence has been transformed into a numeric attributes vector of dimension 210D.

• The principle of this method could be decomposed into two necessary stages:

1. The first step consists in transforming a protein chain *Pi* into another secondary chain *P'i* formed only by symbols *H, N* and *P* following the belonging of each amino acid to its group.
2. The second step is to take this new chain and perform a set of treatments that are calculating the rate of each descriptor *C, T* and *D*.
   i. Encoding of the sequence: The amino acids (AA) are divided into three functional groups, each AA is coded by one of the indices *H*= Hydrophobic (contain C V L I M F W amino acids), *P*= Polar (contain R K E D Q N amino acids) and *N*= Neutral (contain G A S T P H Y amino acids) according to the class to which it belonged.

   The position or variation of these groups in a sequence is the basis for calculating the three local descriptors: Composition *(C)*, Transition *(T)* and Distribution *(D)*.

   ii. Calculating Composition, Transition and Distribution descriptors

   This step serves to find the values of the descriptors so that, starting from a secondary chain P'_i for a given sequence P_i, we will calculate the three descriptors *C, T* and *D* for each amino acid group *H, N* and *P*. Once this calculation effected, the numeric vector will be obtained. Since the amino acids are divided into three groups, the calculation of *C, T* and *D* descriptors generates in total 21 attributes (3 for *C*, 3 for *T* and 15 for *D*). Although this technique is valid if applied throughout the amino acid sequence, we have divided the amino acid sequences into 10 regions [11, 44]. Each descriptor *C, T* and *D* is computed on the 10 subsequences, that giving 210 attributes describing the protein.

- *"Composition C"* Descriptor: This is the overall percentage of each group in the sequence, we can define it by Eq. (3):

$$C\left(x\right) = \frac{n\left(x\right)}{N} \quad (3)$$

Where: x = H, N, P

n(x) = Amino acids number of type x.
N = Sequence size.

So for the composition, we extracted the three first attributes component our vector which are: *C (H), C (P)* and *C (N)*.

- *"Transition T"* Descriptor: The transition from group *H* to *P* is the percentage of frequency at which *H* is followed by *P* or *P* is followed by *H* in the coded sequence, in other words it's the frequency with which the amino acids belonging to a group are followed by amino acids belonging to a different group. We can compute it using Eq. (4):

$$T_x = \frac{N_{rx} + N_{xr}}{N - 1} \tag{4}$$

Where: x ϵ (S = { H, P, N });

r = S - x; N = Sequence size.
N_{xr}, N_{rx} = Number of dipeptide (two amino acids) encoded.

So for the transition, we extracted three other attributes belonging to our vector that are: *T (H), T (P)* and *T (N)*.

- *"Distribution DT"* Descriptor: The distribution descriptor describes the apportionment of each group in the sequence. There are five "distribution" descriptors for each group and these are the percentages of position in the complete sequence for the first, 25, 50, 75 and 100% of the occurrences of a specified group.

At this stage, one feature extraction strategy must be chosen among used PRM for the numerical representation of the GPCR sequences.

3. BIO-INSPIRED FRAMEWORK FOR OPTIMIZING GPCR FUNCTION IDENTIFICATION

As mentioned earlier, in this paper we propose a genetic algorithm for selecting the best couple (FES, DMA) and we adapt the BAT algorithm to solve the FS problem in GPCR function prediction. The main steps of our general architecture system are illustrated in Figure1.

As showed in figure above, the proposed approach starts with a preprocessing step that transform data from a noisy structure to more exploitable form. Next, the feature extraction step transforms each protein sequence from alphabetic to numerical vector. There are several PRM to ensure this transformation. The size of produced numerical vector depends on the adopted FES. After that, the BAT algorithm is used to avoid the combinatorial explosion of DMA, saving time and memory space. This is due to huge size of the numerical protein vector produced in the precedent step. Finally, we use GA to find the best combination between DMA and FES that gives the best efficiency of classification in term of accuracy and error rate. Next sections detail all these proposed steps.

3.1. Pretreatment

This step is devoted to automating the transformation of unstructured stored data from a source file into structured data saved in a database (Figure2), where each sequence has its necessary information: identifier, description, family, sub -family, sub-sub-family also its type. The preprocessing provides a better representation which can be exploited in the subsequent stages.

3.2. Protein Representation

This step transforms the alphabetic chains of protein sequences into digital vectors. These vectors contain features that will be used for building the classification model.

According to Figure3 we can distinguish three necessary treatments explained later.

3.2.1. Selecting Among PRM

Several Protein Representation Methods exist in the literature, each one has its strengths and its weakness. The choice of one method or another is not obvious since it depends on various factors

Figure 1. Steps of the proposed system.

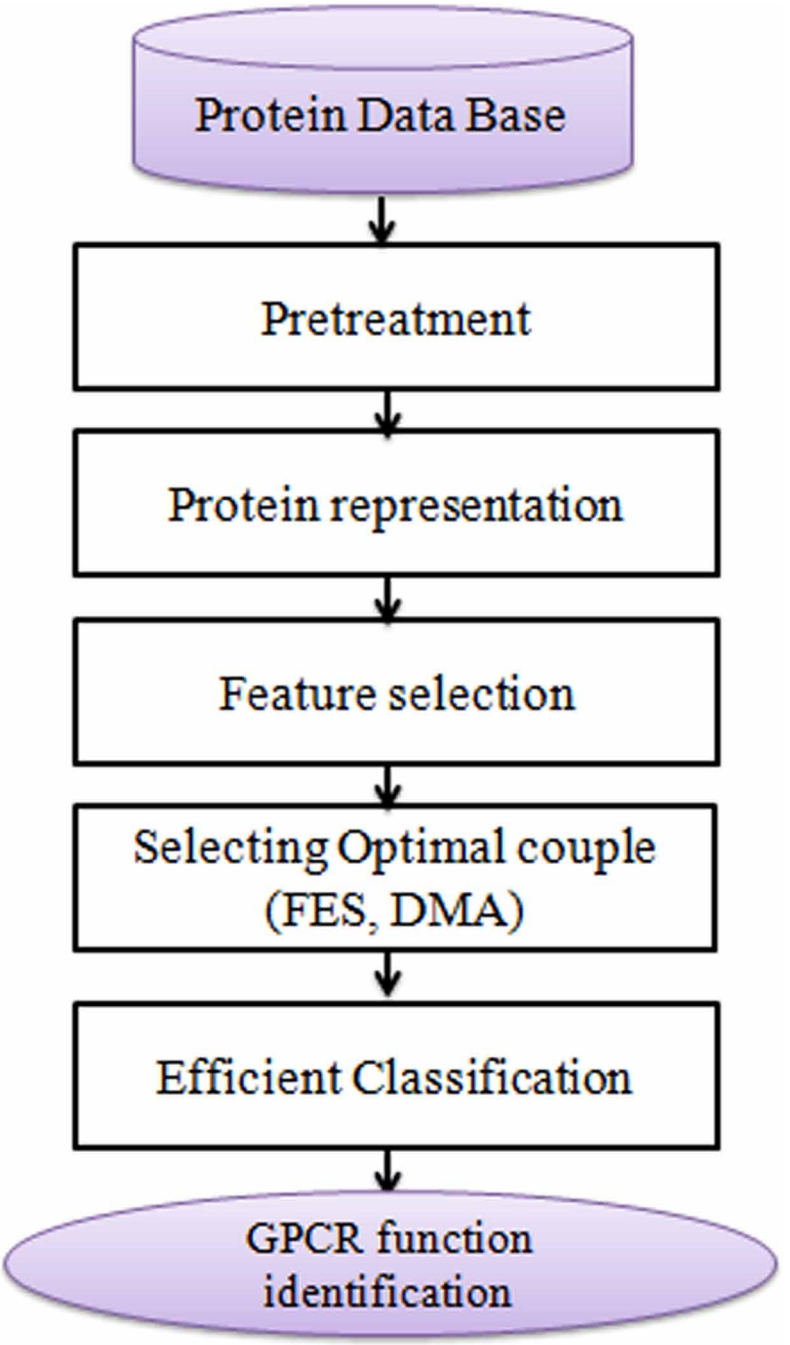
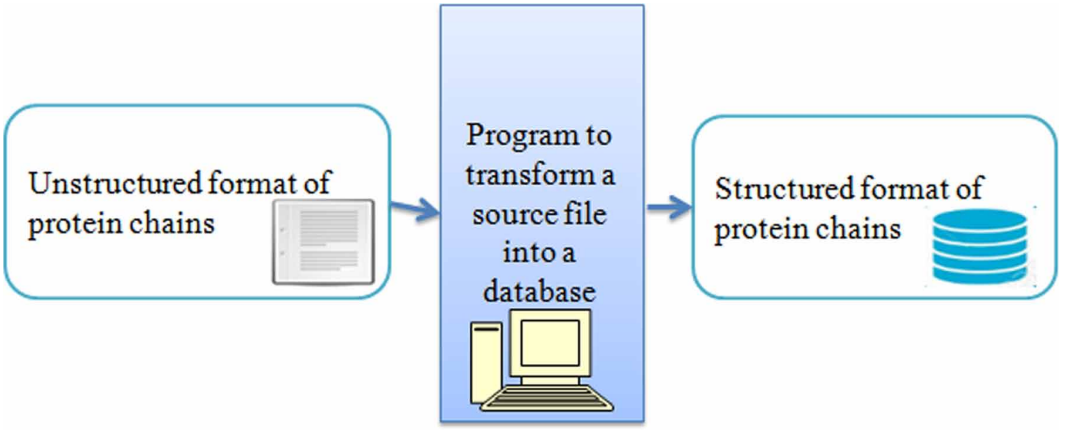


Figure 2. Pretreatment of data.



such as the produced features number, and the information existing in the numerical vectors (Amino Acids order, sequence length …), for this reason, we used genetic algorithm to choose the best feature extraction strategy with the efficient classifier. This, allows to study and observe the variation in the results of the classification.

3.2.2. Construct The New Protein Database

Once a PRM is chosen, we will have a digital vector whose size is closely linked to the selected method. The result will be a training database which contains only the vectors of numeric attributes.

Figure 3. The steps required for feature extraction

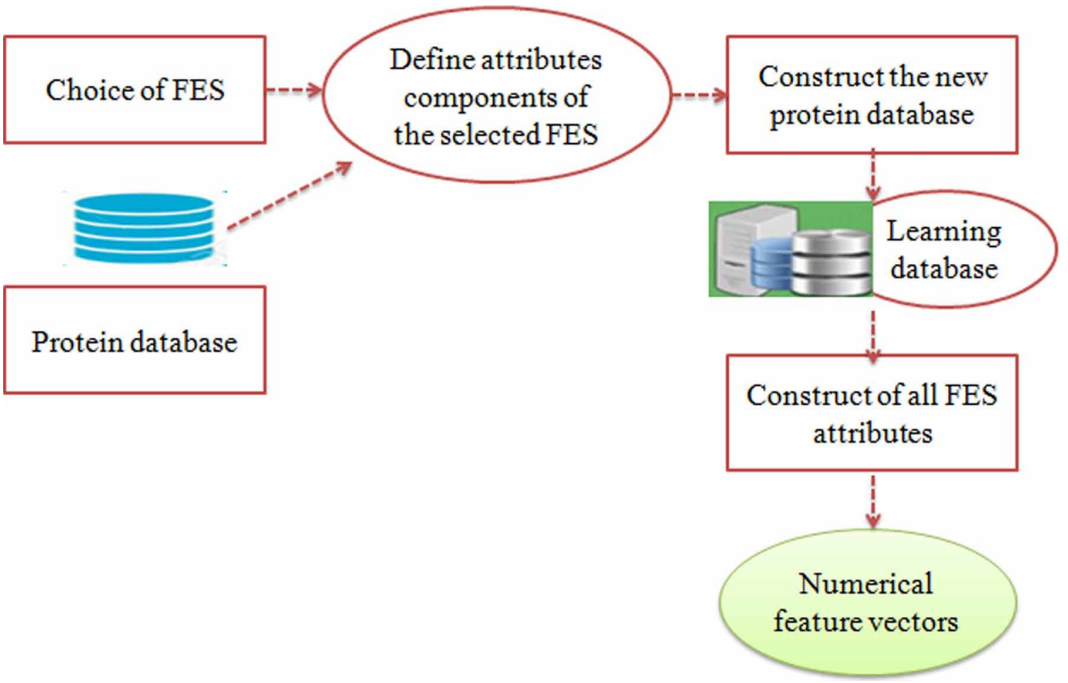
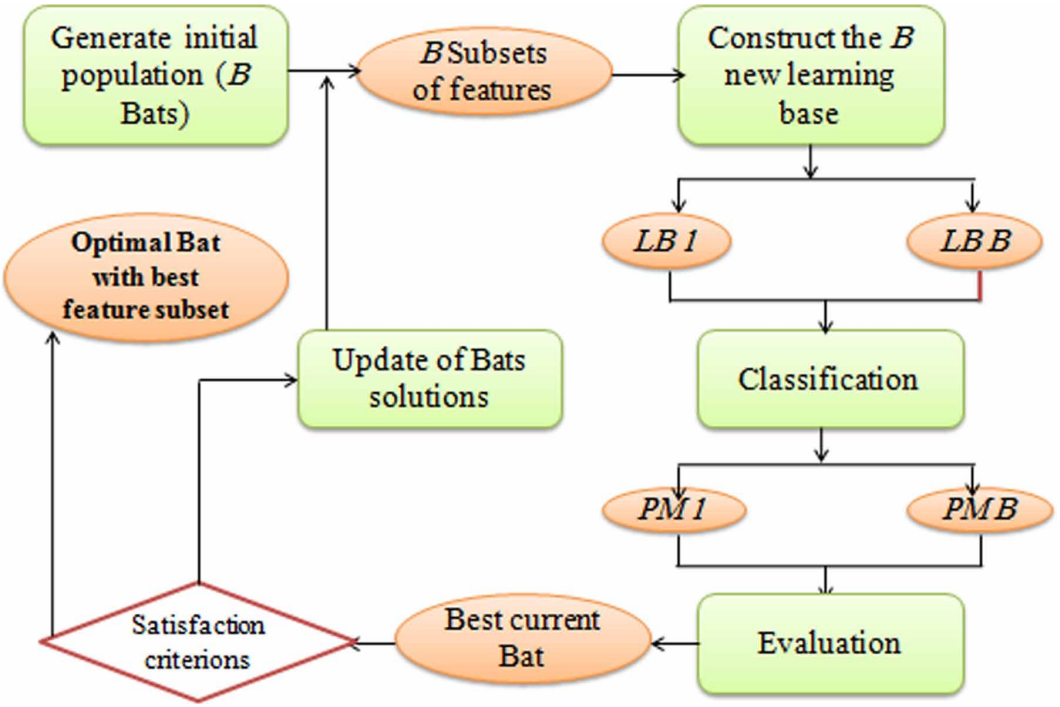


Figure 4. Feature Selection process diagram using BAT Algorithm.



Technically, the latter is in the form of an .arff file which begins with a declaration of all the attributes and then the digital representation of each protein sequence, therefore its family, its sub-family and its sub-sub-family. This file will be the entry into our system to perform the classification step using the Weka package.

3.2.3. Construct of all PRM features

In our work, we have to use the most five commonly used FES in the literature. For each method, we must construct the corresponding learning base in a form of an .arff file as explained previously. Therefore, the feature extraction module produces five learning bases.

The size of a product feature vector varies from a PRM to another and can reach 400D for DC strategy, which may be quite difficult to solve the classification task when we have a huge number of attributes in a given training data.

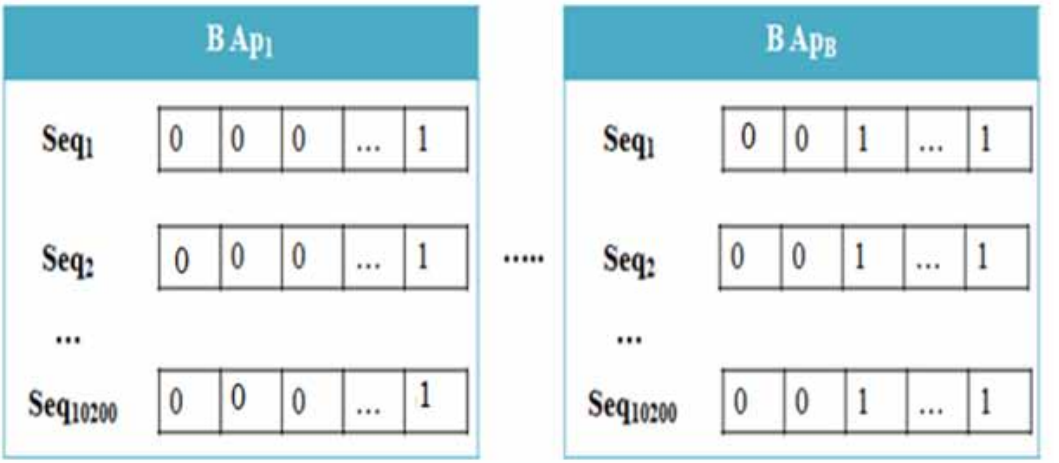
FS may bring lot of advantages such as improving predictive accuracy, avoiding over-fitting, distinguishing pertinent attributes from less important ones and providing a concise understanding of data.

Subsequently, we will detail the step of the attributes selection using bio-inspired meta-heuristic the BAT algorithm.

3.3. Feature Selection

This step focuses on finding the pertinent attributes for each learning base, by eliminating those that are unimportant. A feature selection algorithm called BAT is proposed to generate an optimal subset of attributes from an input numerical vector; it will produce a vector of smaller size with the best attributes at the output.

Figure 5. Randomly initialization of B Bats.



The existing scheme in Figure4 presents the diagram of FS using BAT algorithm. It recapitulates the steps required to complete this process, which we will explain later.

- **Create the initial population:** This initialization phase serves to create an initial population of B bats, each individual represents a solution to the problem that will be initialized randomly by vectors (VB) of binary values where some attributes are activated and the remaining are inactivated.

- **Construct the B new learning base (LB):** Before classification, it is necessary to transform the database into a learning base with a binary presentation for each protein sequence; each attribute having the value ”1” will be taken into consideration.

The vectors number in each learning base (LB) is equal to the number of GPCR sequences in the database. In our case, concerning the LD method, we have attributes vectors of 210D, and a database contains 10200 GPCR sequences, the Figure5 presents the construction of the population having B bats.

After representing Bats and learning bases, we have to initialize parameters corresponding of each one according to following formulas
Such as:

i. *fi* is the pulse emission frequency of the bat, and belongs to the range $[f_{min}, f_{max}]$ corresponding to the wavelength range $[\lambda_{min}, \lambda_{max}]$. In order to simplify the implementation, it has been assumed

Box 1. Formulas

Frequency:	$f_i = f_{min} + (f_{max}-f_{min})\beta,$	**(5)**
Velocity:	$v_i^t = v_i^{t-1}+(x_i^t-x_*)f_i,$	**(6)**
Position:	$x_i^t = x_i^{t-1} - v_i^t$	**(7)**

Table 1. Algorithm 1: BAT Fitness Function

Input: **Position = An Attributes vector; Algo = A chosen classifier;** **Output:** **MinER = Error Rate: Calculated using Eq. (9)** **MaxAcc = Accuracy: Calculated using Eq.(8)** **Begin** **Generate curFile an arff file using activated attributes;** **Compute ER, ACC using Algo on curFile;** **End.**

that $f \epsilon [0, f_{max}]$ Knowing that high frequencies correspond to short wavelengths. For bats, the typical ranges are a few meters. Consequently, the pulse emission rate can be in the range *[0, 1]* whither 0 means that it has no pulsation, and 1 means the maximum rate of pulse emission.

ii. $\beta \epsilon [0, 1]$ is a random vector retrieved from a uniform distribution.

iii. x_* is the best overall position (solution), which will be calculated by comparing all the solutions obtained by each bat.

iv. For local search, once a solution is selected among the best current solutions, a new solution for each bat is generated locally using the random path.

- **Classification:** To evaluate the relevance of the selected attributes, we made a classification step to test its predictive accuracy and its error rate. As much as the error rate is minimal and the accuracy is maximum, the chosen attributes are optimal.

In our work the classification is done using nine DMA: *NB,BN,KNN,J48,DT,RF,BAG,LB and ZR* implemented in Weka package. For all bats of each iteration, we make calculations of performance measurements so that we can compare them and deduce the best bat (best overall solution).

- **Evaluation of population:** To verify the credibility of the current solution, it is necessary to make a local evaluation to extract the best solution in the current iteration and an overall evaluation to obtain the best solution provided from all the iterations, which is done using the objective function based on classification step. Table 1 presents the used algorithm for calculating the fitness function.

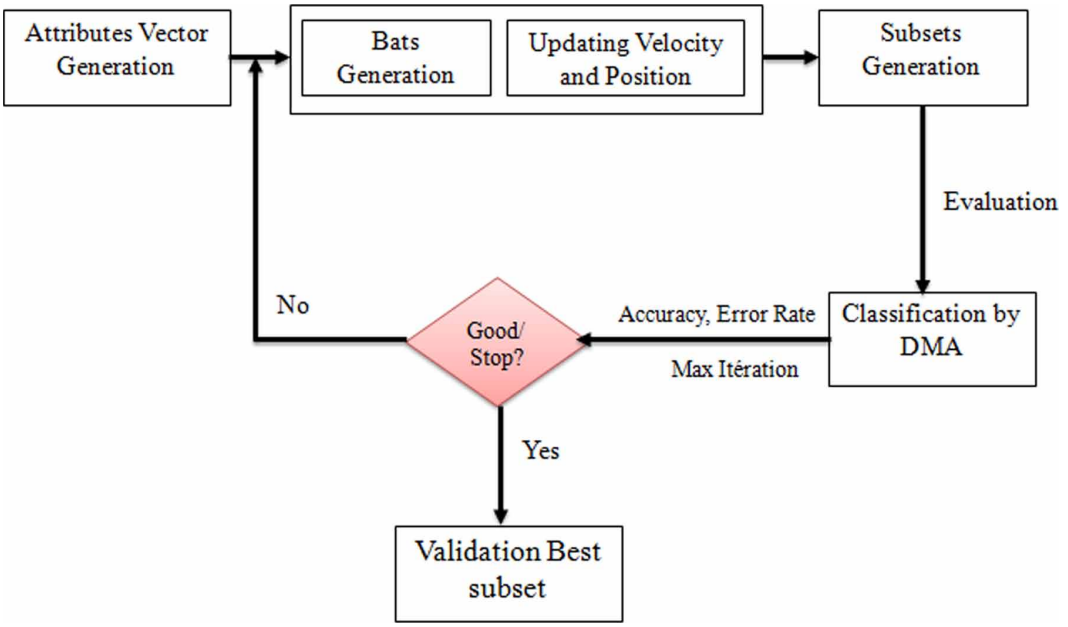
- **Update of Bat solutions:** This step is the most crucial, indeed, the movement of bats is responsible for the effectiveness of the algorithm. The Eq. (5), Eq. (6) and Eq. (7) mentioned above defining the new solution and updating the position and velocity of each bat in a space of dimensions. The next figure (Figure6) shows the flowchart of Bat algorithm.

We adapt the Bat algorithm for the attributes selection in two-phase process. The first phase, called the initialization, is used to construct an initial solution to start the search. Furthermore, we adjust the parameters of this algorithm in order to obtain an initial solution not too bad. In the second phase, which we call improvement phase, we will introduce local search in order to improve the quality of the solution returned in the initialization like mentioned in Table 2.

3.4. Selecting Optimal Couple (FES, DMA)

This section is dedicated to explaining the module responsible of using GA to choose the best FES with the convenient DMA (see Figure 7) as well as all realization steps. The provided results show an optimal classification with the most minimal error rate and maximum accuracy.

Figure 6. The flowchart of Bat algorithm.



GA operates on a population of individuals to produce better approximations. At each generation, a new population is created by the process of selecting individuals according to their level of fitness, and recombining them together using operators borrowed from natural genetics (crossover and mutation). A flow diagram for the training process with the GA is depicted in Figure8.

In our case, each individual in the population represents a predictive model. The number of genes is the total number of used FES and DMA. Genes here are binary values, and represent the activation or not of particular FES, DMA in the model. The number of individuals, or population size, must be chosen for each application. Usually, this is set to be N*M, such as: N the FES number and M the DMA number. Now we are going to describe in detail the operators and the corresponding parameters used by the GA mentioned in table 3.

- Initialization: The first step is to create and initialize the individuals in the population. As the genetic algorithm is a stochastic optimization method, the genes of the individuals are usually initialized at random. In order to illustrate this operator, consider a predictive model represented by the following figure (Figure9). If we generate a population of 4 individuals, then we will have 4 different random FES and DMA. The Figure 9 illustrates this population.

As we can see, each individual is represented by 14 binary genes. The first five genes are devoted to the FES representation and the remaining are dedicated to the DMA. Each positive gene means that the corresponding Strategy/Algorithm is included in the model.

Fitness assignment: Once we have generated and initialized the population, we need to assign the fitness to each individual. To evaluate the fitness, we need to train the predictive model with the training data, and then evaluate its classification error rate and predictive accuracy with the selection individuals. Obviously, a high classification error rate means a low fitness. Those individuals with greater fitness will have a greater probability of being selected for recombination. The fitness value assigned to each individual will be calculated using Algorithm 4 presented in Table 4.

The following example “illustrated in Table 5” depicts the classification error rate (ER) and the accuracy (ACC). Note that the corresponding objective function of each individual is difference between the ACC measurement and the ER.

Table 2. Algorithm 2: BAT algorithm adapted to the FS

Input:
nb = Bats number (population size). na = Number of selected attributes
MaxIter= Maximum Iteration. N = Initial vector size. Algo = Choosing a DMA algorithm.
Output:
S = Subset of selected attributes *(A1.....A_{na})* having the best fitness
Begin
Initialization
Initialize the bats number at nb;
Initialize attributes number at na;
Bat generation;
Randomly initialize the overall fitness FGlob;
For each bat do
Declare a vector *vect* of *N* attributes initialized with the false value;
Randomly activate an attributes subset of vect with size *na* by setting their values to true;
Current position = *vect;*
Initialize random frequency, velocity *(f, V)* using Eq. (5), Eq. (6) respectively;
Initialize $r = 1, A = 1$;
Compute the local fitness *FLoc* "using Algorithm 1 in table 1";
If *FLoc* > *FGlob* then
Updating *FGlob;*
End If; End For;
Improvement
Repeat;
For each bat do
Updating the velocity of bat;
***f* = Compute *FLoc* of bat "using Algorithm 1 in table 1";**
If *f* > *FLoc* then
Updating *FLoc;*
Updating the local position *vect* using Eq. (7);
End If;
If *f* > *FGlob* then
Updating *FGlob;*
Updating overall position;
Updating *r;*
End If; End For;
Until (Max iteration)
End.

Figure 7. General architecture system.

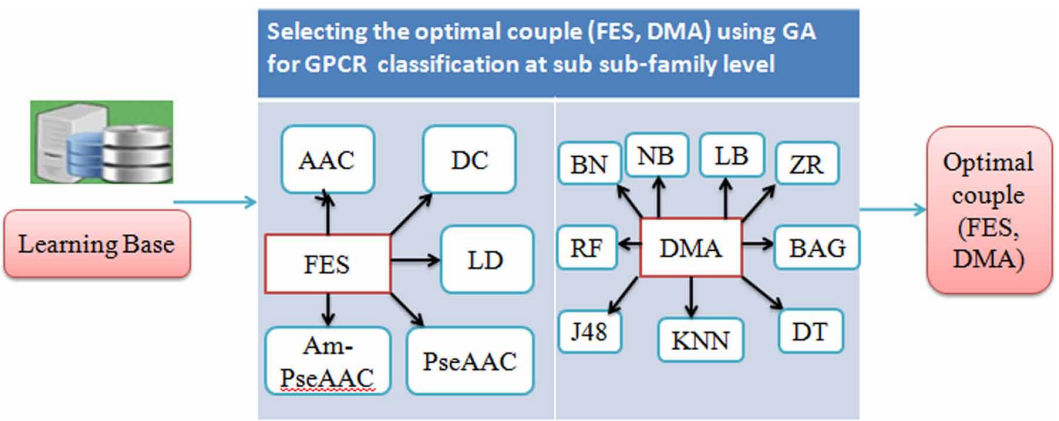


Figure 8. Flow diagram of GA Search.

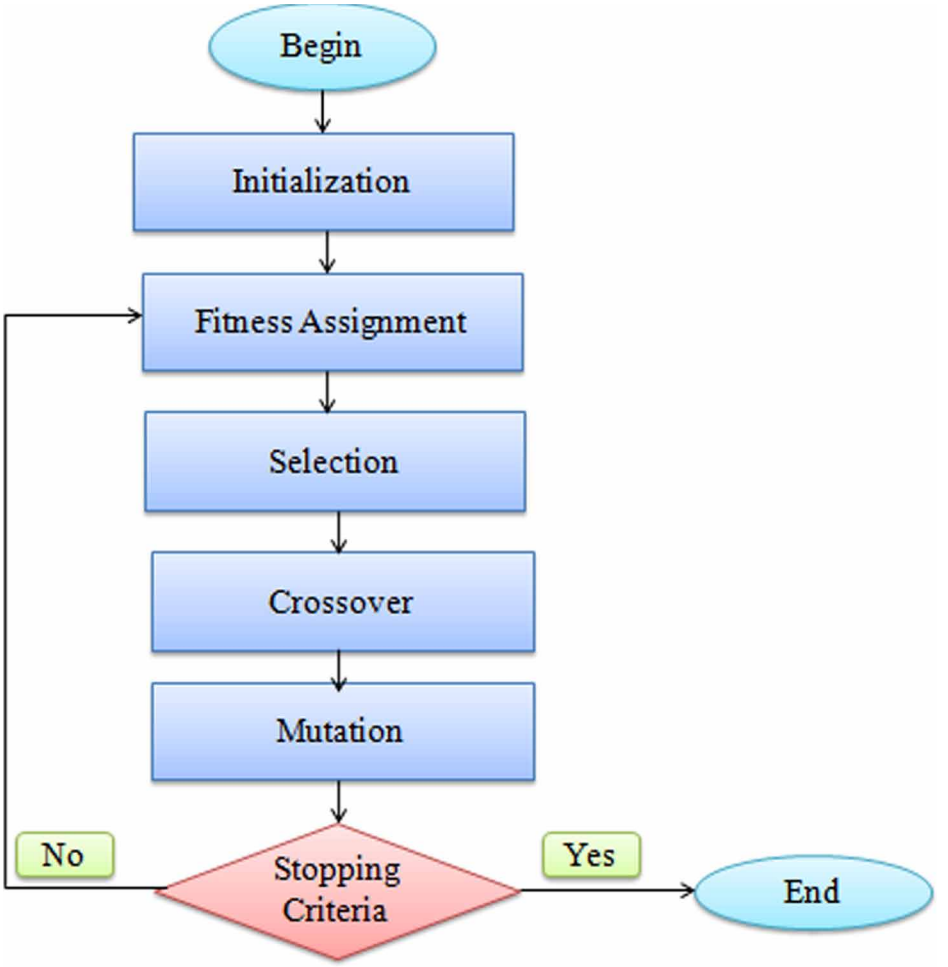
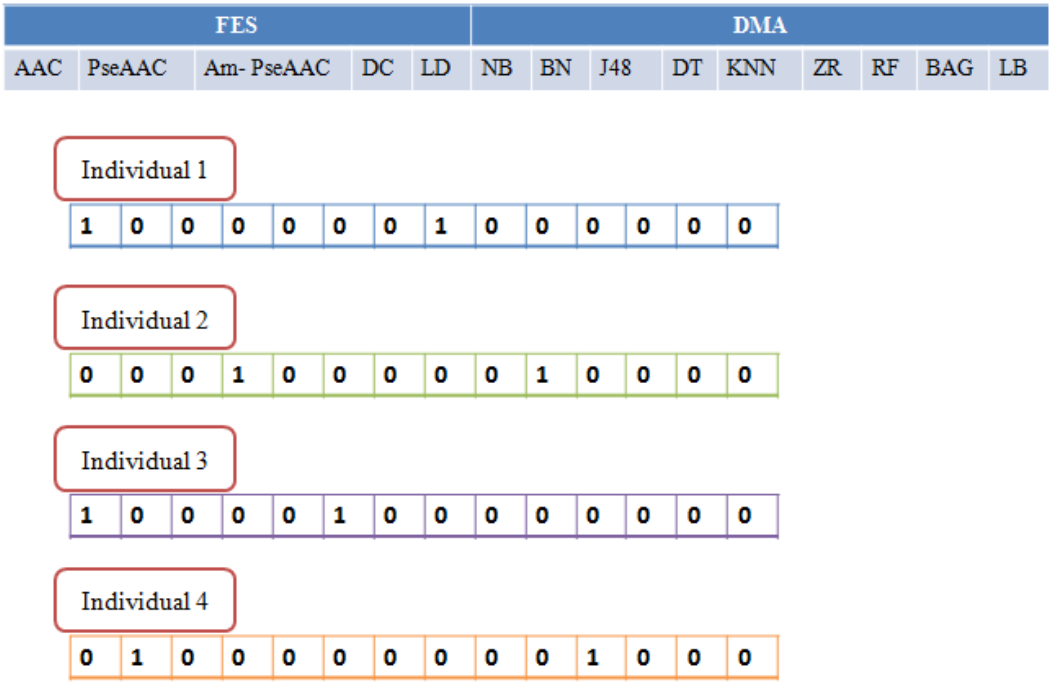


Table 3. Algorithm 3: GA search for selecting the best (FES, DMA)<Tbl_Large></Tbl_Large>

Input:
FES = AAC,PseAAC,Am-PseAAC,DC and LD;
DMA = BN,NB,J48,KNN,DT,RF,BAG,LB and ZR;
MaxIteration; ProbCross; ProbMutation;
Output:
The best set of *(FES, DMA)*
Begin
Initialize Population
While not MaxIteration *do*
Evaluate each individual's fitness using Eq (10).
Selection (CrossOver & Mutation)
Survival (Select a new Population)
EndWhile.
End.

Figure 9. Predictive model for FES and DMA selection.



- Selection: After fitness assignment has been performed, the selection operator chooses the individuals that will recombine for the next generation. The individuals most likely to survive are those more fitted to the environment. Therefore, the selection operator selects the individuals according to their fitness level giving a chance to bad individuals. The number of selected individuals is S/2, being S the population size.
- Crossover: Once the selection operator has chosen half of the population, the crossover operator recombines the selected individuals to generate a new population. This operator picks two individuals at random and combines their features to get four offspring for the new population, until the new population has the same size than the old one.

  We choose the single point crossover method, fixed at point 5 of each individual. The next figure (Figure10) illustrates the crossover step for our example. Here we have generated two children from two selected parents.
- Mutation: The crossover operator can generate children that are very similar to the parents. This might cause a new generation with low diversity. The mutation operator solves this problem by changing the value of some genes in the children randomly (but we required flipping genes either before the crossover point (point 5) or after it).

  The mutation operator alters the characteristics of a solution in a completely random manner,

0	0	0	1	0	1	0	0	0	0	0	0	0

which allows us to introduce and maintain diversity within our population of solutions, it introduces "noise" within the population. In order to decide if a gene will be mutated, we generate a random

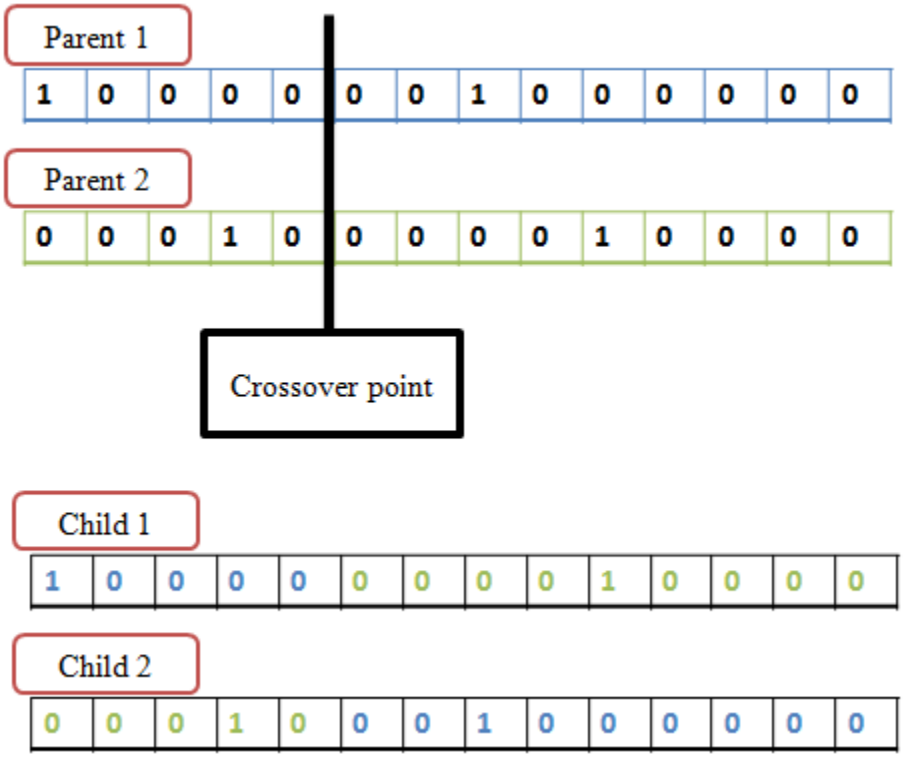
Table 4. Algorithm 4: GA Fitness Function<Tbl_Large></Tbl_Large>

Input: FES = A chosen strategy; DMA = A chosen strategy; Output: MinER = Error Rate: Calculated using Eq (09) MaxAcc = Accuracy: Calculated using Eq (08) Begin Make a classification step according to the selected individuals. Compute ER, ACC using DMA; Compute the fitness function of each selected individual using Eq. (10). End.

Table 5. Example of the fitness function calculation of the 4 individuals

	ER	ACC	Fitness value
Individual 1	0.09	1	99.91
Individual 2	0.07	0.92	91.93
Individual 3	0.11	1	99.89
Individual 4	0.76	0	-0.76

Figure 10. The crossover process.



number between 0 and 1. If this number is lower than a value called the mutation rate, that variable is flipped. The mutation rate is a very low probability pm, generally between 0.001 and 0.01. With that value for the mutation rate, we will mutate one of each individual (statistically).

The next example shows the mutation of the second children of the new generation. As we can see, the sixth and the eighth genes of the child have been flipped.

4. EXPERIMENTAL RESULTS

To prove the effectiveness of our approach, we performed several experiments of the proposed algorithm BAT for feature selection and GA for selecting optimal couple (FES, DMA). This section describes the used materials and methods. After-that, we present an analytical study on protein representation methods using several classifiers for feature selection by BAT algorithm. Also a comparative study with PSO-Search and EA-Search, which are the two bio-inspired versions most used for solving this problem, is given.

4.1. Materials and Methods

Tests took place on a laptop PC with an Intel(R) Core(TM) i5 processor running at 2.3 Ghz and 4 Go of RAM. Programs developed using eclipse neon environment with jdk 1.8.

4.1.1. Data Preparation

The data base that we mainly used for the training and assessment of our classification approach was downloaded from the website[1] mentioned below.

4.1.2. Weka

Weka (Waikato Environment for Knowledge Analysis) is a set of tools for manipulating and analyzing data files, implementing most of the artificial intelligence algorithms, decision trees and the neural networks. It is written in java, available on the website[2]. We used weka for two reasons: First, we have done a classification of the GPCR via nine classifiers and secondly, we have performed a step of attributes selection using EA and PSOSearch in order to compare them with the BAT Algorithm proposed and implemented in our work.

4.1.3. Performance Measurements

The standard metrics used in the flat classification to measure and evaluate the DMA is the jackknife test. In the jackknife test, each protein sequence in the dataset is singled out in turn as a test sample and the remaining protein sequences are used as a training dataset to predict the label of the test sample. Thus, this process is repeated N times for a dataset of N proteins. In this paper, we used the accuracy (ACC) (Eq.8), the error rate (Eq.9) to control their variation with and without FS and the GA fitness (Eq.10).

$$\boldsymbol{Acc} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}} \tag{8}$$

$$\mathrm{ER} = \frac{\mathrm{Numberofmisclassifiedexamples}}{\mathrm{Totalnumberofexamples}} \tag{9}$$

$$Fitness = \text{Acc*}100 - ER \tag{10}$$

4.2. Computational Results and Discussion

In this section, we will first discuss the impact of the different PRM "that is less investigated in the literature" on the task of protein function prediction and we will also discuss the impact of the different DMA. Furthermore, all experiments were performed using 10-fold cross-validation.

The Table 7 presents the results of the GPCR classification at sub sub-family level with FS using BAT algorithm. After several tests, we found that the modification of bat-number and max-iteration parameters can improve the classification results, and the best value obtained was equal to 10 for each. The number of attributes was chosen for each classifier after several experiments in order to obtain the best.

All GA module results are produced with max iteration number equal to 20, the crossover probability = 0.5 and mutation probability = 0.01.

Table 6 shows the best measurements (attribute number, error rate, GA fitness and accuracy) per FES for each DMA:

As shown in the following figures (Figure11, Figure12, Figure13, Figure14, Figure15), we noted that feature selection using BAT algorithm provides better accuracies and error rates compared to classification without feature selection (using all attributes) and this for all DMA.

4.2.1. Impact of The Different PRM

One of the principal contributions of this work is to assess the impact of the choice of the FES for representing protein sequences on the protein function prediction problem. The overall analysis of the results shows some interesting points:

- For the AAC method (Figure 11), the accuracy values are 1 for six classifiers (BN,NB, KNN,J48,DT,LB) using FS by Bat Algorithm, however, this value is included in the range [0.8, 0.95] in a standard classification for algorithms cited above. Unlike the BAG algorithm which marks a slight decrease in the accuracy value using FS technique. As for the error rate values, we can notice a differentiation statistically non significant for all DMA except the DT classifier which significantly improves the error rate using the FS. Concerning the ZR classifier, it provides very poor results either by using the FS or in the standard classification for all FES.
- The results obtained using the PseAAC strategy (Figure12) are more or less effective compared to the AAC method in the standard classification, but the FS techniques by Bat algorithm improved the performance measurements values effectively, so that six DMA are reached the value 1 of accuracies (BN,NB,KNN,J48,DT and LB), note that the error rate values are always in continuous improvement.
- When using Am-PseAAC strategy (Figure13), we notice a statistically significant decrease in the performance measurements values compared to the previous PRMs such that the error rate is high and the accuracy is reduced for all classifiers in the standard classification, but the use of FS technique significantly improves the results, so they became closer to previous results (Figure16). Furthermore, the KNN,J48,DT and BAG classifiers provide accuracies values equal to 1, but the LB algorithm marks a slight decrease in accuracy value using the FS technique.
- Generally, the results of the standard classification by the DC method are good, the accuracies values of the DMA are close, and they are included in the interval [0.83, 0.94] such as the most minimum value is provided by DT classifier. The implementation of the BAT algorithm for FS also improves the GPCR classification performances for all used DMA. Note that RF, BAG and LB classifiers give the best results compared to the remaining DMA like shown in Figure14.
- We can observe the usefulness of the FS technique in proteins classification, using the LD strategy, especially in the NB algorithm (Figure15), which provided accuracy and error rate values equal

Table 7. Evaluation of performance measurements in terms of attributes number.

		FS Using BAT Algorithm			Standard Classification		GA Fitness
FES	**DMA**	**Att-num**	**ER**	**ACC**	**ER**	**ACC**	**Fitness values**
AAC	BN	9	0,11	1	0.11	0.9	99,89
	NB	18	0,12	1	0.13	0.88	99,88
	KNN	13	0,05	1	0.05	0.95	99,95
	J48	18	0,09	1	0.09	0.9	99,91
	DT	18	0,09	1	0.2	0.8	99,91
	RF	13	0.04	0.99	0.05	0.92	98,96
	BAG	12	0.07	0.89	0.08	0.92	88,93
	LB	17	0.07	1	0.08	0.912	99,93
	ZR	/	0,76	0	0.76	0.05	-0,76
PseAAC	BN	9	0.14	1	0.16	0.88	99,86
	NB	20	0.15	1	0.22	0.83	99,85
	KNN	44	0.06	1	0.12	0.88	99,94
	J48	9	0	1	0.09	0.9	100
	DT	18	0.22	1	0.22	0.79	99,78
	RF	24	0.04	0.99	0.05	0.953	98,96
	BAG	39	0.07	0.93	0.07	0.923	92,93
	LB	28	0.07	1	0.07	0.926	99,93
	ZR	/	0.76	0	0.76	0.05	-0,76
Am-PseAAC	BN	26	0.22	0.9	0.27	0.77	89,78
	NB	63	0.25	0.92	0.33	0.73	91,75
	KNN	18/32	0.15	1	0.19	0.8	99,85
	J48	18/43	0.14	1	0.14	0.85	99,86
	DT	14	0.25	1	0.26	0.73	99,75
	RF	41	0.12	0.9	0.12	0.876	89,88
	BAG	43	0.1	1	0.12	0.876	99,9
	LB	50	0.13	0.82	0.13	0.864	81,87
	ZR	/	0.76	0	0.76	0.05	-0,76
DC	BN	199	0.08	0.93	0.11	0.92	92,92
	NB	209	0.11	0.93	0.13	0.91	92,89
	KNN	79	0.07	0.92	0.08	0.92	91,93
	J48	250	0.08	0.91	0.09	0.9	90,92
	DT	235	0.18	0.85	0.18	0.83	84,82
	RF	96	0.06	0.98	0.07	0.91	97,94
	BAG	62	0.06	0.96	0.06	0.93	95,94
	LB	143	0.05	0.96	0.06	0.94	95,95
	ZR	/	0.76	0	0.76	0.05	-0,76
LD	BN	25	0,21	0,83	0.32	0.78	82,79
	NB	36	0,32	0,72	0.48	0.56	71,68
	KNN	169	0,05	0,95	0.05	0.94	94,95
	J48	160	0,09	0,95	0.09	0.9	94,91
	DT	180	0,19	0,88	0.2	0.8	87,81
	RF	141	0.06	0.95	0.07	0.92	94,94
	BAG	32	0.06	0.95	0.07	0.925	94,94
	LB	69	0.06	0.93	0.07	0.926	92,94
	ZR	/	0.76	0	0.76	0.05	-0,76

Figure 11. Evaluation of the accuracies and the error rates values at sub sub-family level for AAC strategy (FS using BAT algorithm versus STD classification)

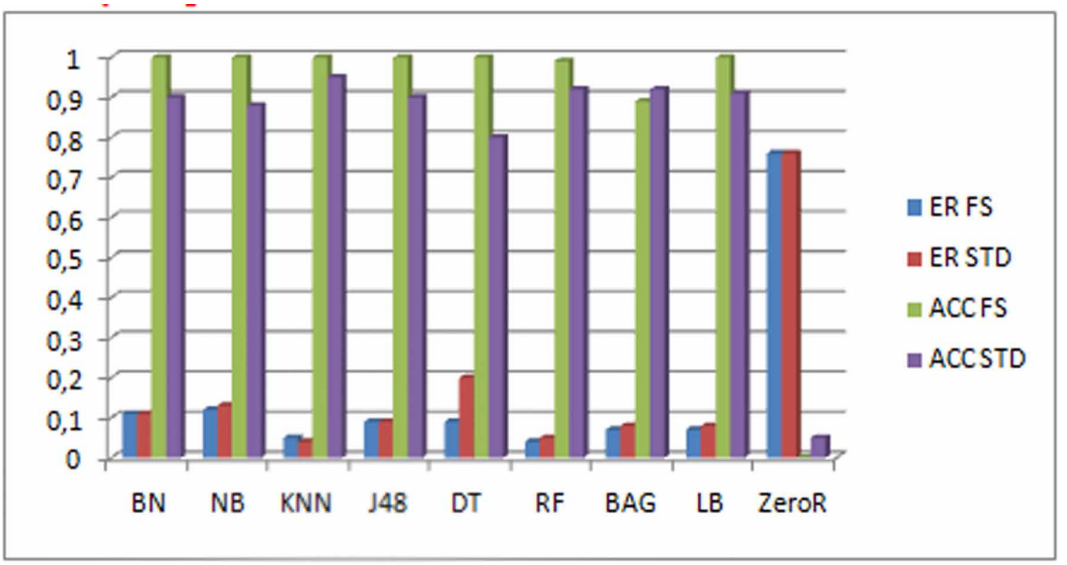
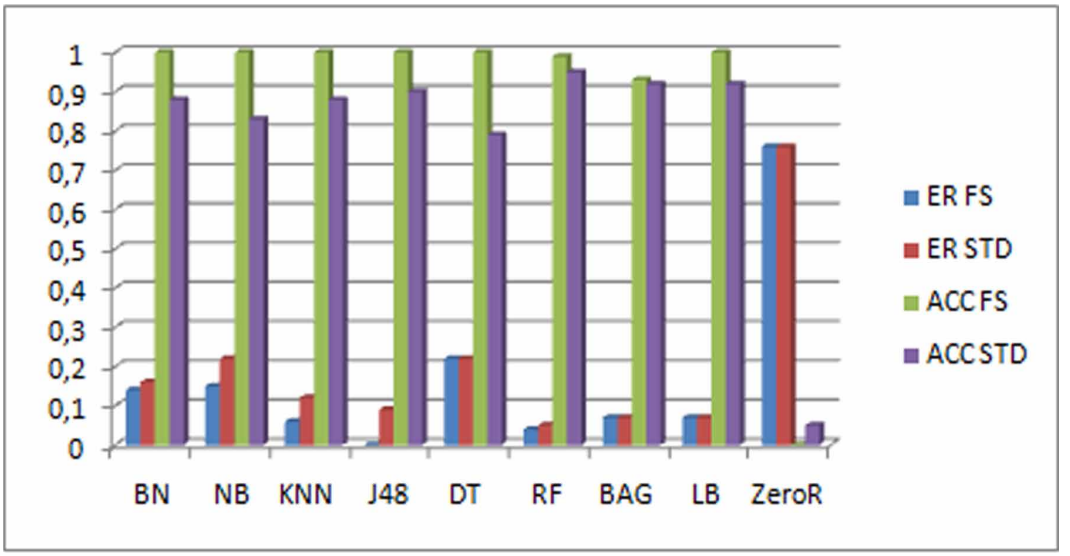


Figure 12. Evaluation of the accuracies and the error rates values at sub sub-family level for PseAAC strategy (FS using BAT algorithm versus STD classification).



to 0.56 and 0.48 respectively in the standard classification, these mesures became 0.72 and 0.32 respectively after FS. BN and NB gave poorer results compared to other DMAs.

4.2.2. Impact of The Different DMA

It is a well-known fact in machine learning that there is no classifier which is the best for all applications. Recall that in this work we are employing the DMA selection using nine classification

Figure 13. Evaluation of the accuracies and the error rates values at sub sub-family level for Am-PseAAC strategy (FS using BAT algorithm versus STD classification).

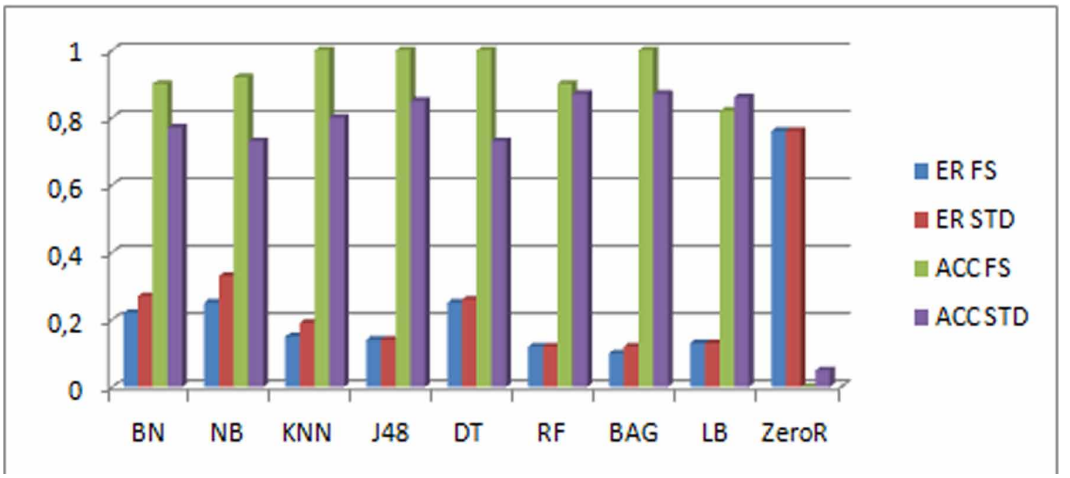
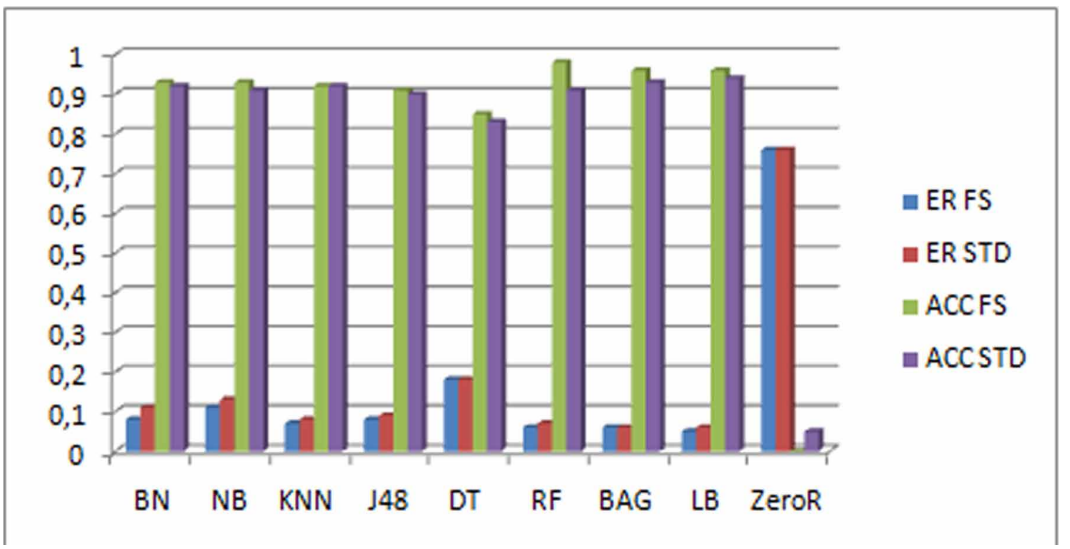


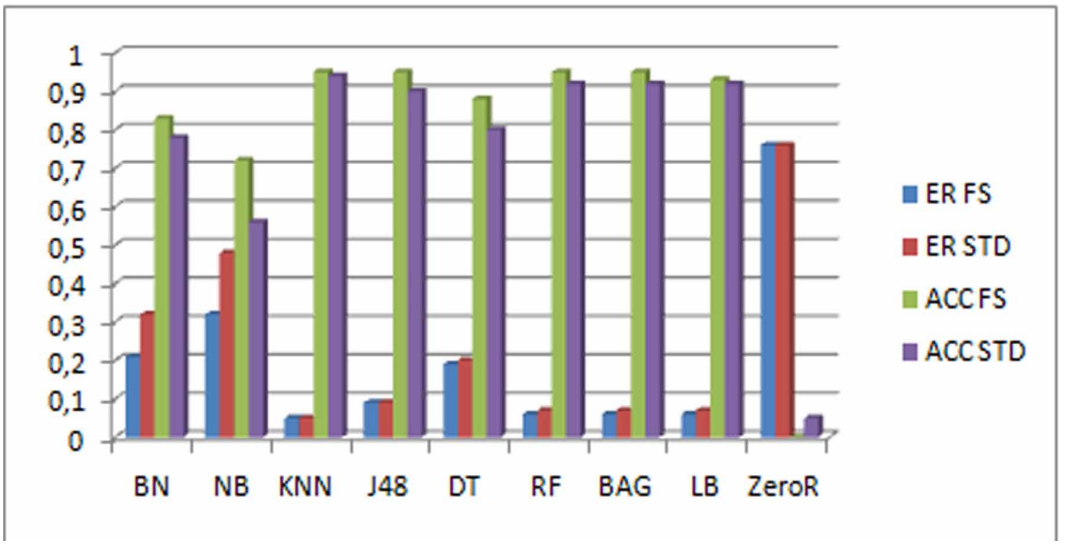
Figure 14. Evaluation of the accuracies and the error rates values at sub sub-family level for DC strategy (FS using BAT algorithm versus STD classification).



algorithms: BN,NB,KNN,J48,DT,RF,BAG,LB and ZR. The choice of these DMAs comes down to their diversification and their different operating principle for good experimentation and precise analysis. The following figure (Figure16) shows the assessment of the accuracies and the ER values at sub sub-family level of the best subset of attributes for all used FES.

According to Figure16 we can extract the following points:

Figure 15. Evaluation of the accuracies and the error rates values at sub sub-family level for LD strategy (FS using BAT algorithm versus STD classification).



1. The behavior of each DMA changes from one strategy to another; on the other hand a classifier can give good results with a strategy (NB with AAC, PseAAC / DT with AAC, PseAAC, Am-PseAAC) and very bad results with another (NB with LD / DT with DC).
2. KNN,J48,DT classifiers give a good result with LD strategy compared to DC method, contrary to BN,NB,RF and LB classifiers.
3. AAC and PseAAC provide almost similar results for all DMA, due to the use of Bat algorithm which is very efficient in FS paradigm.

Figure 16. Evaluation of the accuracies and the error rates values at sub sub-family level of the best subset of attributes.

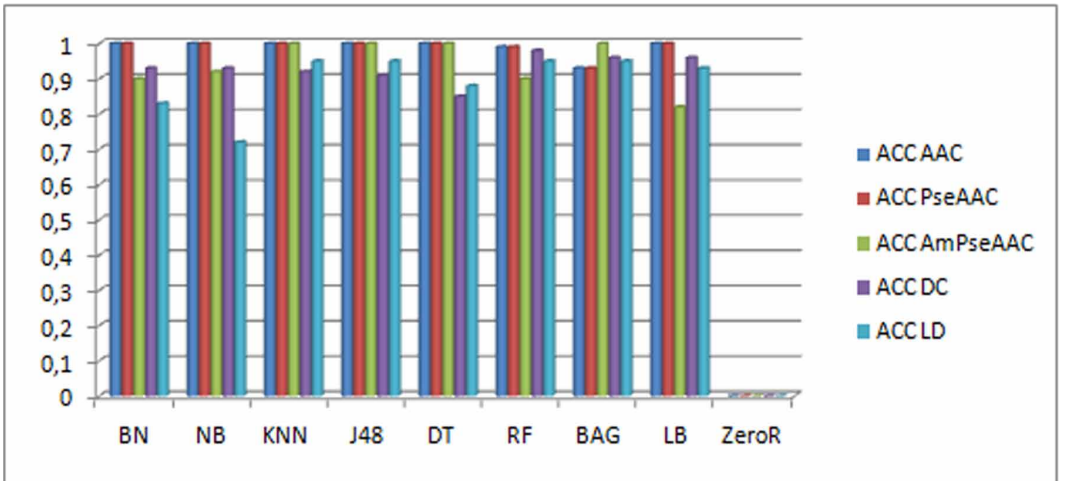
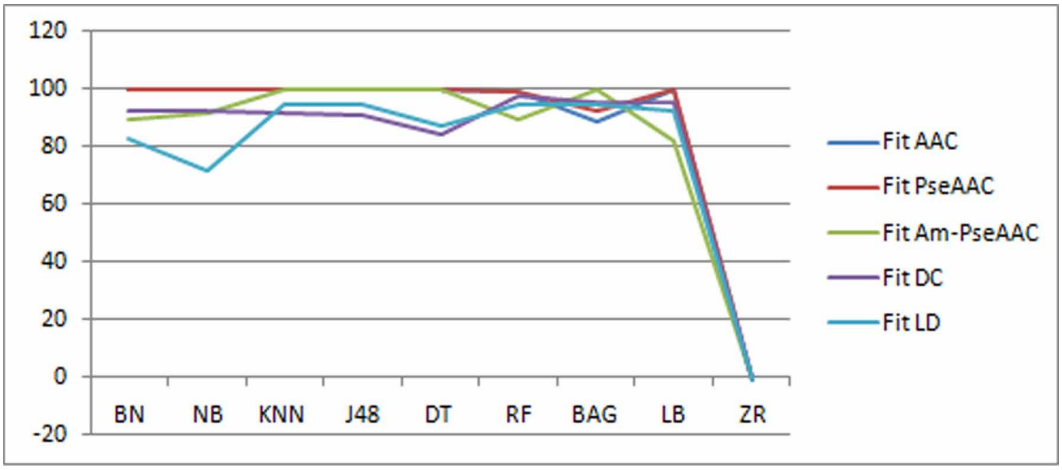


Figure17. Evaluation of fitness values of the used PRM.



4. Despite the fact that all the DMAs have given good results with AAC and PseAAC strategies, BAG classifier behaves less efficiently with these two methods when comparing by the remaining, and it reaches the value 1 of accuracy using Am-PseAAC.
5. Based on experimental results, ZR classifier is not advisable to use in protein classification.

4.2.3. Impact of Fitness Function

The objective of this work is to improve the protein classification for an adequate identification of their function. One of the suggested issues is to use the GA to choose the right couple (FES, DMA) according to the best performance measures.

The fitness function is calculated using Eq.10, it based on accuracy and error rate values. The Figure17 shows us the variation of the fitness values of each DMA for each FES.

The KNN, J48 and DT classifiers provide good fitness using AAC, PseAAC and Am-PseAAC strategies. Note that the AAC, PseAAC methods are very close in terms of fitness, they mark a slight differentiation at the level of BAG algorithm.

Because of the fitness function we can easily choose the best algorithm to use for classifying the GPCR sequences as illustrated in the figure below (Figure18).

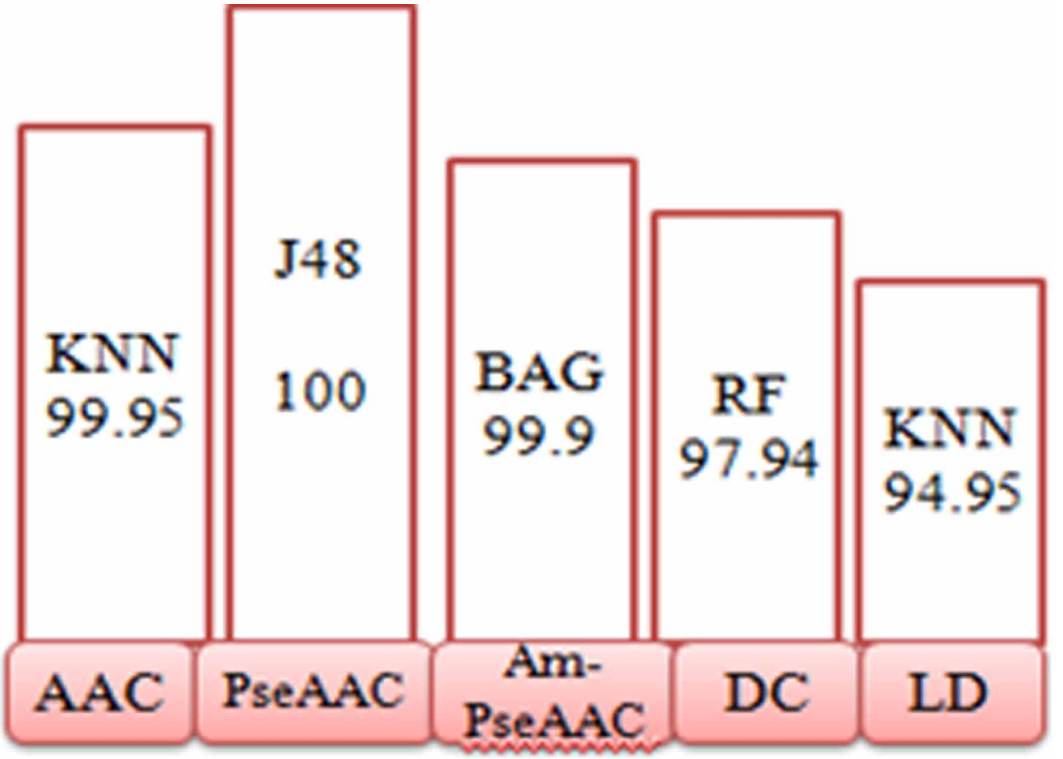
Based on GA fitness values, despite the close nature of measurements, we noted that the best obtained couple (FES, DMA) for the used database is (PseAAC, J48).

4.3. Comparison With Additional Methods

In this section, we present a study that compares our approach of Bat algorithm designed for FS to two existing bio inspired algorithms deployed on Weka environment (PSOSearch and Evolutionary Algorithm (EA)). Table 8 shows the results of the GPCR classification at sub sub-family level with FS using PSO and EA. In our comparison, three parameters are taken into account: attribute number, accuracy and ER. Outcomes (Table 7, Table 8) show that the proposed BAT algorithm extracts relevant subset of features for used database. This is because rigidity and stochastic criteria of BAT algorithm are introduced, so it generates only useful and pertinent subsets. On contrary, PSO and EA approaches can generate subsets of useless features, which diminished the performance of the classification. Note that PSO/EA do not make the FS for the AAC method because of the few number of attributes.

In general, the results reported in Table 8 are approximately similar especially for the first three methods, for example the KNN,J48,DT and ZR classifiers gave the same values of ACC and ER

Figure18. The best fitness of each FES.



for PSO/EA algorithms using PseAAC method and the remaining algorithms mark a differentiation statistically not significant; for the Am-PseAAC and DC methods, we mark slight differentiation can reach 0.03. As for the LD method, PSO produced ER values better than those of EA for BN,NB classifiers, and they are poorer for KNN,J48,DT,RF,BAG,LB classifiers. However, almost all accuracy values are better using EA versus PSO algorithm.

Comparing the results of the classification with FS using PSO/EA algorithms with those of Bat algorithm, we found that in most of the time the BAT algorithm gave the best results; thereafter we will give you more details by analyzing the bellow figures.

The optimization of the GPCR classification with FS does not affect ZR classifier which remains stable using all bio inspired algorithms for all PRM.

Each figure presents the variation of each performance measurements values, the first one (Figure19) is dedicated to the accuracies values, all the bars relating to the bat algorithm indicate that whatever the method used, all DMA achieve the best accuracies compared to EA/PSO, except the NB classifier which produces the lowest accuracy / ER values using LD strategy. Note also that by using the Am-PseAAC method, the BN and NB classifiers give ER values better than EA and bad than PSO algorithm.

The FS using BAT algorithm has significantly improved the performance of GPCR classification especially in the PseAAC and Am-PseAAC methods.

The second figure (Figure20) is devoted to error rate values. It is clear that there are some DMAs which give similar or very close values for all FS algorithms such as: BN, DT, BAG, RF and LB for the PseAAC method, BN, DT, RF, BAG and LB for the Am-PseAAC method. However using DC strategy, all classifiers produce very close error rate values with differentiation statistically not

Table 8. The performance of GPCRs classification at sub sub-family level with FS using PSOSearch/EA algorithms.

	Classifier	ER PSO	ACC PSO	ER EA	ACC EA
PseAAC Att Number PSO/EA = 30/32	BN	0.14	0.88	0.15	0.88
	NB	0.17	0.86	0.18	0.85
	KNN	0.12	0.87	0.12	0.87
	J48	0.09	0.9	0.09	0.9
	DT	0.22	0.79	0.22	0.79
	RF	0.06	0.93	0.06	0.92
	BAG	0.07	0.91	0.08	0.9
	LB	0.07	0.95	0.07	0.94
	ZR	0.76	0	0.76	0
Am-PseAAC Att Number PSO/EA = 12/14	BN	0.2	0.79	0.22	0.78
	NB	0.23	0.75	0.27	0.72
	KNN	0.18	0.81	0.16	0.82
	J48	0.16	0.83	0.17	0.82
	DT	0.26	0.73	0.27	0.71
	RF	0.13	0.86	0.12	0.87
	BAG	0.12	0.87	0.12	0.86
	LB	0.15	0.81	0.14	0.8
	ZR	0.76	0	0.76	0
DC Att Number PSO/EA = 129/126	BN	0.1	0.91	0.09	0.92
	NB	0.12	0.9	0.12	0.9
	KNN	0.06	0.93	0.06	0.93
	J48	0.1	0.9	0.09	0.9
	DT	0.2	0.8	0.19	0.82
	RF	0.07	0.94	0.06	0.95
	BAG	0.07	0.95	0.07	0.95
	LB	0.08	0.96	0.07	0.95
	ZR	0.76	0	0.76	0
LD Att Number PSO/EA = 7/67	BN	0.19	0.8	0.23	0.82
	NB	0.24	0.76	0.35	0.74
	KNN	0.11	0.88	0.05	0.94
	J48	0.13	0.86	0.09	0.9
	DT	0.23	0.76	0.2	0.79
	RF	0.07	0.93	0.06	0.94
	BAG	0.08	0.94	0.07	0.93
	LB	0.08	0.9	0.07	0.91
	ZR	0.76	0	0.76	0

Figure 19. General comparison of the Accuracy values for the used FES using EA/ PSO and Bat Algorithms.

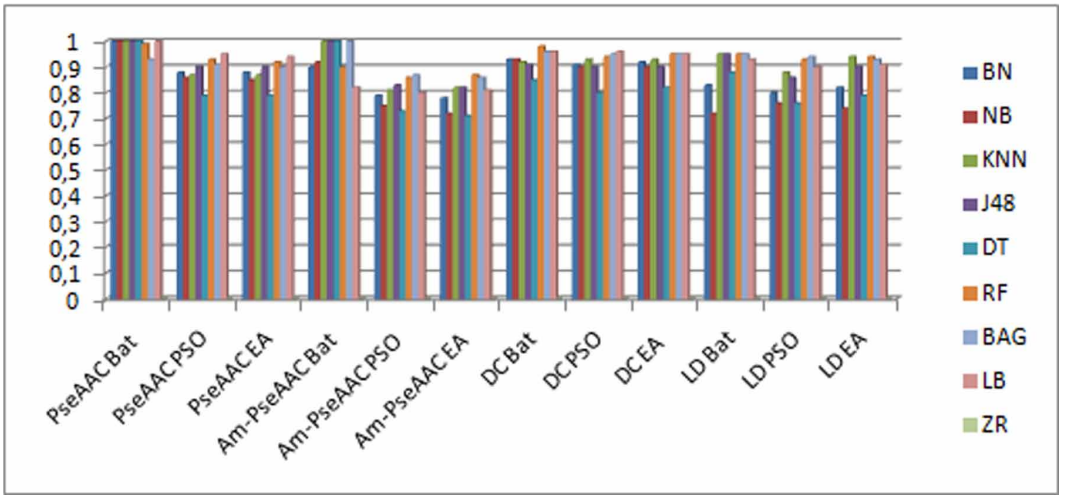
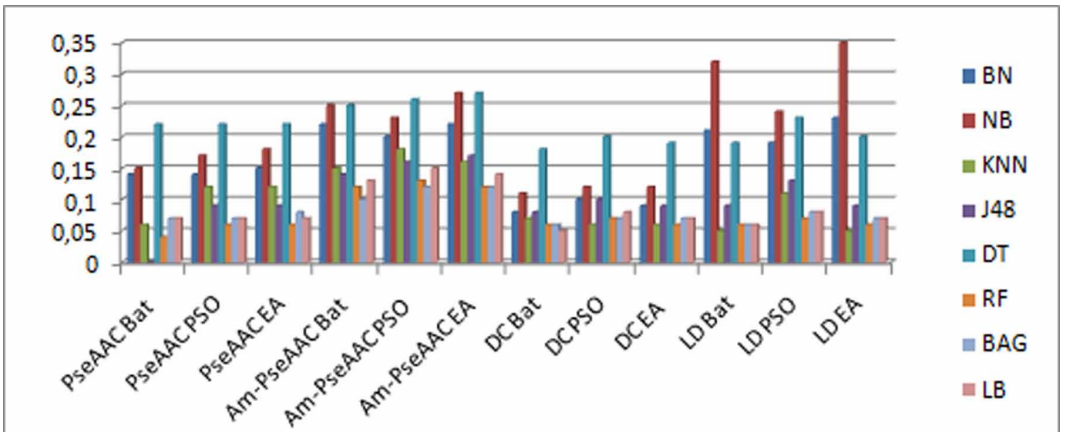


Figure 20. General comparison of the Error Rate values for the used FES using EA/ PSO and Bat Algorithms.



significant can reach 0.02, unlike the LD method which marks a considerable variation especially at the level of NB, BN, KNN, J48, DT algorithms.

Finally, after the comparison of the BAT algorithm with PSO/EA, we found that overwhelmingly, the BAT algorithm gave the best results. Therefore it is suitable and effective in the field of bioinformatics especially for the classification of GPCRs.

5. CONCLUSION AND FUTURE WORKS

In this work, we presented an empirical study analyzing the impact of different PRM and different types of DMA for the task of protein function prediction. We have employed 5 FES, computed from the protein sequence: AAC, PseAAC, Am-PseAAC, DC and LD. To perform the classification we have used 9 classifiers: NB, BN, KNN, J48, DT, RF, LB, BAG and ZR.

Bat Algorithm is a bio-inspired meta-heuristic that uses the benefits of the echolocation ability, it has been used to optimize results in many applications. In this article, we adopted it for GPCR classification using several methods of protein representation. These later, are based on a huge number of features, which could be costly in terms of memory and computing time. For this, we have made a selection of attributes using the BAT algorithm. According to the experimental results, the number of attributes varies with the couple FES, DMA.

We have tried throughout this paper to present our contributions which are mainly the improvement of the quality of the GPCR classification using the minimum of features which will ensure better prediction of GPCRs functions. On the other hand, we tried to find the best couple (FES, DMA) using GA. This will allow us to apply the proposed approach on other protein bases. Therefore, our recommendation (based on our experimental results of fitness function) is that when using AAC, LD strategies, we suggest the use of KNN classifier, when the use of PseAAC, Am-PseAAC is available, we recommend J48, BAG algorithms respectively, and for DC method, the best DMA is RF classifier.

Our experimental results show that in general, regardless of the type of protein, the FS technique using Bat algorithm optimize substantially the GPCR classification; PseAAC is a very good PRM with J48 since it is simple and provides best results; LD is the best improved method by Bat algorithm that can be provides very good results (except for the NB classifier).

Compared with the algorithms proposed in the literature, our approach reached the best performance. In our future work, we would like to study more representation methods and more algorithms, which require more work on the evolutionary part of our approach. We will also try to experiment with several bases.

REFERENCES

Alberts, B., Bray, D., Lewis, J., Raff, M., Roberts, K., & Watson, J. D. (2002). Molecular Biology of the Cell. Garland Publishing.

Bagherzadeh-Khiabani, F., Ramezankhani, A., Azizi, F., Hadaegh, F., Steyerberg, E. W., & Khalili, D. A. (2016). Tutorial on variable selection for clinical prediction models: Feature selection methods in data mining could improve the results. *Journal of Clinical Epidemiology*, *71*, 76–85. doi:10.1016/j.jclinepi.2015.10.002 PMID:26475568

Bhasin, M., & Raghava, G. P. S. (2004). GPCRPred: An SVM-based method for prediction of families and subfamilies of g-protein coupled receptors. *Nucleic Acids Research*, *32*(Web Server), 383–389. doi:10.1093/nar/gkh416 PMID:15215416

Cai, C. Z., Han, L. Y., Ji, Z. L., Chen, X., & Chen, Y. Z. (2003). SVM-Prot: Web-based support vector machine software for functional classification of a protein from its primary sequence. *Nucleic Acids Research*, *31*(13), 3962–3967. doi:10.1093/nar/gkg600 PMID:12824396

Cai, X., Zhang, J., Liang, H., Wang, L., & Wu, Q. (2019). An ensemble bat algorithm for large-scale optimization. *International Journal of Machine Learning and Cybernetics*, *10*(11), 3099–3113. doi:10.1007/s13042-019-01002-8

Chen, S., Montgomery, J., & Bolufe-Rohler, A. (2014). Measuring the curse of dimensionality and its effects on particle swarm optimization and differential evolution. *Applied Intelligence*. Advance online publication. doi:10.1007/s10489-014-0613-2

Cheng, X., Xiao, X., & Chou, K.-C. (2017). pLoc-mEuk: Predict subcellular localization of multi-label eukaryotic proteins by extracting the key GO information into general PseAAC. *Genomics*, *110*. PMID:28818512

Chou, K. C. (2001). Prediction of protein cellular attributes using pseudo-amino acid composition. *Proteins*, *43*(3), 246–255. doi:10.1002/prot.1035 PMID:11288174

Chou, K.-C. (2005). Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes. *Bioinformatics (Oxford, England)*, *21*(1), 10–19. doi:10.1093/bioinformatics/bth466 PMID:15308540

Chou, K. C., & Shen, H.-B. (2006). Predicting protein subcellular location by fusing multiple classifiers. *Journal of Cellular Biochemistry*, *99*(2), 517–527. doi:10.1002/jcb.20879 PMID:16639720

Correa, E. S., Freitas, A. A., & Johnson, C. G. (2007). Particle swarm and Bayesian networks applied to attribute selection for protein functional classification. *Proceedings of the Conference Companion on Genetic and Evolutionary Computation*, 2651-2658. doi:10.1145/1274000.1274081

Costa, E. P., Lorena, A. C., Carvalho, A. C. P. L. F., & Freitas, A. A. (2008). *Top-Down Hierarchical Ensembles of Classifiers for Predicting G-Protein-Coupled-Receptor Functions*. LNBI. doi:10.1007/978-3-540-85557-6_4

Cui, J., Han, L. Y., Li, H., Ung, C. Y., Tang, Z. Q., Zheng, C. J., Cao, Z. W., & Chen, Y. Z. (2007). Computer prediction of allergen proteins from sequence-derived protein structural and physicochemical properties. *Molecular Immunology*, *44*(4), 514–520. doi:10.1016/j.molimm.2006.02.010 PMID:16563508

Davies, M., Secker, A., Freitas, A., Clark, E., Timmis, J., & Flower, D. (2008). Optimizing amino acid groupings for GPCR classification. *Bioinformatics (Oxford, England)*, *24*(18), 1980–1986. doi:10.1093/bioinformatics/btn382 PMID:18676973

Davies, M. N., Gloriam, D. E., Secker, A., Freitas, A. A., Timmis, J., & Flower, D. R. (2011). Present perspectives on the automated classification of the g protein coupled receptors (GPCRs) at the protein sequence level. *Current Topics in Medicinal Chemistry*, *11*(15), 1994–2009. doi:10.2174/156802611796391221 PMID:21470173

Drews, J. (2000). Drug discovery: A historical perspective. *Science*, *287*(5460), 1960–1964. doi:10.1126/science.287.5460.1960 PMID:10720314

Duc, N. M., Kim, H. R., & Chung, K. Y. (2017). Recent Progress in Understanding the Conformational Mechanism of Heterotrimeric G Protein Activation. *Biomolecules & Therapeutics*, *25*(1), 4–11. doi:10.4062/biomolther.2016.169 PMID:28035078

Freitas, A. A., & de Carvalho, A. C. P. L. F. (2007). A tutorial on hierarchical classification with applications in bioinformatics. *Research and Trends in Data Mining Technologies and Application*, *99*(7), 175–208. doi:10.4018/978-1-59904-271-8.ch007

Gandomi, A. H., Yang, X.-S., Alavi, A. H., & Talatahari, S. (2013). Bat algorithm for constrained optimization tasks. *Neural Computing & Applications*, *22*(6), 1239–1255. doi:10.1007/s00521-012-1028-9

Gao, Q. B., Ye, X. F., & He, J. (2013). Classifying G-Protein-Coupled Receptors to the Finest Subtype Level. *Biochemical and Biophysical Research Communications*, *439*(2), 303–308. doi:10.1016/j.bbrc.2013.08.023 PMID:23973783

Gu, Q., & Ding, Y. (2009). Binary particle swarm optimization based prediction of g-protein-coupled receptor families with feature selection. *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, 171-176. doi:10.1145/1543834.1543859

Guo, S. S., Wang, J.-S., & Ma, X.-X. (2019). Improved Bat Algorithm Based on Multi-population Strategy of Island Model for Solving Global Function Optimization Problem. Computational Intelligence and Neuroscience.

Holden, N., & Freitas, A. A. (2006). Hierarchical classification of G-protein coupled receptors with a PSO/ACO algorithm. *Proceedings of the IEEE Swarm Intelligence Symposium*, 77-84.

Holden, N., & Freitas, A. A. (2008). Improving the Performance of Hierarchical Classification with Swarm Intelligence. LNCS, 4973, 48–60. doi:10.1007/978-3-540-78757-0_5

Holden, N., & Freitas, A. A. (2009). Hierarchical classification of protein function with ensembles of rules and particle swarm optimisation. *Soft Computing*, *13*(3), 259–272. doi:10.1007/s00500-008-0321-0

Huang, Y., Cai, J., Ji, L., & Li, Y. (2004). Classifying G-protein coupled receptors with bagging classification tree. *Computational Biology and Chemistry*, *28*(4), 275–280. doi:10.1016/j.compbiolchem.2004.08.001 PMID:15548454

Judson, R. S. (1997). Genetic algorithms and their use in chemistry. In K. B. Lipkowitz & D. B. Boyd (Eds.), *Rev. Computational Chemistry* (Vol. 10, pp. 1–73). Wiley-VCH.

Judson, R. S. (2008). Genetic Algorithms for Protein Structure Prediction. In C. Floudas & P. Pardalos (Eds.), *Encyclopedia of Optimization*. Springer.

Khan, A., Majid, A., & Choi, T.-S. (2010). Predicting protein subcellular location: Exploiting amino acid based sequence of feature spaces and fusion of diverse classifiers. *Amino Acids*, *38*(3), 47–35. doi:10.1007/s00726-009-0238-7 PMID:19198979

Khan, M., Hayat, M., Khan, S. A., & Iqbal, N. (2017). Unb-DPC: Identify mycobacterial membrane protein types by incorporating un-biased dipeptide composition into Chou's general PseAAC. *Journal of Theoretical Biology*, *415*, 13–19. doi:10.1016/j.jtbi.2016.12.004 PMID:27939596

Khan, Z. U., Hayat, M., & Khan, M. A. (2015). Discrimination of acidic and alkaline enzyme using Chou's pseudo amino acid composition in conjunction with probabilistic neural network model. *Journal of Theoretical Biology*, *365*, 197–203. doi:10.1016/j.jtbi.2014.10.014 PMID:25452135

King, R. D., Karwath, A., Clare, A., & Dehaspe, L. (2001). The utility of different representations of protein sequence for predicting functional class. *Bioinformatics (Oxford, England)*, *17*(5), 445–454. doi:10.1093/bioinformatics/17.5.445 PMID:11331239

Kumar, R., Kumari, B., & Kumar, M. (2017). Prediction of endoplasmic reticulum resident proteins using fragmented amino acid composition and support vector machine. *PeerJ*, *4*, e3561. Advance online publication. doi:10.7717/peerj.3561 PMID:28890846

Kumar, R., Srivastava, A., Kumari, B., & Kumar, M. (2015). Prediction of β-lactamase and its class by Chou's pseudo-amino acid composition and support vector machine. *Journal of Theoretical Biology*, *365*, 96–103. doi:10.1016/j.jtbi.2014.10.008 PMID:25454009

Kumari, T., Pant, B., & Pardasani, K. R. (2010). A SVM Model for AAC Based Classification of Class B GPCRs. *World Congress of Biomechanics (WCB),* 1607–1610. doi:10.1007/978-3-642-14515-5_409

Li, J., & Liu, H. (2017). Challenges of Feature Selection for Big Data Analytics. *IEEE Computer Society*, *17*(2), 1541–1672. doi:10.1109/MIS.2017.38

Li, Z., Zhou, X., Dai, Z., & Zou, X. (2010). Classification of G-Protein Coupled Receptors based on Support Vector Machine with Maximum Relevance Minimum redundancy and Genetic Algorithm. *BMC Bioinformatics*, *11*(1), 325–340. doi:10.1186/1471-2105-11-325 PMID:20550715

Liu, B., Yang, F., & Chou, K.-C. (2017). 2L-piRNA: A Two-Layer Ensemble Classifier for Identifying Piwi-Interacting RNAs and Their Function. *Molecular Therapy. Nucleic Acids*, *7*, 267–277. doi:10.1016/j.omtn.2017.04.008

Munoz, S., Guerrero, F. D., Kellogg, A., Heekin, A. M., & Leung, M.-Y. (2017). Bioinformatic prediction of G protein Coupled receptor encoding sequences from the transcriptome of the foreleg, including the Haller's organ, of the cattle tick, Rhipicephalus australis. *PLoS One*, *12*(2), e0172326. doi:10.1371/journal.pone.0172326

Nakano, F. K., Mastelini, S. M., Barbon, S., & Cerri, R. (2017). Stacking Methods for Hierarchical Classification. *Proceedings of 16th IEEE International Conference on Machine Learning and Applications*.

Naveed, M., & Khan, A. U. (2012). GPCR-MPredictor: Multi-level prediction of G protein-coupled receptors using genetic ensemble. *Amino Acids*, *42*(5), 1809–1823. doi:10.1007/s00726-011-0902-6 PMID:21505826

Nayyar, A., Garg, S., Gupta, D., & Khanna, A. (2018a). Evolutionary computation: theory and algorithms. In *Advances in Swarm Intelligence for Optimizing Problems in Computer Science* (pp. 1–26). Chapman and Hall/CRC. doi:10.1201/9780429445927-1

Nayyar, A., & Nguyen, N. G. (2018). Introduction to Swarm Intelligence. *Advances in Swarm Intelligence for Optimizing Problems in Computer Science*, 53-78.

Nemati, S., Basiri, M. E., Ghasem-Aghaee, N., & Aghdam, M. H. (2009). A novel ACO–GA hybrid algorithm for feature selection in protein function prediction. *Expert Systems with Applications*, *36*(10), 12086–12094. doi:10.1016/j.eswa.2009.04.023

Papasaikas, P. K., Bagos, P. G., Litou, Z. I., Promponas, V. J., & Hamodrakas, S. J. (2004). PRED-GPCR: GPCR recognition and family classification server. *Nucleic Acids Research*, *32*(Web Server), 380–382. doi:10.1093/nar/gkh431

Pedersen, J., & Moult, J. (1996). Genetic algorithms for protein structure prediction. *Current Opinion in Structural Biology*, *6*(2), 227–231. doi:10.1016/S0959-440X(96)80079-0 PMID:8728656

Peng, . (2010). An improved classification of G-protein-coupled receptors using sequence-derived features. *BioMed Central Bioinformatics*, *11*, 420. PMID:20696050

Rehman, Z. U., Mirza, M. T., Khan, A., & Xhaard, H. (2013). Predicting G-Protein-Coupled Receptors Families Using Different Physiochemical Properties and Pseudo Amino Acid Composition. *Methods in Enzymology*, *522*, 61–79. doi:10.1016/B978-0-12-407865-9.00004-2 PMID:23374180

Saeys, Y., Inza, I., & Larranaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics (Oxford, England)*, *23*(19), 2507–2517. doi:10.1093/bioinformatics/btm344 PMID:17720704

Santos, B. C., Rodrigues, M. W., Pinto, L. N. C., Nobre, C. N., & Zárate, L. E. (2019). Feature selection with genetic algorithm for protein function prediction. *Proceedings IEEE International Conference on Systems, Man and Cybernetics (SMC)*. doi:10.1109/SMC.2019.8914587

Secker, A., Davies, M., Freitas, A., Timmis, J., Mendao, M., & Flower, D. (2007). An experimental comparison of classification algorithms for the hierarchical prediction of protein function. *The BCSSGAI Magazine*, *9*(3), 17–22.

Secker, A., Davies, M. N., Freitas, A. A., Clark, E., Timmis, J., & Flower, D. R. (2010). Hierarchical classification of G-Protein-Coupled Receptors with data-driven selection of attributes and classifiers. *International Journal of Data Mining and Bioinformatics*, *4*(2), 191–210. doi:10.1504/IJDMB.2010.032150 PMID:20423020

Secker, A., Davies, M. N., Freitas, A. A., Timmis, J., Clark, E., & Flower, D. R. (2009). An Artificial Immune System for Clustering Amino Acids in the Context of Protein Function Classification. *Journal of Mathematical Modelling and Algorithms*, *8*(2), 103–123. doi:10.1007/s10852-009-9107-3

Selzer, P., & Ertl, P. (2005). Identification and classification of GPCR ligands using self-organizing neural networks. *QSAR & Combinatorial Science*, *24*(2), 270–276. doi:10.1002/qsar.200420071

Shrivastava, S., Pardasani, K. R., & Malik, M. M. (2010). SVM Model for Identification of human GPCRs. *Journal of Computing*, *2*(2).

Silla, C. N. Jr, & Freitas, A. A. (2011). Selecting Different Protein Representations and Classification Algorithms in Hierarchical Protein Function Prediction. *Intelligent Data Analysis*, *15*(6), 979–999. doi:10.3233/IDA-2011-0505

Tong, J. C., & Tammi, M. T. (2008). Prediction of protein allergenicity using local descriptions of amino acid sequence. *Frontiers in Bioscience*, *13*(13), 6072–6078. doi:10.2741/3138 PMID:18508644

Tupe, K. A., & Wakchaure Prof, M. A. (2017). Big Data Feature Selection Data Stream Mining. *International Journal Of Engineering And Computer Science*, *6*(7), 22041–22044.

ur-Rehman, Z., & Khan, A. (2011). G-protein-coupled receptor prediction using pseudo-amino-acid composition and multiscale energy representation of different physiochemical properties. *Analytical Biochemistry, 412*, 173–182.

Yang, X.-S. (2010). A new metaheuristic bat-inspired algorithm. In *Nature inspired cooperative strategies for optimization* (Vol. 284, pp. 65–74). Studies in Computational Intelligence.

Yang, X.-S., & Gandomi, A. H. (2012). Bat algorithm: A novel approach for global engineering optimization. *Engineering Computations*, *29*(5), 464–483.

Zekri, M., Alem, K., & Souici-Meslati, L. (2011). Identification methods of G Protein-Coupled Receptors. *International Journal of Knowledge Discovery in Bioinformatics*, *2*(4), 35–52.

Zhang, Z., Wu, J., Yu, J., & Xiao, J. (2012). A brief review on the evolution of GPCR: Conservation and diversification. *Open Journal of Genetics*, *2*, 11–17.

ENDNOTES

[1] www.gpcrdb.org
[2] www.cs.waikato.ac.nz/ml/weka

Safia Bekhouche is a PhD student at Computer Science Department, Faculty of Science and Technology, Badji Mokhtar University, Annaba, Algeria. He received his Master in Computer Science from the Badji Mokhtar University, in 2013. In 2014, he joined at the Laboratory of Research in Informatics (LRI) at UBMA, Algeria. His research interests include data mining, artificial intelligence, and evolutionary computing, bio-inspired computation, and bioinformatics.

Yamina Mohamed Ben Ali is a Full Professor at the Department of Computer Science at the University Badji Mokhtar Annaba (Algeria). She earned her PhD in Computer Science from UBMA, Algeria. Since 2008, she is the chief and the head of the specialty of Pattern Recognition and Artificial Intelligence (RFIA), she is member at the Laboratory of Research in Informatics (LRI) at UBMA . Her main research interests are related to artificial intelligence and data mining, Pattern recognition, Image processing, evolutionary computing, bio-inspired computation, and bioinformatics.