Computational and Structural Biotechnology Journal 18 (2020) 455–463

COMPUTATIONAL AND STRUCTURAL BIOTECHNOLOGY JOURNAL

journal homepage: www.elsevier.com/locate/csbj

An automatic representation of peptides for effective antimicrobial activity classification

Jesus A. Beltran [a], Gabriel Del Rio [b], Carlos A. Brizuela [a,*]

[a] Computer Science Department, Cicese Research Center, Ensenada, Baja California 22860, Mexico
[b] Department of Biochemistry and Structural Biology, Instituto de Fisiologia Celular, Universidad Nacional Autónoma de México, 04510, Mexico

ARTICLE INFO

Article history:
Received 15 October 2019
Received in revised form 30 January 2020
Accepted 1 February 2020
Available online 26 February 2020

Keywords:
Antimicrobial peptide
Feature selection
Wrapper method
Genetic algorithm

ABSTRACT

Antimicrobial peptides (AMPs) are a promising alternative to small-molecules-based antibiotics. These peptides are part of most living organisms' innate defense system. In order to computationally identify new AMPs within the peptides these organisms produce, an automatic AMP/non-AMP classifier is required. In order to have an efficient classifier, a set of robust features that can capture what differentiates an AMP from another that is not, has to be selected. However, the number of candidate descriptors is large (in the order of thousands) to allow for an exhaustive search of all possible combinations. Therefore, efficient and effective feature selection techniques are required.

In this work, we propose an efficient wrapper technique to solve the feature selection problem for AMPs identification. The method is based on a Genetic Algorithm that uses a variable-length chromosome for representing the selected features and uses an objective function that considers the Mathew Correlation Coefficient and the number of selected features. Computational experiments show that the proposed method can produce competitive results regarding sensitivity, specificity, and MCC. Furthermore, the best classification results are achieved by using only 39 out of 272 molecular descriptors.

1. Introduction

Antimicrobial peptides (AMPs) are a promising alternative for combating pathogens resistant to conventional antibiotics, mainly because of their multiple direct action mechanisms against microbes (*e.g.*, bacteria, fungi, and virus) and in consequence their low susceptibility of antimicrobial resistance; they also have been an effective weapon to fight against multi-drug-resistant microbial pathogens in vitro tests [1] and few others are currently being used to treat microbial infections in humans [2].

Next-Generation Sequencing (NGS) technologies are generating a vast amount of data (*e.g.,* DNA, RNA, or protein) where peptides with antimicrobial activity might be found. Identifying these peptides will only be possible through the development of computer-assisted strategies. These strategies can automatically evaluate a large amount of data and identify candidates to antimicrobial peptides before their biological evaluation in the wet lab. In this context, an important aspect is the development of machine learning models that determine whether or not an amino acid sequence is antimicrobial. Quantitative Structure-Activity Relationship (QSAR) modeling has been widely applied to AMP discovery for the development of classification models [3]. QSAR mathematically relates the quantitative physicochemical properties extracted from the peptides, termed molecular descriptors, with their corresponding biological activity through a predictive mathematical model. There are two crucial aspects to QSAR modeling: the choice of the descriptors set that defines the feature of the peptides of interest and the selection of the statistical learning technique to create a model [4,5]. Computational research has mainly focused on the second aspect, where several machine learning algorithms (MLAs) have been proposed for this purpose. Examples of these MLAs includes Discriminant Analysis (DA) [6], Random Forest (RF) [6,7], Support Vector Machine (SVM) [6,8,7], Artificial Neural Network (ANN) [5,8,7], Adaptive Neuro-Fuzzy Inference System (ANFIS) [4], Binary Logic Regression (BLR) [9] and Fuzzy K-Nearest Neighbor (FKNN) [10]. In overall, the proposed algorithms allow generating models with a predictive accuracy of up to 96%. However, these studies used different databases to measure the performances of their approaches for AMP's recognition.

* Corresponding author.
E-mail addresses: armando.3eltran@gmail.com (J.A. Beltran), gdelrio@ifc.unam.mx (G. Del Rio), cbrizuel@cicese.mx (C.A. Brizuela).

https://doi.org/10.1016/j.csbj.2020.02.002

On one hand, an amino acid sequence is considered to be AMP if it is labeled as such in a given database that collects only experimentally validated sequences. On the other hand, due to the difficulty to guarantee that a given sequence is not AMP, the databases define as such, sequences that passed through a strict filtering process aimed at increasing the prob-ability that they will not have antimicrobial properties. In our case, three databases DAT1, DAT2 and DAT3, used in the literature are considered. These databases are explained in subSection 3.1. It is important to mention that a more precise definition of what an AMP is, in terms of its MIC, is required to advance to the next level of granularity in the prediction of AMP activity.

There are many methods for the selection of descriptors for peptide representation, they are mainly based on two approaches: expert's knowledge [11,8] or filtering methods [12,8,7,13,4]. However, these methods do not consider complex interactions in a set of descriptors. A way to overcome this limitation is by the use of wrapper methods, which has received little attention from the AMP's research community, although it is an essential aspect for determining the performance of predictive models since those descriptors define the chemical space where each peptide is projected and in consequence the efficiency of the classification depends on it. Furthermore, currently, a large number of descriptors can be calculated for peptides. In earlier studies, the selection of molecular descriptors has often been made based on chemical intuition or observed properties that give rise to the antimicrobial activity [11,8]. On the other hand, recent studies employ hand-picked features (descriptors) procedures or filtering methods that independently evaluate the features according to a given criterion and select the top k features [8,7,4]. However, most of these approaches focused on the pairwise relationship and interaction of the descriptors, while the biological activity might depend on the relation of three or more descriptors.

Therefore, a feature selection procedure is needed in order to improve the performance of learning models [14]. In this paper, we propose a novel method to automatically select a peptide representation, based on molecular descriptors, that efficiently performs the classification of the peptide's antimicrobial activity. For this purpose, our method combines what we call a species adaptive genetic algorithm (SAGA) and a machine learning model to efficiently search promising solutions and to estimate the fitness directly for each subset of molecular descriptors. We systematically evaluate our proposed method and compared it with the state-of-the-art AMP classification methods on three well-known benchmarks.

2. Materials and methods

The aim of our approach is choosing a molecular descriptors' representation of peptides to discern between AMP and non-AMP sequences. The choice of descriptors can be formulated as a feature subset selection problem (FSSP). In supervised learning, the FSSP can be defined as: given a dataset described by a set of features, select those features that are useful for building a good classifier [15]. In general, the usefulness is given by the predictive power of the classifier instead of the relevance of individual features. Next, we introduce some notation and formally define the FSSP.

2.1. FSSP formulation

Consider a labeled dataset $\mathcal{D} = \{(\mathbf{x}^1, y^1), \ldots, (\mathbf{x}^n, y^n)\}$ of n peptides described by a set $X = \{X_1, \ldots, X_m\}$ of m input molecular descriptors and a label set $Y = \{\text{AMP}, \text{non} - \text{AMP}\}$. Here $\mathbf{x}^i = [x_1^i, x_2^i, \ldots, x_m^i]^T$ is an m-dimensional vector with a class label y_i from Y. The component x_j^i is the measurement of the j-th molecular descriptor for the i-th peptide.

Statement. Let $\mathcal{I}$ be a machine learning algorithm, $\mathcal{D}$ a dataset and J a performance criterion measured over all classifiers $\mathcal{I}(\mathcal{D})$, induced by $\mathcal{I}$ and $\mathcal{D}$, then, the formal definition of the FSSP is [16]:

$$\begin{aligned} &\underset{X'}{\text{maximize}} \quad f(X', \mathcal{D}) = J(\mathcal{I}(\mathcal{D}(X'))) \\ &\text{subject to} \quad X' \subseteq X. \end{aligned} \tag{1}$$

where $\mathcal{D}(X') \subseteq \mathcal{D}$ is a dimensional reduction of the dataset $\mathcal{D}$ obtained by removing the values of variables that are not in X' from each $\vec{x_i} \in \mathcal{D}$. It is important to note that the optimal subset feature X_{opt} is not necessarily unique, *i.e.,* it is possible to achieve the same value for the performance criterion using different subsets of features [16]. Notice also that the size of X_{opt} is unknown a *priori*, this makes FSSP harder than a related problem where the size of the desired feature set is given [17].

2.2. Characterization of wrapper methods

The formulation of the FSSP in this manner allows for the use of well-known optimization techniques that use, in their inner loops, machine learning algorithms to evaluate the quality of subsets of features. Methods that use the classification performance of the machine learning algorithm to guide the search towards the optimal subset of features are categorized as wrapper methods. According to [15], there are three considerations to characterize a wrapper method: (i) a search strategy; (ii) a performance estimation method and (iii) a machine learning algorithm.

(i) *Search strategies*. These define how to search through the space of feature subsets (there are $2^n - 1$ candidate feature subsets). In general, search strategies partially sample the search space, since for large values of n (*i.e.,* more than 40 features), the space becomes infeasible to be exhaustively explored [18]. The problem of finding the optimal feature subset is NP-hard [19]. Search strategies can be divided into three broad categories: exponential, greedy, and randomized. In short, the exponential search guarantees to find the optimal subset from a feature set. This strategy includes such searches as exhaustive enumeration, branch and bound [20] and beam search [21]. On the contrary, the greedy and randomized searches cannot guarantee to find the optimal subset. However, they are valid alternatives when the number of features is high. On the one hand, the greedy search makes a locally optimal choice, *i.e.,* it always selects a feature sequentially, for adding or removing, in order to maximize the current objective function (it adds or removes a single feature at a time to maximize the objective function). Some examples of greedy search algorithms are a sequential forward selection, sequential backward selection, bidirectional search, and greedy hill-climbing search [18]. On the other hand, the randomized search uses a sampling of the space of possible subsets for searching the optimal feature subset. The advantages of this approach are: it is possible to find a solution quickly, and it is capable of avoiding getting trapped at local optima [22]. Some example of randomized search are: MC1 [23], random mutation hill climbing [24], ant colony [25], simulated annealing [26] and genetic algorithms [27].

(ii) *Performance estimation methods*. To measure the quality of a feature subset, we need a performance estimation method that measures the predictive ability of the classifier induced by a particular machine learning algorithm and a dataset represented by the reduced feature set. Accordingly, the

optimal subset is the one with the best performing classifier. The performance estimation method employs a metric (*e.g.*, accuracy, MCC, precision, recall) and a re-sampling method to partition the dataset into a training and test sets. In the re-sampling method, the split process can be repeated multiple times. The most common methods are cross-validation and bootstrap [15].

(iii) *A machine learning algorithm*. Wrapper methods need a machine learning algorithm to build a classifier. Examples of machine learning algorithms includes support vector machine (SVM), random forest (RF), *k*-nearest neighbor (*k*-NN), multilayer perceptron (MLP) and c4.5 algorithm, among others.

2.3. The feature selection approach

In this subsection, we present a wrapper method to solve the FSSP problem. The three components exposed earlier for the AMP's classification problem is described next.

- **Search strategy**. We propose a Species Adaptive Genetic Algorithm for Feature Selection (SAGAFS). SAGAFS is an adapted version of two well-known algorithms: a Genetic Algorithm (GA) and Variable Length Representation Evolutionary Algorithm (VLREA) [28,29]. GA is commonly recommended for large-scale feature selection problems (*i.e.*, from now on 50 or more candidate features) [27]. On the other hand, VLREA is appropriate for problems where the solution length contributes to its fitness, as it happens in our case. To the best of our knowledge, this is the first time a VLR evolutionary algorithm is applied to solve the feature selection problem.
  The proposed SAGAFS algorithm includes a variable length representation and neighboring spaces strategies to efficiently sample the vast search space.
- **Performance estimation method.** We used *k*-fold cross-validation to estimate the average Matthews correlation coefficient (MCC) of the induced classifier. In the *k*-fold cross-validation, the dataset is partitioned into *k* non-empty disjoint subsets $\mathcal{D}_1, \ldots, \mathcal{D}_k$. Each subset (*i.e.*, fold) has roughly equal size. Then, we repeat *k* times the following procedures: the machine learning algorithm induces a classifier using the dataset D/D_i, and the classifier is tested on the subset D_i. The MCC estimation is calculated by averaging it over the *k* runs. This coefficient is given by

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

  where TP, TN, FP, and FN are the number of True Positives, True Negatives, False Positive, and False Negatives, respectively.
- **A machine learning algorithm.** For the generation of a binary classifier, we used two machine learning algorithms: the first one, a linear classifier, Support Vector Machine-linear (SVM-L); the second one, a non-linear classifier, Random Forest (RF).

The methodology adopted in this study is described in the following sections and a scheme of this is shown in Fig. 1.

2.4. From peptide sequences to molecular descriptors

Several studies have been found five major properties related to the antimicrobial activity of peptides; these include conformation, charge, hydrophobic character and secondary structure [42,43,11,44]. In this direction, molecular descriptors have been widely applied for extracting these properties from peptides in a quantitative way.

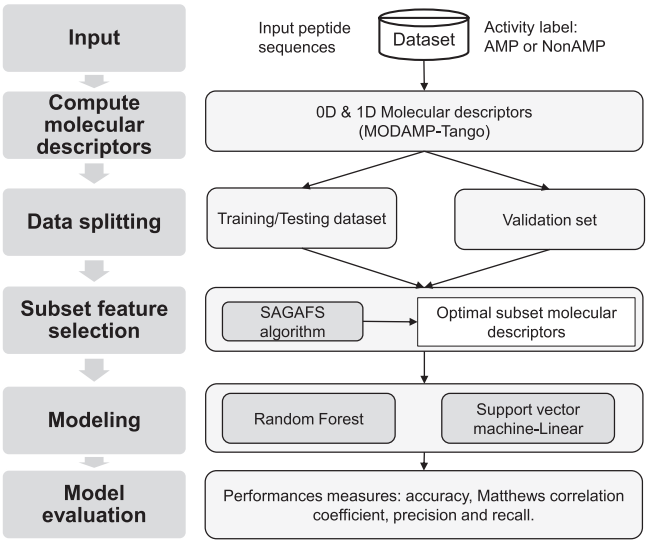

Fig. 1. Schematic process of the automatic selection of peptide representation based on molecular descriptors and the antimicrobial activity classification.

In this study, a total of 122 molecular descriptors were collected, from which 272 values were derived. The number of molecular descriptors for each property was: 74 at amino acid composition, 10 at charge, 31 at hydrophobic character, 5 at secondary structure and 2 at other properties. The molecular descriptors included in this work have been used in previous antimicrobial peptide studies (see Table 1). These can be calculated from peptide sequences.

To compute the molecular descriptors, we used two different software packages: Tango software [40,41,45] and an in–house MOlecular Descriptor for AntiMicrobial Peptides (MODAMP). Tango was used to calculate four descriptors related to the secondary structure (AGG, TURN, BETA, HELIX), whereas MODAMP was used to compute the remaining 268 descriptors which are listed in Supplementary File 1.

In this step, we assumed that each peptide, from the input dataset, was a valid sequence $S_i = s_1, \ldots, s_l$, i.e., each symbol s_j comes from the standard amino acid alphabet of size 20. Consider the set of molecular descriptors $\{X_1, \ldots, X_{272}\}$, we convert each sequence S_i into a 272-dimensional vector $\mathbf{x}^i = [x_1^i, \ldots, x_m^i]^T$, each component x_j^i encodes the value for the molecular descriptor X_j of sequence S_i.

2.5. Feature subset selection algorithm

2.5.1. Solution representation

The design of a suitable representation for candidate solutions is an essential step in a genetic algorithm; since it defines a mapping of candidate-solutions space, referred to as phenotypic space, to the problem-solving space, referred to as genotypic space. As we described earlier, the phenotype space for the FSSP (1) is the collection of all subsets of the input feature set X, excluding the empty set. Previous works on genetic algorithms for the FSSP have considered a fixed-length binary string to represent the phenotype, where each bit position (fixed to 1 or 0) encodes whether each one of the m features of X is selected or not [46,27]. However, taking into consideration a large number of molecular descriptors that are computable in peptides and that the candidate solutions are just a subset of them, the binary encoding might generate large chromosomes with only a few bits encoding the features for a candidate solution. For this reason, we considered a variable length

Table 1
Summary of the 272 molecular descriptors considered as a the universe set of feature for the peptide representation, these are grouped by dimensionality into 0D and 1D.

Group	Name	No. of molecular descriptors	No. of descriptors' values	Reference
0D	Standard amino acid composition	1	20	[30]
	Reduce amino acid composition	10	41	[31,32]
	Aliphatic index	1	1	[30]
	Net charge and mean net charge	6	6	[33]
	Grand Average of Hydrophilicity	2	2	[33]
	Grand Average of Hydropathy (GRAVY)	1	1	[30]
	Grand Average of Hydrophobicity	23	23	[33]
	Charge at different pH values (5, 7, and 9)	3	3	[34]
	Boman index	1	1	[35]
	Molecular weight	1	1	[30]
	Number of amino acids	1	1	[30]
1D	Instability index	1	1	[30,36]
	Reduced amino acid Transition	10	21	[31,32]
	Reduced amino acid distribution	50	105	[31,32]
	Dipeptide	1	9	[32]
	Tripeptide	1	27	[32]
	Max mean hydrophobicity	1	1	[37]
	Hydrophobic moment	3	3	[38]
	Isoelectric Point	1	1	[39,30]
	In vitro aggregation	1	1	[40,41]
	turn structure propensity	1	1	[40,41]
	β-sheet propensity	1	1	[40,41]
	α-helix propensity	1	1	[40,41]
	Total	**122**	**272**	

representation (VLR) that allows encoding only the features related to the candidate solution.

In SAGAFS, a chromosome g is a subset of integers {1,...,m} that encodes the index for each selected feature. Then a given genotype $g = \{g_1, g_2, \ldots, g_k\}, g_i \in \{1, \ldots, m\}$ of cardinality k represents the subset $X_g = \{X_{g1}, X_{g2}, \ldots, X_{gk}\}$. Next we show an example of an individual and its corresponding solution (phenotpye).

Chromosome Candidate solution

$g = \{1, 3, 5\} \quad \rightarrow \quad X_g = \{X_1, X_3, X_5\}$

2.5.2. Fitness function

The quality of a subset X_g is based on the performance of induced classifiers by a machine learning algorithm and the training dataset with only that subset of features. Additionally, we include a second term, which measures the model complexity in terms of the number of features. Hence, higher performance of X_g indicates a better candidate solution, and if two subsets have the same performance, the simplest one indicates a more suitable candidate. The quality of a given subset X_g, represented by a chromosome g, is defined as,

$$f(X_g) = J(\mathcal{D}(X_g)) + \lambda \frac{|X_g|}{m} \tag{2}$$

with

$$J(\mathcal{D}(X_g)) = \frac{1}{k}\sum_{i=1}^{k} |MCC_i(\mathcal{I}(\mathcal{D}(X_g) - \mathcal{D}_i(X_g), \mathcal{D}_i(X_g)))|$$

where $J(\mathcal{D}(X_g))$ is the *MCC* estimated by k-fold cross-validation. Here, $|.|$ is the absolute value of MCC_i obtained by testing the induced classifier $\mathcal{I}$ with the validation set $\mathcal{D}_i$, where $\mathcal{I}$ is trained with the set $\mathcal{D} - \mathcal{D}_i$. The second term in (2) is a tiebreaker criterion to encourage small subsets, where, λ is a value in the range $[10^{-2}, 10^{-4}]$ and m is the cardinality of the universe of features.

2.5.3. Main steps of SAGAFS

- At $t = 0$, a population $P(t)$ of N_{pop} individuals was randomly generated. In each chromosome ($g \in P(t)$) k-integer values out of m available are selected at random, where k is restricted by lower L_i and upper U_i bounds. These bounds are employed to restrict the size of individuals delimiting the dimensionality of the feature space that is sampled by SAGAFS at the moment. Subsequently, the fitness value of each g in $P(t)$ is computed by using the function (2).
- From the current population P(t), μ individuals are obtained by the standard binary tournament with replacement scheme [47]. The obtained individuals are added and scrambled in the parent set, denoted as $M(t)$.
- Each pair of consecutive parents (g_i and g_{i+1} in $M(t)$ for $i \leqslant \mu - 1$) is recombined with probability p_c by using the subset size-oriented common feature crossover operator (SSOCF) [48]. The SSOCF is adapted for our representation (*i.e.,* VLR) because the original version has been designed for fixed-length representations (the details of adapted SSOCF is shown in Supplementary File 2. The SSOCF is used to preserve the common features of the parent into their offspring. As a result, this operator produces two children (o_i and o_{i+1}) for each pair of parents.
- For each offspring in the offspring set, denoted as $O(t)$, a k-indel mutation with probability p_m is applied. Conventionally, p_m is a user-defined and a static value, however in SAGAFS, it was dynamically estimated by a self-adaptive mutation method [49]. This method is used to increase the p_m if the current population $P(t)$ is over a similarity threshold (*i.e,*low diversity), otherwise p_m is decreased. In detail, the similarity value of a population is given by $s(P(t)) = \frac{s}{N_{pop}}$, where s is the number of identical individuals in $P(t)$. The self-adaptive mutation probability p_m was calculated as follows:

$$p_m = p_0 + sgn(s) \times \sigma \tag{3}$$

with

$$sgn(s) = \begin{cases} -1 & \text{if } s < \theta \\ 0 & \text{if } s = \theta \\ 1 & \text{if } s > \theta \end{cases} \tag{4}$$

where θ is a specified threshold, p_0 is the initial mutation probability and σ is the step size.

We developed a mutation operator, named k-insertions/deletions (k-indel), for the dimensional variation of a particular offspring. k-indel works by randomly picking k integers from

[1,m]. Each integer is inserted or deleted into the offspring, depending on, whether or not the integer is in the offspring. To illustrate this operator, we introduce the following example:

$$o = \{2,3,8,10\} \rightarrow o = \{3,\mathbf{6},8,\mathbf{9},10\} \quad \text{where} \quad k \in \{2,6,9\}$$

Note that in this example feature number 2 is deleted while features number 6 and 9 are added. As a next step in the SAGAFS, we compute the fitness value of each o in the offprings population $O(t)$.

- To select the chromosomes that will form part of the population $P(t+1)$ in the next generation, we performed the standard survival selection elitism [47], thus, the current population set $P(t)$ and the offspring set $O(t)$, each comprises of N_{pop} and μ individuals, were merged and sorted by their fitness values. After that, the N_{pop} top individuals were selected as the new population ($P(t+1)$).
- We defined a stop criterion, in accordance with the maximum number of generations n_g and number of generations without improvement n_{gwi} in the objective function given by 2.

The best feature subset X_{opt} and its fitness values are provided as an output of SAGAFS. From this information, both, the validation set and the training set were reduced in size (*i.e.*, the dataset set were represented only with the optimal subset of features).

3. Results and discussions

To evaluate the SAGAFS performance, we run it 30 times for each dataset. Then we selected the best solution obtained for each dataset and compared them with the state-of-art AMP classification methods. The best solution obtained for SAGAFS was compared with publicly available AMP prediction tools. The implemented algorithm and the evaluated datasets are available for download at https://github.com/gdelrioifc/AMPFeatureSelection. The evaluation of this algorithm along with the main results are described next.

3.1. Peptide datasets (Benchmarks)

We considered three benchmark datasets widely used for the binary antimicrobial classification task. We used these datasets to measure, in an unbiased way, the performance of molecular descriptors obtained by SAGAFS. They are: DAT1 [4], DAT2 [6], and DAT3 used in [10] was taken from [50,51]. Fig. 2 shows the overlapping among these datasets, the left part shows the intersection of all datasets, while the right part shows the intersections of their partitions in AMPs and Non-AMPs. It can be observed that the overlap is only among the sets of antimicrobial peptides (AMPs), even though the three datasets used a similar methodology to retrieve non-antimicrobial (Non-AMPs) sequences. A criterion to measure how difficult it is to discriminate a set of AMPs from Non-AMPs, at the sequence level, is by computing their similarity. If this similarity is close to zero then the set is not challenging since a simple sequence-similarity-based algorithm will be able to separate the classes. On the contrary, if this measure is large then the dataset will be difficult to separate at the sequence level. After computing a similarity measure, with Dover Analyzer software [52] at a 30% threshold, we found the following similarity values: DAT1 has 0.88%, DAT2 36.56%, and DAT3 18.83%. This means that, at sequence level, DAT1 should be the easiest dataset to discriminate, while DAT2 should be the hardest one.

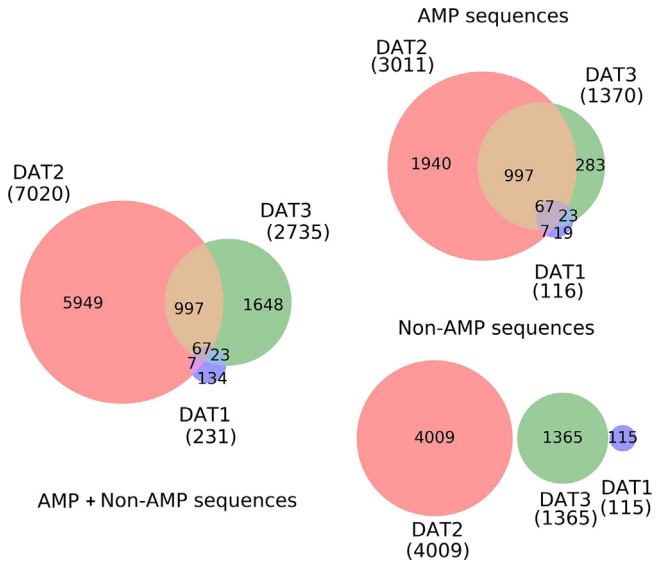

Fig. 2. Venn diagram of considered benchmark datasets for SAGAFS's test. The level of overlap among datasets DAT1 [4], DAT2 [6], and DAT3 [10] corresponds only to AMPs, *i.e.*, there is no intersection between non-antimicrobial peptides of any pair of datasets.

3.2. AMP prediction methods

Many methods for AMPs' classification have been described in the literature, unfortunately, only a few of them are publicly available. We analyzed the performance of four state-of-art AMP classification methods and six publicly available AMP tools.

We compared the performance of our approach with the following methods: ANFIS [4], CAMP [7], iAMP-2L [10], and MLAMP [53]. The same datasets reported by these methods were used to perform such comparison; DAT1 was used to compare our method with ANFIS [4], DAT2 was used to compare with CAMP [7], and DAT3 to compare with iAMP-2L [10] and with MLAMP [53].

3.3. Classification results

Table 2 shows the mean results obtained by SAGAFS after 30 runs for each dataset. The fitness function computation was performed by using 10-fold cross-validation over 75% of the data for DAT1, 70% for DAT2, and 100% for DAT3, following what the authors of the methods using these data did. In general, SAGAFS has a uniform performance through the 30 runs (*i.e.*, its standard deviation is small). Furthermore, the machine learning algorithm with best performances is Random Forest (RF). The best performance was observed for DAT3 with Acc(%) of 96.28 ± 0.61 and the MCC was 0.93 ± 0.01. These results outperform the ones recently presented by a method that used a linear projection of features' subspaces [13] instead of a wrapper to achieve the same goal; we gained this improvement at the expense of a higher computational cost.

To study the impact of the optimal set of descriptors obtained by SAGAFS on the classifiers' performances, we compared the performances achieved when using all candidate molecular descriptors with the results shown in Table 2. The comparison is shown in Fig. 3 and indicates that, on average, the classifier constructed using the solution of SAGAFS is competitive with respect to the base-line classifier, i.e., the classifier that uses the 272 molecular descriptors (*i.e.*, SVM-control, RF-control), the performances are indicated by triangles in Fig. 3. In all cases, the cardinality of the optimal descriptors sets represent a reduction of at least 75% with

Table 2
Mean performance values with their respective standard deviation of the best solutions obtained with the SAGAFS algorithm for the three benchmark datasets after 30 runs. The results are presented as the mean ± one standard deviation.

Dataset	MLA*	Acc (%)	Sn	Sp	F-score	MCC	ROC area
DAT1	SVM-L	92.70(±1.51)	0.91(±0.05)	0.94(±0.04)	0.93(±0.02)	0.86(±0.03)	0.93(±0.02)
	RF	93.76(±1.01)	0.93(±0.02)	0.94(±0.02)	0.94(±0.01)	0.88(±0.02)	0.95(±0.01)
DAT2	SVM-L	82.01(±0.73)	0.81(±0.03)	0.83(±0.03)	0.82(±0.01)	0.63(±0.02)	0.82(±0.01)
	RF	92.50(±0.40)	0.91(±0.04)	0.93(±0.03)	0.92(±0.00)	0.85(±0.01)	0.97(±0.00)
DAT3	SVM-L	95.12(±0.42)	0.94(±0.01)	0.96(±0.00)	0.95(±0.00)	0.90(±0.01)	0.95(±0.00)
	RF	96.28(±0.61)	0.96(±0.01)	0.96(±0.01)	0.96(±0.01)	0.93(±0.01)	0.99(±0.00)

* MLA, Machine Learning Algorithm: RF = Random Forest; SVM-L = Support Vector Machine-Linear.

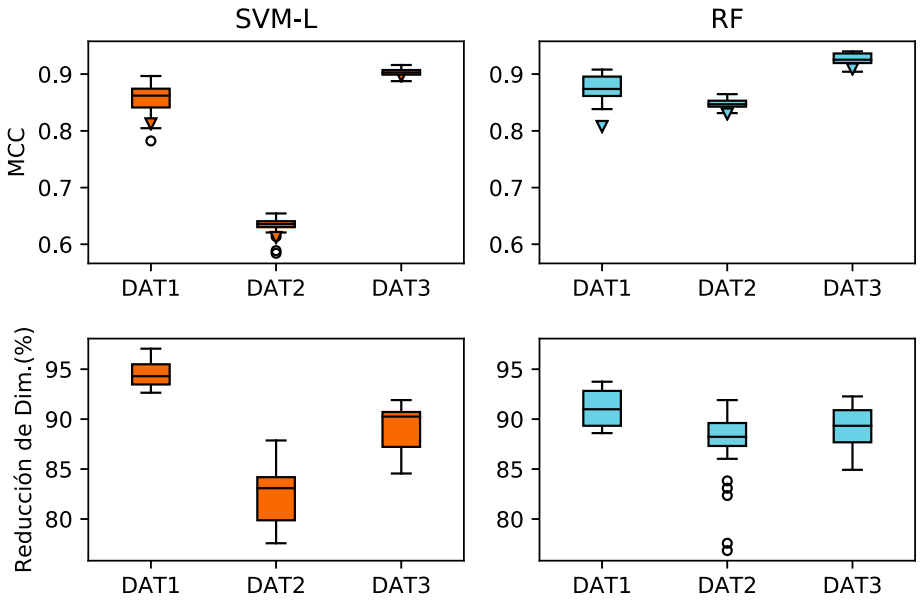

Fig. 3. Performance comparison among the best solutions obtained by SAGAFS + SVM-L and SAGAFS + RF after 30 runs. The triangles indicate the MCC for the base-line model (upper left and right figures). The lower part (left and right) depicts the percentage of reduction in number of descriptors with respect to the base-line (272 descriptors).

respect to the size of 272 descriptors. For instance, the best-obtained solution, for DAT3, needed only 39 molecular descriptors to achieve an accuracy of 96.64% with an MCC of 0.94 and an ROC area of 0.99.

Previous works that used DAT1 [4] and DAT2 [6] started with a universe of features that are a subset of the starting set of features used here. However, the work that used DAT3 [10] built a set of features based on amino acid composition, measuring five physicochemical properties (hydrophobicity, pK1, pK2, pI, and molecular weight) for each of the 20 amino acids, for an initial set of 100 descriptors. The final set of features for the work dealing with DAT1 was composed of two features, in vitro aggregation and peptide length, the latter also selected by our wrapper method, while that using DAT2 [6] ends up with 64 features, unfortunately these were not provided by the authors. Hence, at this point, we could see some similarities of the features previously selected to model AMP, but it is not possible to fully compare them with those observed in our study.

3.4. Relevance of selected features

Since the feature selection algorithm (SAGAFS) is run 30 times, each one of these runs generates a set of features that corresponds to the fittest individual found in that run. Then, for every feature in the 30 sets of features we compute its relative frequency. Fig. 4 shows the indices of the most frequently selected features. Each graph depicts the top 10 most frequently selected features when SAGAFS is run. The top rows show the results when SAGAFS uses a SVM method applied to DAT1 (first column), DAT2 (second column), and DAT3 (third column). The bottom row shows the same results but now when SAGAFS uses a Random Forest classifier. For instance, for DAT1, the most selected feature for both models (SVM and RF) is the feature given by the index number 268 which corresponds to in vitro aggregation.

If we analyze the top 10 features, as they are selected from the best solution in every one of the 30 SAGAFS runs, that simultaneously appear under both learning models (i.e., under SVM-L and RF), we found the following coincidences.

For DAT1, in vitro aggregation (268), length (257), electric charge (27), and maximum of the mean hydrophobicity (263) appear in both models. For DAT2, molecular weight (258), length (257), and electric charge (22). For DAT3, frequency of Metionine (10), amphipathicity (28), frequency of Tryptophane (18), and solvent accessibility of certain k-mers (92). The biological significance of some of these features has already been indentified in previous works, for instance, net charge, amphipathicity and hydrophobicity properties were found to be relevant for the antimicrobial activity [4]. On the other hand, Tryptophan has already noted to be present in a family of archetypal AMP [54], yet Methionine has not (see for instance [55,56]). Our results suggest that Methionine may be enriched on AMP with respect to the non-AMP, despite being under-represented in AMP. We believe these results may promote further investigation onto the role of such amino acid on AMP function.

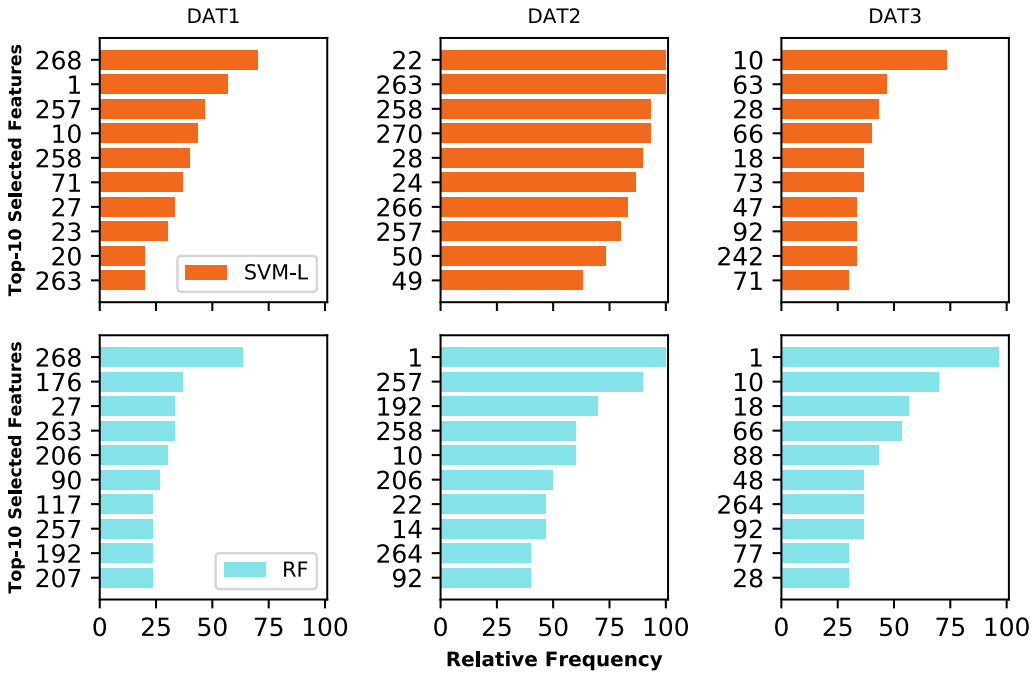

Fig. 4. Most frequently selected features for SAGAFS on each dataset. The plots in the lower part represent the indices for the most frequent features for the model generated by Random Forest (RF), while the plots in the upper part show the indices for the SVM-L.

Table 3
Performance comparison of SAGAFS method with ANFIS [4] on the dataset DAT1.

Method	MLA[a]	Dataset	ACC(%)	Sn	Sp	F1-score	MCC
[4]	ANFIS	Training	96.23	1.00	0.93	0.96	0.93
		Testing	100	1.00	1.00	1.00	1.00
		Validation	94.34	0.96	0.92	0.95	0.89
		Overall	96.73	**0.99**	0.95	**0.97**[b]	**0.94**
SAGAFS	RF	Training	100	1.00	1.00	1.00	1.00
		Testing	84.48	0.88	0.79	0.84	0.70
		Validation	100	1.00	1.00	1.00	1.00
		Overall	**96.89**	0.97	**0.97**	**0.97**	**0.94**

[a] Machine Learning Algorithm (MLA): RF = Random Forest; ANFIS = Adaptive Neuro-fuzzy Inference System.
[b] Bold font indicates the best value per measure.

Table 4
Performance comparison of SAGAFS method and CAMP [7] on the dataset DAT2.

Method	MLA[a]	MCC		Performance in (%)			10-fold CV
		Train	Test	Sn	Sp	ACC	ACC(%)
[7]	RF	0.82	**0.84**	**90.8**	93.7	**92.5**	**93.4**
	SVM	**0.91**	0.83	89.7	93.1	91.6	92.6
	ANN	0.72	0.72	82.9	88.9	86.3	86.9
SAGAFS	RF	0.87	**0.84**	88.5	**95.14**	92.4	93.3

[a] Machine Learning Algorithm (MLA): RF = Random Forest; SVM = Support Vector Machine with polynomial kernel (degree 4); ANN = Artificial Neural Network. Bold font indicates the best value per measure.

Table 5
Performance comparison of SAGAFS method with iAMP-2L [10] and MLAMP [53] on the dataset DAT3.

Method	MLA[a]	Sn(%)	Sp(%)	ACC(%)	MCC
iAMP-2L [10]	FKNN	87.13	86.03	86.32	0.727
MLAMP [53]	RF	77.00	94.60	89.90	0.737
SAGAFS	RF	**96.64**	**97.36**	**97.00**	**0.940**

[a] Machine Learning Algorithm (MLA): RF = Random Forest; FKNN = Fuzzy K-Nearest Neighbor. Bold font indicates the best value per measure.

3.5. Performance comparison with state-of-art classifiers

Table 3 compares the performances of SAGAFS witn ANFIS [4] on DAT1, where ANFIS outperforms SAGAFS on the testing dataset and the opposite occurs on the validation dataset, while the overall performance remains similar for both algorithms. The results are in accordance with the low similarity between AMPs and Non-AMPs' sequences for this dataset, i.e., we expected to have this high performance results since the classes are not hard to separate even at the sequence level.

The comparison of SAGAFS with CAMP on DAT2 is shown in Table 4. This is the hardest dataset according to the similarity measure used. The performances of SAGAFS and CAMP are similar in MCC and ACC(%) metrics.

Table 5 compares the performance of SAGAFS with iAMP-2L [10] and MLAMP [53] on the DAT3. The performance achieved by SAGAFS is higher than the performances reported by iAMP-2L and MLAMP, over all performance measures. This is the second hardest dataset according to the similarity measure employed.

4. Conclusion

A novel and effective evolutionary algorithm to solve the feature selection problem in antimicrobial peptides classification has been proposed. The approach combines two algorithms, a Genetic Algorithm and a variable length evolutionary algorithm with an objective function that combines the classifier's MCC measure with the chromosome length, i.e. the number of selected descriptors. Results from computational experiments show that the proposed method is able to find a representation for the peptides capable of generating models that outperform state-of-the-art methods that are publicly available for AMP classification. Our findings suggest that our approach could be used in preliminary computational screening in order to identify novel antimicrobial peptides, efficiently.

Future research is aimed at extending the comparison with other available AMP predictors over a larger dataset. We are also planning to apply our SAGAFS algorithm to multi-class classification of AMPs, i.e., once you know that a given peptide is AMP, identify its specific function.

CRediT authorship contribution statement

Jesus A. Beltran: Conceptualization, Methodology, Writing - original draft, Software, Validation. **Gabriel Del Rio:** Conceptualization, Writing - review & editing, Validation. **Carlos A. Brizuela:** Conceptualization, Methodology, Validation, Writing - review & editing, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

CAB acknowledges the support of CONACYT under grant A1-S-20638. JAB, GDR and CAB acknowledge the support of CONACyT under grant FOIN-219.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.csbj.2020.02.002.

References

[1] Cherkasov A, Hilpert K, Jenssen H, Fjell CD, Waldbrook M, Mullaly SC, Volkmer R, Hancock RE. Use of artificial intelligence in the design of small peptide antibiotics effective against a broad spectrum of highly antibiotic-resistant superbugs. ACS Chem Biol 2008;4(1):65–74.
[2] Usmani SS, Bedi G, Samuel JS, Singh S, Kalra S, Kumar P, Ahuja AA, Sharma M, Gautam A, Raghava GP. Thpdb: database of fda-approved peptide and protein therapeutics. PLoS One 2017;12(7): e0181748.
[3] Jenssen H. Descriptors for antimicrobial peptides. Expert Opin Drug Discovery 2011;6(2):171–84.
[4] Fernandes FC, Rigden DJ, Franco OL. Prediction of antimicrobial peptides based on the adaptive neuro-fuzzy inference system application. Peptide Sci 2012;98(4):280–7.
[5] Fjell CD, Jenssen H, Hilpert K, Cheung WA, Pante N, Hancock RE, Cherkasov A. Identification of novel antibacterial peptides by chemoinformatics and machine learning. J Med Chem 2009;52(7):2006–15.
[6] Thomas S, Karnik S, Barai RS, Jayaraman VK, Idicula-Thomas S. Camp: a useful resource for research on antimicrobial peptides. Nucleic Acids Res 2009(suppl 1):D774–80.
[7] Waghu FH, Gopi L, Barai RS, Ramteke P, Nizami B, Idicula-Thomas S. Camp: Collection of sequences and structures of antimicrobial peptides. Nucl Acids Res 2014;42(D1):D1154–8.
[8] Torrent M, Andreu D, Nogués VM, Boix E. Connecting peptide physicochemical and antimicrobial properties by a rational prediction model. PloS one 2011;6(2): e16968.
[9] Randou EG, Veltri D, Shehu A. Binary response models for recognition of antimicrobial peptides. In: Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics. ACM; 2013. p. 76.
[10] Xiao X, Wang P, Lin W-Z, Jia J-H, Chou K-C. iamp-2l: a two-level multi-label classifier for identifying antimicrobial peptides and their functional types. Anal Biochem 2013;436(2):168–77.
[11] Fjell CD, Hiss JA, Hancock RE, Schneider G. Designing antimicrobial peptides: form follows function. Nat Rev Drug Discovery 2012;11(1):37–51.
[12] Veltri D, Kamath U, Shehu A. Improving recognition of antimicrobial peptides and target selectivity through machine learning and genetic programming. IEEE/ACM Trans Comput Biol Bioinf 2015;14(2):300–13.
[13] Beltran JA, Aguilera-Mendoza L, Brizuela CA. Optimal selection of molecular descriptors for antimicrobial peptides classification: an evolutionary feature weighting approach. BMC Genomics 2018;19(7):672.
[14] Gabere MN, Noble WS. Empirical comparison of web-based antimicrobial peptide prediction tools. Bioinformatics 2017;33(13):1921–9.
[15] Guyon I, Elisseeff A. An introduction to variable and feature selection. J Mach Learn Res 2003;3(Mar):1157–82.
[16] Kohavi R, John GH. Wrappers for feature subset selection. Artif Intell 1997;97(1–2):273–324.
[17] Webb AR. Statistical pattern recognition. 2nd ed. John Wiley & Sons; 2003. Ch. 9, pp. 305–360.
[18] James G, Witten D, Hastie T, Tibshirani R. Linear model selection and regularization. In: An Introduction to Statistical Learning. Springer; 2013. p. 203–64.
[19] Amaldi E, Kann V. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. Theor Comput Sci 1998;209(1–2):237–60.
[20] Somol P, Pudil P, Kittler J. Fast branch & bound algorithms for optimal feature selection. IEEE Trans Pattern Anal Mach Intell 2004;26(7):900–12.
[21] Lowerre BT. The harpy speech recognition system, Ph.D thesis. Carnegie-Mellon University; 1976.
[22] Stracuzzi DJ. Randomized feature selection. In: Computational Methods of Feature Selection. Chapman and Hall/CRC; 2007. p. 57–78.
[23] Prototype and feature selection by sampling and random mutation hill climbing algorithms. In: Cohen WW, Hirsh H (Eds.), Machine Learning Proceedings 1994, Morgan Kaufmann, San Francisco (CA); 1994. p. 293–301. .
[24] Skalak DB. Prototype and feature selection by sampling and random mutation hill climbing algorithms. In: Proceedings of the eleventh international conference on machine learning. p. 293–301.
[25] Ali SI, Shahzad W. A feature subset selection method based on conditional mutual information and ant colony optimization. Methods 2012;1(2):3–4.
[26] Doak J. An evaluation of feature selection methods and their application to computer security. UC Davis Dept of Computer Science tech reports 1992.
[27] Huang J, Cai Y, Xu X. A hybrid genetic algorithm for feature selection wrapper based on mutual information. Pattern Recogn Lett 2007;28(13):1825–44.
[28] Zebulum RS, Vellasco M, Pacheco MA. Variable length representation in evolutionary electronics. Evol Comput 2000;8(1):93–120.
[29] Harvey I. Species adaptation genetic algorithms: a basis for a continuing saga. In: Toward a practice of autonomous systems: Proceedings of the first european conference on artificial life. p. 346–54.
[30] Gasteiger E, Hoogland C, Gattiker A, Wilkins MR, Appel RD, Bairoch A, et al. Protein identification and analysis tools on the expasy server. In: The proteomics protocols handbook. Springer; 2005. p. 571–607.
[31] Dubchak I, Muchnik I, Holbrook SR, Kim S-H. Prediction of protein folding class using global description of amino acid sequence. Proc Natl Acad Sci 1995;92(19):8700–4.
[32] Li Z-R, Lin HH, Han L, Jiang L, Chen X, Chen YZ. Profeat: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence. Nucl Acids Res 2006;34(suppl_2):W32–7.

[33] Klein P, Kanehisa M, DeLisi C. Prediction of protein function from sequence properties: Discriminant analysis of a data base. Biochimica et Biophysica Acta (BBA)-Protein Structure and Molecular 1984;787(3):221–6.
[34] Piotto SP, Sessa L, Concilio S, Iannelli P. Yadamp: yet another database of antimicrobial peptides. Int J Antimicrobial Agents 2012;39(4):346–51.
[35] Boman H. Antibacterial peptides: basic facts and emerging concepts. J Internal Med 2003;254(3):197–215.
[36] Guruprasad K, Reddy BB, Pandit MW. Correlation between stability of a protein and its dipeptide composition: a novel approach for predicting in vivo stability of a protein from its primary sequence. Protein Eng, Des Selection 1990;4 (2):155–61.
[37] Eisenberg D, Schwarz E, Komaromy M, Wall R. Analysis of membrane and surface protein sequences with the hydrophobic moment plot. J Mol Biol 1984;179(1):125–42.
[38] Eisenberg D, Weiss RM, Terwilliger TC, Wilcox W. Hydrophobic moments and protein structure. Faraday Symposia of the Chemical Society, vol. 17. Royal Society of Chemistry; 1982. p. 109–20.
[39] Kozlowski LP. Ipc-isoelectric point calculator. Biol Direct 2016;11(1): 55.
[40] Conchillo-Solé O, de Groot NS, Avilés FX, Vendrell J, Daura X, Ventura S. Aggrescan: a server for the prediction and evaluation of hot spots of aggregation in polypeptides. BMC Bioinf 2007;8(1):65.
[41] Fernandez-Escamilla A-M, Rousseau F, Schymkowitz J, Serrano L. Prediction of sequence-dependent and mutational effects on the aggregation of peptides and proteins. Nat Biotechnol 2004;22(10):1302.
[42] Kang H-K, Kim C, Seo CH, Park Y. The therapeutic applications of antimicrobial peptides (amps): a patent review. J Microbiol 2017;55(1):1–12.
[43] Mahlapuu M, Håkansson J, Ringstad L, Björn C. Antimicrobial peptides: an emerging category of therapeutic agents. Front Cell Infection Microbiol 2016;6.
[44] Wang G, Li X, Zasloff M, et al. A database view of naturally occurring antimicrobial peptides: nomenclature, classification and amino acid sequence analysis, Antimicrobial peptides: discovery, design and novel therapeutic strategies; 2010, p. 1–21.
[45] Linding R, Schymkowitz J, Rousseau F, Diella F, Serrano L. A comparative study of the relationship between protein structure and β-aggregation in globular and intrinsically disordered proteins. J Mol Biol 2004;342(1):345–53.
[46] Kabir MM, Shahjahan M, Murase K. A new local search based hybrid genetic algorithm for feature selection. Neurocomputing 2011;74(17):2914–28.
[47] Eiben AE, Smith JE, et al. Introduction to evolutionary computing, vol. 53. Springer; 2003.
[48] Emmanouilidis C, Hunter A, MacIntyre J. A multiobjective evolutionary setting for feature selection and a commonality-based crossover operator. In: Evolutionary Computation, 2000. Proceedings of the 2000 Congress on, vol. 1, IEEE; 2000. p. 309–16. .
[49] Smullen D, Gillett J, Heron J, Rahnamayan S. Genetic algorithm with self-adaptive mutation controlled by chromosome similarity. In: Evolutionary Computation (CEC), 2014 IEEE Congress on, IEEE; 2014. p. 504–11. .
[50] Wang Z, Wang G. Apd: the antimicrobial peptide database. Nucl Acids Res 2004;32(suppl 1):D590–2.
[51] Wang G, Li X, Wang Z. Apd2: the updated antimicrobial peptide database and its application in peptide design. Nucl Acids Res 2009;37(suppl 1):D933–7.
[52] Aguilera-Mendoza L, Marrero-Ponce Y, Tellez-Ibarra R, Llorente-Quesada MT, Salgado J, Barigye SJ, Liu J. Overlap and diversity in antimicrobial peptide databases: compiling a non-redundant set of sequences. Bioinformatics 2015;31(15):2553–9.
[53] Lin W, Xu D. Imbalanced multi-label learning for identifying antimicrobial peptides and their functional types. Bioinformatics 2016;32(24):3745–52.
[54] Shagaghi N, Palombo EA, Clayton AH, Bhave M. Archetypal tryptophan-rich antimicrobial peptides: properties and applications. World J Microbiol Biotechnol 2016;32(2):31.
[55] Wang X, Mishra B, Lushnikova T, Narayana JL, Wang G. Amino acid composition determines peptide activity spectrum and hot-spot-based design of merecidin. Adv Biosyst 2018;2(5):1700259.
[56] Jhong J-H, Chi Y-H, Li W-C, Lin T-H, Huang K-Y, Lee T-Y. dbamp: an integrated resource for exploring antimicrobial peptides with functional activities and physicochemical properties on transcriptome and proteome data. Nucl Acids Res 2019;47(D1):D285–97.