Expert Systems with Applications 36 (2009) 12086–12094

Contents lists available at ScienceDirect

Expert Systems with Applications

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

A novel ACO–GA hybrid algorithm for feature selection in protein function prediction

Shahla Nemati[a], Mohammad Ehsan Basiri[a,*], Nasser Ghasem-Aghaee[b], Mehdi Hosseinzadeh Aghdam[c]

[a] Young Research Club, Islamic Azad University, Arsenjan Branch, Fars, Iran
[b] Computer Engineering Department, University of Isfahan, Hezar Jerib Avenue, Isfahan, Iran
[c] Computer Engineering Department, Technical & Engineering Faculty of Bonab, University of Tabriz, Tabriz, Iran

ARTICLE INFO

Keywords:
Protein function prediction
Ant colony optimization (ACO)
Genetic algorithm (GA)
Feature selection (FS)
Hierarchical classification
Bioinformatics

ABSTRACT

Protein function prediction is an important problem in functional genomics. Typically, protein sequences are represented by feature vectors. A major problem of protein datasets that increase the complexity of classification models is their large number of features. Feature selection (FS) techniques are used to deal with this high dimensional space of features. In this paper, we propose a novel feature selection algorithm that combines genetic algorithms (GA) and ant colony optimization (ACO) for faster and better search capability. The hybrid algorithm makes use of advantages of both ACO and GA methods. Proposed algorithm is easily implemented and because of use of a simple classifier in that, its computational complexity is very low. The performance of proposed algorithm is compared to the performance of two prominent population-based algorithms, ACO and genetic algorithms. Experimentation is carried out using two challenging biological datasets, involving the hierarchical functional classification of GPCRs and enzymes. The criteria used for comparison are maximizing predictive accuracy, and finding the smallest subset of features. The results of experiments indicate the superiority of proposed algorithm.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

Protein function prediction is an important problem in functional genomics. Proteins are large molecules that perform nearly all of the functions of a cell in a living organism (Alberts et al., 2002). The primary sequence of a protein consists of a long sequence of amino acids. Proteins are the most essential and versatile macromolecules of life, and the knowledge of their functions is a crucial link in the development of new drugs, better crops, and even the development of synthetic biochemical such as biofuels.

Over the past few decades, major advances in the field of molecular biology, coupled with advances in genomic technologies, have led to an explosive growth in the biological information generated by the scientific community. Although the number of proteins with known sequence has grown exponentially in the last few years, due to rapid advances in genome sequencing technology, the number of proteins with known structure and function has grown at a substantially lower rate (Freitas & de Carvalho, 2007).

Searching for similar sequences in protein databases is a common approach used in the prediction of a protein function. The objective of this search is to find a similar protein whose function is known and assigning its function to the new protein. Despite the simplicity and usefulness this method in a large number of situations, it has also some limitations (Freitas & de Carvalho, 2007). For instance, two proteins might have very similar sequences and perform different functions, or have very different sequences and perform a similar function. Additionally, the proteins being compared may be similar in regions of the sequence that are not determinants of their function.

Another approach that may be used alternatively or in complement to the similarity-based approach is to build a model for predictive classification. The goal of such a model is to classify data instances into one of a predefined set of classes or categories. In this approach a feature vector represents each protein, a learning algorithm captures the most important relationships between the features, and the classes present in the dataset. A major problem in protein datasets is the high dimensionality of the feature space. Most of these dimensions are not relative to protein function; even some noise data hurt the performance of the classifier. Hence, we need to select some representative features from the original feature space to reduce the dimensionality of feature space and improve the efficiency and performance of classifier.

Feature selection (FS) is of considerable importance in bioinformatics (Basiri, Ghasem-Aghaee, & Aghdam, 2008; Saeys, Inza, & Larranaga, 2007), signal processing (Nemati, Boostani, & Jazi, 2008), text categorization (Aghdam, Ghasem-Aghaee, & Basiri, 2008), data mining and pattern recognition (Jensen, 2005). Among too many methods that are proposed for FS, population-based optimization algorithms such as genetic algorithm (GA), ant colony

[*] Corresponding author. Tel.: +98 3117932671; fax: +98 3117932670.
E-mail address: basiri@eng.ui.ac.ir (M.E. Basiri).

0957-4174/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2009.04.023

optimization (ACO) and particle swarm optimization (PSO) have attracted a lot of attention (Basiri et al., 2008; Liu, Qin, Xu, & He, 2004; Punch, Goodman, Pei, Hovland, & Enbody, 1993). These methods are stochastic optimization techniques attempt to achieve better solutions by application of knowledge from previous iterations.

Genetic algorithms are optimization techniques based on the mechanism of natural selection. They used operations found in natural genetics to guide itself through the paths in the search space (Siedlecki & Sklansky, 1989). Because of their advantages, recently, GAs have been widely used as a tool for feature selection in data mining (Srinivas & Patnik, 1994).

ACO is a branch of newly developed form of artificial intelligence called Swarm Intelligence (SI). ACO algorithm is inspired by social behavior of ant colonies. Although they have no sight, ants are capable of finding the shortest route between a food source and their nest by chemical materials called pheromone that they leave when moving (Liu, Abbass, & McKay, 2004).

ACO algorithm was firstly used for solving Traveling Salesman Problem (TSP) (Dorigo, Maniezzo, & Colorni, 1996) and then has been successfully applied to a large number of difficult problems like the Quadratic Assignment Problem (QAP) (Maniezzo & Colorni, 1999), routing in telecommunication networks, graph coloring problems, scheduling, etc. This method is particularly attractive for feature selection, as there seems to be no heuristic that can guide search to the optimal minimal subset every time (Aghdam et al., 2008).

In our previous work (Basiri et al., 2008), we have proposed an ACO algorithm for feature selection in prediction postsynaptic activity of proteins. In this paper, we intend to hybridize ACO and genetic algorithms to obtain their excellent features by synthesizing them. More specifically, ACO offers a critical advantage of local searching, not found in GA. On the other hand, GA considers a global perspective by operating on the complete population from the very beginning. Therefore, ACO and GA can nullify each others drawbacks when hybridized.

The rest of this paper is organized as follows. Section 2 presents a brief overview of protein function prediction. Feature selection methods are shortly discussed in Section 3. Ant colony optimization is described in Section 4. Genetic algorithms are addressed in Section 5. Section 6 explains the proposed hybrid feature selection algorithm. Section 7 reports computational experiments. It also includes a brief discussion of the results obtained and finally the conclusion and future works are offered in the last section.

2. Protein function prediction

Proteins are the most essential and versatile macromolecules of life and serve as building blocks and functional components of a cell, and account for the second largest fraction of the cellular weight after water. They are polypeptides, formed within cells as a linear chain of amino acids (Setubal & Meidanis, 1999). Twenty different amino acids are available, which are denoted by 20 different letters of the alphabet and a linear sequence of these amino acids is known as the primary structure.

Some patterns may be common to multiple proteins. These common patterns and domains include helixes, sheets, various sites, which allow functions of a protein to be turned on and off, etc. From a data mining point of view these regions are very interesting since they work together to produce the function of proteins and so must produce patterns that can be analyzed. Some databases of these common patterns have been created, including the Prosite database (Hulo, 2006), which is used in this work. This database contains unique fingerprint style entries, which are designed to be used to identify the function of unknown proteins.

The concept of protein function is highly context sensitive and not very well defined. Rost, Liu, Nair, Wrzeszczynski, and Ofran (2003) in their survey defined protein function as follows: “function is everything that happens to or through a protein”. From a data mining point of view, protein function prediction can be regarded as a classification problem, namely to correctly classify a newly discovered protein into its functional class. Typically, protein datasets are organized as a hierarchy of classes. The classification of data in such a hierarchy poses some unique challenges to data miners such as the need for classification at different levels, which may require the use of different characteristics of the data.

In this paper two families of proteins are used to investigate performance of proposed approach involve G-protein-coupled receptor (GPCR) and enzyme protein families. G-protein-coupled receptors (GPCR) are proteins involved in signaling. They span cell walls so that they influence the chemistry inside the cell by sensing the chemistry outside the cell. More specifically, when a ligand (a substance that binds to a protein) is received by a GPCR, it causes the attached G-proteins to activate and detach. This mechanical biological switch causes the released G-Protein to affect other reactions within the cell. This kind of protein is particularly important for medical applications because it is believed that 40–50% of current medical drugs target GPCR activity (Freitas & de Carvalho, 2007).

Enzymes are a sub set of proteins; they are essential proteins responsible for the catalysis of metabolic reactions. They are used to speed up and make possible many of the chemical reactions that take part within the cell, without being altered themselves. Enzymes are assigned EC codes (enzyme commission numbers), which are four digit numbers that represent the type of chemical reaction the enzyme in question catalyzes (Shah & Hunter, 1998). Each digit corresponds to a level in the hierarchy. For instance, EC 1.2.3.4 is an enzyme with class value 1 in the first level, class value 2 in the second level, etc.

Techniques that predict protein function from sequence can be categorized into three classes, namely, sequence homology-based approaches, subsequence-based approaches and feature-based approaches (Pandey, Kumar, & Steinbach, 2006). The subsequence and feature-based approaches can be grouped into the category of model-based approaches since they are very similar at the fundamental level. All these approaches involve construction of a model for mapping a feature vector to a function and they follow the route shown in Fig. 1.

This paper mainly focuses on the feature selection step in construction of a model for protein function classification (shown in Fig. 1) and we will discuss in detail about that step in the next Section.

3. Feature selection approaches

During the last decade, application of feature selection (FS) techniques in bioinformatics has become a real prerequisite for model building. In particular, the high dimensional nature of many modeling tasks in bioinformatics, going from sequence analysis over microarray analysis to spectral analysis and literature mining has given rise to a wealth of feature selection techniques being presented in the field (Blum & Dorigo, 2004).

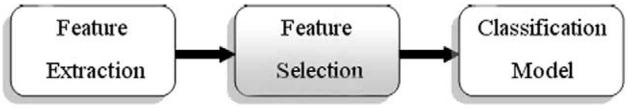

Fig. 1. Model-based approach for protein function prediction.

Feature selection is a problem of global combinatorial optimization in machine learning, which reduces the number of features, removes irrelevant, noisy and redundant data, and results in acceptable classification accuracy. The whole search space for optimization contains all possible subsets of features, meaning that its size is:

$$\sum_{s=0}^{D} \binom{D}{s} = \binom{D}{0} + \binom{D}{1} + \cdots + \binom{D}{n} = 2^D, \quad (1)$$

where D is the dimensionality (the number of features) and s is the size of the current feature subset (Mladenić et al., 2006). Usually FS algorithms involve heuristic or random search strategies in an attempt to avoid this prohibitive complexity. However, the degree of optimality of the final feature subset is often reduced (Jensen, 2005).

The objectives of feature selection are manifold, the most important ones being: improving models performance, providing faster and more cost-effective models, and to gain a deeper insight into the underlying processes that generated the data.

In the context of classification, FS techniques differ from each other in the way they interact with classifiers. Feature selection techniques can be organized into three categories, depending on their evaluation procedure: filter methods, wrapper methods and embedded methods (Duda & Hart, 1973). Filter techniques assess the relevance of features by looking only at the intrinsic properties of the data. These approaches mostly include selecting features based on inter-class separability criterion (Duda & Hart, 1973). If the evaluation procedure is tied to the task (e.g. classification) of the learning algorithm, the FS algorithm is a sort of wrapper approach. This method searches through the feature subset space using the estimated accuracy from an induction algorithm as a measure of subset suitability. If the feature selection and learning algorithm are interleaved then the FS algorithm is a kind of embedded approach (Mladenić et al., 2006). A common disadvantage of filter methods is that they ignore the interaction with the classifier. The wrapper method is computationally more involved, but takes the dependency of the learning algorithm on the feature subset into account (Jensen, 2005).

In the wrapper approach the evaluation function calculates the suitability of a feature subset produced by the generation procedure and it also compares that with the previous best candidate, replacing it if found to be better. A stopping criterion is tested in each of iterations to determine whether or not the FS process should continue.

Famous population-based FS approaches are based on the genetic algorithm (GA) (Siedlecki & Sklansky, 1989), simulated annealing (SA), particle swarm optimization (PSO) (Wang, Yang, Teng, Xia, & Jensen, 2007) and ant colony optimization (ACO) (Aghdam et al., 2008; Basiri et al., 2008; Nemati et al., 2008). For a review of application of feature selection techniques in bioinformatics see (Siedlecki & Sklansky, 1989).

4. Ant colony optimization

Ant colony optimization (ACO) algorithms were introduced by Marco Dorigo (Dorigo et al., 1992) in the early 1990s. While moving, ants leave a chemical pheromone trail on the ground. Ants are guided by pheromone smell. Ants tend to choose the paths marked by the strongest pheromone concentration. The indirect communication between the ants via pheromone trails enables them to find shortest paths between their nest and food sources (Dorigo, Bonaneau, & Theraulaz, 2000).

The basic idea of ACO is to model the problem to solve as the search for a minimum cost path in a graph, and to use artificial ants to search for good paths. The behavior of artificial ants is inspired from real ants; they lay pheromone on edges and/or vertices of the graph and they choose their path with respect to probabilities that depend on pheromone trails that have been previously laid by the colony; these pheromone trails progressively decrease by evaporation (Dorigo et al., 1996).

Artificial ants also have some extra features that do not find their counterpart in real ants. In particular, they are usually associated with data structures that contain the memory of their previous actions, and they may apply some daemon procedures, such as local search, to improve the quality of computed paths. In many cases, pheromone is updated only after having constructed a complete path and the amount of pheromone deposited is usually a function of the quality of the complete path. Finally, the probability for an artificial ant to choose a component often depends not only on pheromone, but also on problem-specific local heuristics (Dorigo et al., 1996).

4.1. Ant colony optimization for feature selection

As mentioned earlier given a feature set of size D, the FS problem is to find a minimal feature subset of size $s(s < D)$ while retaining a suitably high accuracy in representing the original features. Therefore, there is no concept of path. A partial solution does not define any ordering among the components of the solution, and the next component to be selected is not necessarily influenced by the last component added to the partial solution (Blum & Dorigo, 2004; Leguizamon & Michalewicz, 1999). Furthermore, solutions to an FS problem are not necessarily of the same size. To apply an ACO algorithm to solve a feature selection problem, these aspects need to be addressed. The first problem is addressed by redefining the way that the representation graph is used.

4.1.1. Graph representation

The feature selection problem may be reformulated into an ACO-suitable problem. The main idea of ACO is to model a problem as the search for a minimum cost path in a graph. Here nodes represent features, with the edges between them denoting the choice of the next feature. The search for the optimal feature subset is then an ant traversal through the graph where a minimum number of nodes are visited that satisfies the traversal stopping criterion. Fig. 2 illustrates this setup. Nodes are fully connected to allow any feature to be selected next. The ant is currently at node f_1 and has a choice of which feature to add next to its path (dotted lines). It chooses feature f_2 next based on the transition rule, then f_3 and then f_4. Upon arrival at f_4, the current subset $\{f_1,f_2,f_3,f_4\}$ is determined to satisfy the traversal-stopping criterion (e.g. suitably high classification accuracy has been achieved with this subset). The ant terminates its traversal and outputs this feature subset as a candidate for data reduction (Basiri et al., 2008).

Based on this reformulation of the graph representation, the transition rules and pheromone update rules of standard ACO algorithms can be applied. In this case, pheromone and heuristic value

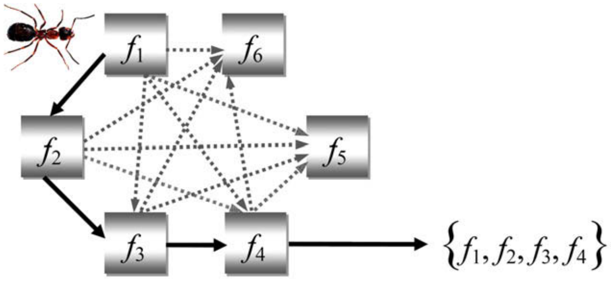

Fig. 2. ACO problem representation for FS.

are not associated with links. Instead, each feature has its own pheromone value and heuristic value.

4.1.2. Heuristic desirability

The basic ingredient of any ACO algorithm is a constructive heuristic for probabilistically constructing solutions. A constructive heuristic assembles solutions as sequences of elements from the finite set of solution components. A solution construction starts with an empty partial solution. Then, at each construction step, the current partial solution is extended by adding a feasible solution component from the set of solution components (Dorigo & Blum, 2005). A suitable heuristic desirability of traversing between features could be any subset evaluation function for example, an entropy-based measure or rough set dependency measure (Jensen, 2005). In proposed algorithm, classifier performance is mentioned as heuristic information for feature selection. The heuristic desirability of traversal and node pheromone levels are combined to form the so-called *probabilistic transition rule*, denoting the probability that ant k will include feature i in its solution at time step t:

$$P_i^k(t) = \begin{cases} \frac{[\tau_i(t)]^\alpha . [\eta_i]^\beta}{\sum_{u \in J^k} [\tau_u(t)]^\alpha . [\eta_u]^\beta} & \text{if } i \in J^k, \\ 0 & \text{otherwise}, \end{cases} \tag{2}$$

where J^k is the set of feasible features that can be added to the partial solution; τ_i and η_i are, respectively, the pheromone value and heuristic desirability associated with feature i. α and β are two parameters that determine the relative importance of the pheromone value and heuristic information.

The transition probability used by ACO is a balance between pheromone intensity (i.e. history of previous successful moves), τ_i, and heuristic information (expressing desirability of the move), η_i. This effectively balances the exploitation–exploration trade-off. The best balance between exploitation and exploration is achieved through proper selection of the parameters α and β. If $\alpha = 0$, no

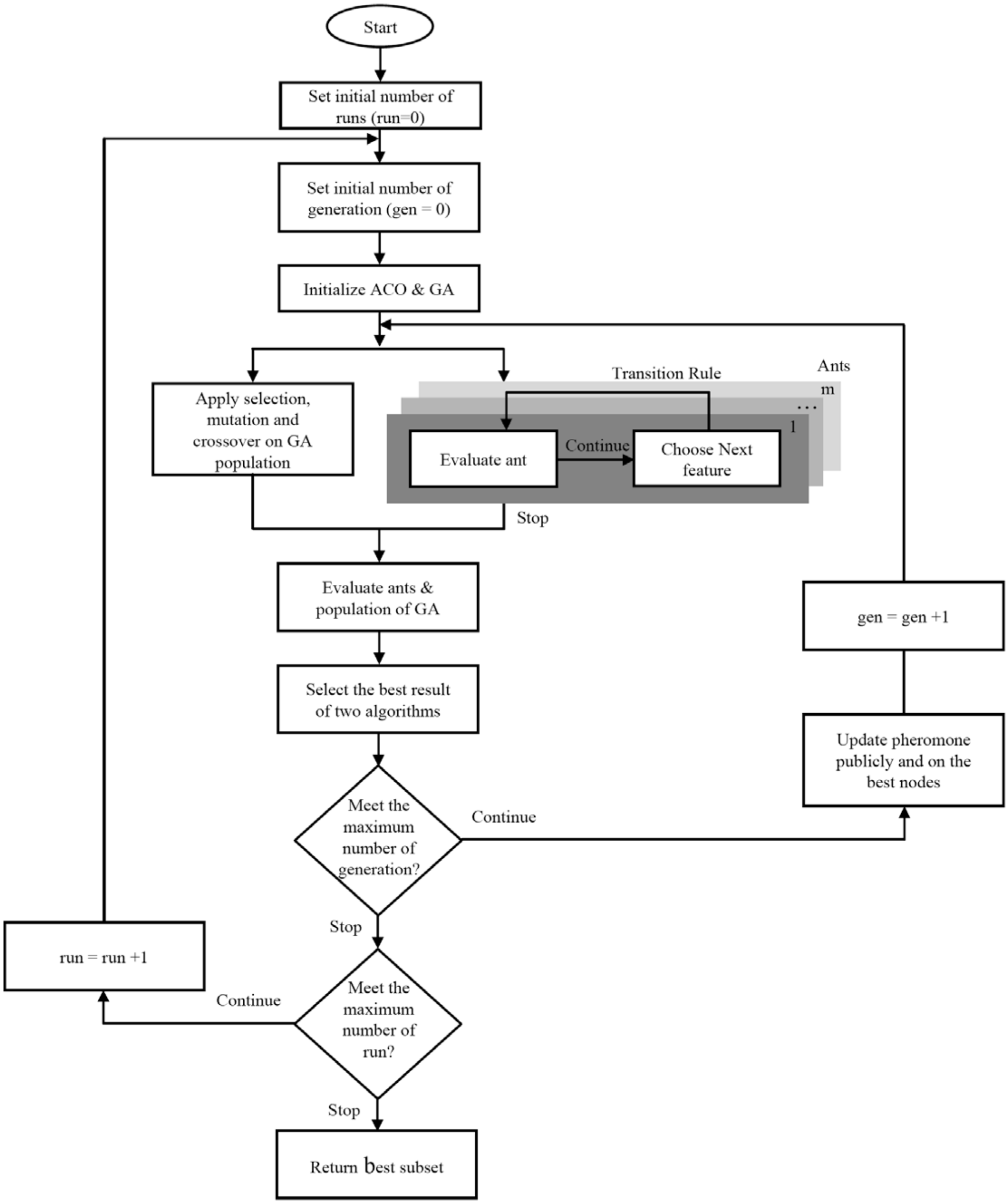
Start
Set initial number of runs (run=0)
Set initial number of generation (gen = 0)
Initialize ACO & GA
Apply selection, mutation and crossover on GA population
Transition Rule
Ants
m
...
1
Evaluate ant
Continue
Choose Next feature
Stop
Evaluate ants & population of GA
Select the best result of two algorithms
Meet the maximum number of generation?
Continue
Update pheromone publicly and on the best nodes
gen = gen +1
Stop
run = run +1
Continue
Meet the maximum number of run?
Stop
Return best subset

Fig. 3. Proposed ACO-GA feature selection algorithm.

pheromone information is used, i.e. previous search experience is neglected. The search then degrades to a stochastic greedy search. If $\beta = 0$, the attractiveness (or potential benefit) of moves is neglected.

4.1.3. Pheromone update rule

After all ants have completed their solutions, pheromone evaporation on all nodes is triggered, and then according to Eq. (3) each ant k deposits a quantity of pheromone, $\Delta\tau_i^k(t)$, on each node that it has used

$$\Delta\tau_i^k(t) = \begin{cases} \phi \cdot \gamma(S^k(t)) + \frac{\varphi\cdot(n-|S^k(t)|)}{n} & \text{if } i \in S^k(t), \\ 0 & \text{otherwise}, \end{cases} \tag{3}$$

where $S^k(t)$ is the feature subset found by ant k at iteration t, and $|S^k(t)|$ is its length. The pheromone is updated according to both the measure of the classifier performance, $\gamma(S^k(t))$, and feature subset length. $\phi \in [0,1]$ and $\varphi = 1 - \phi$ are two parameters that control the relative weight of classifier performance and feature subset length. This formula means that the classifier performance and feature subset length have different significance for feature selection task. In our experiment we assume that classifier performance is more important than subset length, so they were set as $\phi = 0.8$, $\varphi = 0.2$.

In practice, the addition of new pheromone by ants and pheromone evaporation are implemented by the following rule applied to all the nodes:

$$\tau_i(t+1) = (1-\rho)\tau_i(t) + \sum_{k=1}^{m} \Delta\tau_i^k(t) + \Delta\tau_i^g(t), \tag{4}$$

where m is the number of ants at each iteration and $\rho \in (0,1)$ is the pheromone trail decay coefficient. The main role of pheromone evaporation is to avoid stagnation, that is, the situation in which all ants constructing the same solution. g indicates the best ant at each iteration. All ants can update the pheromone according to Eq. (4) and the best ant deposits additional pheromone on nodes of the best solution. This leads to the exploration of ants around the optimal solution in next iterations.

5. Genetic algorithm (GA)

Genetic algorithms belong to a class of population-based stochastic search algorithms that are inspired from principles of natural evolution known as evolutionary algorithms (EA) (Choenauer & Michalewicz, 1997). GA is based on the principle of "survival of fittest", as in the natural phenomena of genetic inheritance and Darwinian strife for survival. These algorithms are general-purpose optimization algorithms with a probabilistic component that provide a means to search poorly understood, irregular spaces.

Primarily, GA was designed to optimally solve sequential decision processes more than to perform function optimization but over the years, it has been used widely in both learning and optimization problems (Sheta & Turabieh, 2006).

Instead of working with a single point, GAs work with a population of points. Each point is a vector in hyperspace representing one potential (or candidate) solution to the optimization problem. A population is, thus, just an ensemble or set of hyperspace vectors. Each vector is called a chromosome in the population. The number of elements in each chromosome depends on the number of parameters in the optimization problem and the way to represent the problem.

The operators of the GA are selection, crossover and mutation. Selection is a process in which p individuals are selected with a probability proportional to their fitness to be parents. Crossover is a process in which two different parents are iteratively selected from the set of p parents to swap information between them to generate two new individuals (offspring). This is done by choosing randomly a point of break for the parents and swapping parts between them. Then mutation is applied to every offspring. Mutation is the alteration of the bits of an individual with a small predefined probability, sometimes known as mutation coefficient (mc). These new altered individuals compose the new population P.

5.1. Genetic algorithm for feature selection

Several approaches exist to use GAs for feature subset selection. The two main methods that have been widely used in the past are as follow. First is due to (Siedlecki & Sklansky, 1989), of finding an optimal binary vector in which each bit corresponds to a feature (Binary Vector Optimization (BVO) method). A '1' or '0' suggests that the feature is selected or dropped, respectively. The aim is to find the binary vector with the smallest number of 1's such that the classifier performance is maximized. This criterion is often modified to reduce the dimensionality of the feature vector at the same time (Yang & Honavar, 1998). The second and more refined technique uses an m-ary vector to assign weights to features instead of abruptly dropping or including them as in the binary case (Punch et al., 1993). This gives a better search resolution in the multidimensional space (Raymer, Punch, Goodman, Kuhn, & Jain, 2000).

6. Proposed ACO–GA algorithm

As mentioned earlier, in this paper we intend to hybridize ACO and GA in such a manner that they complement each other for feature selection in protein function prediction. The main steps of proposed feature selection algorithm are shown in Fig. 3.

ACO and GA are used to explore the space of all subsets of given feature set. The performance of selected feature subsets is measured by invoking an evaluation function with the corresponding reduced feature space and measuring the specified classification result. The best feature subset found is then output as the recommended set of features to be used in the actual design of the classification system.

The process begins by generating a number of ants and a population in GA. ACO and GA generate feature subset in parallel and the resulting subsets are gathered and then evaluated at the end of iterations. The best subset is selected according to evaluation measures. If an optimal subset has been found or the algorithm has executed a certain number of runs, then the process halts and outputs the best feature subset encountered. If none of these conditions hold, then all ants can update the pheromone according to Eq. (4), the best ant deposits additional pheromone on nodes of the best solution. The best solution may be generated by either ACO or GA. Therefore, ACO can utilize the GA's cross-over and mutation operations. This leads to the exploration of ants around the optimal solution in next iterations. After updating pheromone, the process iterates once more.

7. Experimental results

A series of experiments was conducted to show the utility of proposed feature selection algorithm. All experiments have been run on a machine with 3.0 GHz CPU and 512 MB of RAM. We implement proposed ACO–GA, ACO-based and GA-based feature selection algorithms in Matlab R2006a. The operating system was Windows XP Professional. For experimental studies, we have considered two datasets; GPCR-PROSITE and ENZYME-PROSITE. The following sections describe these two datasets and implementation results.

7.1. Datasets

The datasets used in this paper employ signatures (describing sequence similarity) generated directly from protein sequences to attempt to predict a given proteins function. We use two datasets in this paper involve GPCR-PROSITE dataset previously mined in (Correa, Freitas, & Johnson, 2007; Holden & Freitas, 2006) and ENZYME-PROSITE dataset.

The GPCR-PROSITE dataset contains 190 proteins. The proteins are represented by a set of 127 Prosite patterns (Hulo, 2006). Prosite is a database of protein families and domains. It is based on the observation that, while there are a huge number of different proteins, most of them can be grouped, based on similarities in their sequences, into a limited number of families (a protein consists of a sequence of amino acids). Prosite patterns are small regions within a protein that present a high sequence similarity when compared to other proteins. The proteins in this data set are grouped into families and subfamilies in a hierarchical fashion. There are three levels of hierarchy. The first level has eight classes (families), the second and third levels have 32 classes (subfamilies) each one (some proteins are classified only up to the second hierarchical level and have no class at the third level). The objective of our algorithms is to classify each protein into its most suitable family in each level (Correa et al., 2007).

The classes to be predicted in the second dataset are four digit EC numbers (enzyme commission number), and the predictor features are Prosite patterns. This dataset was extracted from the UniProt and Prosite databases and it contains 22,500 records. In both datasets the absence of a given Prosite pattern is indicated by a value of ‘0’ for the feature corresponding to that Prosite pattern and its presence is indicated by a value of ‘1’. As a pre-processing step, classes that contain less than 10 records were merged with their most similar sibling. The similarity between two classes was measured simply as the average number of matching attribute values between all records in either class (Holden & Freitas, 2006). The total number of classes after this process was 750, with six classes at the first level, 44 at the second, 106 at the third and 594 at the fourth.

7.2. Experimental methodology

There are several strategies available for predicting hierarchical classes; the first method is to flatten the dataset to one single level, then use one of the standard classification algorithms to predict the class. The second approach is to divide a hierarchical problem into a set of flat classification problems. The third method is to use the divide-and-conquer principle (Sun & Lim, 2001) finally, the forth approach is to employ the big-bang approach in which only a single classifier is used in the classification process.

In this work, we follow the third approach. In this approach, Top-Down approach, one or more classifiers are trained for each level of the hierarchy. This produces a tree of classifiers. The root classifier is trained with all training instances. Then, at the next class level, a classifier is training with just the subset of instances belonging to the classes predicted by the classifier (Freitas & de Carvalho, 2007). In the test phase, beginning at the root node, an instance is classified in a Top-Down manner. When assigned to one class, the instance is then submitted to a new classifier in order to predict to which subclasses of this class it belongs. This procedure is repeated until a leaf-node class is reached or until no additional prediction can be made from an internal node, such that the reliability is not affected. For a comprehensive review of hierarchical classification approaches see (Freitas & de Carvalho, 2007).

The computational experiments involved a 10-fold stratified cross-validation method (Witten & Frank, 2005). First, the 190 records in the GPCR-PROSITE data set and 22500 records in the second dataset were divided into 10 equally sized folds. Each entire 10-fold cross validation test was repeated 30 times.

In each of the 10 iterations of the cross-validation procedure, the predictive accuracy of the classification is compared between four different algorithms, as follows. (1) By baseline algorithm, using all original features. (2) By GA-based algorithm, where GA is used for feature selection. (3) By ACO algorithm, where only the features selected by ACO are used for classification. (4) By ACO–GA algorithm where proposed hybrid algorithm is applied for feature selection. All these approaches use the divide and conquer principle, Top-Down approach, as discussed earlier.

Various values were tested for the parameters of ACO and GA. The experiments show that the highest performance is achieved by setting the control parameters to values shown in Table 1.

Parameter values were empirically determined in our preliminary experiments for leading to better convergence; but we make no claim that these are optimal values. Parameter optimization is a topic for future research. To show the effectiveness of proposed algorithm, we use a simple classifier (nearest neighbor classifier) in that. The use of this simple classifier in our algorithm can affect the classification performance and taking the advantage of using classifiers that are more complex in that is another research direction. On our experiments, we use a measurement for the accuracy rate of a classification model which has also been used before in (Basiri et al., 2008; Correa et al., 2007). This measurement is given by the Eq. (5).

$$\text{Predictive accuracy rate} = TPR \times TNR, \tag{5}$$

where

$$TPR = \frac{TP}{TP + FN}, \quad TNR = \frac{TN}{TN + FP}, \tag{6}$$

TP, TN, FP and FN are the numbers of true positives, true negatives, false positives and false negatives, respectively (Basiri et al., 2008).

7.3. Results

The classification quality and feature subset length are two criteria that are considered to assess the performance of algorithms. Comparing the first criterion, predictive accuracy, we noted that ACO, GA, and proposed ACO–GA algorithms did better (in all class levels) than the baseline algorithm using all features. Furthermore, ACO did slightly better than GA and proposed ACO–GA outperforms both ACO and GA algorithms in term of predictive accuracy. Tables 2 and 3 compare the predictive accuracy results of four algorithms.

Nevertheless, the difference in the accuracy between these algorithms is, in some cases, not statistically significant. Table 4

Table 1
ACO and GA parameter settings.

	Population	Iteration	Crossover probability	Mutation probability	Initial pheromone	α	β	ρ
GA	50	100	0.7	0.005	–	–	–	–
ACO	50	100	–	–	1	1	0.1	0.2

Table 2
Results for predictive accuracy in the GPCR-PROSITE dataset.

Level number	Baseline	GA	ACO	ACO–GA
1	70.36 ± 2.96	79.28 ± 2.68	80.12 ± 2.98	82.26 ± 2.12
2	32.27 ± 6.56	41.27 ± 5.65	46.78 ± 5.46	50.12 ± 4.86
3	20.53 ± 2.34	24.85 ± 3.24	28.26 ± 2.88	30.12 ± 3.14

Table 3
Results for predictive accuracy in the ENZYME-PROSITE dataset.

Level number	Baseline	GA	ACO	ACO–GA
1	82.25 ± 4.26	88.28 ± 2.68	96.85 ± 2.66	98.26 ± 2.12
2	70.68 ± 7.36	80.26 ± 7.62	85.88 ± 8.68	87.62 ± 6.86
3	48.62 ± 6.86	61.85 ± 4.24	65.04 ± 4.34	67.12 ± 4.14

Table 4
Paired two-tailed *t*-test for the predictive accuracy with significance level 0.05.

Level number	GPCR	ENZYME
1	$t(9) = 1.568$, $p = 0.15$	$t(9) = 2.068$, $p = 0.065$
2	$t(9) = 6.366$, $p = 4.4\text{E}{-5}$	$t(9) = 2.179$, $p = 0.051$
3	$t(9) = 7.664$, $p = 4.8\text{E}{-6}$	$t(9) = 2.168$, $p = 0.061$

shows the results of a paired two-tailed *t*-test for the accuracy of the proposed ACO–GA algorithm versus the accuracy of the ACO (at a significance level of 0.05).

According to Table 4, difference in predictive accuracy between the algorithms in GPCR-PROSITE at only the first level is not statistically significant while, this difference in ENZYME-PROSITE dataset is not statistically significant at all class levels.

Second, we compare the other criterion, number of selected features. As could be seen in Tables 5 and 6, ACO–GA outperforms the ACO and GA in selecting smaller subset of features in all class levels in both datasets. Therefore, the second comparison criterion is the discriminating factor between the performances of these algorithms. Table 7 shows that, this difference in number of selected features is statistically significant in all levels.

Table 5
Results for number of selected features in the GPCR-PROSITE dataset.

Level number	GA	ACO	ACO–GA
1	6.8 ± 1.68	5.8 ± 1.43	4.6 ± 1.12
2	13.7 ± 1.65	10.9 ± 1.87	7.1 ± 1.66
3	16.5 ± 2.24	14.4 ± 2.40	12.2 ± 2.14

Table 6
Results for number of selected features in the ENZYME-PROSITE dataset.

Level number	GA	ACO	ACO–GA
1	12.2 ± 2.08	8 ± 1.73	6.6 ± 1.12
2	44.3 ± 3.15	30.8 ± 3.94	28.1 ± 3.16
3	101.2 ± 5.46	90.8 ± 5.44	85.4 ± 5.04

Table 7
Paired two-tailed *t*-test for the number of selected features with significance level 0.05.

Level number	GPCR	ENZYME
1	$t(9) = 5.266$, $p = 3.2\text{E}{-4}$	$t(9) = 5.448$, $p = 3.6\text{E}{-4}$
2	$t(9) = 7.141$, $p = 4.8\text{E}{-5}$	$t(9) = 10.118$, $p = 3.2\text{E}{-6}$
3	$t(9) = 9.244$, $p = 2.2\text{E}{-6}$	$t(9) = 12.822$, $p = 1.2\text{E}{-7}$

The high performance obtained by ACO, GA and ACO–GA algorithms in the higher levels of datasets, showed in Tables 2 and 3, occurred because of two reasons. First, the number of classes per level increases at deeper levels, with a corresponding decrease in the number of examples per class, making an accurate prediction at deeper levels more unlikely. Second, it is an inevitable result of using a divide and conquer type algorithm, as once an incorrect prediction has been made at a higher level it cannot be rectified, this leads to the accuracy being at best the same as the level above.

Figs. 4 and 5 show the predictive accuracy for each of the feature selection algorithms as we change the number of selected features for the last level of datasets. The results show that as the percentage of selected features exceeds 10%, the ACO–GA algorithm outperforms genetic algorithm and ACO.

7.4. Discussion

Experimental results show that the use of unnecessary features hurt classification accuracy and FS is used to reduce redundancy in the information provided by the selected features. Using only a small subset of selected features, the ACO–GA, the GA and the

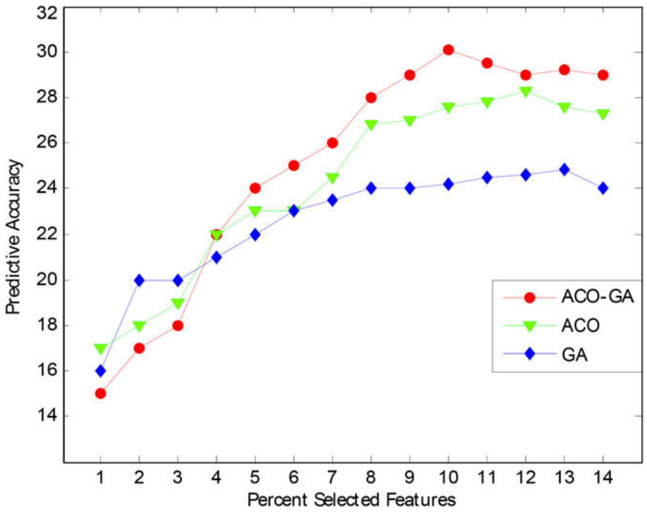

Fig. 4. Comparison of ACO–GA, ACO and GA algorithms in GPCR-PROSITE dataset.

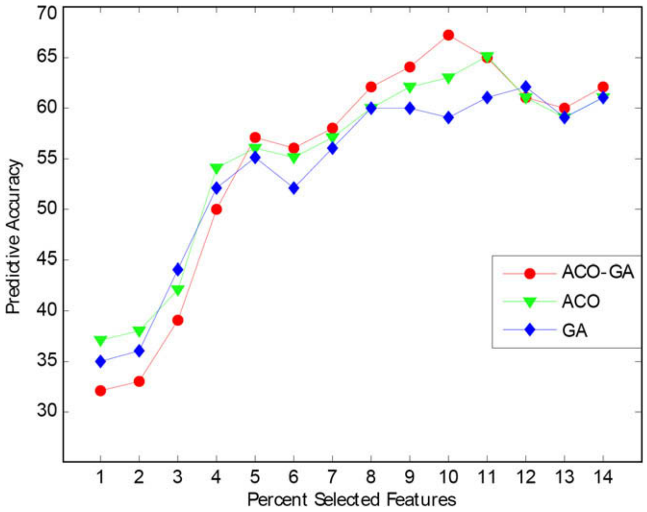

Fig. 5. Comparison of ACO–GA, ACO and GA algorithms in ENZYME-PROSITE dataset.

ACO algorithms obtained better classification accuracy than the baseline algorithm using all features. Previous work had already shown that application of FS in biological datasets can improve the performance of classification process (Basiri et al., 2008).

The strength of GAs is in the parallel nature of their search. GAs implement a powerful form of hill climbing that preserves multiple solutions, eradicates unpromising solutions, and provides reasonable solutions. ACO shares many similarities with evolutionary computation (EC) techniques in general and GAs in particular. These techniques begin with a group of a randomly generated population and utilize a fitness value to evaluate the population. They all update the population and search for the optimum with random techniques.

Both ACO and GA are stochastic population-based search approaches that depend on information sharing among their population members to enhance their search processes using a combination of deterministic and probabilistic rules. They are efficient, adaptive and robust search processes, producing near optimal solutions, and have a large degree of implicit parallelism. The main difference between the ACO compared to GA, is that ACO does not have genetic operators such as crossover and mutation. Ants update themselves with the pheromone update rule; they also have a memory that is important to the algorithm.

Compared to GAs, the ACO has a much more intelligent background and can be implemented more easily. The computation time used in ACO is less than in GAs. The parameters used in ACO are also fewer. However, if the proper parameter values are set, the results can easily be optimized. The decision on the parameters of the ant colony affects the exploration–exploitation tradeoff and is highly dependent on the form of the objective function. Successful feature selection was obtained even using conservative values for the ACO basic parameters.

In this paper, we hybridized the two approaches (ACO and GA) in such a manner that they complement each other for classification of protein functions. More specifically, ACO offers a critical advantage of local searching, not found in GA, i.e. searching for local optimality which can optimize the global or the final solution. On the other hand, GA takes a global perspective into account by operating on the complete population from the very beginning. Thus, by hybridizing these approaches, they can nullify each others drawbacks. Also, quick convergence provided by the ACO component can be advantageous for time constrained problems.

8. Conclusion and future research

In this paper, we present a hybrid ACO–GA feature selection algorithm and adapt it for hierarchical classification of biological data in a Top-Down manner. This algorithm, ACO–GA, was compared with an ordinary ACO-based algorithm and a classical genetic algorithm for hierarchical classification of proteins. Proposed algorithm has the ability to converge quickly; it has a strong search capability in the problem space and can efficiently find minimal feature subset. Experimental results demonstrate competitive performance.

In order to evaluate the performance of these approaches, experiments were performed using two bioinformatics datasets, which are related with G-Protein-Coupled Receptor (GPCR) and Enzyme protein families and the predictor features were Prosite patterns. According to the experimental results, the use of unnecessary features decreases classification accuracy and FS is used to reduce redundancy in the information provided by the selected features. Furthermore, results of experiments indicate that proposed feature selection algorithm outperforms both ACO and GA algorithms in GPCR-PROSITE and ENZYME-PROSITE datasets.

More experimentation and further investigation into this technique may be required. The pheromone trail decay coefficient (ρ) and pheromone amount ($\Delta\tau_i^k(t)$) in the ACO component have an important impact on the performance of ACO–GA. The selection of the parameters may be problem-dependent. The deposited pheromone, $\Delta\tau_i^k(t)$, calculated using Eq. (3), expresses the quality of the corresponding solution. ρ simulates the pheromone evaporation. Evaporation becomes more important for more complex problems. If $\rho = 0$, i.e. no evaporation, the algorithm does not converge. If pheromone evaporates too much (a large ρ is used), the algorithm often converged to sub-optimal solutions for complex problem. In many practical problems, it's difficult to select the best ρ without trial-and-error. α and β are also key factors in ACO for feature selection.

For future research, we will test proposed algorithms on other kinds of biological data. Other hierarchical classification algorithms will also be investigated. To show the effectiveness of proposed algorithm, we use a simple classifier (nearest neighbor classifier) in that which can affect the classification performance. For future work, the authors intend to investigate the performance of proposed feature selection algorithm by taking advantage of using more complex classifiers in that. Finally, the authors plan to combine hierarchical classification with multi-label classification.

References

Aghdam, M. H., Ghasem-Aghaee, N., & Basiri, M. E. (2008). Application of ant colony optimization for feature selection in text categorization. In *Proc. CEC 2008, Proceeding of the fifth IEEE congress on evolutionary computation*, IEEE Press, Hong Kong.

Alberts, B., Bruce, A., Johnson, A., Lewis, J., Raff, M., & Roberts, K. (2002). *The molecular biology of the cell* (4th ed.). Garland Press.

Basiri, M. E., Ghasem-Aghaee, N., & Aghdam, M. H. (2008). *Using ant colony optimization-based selected features for predicting post-synaptic activity in proteins, EvoBIO 2008* (pp. 12–23), LNCS 4973, Berlin, Heidelberg, Italy: Springer-Verlag.

Blum, C., & Dorigo, M. (2004). The hyper-cube framework for ant colony optimization. *IEEE Transaction on Systems, Man, and Cybernetics – Part B, 34*(2), 1161–1172.

Choenauer, M., & Michalewicz, Z. (1997). Evolutionary computation control and cybernetics. *Proceedings of the IEEE, 26*(3), 307–338.

Correa, S., Freitas, A. A., & Johnson, C. G. (2007). Particle Swarm and Bayesian networks applied to attribute selection for protein functional classification. In *Proc. of the GECCO-2007 workshop on particle swarms: The second decade* (pp. 2651–2658).

Dorigo, M. (1992). *Optimization, learning and natural algorithms*. PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: Optimization by a colony of cooperating agents. *IEEE Transaction on Systems, Man, and Cybernetics – Part B, 26*(1), 29–41.

Dorigo, M., Bonaneau, E., & Theraulaz, G. (2000). Ant algorithms and stigmergy. *Future Generation Computer Systems, 16*, 851–871.

Dorigo, M., & Blum, C. (2005). Ant colony optimization theory: A survey. *Theoretical Computer Science*, 243–278.

Duda, R. O., & Hart, P. E. (1973). *Pattern classification and scene analysis*. John Wiley & Sons.

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 Applications, 99*(7), 175–208.

Holden, N., & Freitas, A. A. (2006). Hierarchical classification of *g*-protein-coupled receptors with a pso/aco algorithm. In *Proc. IEEE swarm intelligence symposium (SIS-06)* (pp. 77–84).

Hulo, N., et al. (2006). *The PROSITE database. Nucleic acids research* (Vol. 34) (pp. D227–D230). Database issue.

Jensen, R. (2005). *Combining rough and fuzzy sets for feature selection*. PhD thesis, University of Edinburgh.

Leguizamon, G., & Michalewicz, Z. (1999). A new version of ant system for subset problems. In *Proceedings of IEEE congress on evolutionary computation* (pp. 1458–1465).

Liu, Y., Qin, Z., Xu, Z., & He, H. (2004). *Feature selection with particle swarms* (pp. 425–430). CIS 2004, LNCS 3314, Berlin, Heidelberg: Springer-Verlag.

Liu, B., Abbass, H. A., & McKay, B. (2004). Classification rule discovery with ant colony optimization. *IEEE Computational Intelligence Bulletin, 3*(1), 31–35.

Maniezzo, V., & Colorni, A. (1999). The ant system applied to the quadratic assignment problem. *IEEE Transaction on Knowledge and Data Engineering, 11*(5), 769–778.

Mladenić, D. (2006). *Feature selection for dimensionality reduction. subspace, latent structure and feature selection, statistical and optimization, perspectives workshop,* *SLSFS 2005* (pp. 84–102), Bohinj, Slovenia, Lecture notes in computer science 3940, Springer.

Nemati, S., Boostani, R., & Jazi, M. D. (2008). *A novel text-independent speaker verification system using ant colony optimization algorithm* (pp. 421–429). ICISP2008, LNCS 5099, Berlin, Heidelberg, France: Springer-Verlag.

Pandey, G., Kumar, V., & Steinbach, M. (2006). *Computational approaches for protein function prediction: A survey*. Technical report, University of Minnesota.

Punch, W. F., Goodman, E. D., Pei, L. C. S. M., Hovland, P., & Enbody, R. (1993). Further research on feature selection and classification using genetic algorithms. In *Proceedings international conference on genetic algorithms* (pp. 557–564).

Raymer, M., Punch, W., Goodman, E., Kuhn, L., & Jain, A. K. (2000). Dimensionality reduction using genetic algorithms. *IEEE Transactions on Evolutionary Computing, 4*, 164–171.

Rost, B., Liu, J., Nair, R., Wrzeszczynski, K. O., & Ofran, Y. (2003). Automatic prediction of protein function. *Cellular and Molecular Life Sciences, 60*(12), 2637–2650.

Saeys, Y., Inza, I., & Larranaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics, 23*(19), 2507–2517.

Setubal, J., & Meidanis, J. (1999). *Introduction to computational molecular biology*. Boston, MA: Thomson.

Shah, I., & Hunter, L. (1998). Visualization based on the enzyme commission nomenclature. *Pacific Symposium on Biocomputing, 3*(2), 142–152.

Sheta, A., & Turabieh, H. (2006). A comparison between genetic algorithms and sequential quadratic programming in solving constrained optimization problems. *ICGST International Journal on Artificial Intelligence and Machine Learning (AIML), 6*(1), 67–74.

Siedlecki, W., & Sklansky, J. (1989). A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters, 10*(5), 335–347.

Srinivas, M., & Patnik, L. M. (1994). *Genetic algorithms: A survey*. Los lamitos: IEEE Computer Society Press.

Sun, A., & Lim, E. (2001). Hierarchical text classification and evaluation. *Proc. IEEE ICDM*, 521–528.

UniProt. <http://www.ebi.uniprot.org/>, visited on April 2008.

Wang, X., Yang, J., Teng, X., Xia, W., & Jensen, R. (2007). Feature selection based on rough sets and particle swarm optimization. *Pattern Recognition Letters, 28*(4), 459–471.

Witten, I. H., & Frank, E. (2005). *Data mining: Practical machine learning tools and techniques*. San Francisco: Morgan Kaufmann.

Yang, J., & Honavar, V. (1998). Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems, 13*, 44–49.