ISTITUTO DI ANALISI DEI SISTEMI ED INFORMATICA
"Antonio Ruberti"
CONSIGLIO NAZIONALE DELLE RICERCHE

P. Bertolazzi, G. Felici, G. Lancia

APPLICATION OF FEATURE SELECTION AND CLASSIFICATION TO COMPUTATIONAL MOLECULAR BIOLOGY

R. 08-02, 2008

Paola Bertolazzi – Istituto di Analisi dei Sistemi ed Informatica del CNR, viale Manzoni 30 - 00185 Roma, Italy. Email: `bertola@iasi.cnr.it`.

Giovanni Felici – Istituto di Analisi dei Sistemi ed Informatica del CNR, viale Manzoni 30 - 00185 Roma, Italy. Email: `felici@iasi.cnr.it`.

Giuseppe Lancia – Dipartimento di Informatica e Matematica, Università di Udine, Udine, Italy. EMAIL: `lancia@dei.unipd.it`.

ISSN: 1128–3378

Collana dei Rapporti dell'Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti", CNR

viale Manzoni 30, 00185 ROMA, Italy

tel. ++39-06-77161
fax ++39-06-7716461
email: iasi@iasi.cnr.it
URL: http://www.iasi.cnr.it

Abstract

One of the main challenges of computational molecular biology lies in the fact that most experiments produce data sets of considerable size. This is true, for instance, of DNA sequencing, microarray hybridization, and SNP identification and mapping. The large data sets must be analyzed and interpreted to extract all relevant information they can provide, thus separating it from background noise and extra information of little practical use. *Feature Selection* and *Classification* techniques are the main tools to pursue this task. *Feature selection* techniques are meant at identifying a small subset of important data within a large data set. *Classification* techniques are designed to identify, within the analyzed data, synthetic models that are able to explain some of the relevant characteristics contained therein. Classification techniques are often considered *learning* methods, in the sense that they are used to learn new knowledge from data.

In this chapter, we survey several feature selection and classification methods and their use in
(i) microarray analysis, (ii) haplotype analysis and tag SNP selection, (iii) string barcoding.

1. Introduction

Modern technology, where sophisticated instruments are coupled with the massive use of computers, has made molecular biology a science where the size of the data to be gathered and analyzed poses serious computational problems. Very large data sets are ubiquitous in computational molecular biology: The European Molecular Biology Laboratory Nucleotide Sequence Database (EMBL) has almost doubled its size every year in the past ten years, and, currently, the archive comprises over 1.7 billion records covering almost 1.7 trillion base pairs of sequences. Similarly, the Protein Data Bank (PDB) has seen an exponential growth, with the number of protein structures deposited (each of which is a large data set by itself) currently at over 50,000. An assembly task can require to reconstruct a large genomic sequence starting from hundreds of thousands of short (100 to 1000 bp) DNA fragments. Microarray experiments produce information about the expression of hundreds of thousands of genes in hundreds of individuals at once (data set in the order of Gigabytes), and the list goes on and on.

This abundance of large bodies of biological data calls for effective methods and tools for their analysis. The data, both structured or semi-structured, have in many cases the form of two dimensional arrays, where the rows correspond to *individuals* and the columns are associated with some *features*. While in other fields (see for instance medical data) a data set contains a large number of individuals and a small set of features in the field of molecular biology this situation is reversed, and the number of individuals is small while the number of features is very large. This is mainly due to the cost of the experiments (for instance, the DNA sequencing procedure or the phasing of genotypes require a lot of time and very complex computational procedures) and to the complexity of the molecules.

These large data sets must be analyzed and interpreted to extract all relevant information they can provide, thus separating it from extra information of little practical use. *Feature Selection* and *Classification* techniques are the main tools to pursue this task. *Feature selection* techniques are meant at identifying a small subset of important data within a large data set. *Classification* techniques are designed to identify, within the analyzed data, synthetic models that are able to explain some of the relevant characteristics contained therein. The two techniques are indeed strongly related: the selection of few relevant features among the many ones available can be considered - *per se* - already a simple model of the data, and thus an application of learning; on the other hand, feature selection is always used on large bodies of data to identify those features on which to apply a classification method to identify meaningful models.

In this chapter we consider three different problems arising in computational molecular biology: *Classification*, *Representation* and *Reconstruction*.

Classification. When the experiments are divided into two or more classes associated with some characteristics of the individuals, it is of interest to identify those rules that put the values of the features of an individual in relation with its class; such rules are then used to shed light on the studied characteristic (e.g., if the individuals are divided into *healthy* and *ill* classes, one may want to know what genes are over-expressed in the ill individuals while being *under-expressed* in the healthy ones). Besides, when the rules exhibit a sufficient precision, they can be used to classify new individuals, e.g., to *predict* the class of an individual of unknown class. This is the case of *classification* or *supervised learning* problems. Here the relevance of the features to be selected is related with their ability to discriminate between individuals that belong to different classes; in addition, a classification method may be needed to identify with more precision the classification rules on the selected features.

Representation. When a set of individuals has to be analyzed according to the values of its features to discover relevant phenomena, a crucial step is to restrict the analysis to a small and treatable number of relevant features. In this case the final scope of the analysis is not known in advance; the relevance of the features to be selected is then related to the way they *represent* the overall information gathered in the experiments; such type of problems are usually referred to as *unsupervised learning*.

Reconstruction. This class of problems arise when the objective is to find a possibly small subset of the available features from which it is possible to derive the values of all the remaining features with a good degree of precision. Here the feature selection method has to be combined with a *reconstruction* algorithm, i.e., some rules that link the values of a non-selected feature with the values of one or more selected features.

The chapter is organized as follows. We give a general overview on Feature Selection (FS) methods both for unsupervised and supervised learning problems in Section 2. Then we review the main Classification methods that are used in biological applications (Section 3). We then describe in greater detail two biological data analysis problems, where FS and classification methods have been used with success: in Section 4 we consider *Microarray analysis*, where the profiles (expression levels) of mRNA (transcriptomics), proteins (proteomics) or metabolites (metabolomics) in a given cell population are studied to identify those genes (or proteins or metabolites) that appear to be correlated with the studied phenomena; Section 5 is dedicated to *Species discrimination through DNA Barcode*, a method that aims at using genotypes and haplotypes information in place of phenotype information to classify species. Finally we consider, in Section 6 a typical problem of representation and reconstruction: *TAG-SNPs selection*, that concerns the problem of finding a subset of the loci of a chromosome whose knowledge allows to derive all the others. Given the breadth and the complexity of the material considered, we provide the reader with a minimal level of technical details on the methods described, and give references to the appropriate literature.

2. Feature selection methods

As already pointed out, the identification of a subset of relevant features among a large set (*Feature Selection problem* or FS) is a typical problem in the analysis of biological data. The selected subset has to be small in size and must retain the information that is most useful for the specific application. The role of Feature Selection is particularly important when the data collection process is difficult or costly, as it is the case in most molecular biology contexts.

In this section we look at FS as an independent task that pre-processes the data before classification, reconstruction or representation algorithms are employed. However, the way the target features are chosen strongly depends on the scope of the analysis. When the analysis is of *unsupervised type* - as it is the case for representation or reconstruction - the ideal subset of features should be composed by features that are pairwise uncorrelated and able to differentiate each element from the others; or, put it differently, to have a high degree of variability over the individuals. In absence of other requirements, the best that can happen is that the information (i.e., the variability over the columns of the original data matrix) is retained in the matrix projected over the columns that represent the selected features. On the other hand, in the case of *supervised* analysis (e.g., classification), good features are those that have different values in individuals that belong to different classes, while having equal (or similar) values for individuals in the same class. Obviously, also in this case it is appropriate to require the selected features to be pairwise uncorrelated: the discriminating power of a pair of strongly correlated feature would be in fact equivalent to that of only one of the feature in that pair.

In both cases, the main difficulty of FS lies in the fact that its goal is to select a subset of a larger set that has some desirable properties, where such properties strongly depend on the whole subset and it is thus not always appropriate to measure them by means of simple or low order functions in the elements. Moreover, the number of candidate subsets is exponential in the size of the original set. Many successful methods thus propose heuristic approaches, typically greedy, where the final subset is not guaranteed to be the best possible one, but is verified, by some proper method, to function ”well”. On the other hand, optimal approaches, that guarantee the minimization of some quality function, need some approximation in the evaluation function to become tractable.

In the remainder of this section we introduce a general paradigm for FS, and then we illustrate some of the most established search methods for FS in supervised or unsupervised settings (part of the material of this Section is derived from [25]). In order to give a general overview of the methods available, we refer to the work of [49], according to whom a FS method is based on four main steps, as follows:

1. generation procedure;

2. evaluation function;

3. stopping criterion;

4. validation procedure.

The *generation procedure* is in charge of generating the subsets of features to be evaluated. From the computational standpoint, the number of possible subsets from a set of N features is 2^N. It is therefore very crucial to generate good subsets by trying to avoid the exploration of the entire search space, using heuristic strategies, that at each step select a new feature amongst the available ones to be added to the existing set, or random strategies, where a given number of subsets is generated at random, and the one with the best evaluation value is chosen. The generation starts with the empty set, and then adds a new feature at each iteration (*forward strategy*). Alternatively, it may start from the complete set of features removing one feature at each step (*backward strategy*). Finally, some methods propose to start from a randomly generated subset to which forward or backward strategy is applied.

The *evaluation function* is used to measure the quality of a subset. Such value is then compared with the best available value obtained, and the latter is updated if appropriate. More specifically, the evaluation function measures the relevance of a single feature - or of a subset of features - with respect to the final task of the analysis. An interesting classification for FS methods in the case of supervised learning is given by [23], who propose four classes based on the type of evaluation functions:

- *distance* measures: given 2 classes C_1 and C_2, feature X is preferred to Y if $P(C_1|X) - P(C_2|X) >$ $P(C_1|Y)$ - $P(C_2|Y)$, that is, if X induces a larger increase in the class conditional probabilities with respect to Y;
- *information* measures, that tend to indicate the quantity of information retained by a given feature. For example, feature X is preferred to feature Y if the improvement in the entropy function obtained by adding X is larger that the one obtained by adding Y;
- *dependence* or *correlation* measures: they indicate the capability of a subset of features to predict the value of other features. In this setting, X is preferred to Y if its correlation with the class to be predicted is larger. These measures may also indicate redundancies in the features, based on the cross-correlation between the features themselves;
- *consistency* measures: their purpose is to evaluate the capacity of the selected feature to separate the objects in different classes. For example, a particular feature may be considered uninteresting if two elements of the data set have the same value for that feature but belong to different classes.

The *stopping criterion* is needed to avoid time consuming exhaustive search of the solution space without a significant improvement in the evaluation function. The search may be stopped when a given number of features has been selected, or when the improvement obtained by the new subset is not relevant.

Finally, the *evaluation procedure* measures the quality of the selected subset. This is typically accomplished by running classification or reconstruction methods algorithms by using only the selected features on additional data. According to the type of evaluation function adopted, FS methods are divided into two main groups: *filter methods* and *wrapper methods*. In the former, the evaluation function is independent from the classification or reconstruction algorithm that is to be applied. In the latter, the algorithms are, to a certain extent, the essence of the evaluation function: each candidate subset is tested by using the algorithm and then evaluated on the basis of its performance.

In Figure 1, the general design of a filter and a wrapper method is depicted, where the algorithm is represented by a generic classifier. The opinion that wrapper methods can provide better results in term of final accuracy is widely shared by the scientific community. However, these methods are extremely expensive from the computational standpoint.

Figure 1: Wrappers and Filters.

The computational complexity of filter methods turns out to be very important in the analysis of biological large datasets, where the dimensions involved forbid the application of certain sophisticated classification techniques which use the complete set of features. In such cases, wrapper methods could not be applied in the first place.

For both FS approaches different methods to evaluate the quality of a partial or complete solution are proposed. Such methods are part of the FS algorithm itself, as they direct the search for a solution or the choice of a best solution among the ones available. Nonetheless, their relative importance in wrapper and filter methods is different: in the former, the evaluation function is not used to select the final solution among many, but only to obtain each of them; for this approach, in fact, the effective classification performances are those that drive the choice of the solution. In the latter (filter methods) the choice of the feature set cannot benefit of the result of the classification algorithm, and such task is entirely accomplished by the evaluation function. For this reason it is appropriate to stress the importance of a good choice of the evaluation function (and thus of the FS method). Below, we analyze few of the more relevant methods - the list far from being exhaustive.

Methods based on consistency. Such methods have been conceived for classification problem (supervised learning). The main idea behind this class of methods is that of searching for the smallest subset of the available features that is as consistent as possible with the class variable. The work [3] proposes FOCUS, a method conceived for Boolean domains. The method searches the solution space until the feature subset is such that each combination of feature values belongs to one and only one class. The main drawback of this approach is the explosion of the dimension of the search space when the number of original features increases. They also propose some variants of the original algorithm to speed-up the search procedure. One of them is sketched below. Given a subset of features S, the available data is divided into a number of groups, each of them having the same values for the features in S. Assuming that p_i and n_i represent the number of positive and negative examples in the $i-th$ group respectively, N is the number of individuals, the formula:

$$E(S) = -\sum_{i=0}^{2^{|S|}-1} \frac{p_i + n_i}{N} \left[\frac{p_i}{p_i + n_i} \log_2 \frac{p_i}{p_i + n_i} + \frac{n_i}{p_i + n_i} \log_2 \frac{n_i}{p_i + n_i} \right] \tag{1}$$

is used to evaluate all candidate features that may be added to the current subset and then select the one that shows the lowest value of $E(S)$.

A similar approach has been exploited by [51] with the LVF algorithm, where they measure inconsistency through the following formula:

$$I(S) = \sum_{g=1}^{G_s} \frac{n_g - f_g}{n}, \tag{2}$$

where G_s is the number of different groups of objects defined by the features in S, n_g is the number of objects in group g, f_g is the number of objects in group g that belong to the most frequent class, and n is the total number of objects in the training set. The LVF algorithm then proceeds with the following steps:

- the best subset B is filled with all the original features, and $I(B)$ is then computed;
- a random subset S of the features is chosen;
- if the cardinality of S is less than or equal to the cardinality of B, then $I(S)$ is computed;
- if $I(S) \leq I(B)$, then $B \leftarrow S$, and iterate.

This method can have good behavior in the presence of noisy data, and may be efficient in practice. It may although be misled by features that take on a large number of different values in the training set; in these cases such a feature would provide a high contribution to the consistency measure, but would not be particularly effective for prediction purposes. Similar techniques have been investigated also by [65] and [56].

Methods based on Information Theory. Also in this case, we present methods designed to operate in the presence of a supervised learning task, where a class is value is associated to each individual and the final task is to learn how to predict such class from the selected features. A measure of the information conveyed by a subset is used to direct the search of the final features. Good examples of such methods are the Minimum Description Length Method (MDLM) [67] and the probabilistic approach by [46], that we briefly describe below.

The main idea in [46] is that a good subset of the features should present a class probability distribution as close as possible to the distribution obtained with the original set of features. More formally, let C be the set of classes, V the set of the features, X is a subset of V, $v = (v_1,...,v_n)$ the values taken on by the features V, and v_x the projection of v on X. Then, FS should aim at finding a subset S such that $Pr(C|X = v_x)$ is as close as possible to $Pr(C|V = v)$.

The proposed algorithm starts with all the features and applies backward elimination. At each step, it removes the feature that minimizes the distance between the original and the new class probability distribution. Such distance is measured by means of *cross-entropy*, defined as follows:

$$\begin{aligned} D(Pr(C|V_i = v_i, V_j = v_j), Pr(C|V_j = v_j)) = \\ \sum_{c \in C} p(c|V_i = v_i, V_j = v_j) \log_2 \frac{p(c|V_i = v_i, V_j = v_j)}{p(c|V_j = v_j)}. \end{aligned} \tag{3}$$

Features are then removed iteratively until the desired number of features is reached. Given the nature of the formulas involved, the method must operate on binary features, and thus may require additional transformations of the data.

Methods based on Correlation. The FS process for classification problems is strongly related to the correlation among the features and to the correlation of the features with the class attribute, as in

[31]. Thus, a feature is useful if it is highly correlated with the class attribute. In this case, it will have a good chance of correctly predicting its value. Conversely, a feature will be redundant if its value can be predicted from the values of other features, that is, if it is highly correlated with other features. Such considerations lead to the claim that a good subset of features is composed of those features that are strongly correlated with the class attribute and very poorly correlated amongst themselves. One example of such methods is the Correlation-based Feature Selector method (CFS), proposed in [36], where features are selected on the basis of the correlation amongst nominal attributes. A similar method is presented in [25], where the final set of features is found by searching for a particular maximum-weight $k - subgraph$ in a graph whose weighted nodes are associated with the features and whose arcs represent correlations between the features.

Combinatorial Approaches to FS. In [19] the following combinatorial problem is analyzed and discussed: given a set S, select a subset K such that a number of properties $\Pi_i, i = 1, \ldots n$ held by S are maintained in K. According to the nature of the problem, the dimension of K is to be maximized or minimized. They consider such problem a fundamental model for FS, and state two main variants:

1. subspace selection: S does not satisfy some Π; identify the largest subset $K \in S$ such that $S_{|K}$ (S projected onto K) satisfies all Π;

2. dimension reduction: S satisfies all Π; identify the smallest subset K such that $S_{|K}$ satisfies all Π.

Such setting appears to be very interesting from the formal point of view. Among the different approaches, the idea of formulating the FS problem as a mathematical optimization problem where the number of selected features is to be minimized under some constraints has received some attention in the literature, and proven to be effective in many situations. In [18] the authors adopt such an approach for the selection of Tag SNPs; the mathematical model adopted turns out to be a linear problem with binary variables whose structure is well known in the combinatorial optimization literature as the *set covering problem*. Several similar models where also treated in [30], where large set covering models where proposed (a.k.a. the *test cover* problems). An interesting characteristic of this class of models is that it can be used profitably for feature selection both in supervised and unsupervised learning tasks, simply by changing the set of individual pairs on which the constraints of the model are defined: in the unsupervised case a constraint is generated for each pair of individuals, while in the supervised case only pairs of individuals that belong to different classes are associated with a constraint. The main drawback of this approach, and of the many variants that have been then proposed, lays in the fact that it uses one constraint of the integer programming problem for each pair of elements of the data set that belong to different classes. Such fact implies a rapid growth of the problem dimension, and thus of its intractability, that then requires the use of non optimal solution algorithms. In [7] the computational problems related with the large dimension of the integer programming formulations used for FS are tackled by the use of an efficient GRASP heuristic that provides solutions of good quality in reasonable time. In the same work, an alternative and simplified model based is proposed for the supervised learning case. Such method has been used in some of the applications later described in this chapter, and we give below a short description of it assuming that the available individuals are described by aligned DNA fragments.

For simplicity, we assume the individuals to belong to only two classes, class A and class B. Given a feature f_j, we define $P_A(j,k)$ and $P_B(j,k)$ be the proportion of individuals where feature f_j has value k (for $k \in (A, C, G, T)$) in sets A and B, respectively. If $P_A(j,k) > P_B(j,k)$ (resp. $P_B(j,k) > P_A(j,k)$), then the presence of f_j with value k is likely to characterize items that belong to class A (resp. B). To better qualify the strict inequality between $P_B(j,k)$ and $P_A(j,k)$ we introduce an additional parameter $\lambda > 1$, and then define, for each feature j and for each individual i in class A the vector d_{ij} as follows.

$$d_{ij} = \begin{cases} 1, & \text{if } f_{ij} = k \text{ and } P_A(j,k) \geq \lambda P_B(j,k); \\ 0, & \text{if } f_{ij} = k \text{ and } \lambda P_A(j,k) \leq P_B(j,k); \\ 1, & \text{if } f_{ij} \neq k \text{ and } \lambda P_A(j,k) \leq P_B(j,k); \\ 0, & \text{if } f_{ij} \neq k \text{ and } P_A(j,k) >\geq \lambda P_B(j,k); \end{cases}$$

While, for individuals i in class B, the value of d_{ij} will be:

$$d_{ij} = \begin{cases} 1, & \text{if } f_{ij} = k \text{ and } \lambda P_A(j,k) \leq P_B(j,k); \\ 0, & \text{if } f_{ij} = k \text{ and } P_A(j,k) \geq \lambda P_B(j,k); \\ 1, & \text{if } f_{ij} \neq k \text{ and } P_A(j,k) \geq \lambda P_B(j,k); \\ 0, & \text{if } f_{ij} \neq k \text{ and } \lambda P_A(j,k) \leq P_B(j,k); \end{cases}$$

In the practical application, the parameter λ directly influences the density of the matrix composed of d_{ij} and can be adjusted to obtain a reasonable value for the density itself (say 20%). According to this definition, we assume that the number of ones in vector $d_{.j}$ is positively correlated with the capability of feature f_j to discriminate between classes A and B. We would then like to select a subset of the features that exhibits, as a set, a good discriminating power for all the items considered, so that we may use more features combined together to build rules that perform a complete separation between A and B. The purpose of the feature selection model is then to select a given and small number of features that guarantee a good discriminating power for all the elements of the data sets. This can be formally stated asking to select a given number of features (say, β) that maximize the minimum of the discriminating power over all the items. As is the case in the combinatorial FS models, we define the binary decision variable $x_j = \{0, 1\}$ with the interpretation that $x_j = 1$ (resp. $x_j = 0$) means that feature j is selected, (resp., is not selected). The binary integer optimization problem can then be defined as follows:

$$\begin{array}{lll} \max & \alpha & \\ s.t. & \displaystyle\sum_{i=1}^{m} d_{ij} x_j \geq \alpha & i = 1 \ldots n \\ & \displaystyle\sum_{j=1}^{m} x_j \leq \beta & \\ & x_j \in \{0,1\} & j = 1 \ldots m, \end{array} \tag{4}$$

The optimal solution of the above problem would then select the β features that guarantee the largest discriminating power over all the elements in the data (we note that β is a parameter of the problem, and not a variable).

The number of variables of the problem is given by the number of features (m), and the number of rows by the number of individuals (n), keeping the size of the problem in a linear relation with the size of the data.

The use of the integer optimization models proposed above makes it possible to orient the type of the solution identified by the model. As pointed out before, FS aims at identifying a small number of features with high discrimination power. The fact that these two elements are in a natural position of tradeoff is well represented by the IP models above, where the dimension of the feature set and the dicriminating power (approximated by the right hand side values α of model (4)) may be, in turn, an objective or a constraint.

In a real applications one has no clear clue of what would be the right treshold value to assingn to one parameter while optimizing the other. For example, in solving model 4, one may obtain a value of $\alpha = 3$ having fixed the parameter value of $\beta = 10$. Clearly, a sebsequent run of the optimization algorithm with a larger value of β would provide a solution with $\alpha \geq 3$, while a smaller value of β should result in $\alpha \leq 3$.

on the other hand, if the objective is to guarantee the separability of the sets with the smallest number of features, the classical *minimal test collection* model (as described, among others, in [30]) is the natural choice. In this case the discriminating power is kept to a minimum ($\alpha = 1$) and such fact may results in harder classification problems or more complicated separating models. In other cases the limitation on the number of features that can be selected are made available already by the application - e.g., one may desire to find models that use no more than β^* features. Here the proper route would be to maximize the discriminating power under the constraints that no more than β^* features are selected.

It is important to point out in this context that the ideal solution for a given problem - in terms of number of features and discriminating power - may need few iterations of search on the value of the parameter (be it α or β), and may strongly depend on the nature of the problem and on the type of classification method used after the FS has taken place.

For example, in the applications discussed at the end of this chapter, the winning choice has been to select a value of β sufficiently small to produce simple logic models provided that the associated α exhibits some degrees of redundacy in the discriminating power. The wisdom of this choice is eventually confirmed by the good predictive performances of the models extracted.

3. Classification methods

Let us assume that the objects to classify are vectors of R^n (this is equivalent to saying that, for each object, we know the status, or the level, of n features). Assume furthermore that there are m classes, and that each point x belongs to exactly one class. Given an object $x \in R^n$, let us denote its "true" class by $c(x) \in \{1, \ldots, m\}$.

Typically, a classification problem is formulated as follows. Given t points $x1, \ldots, x^t$, whose class is known, build a *classifier* and use it to classify new points, whose class is unknown. A classifier is a function $\tilde{c} : R^n \mapsto \{1, \ldots, m\}$. The goal is to have $\tilde{c}(x) = c(x)$, for all x, i.e., the hypothesized class should be the true class. Clearly, it is easy to guarantee that $\tilde{c}(x^i) = c(x^i)$ for all $i = 1, \ldots, t$, but it may be hard to guess the true c by the observation of only t points.

A classification function is then *learned from*, or *fitted to*, a training data set. In the above example, for instance, we could take any subset of t' points and construct a function $\tilde{c}'$ from it, and test the validity of the function on the remaining $t - t'$ points. Given that the true state of some points is known and that the function is built by exploiting this knowledge, classification methods of this type are so called *supervised.*

In other situations, the class label for the testing data set is unknown, or voluntarily ignored. In this case, the classification algorithm intuitively tries to label "similar" points (e.g., points whose distance is small under some metric) with the same label. The resulting classification methods of this type are called *unsupervised*, and the classification problem is also called *clustering.* Basically, the classification function defines a set of clusters. New points are then assigned to the cluster to which they have, e.g., the smallest average distance.

We now give a very brief, schematic description, of some of the most widely used classification and clustering procedures. In particular, we pay attention to those procedures that have been widely used in bioinformatics applications (see, e.g., Section 4 on microarrays in this chapter).

Support Vector Machines (SVM). Support vector machines [70, 20] are particularly suited for binary classification tasks. In this case, the input data are two sets of n-dimensional vectors, and a SVM tries to construct a separating hyperplane, i.e., one which maximizes the margin (defined as the minimum distance between the hyperplane and the closest point in each class) between the two data sets.

More, formally, given a training data T consisting of t points, where $T = \{(x^i, c(x^i)) \,|\, x^i \in R^n, c(x^i) \in \{-1, 1\}\}_{i=1}^t$ (in this case, the class for each point is labeled -1 or 1, rather than 1 or 2) we want to find a hyperplane $ax = b$ which separates the two classes. Ideally, we want to impose the constraints $ax^i - b \geq 1$ for x^i in the first class and $ax^i - b \leq -1$ for x^i in the second class. These constraints can be relaxed by the introduction of non-negative slack variables s_i and then rewritten as

$$c(x^i)(ax^i - b) \geq 1 - s_i \tag{5}$$

for all x^i.

A training datum with $s_i = 0$ is closest to the separating hyperplane $\{x : ax = b\}$ and its distance from the hyperplane is called margin. The hyperplane with the maximum margin is the optimal separating hyperplane. It turns out that the optimal separating hyperplane can be found through a *quadratic optimization* problem, i.e., minimize $\frac{1}{2}\,||a||\,2 + C\sum_{i=1}^t s_i$ over all (a, b, s) which satisfy (5). Here, C is a margin parameter, used to set a tradeoff between the maximization of the margin and minimization of classification error. The points that satisfy equality in (5) are called support vectors.

A variant of SVMs, called *discrete support vector machines* (DSVMs) is introduced in [76] and is motivated by an alternative classification error function which counts the number of misclassified points. The proposed approach has been later extended for several classification tasks (see [77]).

SVMs can also be used, albeit less effectively, for multiclass classification. Basically, in the multiclass case, the standard approach is to to reduce the classification to a series of binary decisions, decided by standard SVM. Two such approaches are *one-vs-the-rest* and *pairwise comparison* (other, similar approaches exist in the literature). Assume there are m classes. In the one-vs-the-rest approach, we build a binary classifier that separates each single class from the union of the remaining classes. Then, to classify a new point x, we run each of the m classifiers and look at the output. For each class k, assuming x is in k, the answers may have some inconsistencies. Let $\epsilon(x, k)$ be the number of errors that the m classifiers made under the hypothesis that x does in fact belong to k. Then, x is eventually assigned to the class $\widehat{k}$ for which $\epsilon(x, \widehat{k})$ is minimum (i.e., the class consistent with most of the answers given by the classifiers). This classification approach has several problems and therefore, although widely used, has also been widely criticized [2].

In the pairwise comparison method, one trains a classifier for each pair of classes, so that there are $m(m-1)$ independent binary classifiers. To classify a new data point x, each of these classifiers is run, and its output is viewed as a vote to put x in a certain class. The point is eventually assigned to the class receiving most votes (ties are broken randomly). As for the previous one, also for this classification method several problems have been pointed out.

Error Correcting Output Codes. The ideas underlying Error Correcting Output Codes (ECOC) classification stem from research on the problem of data transmission over noisy channels. The method is based on d binary classifiers, each of which defines a binary partition of the union of the m classes. For each input object, these classifiers produce as output a d-ary vector over $C = \{-1, 1\}^d$, which can be seen as the "codeword", or "signature", of the object. Each class is assigned a unique codeword in C. The class codewords can be arranged in an $m \times d$ matrix, which has the property that *(i)* no column is made of only 1's or -1's (that classifier would not distinguish any class from the others); *(ii)* no two columns are identical or complementary (they would be the same classifier). Under these conditions, the maximum number of classifiers (columns) possible is $2^{m-1} - 1$. If the minimum Hamming distance between any two matrix rows is h, then even if a row undergoes up to $\lfloor (h-1)/2 \rfloor$ bit-flips, its original correct value can still be recovered.

The classification is performed as follows. Given a new input object x, its codeword $w(x)$ is computed via the d classifiers. Then, the object is assigned to the class whose codeword has smallest Hamming distance to $w(x)$. The method performs its best with codewords of maximum length (exhaustive coding). In this case, codewords for each class can be built so that the Hamming distance between each two codewords is $(2^{m-1} - 1)/2$. The method was proposed in [24]. One of its limitations is that the number of classifiers is exponential in the number of classes.

Decision trees. A decision tree (sometimes also called classification tree) is a binary tree whose leaves are labeled by the classes, and whose internal nodes are associated to binary predicates related to the features defining the objects. At each node, one of the outgoing branches corresponds to objects whose features satisfy the predicate, while the other corresponds to objects which do not satisfy the predicate. The path from the root to each node corresponds therefore to a set of conditions which must be met by all objects associated to the node.

Decision trees are built (learned) recursively from the training data via some standard procedures. One such rule (entropy rule) is to find at each tree node a predicate which optimizes an entropy function, or information gain, of the partition it induces (intuitively, a predicate which splits a group in roughly two halves has a good information content). Popular tree decision software packages, widely used in bioinformatics applications, such as C4.5 [62], are based on entropy rules.

To predict the class label of a new input object, the predicates are applied to the input starting at the tree root, one at a time. The responses to the predicates define a path from the root to one leaf. The object is then classified with the class labeling the leaf.

K-Nearest Neighbor. The k-nearest neighbor algorithm (KNN) is a classifier based on the closest training data in the feature space [21]. This is a very simple classifier which has been applied to a large number of classification problems, from document retrieval, to computer vision, bioinformatics, computational physics, etc.

Given a training set of objects whose class label is known, a new input object is classified by looking at its k closest neighbors (typically in the Euclidean distance) of the training set. Each of this neighbors can be imagined as casting a vote for its class. The input is eventually assigned to the class receiving the most votes (if $k = 1$, then the object is simply assigned to the class of its nearest neighbor).

k is a positive integer, usually small, and its best value depends upon the data. While larger k reduce the influence of noise in classification, they tend to degrade clear-cut separation between classes. Usually the best k for a particular problem is found by some heuristic ad-hoc approach.

One of the main drawbacks of KNN is that it may be biased, i.e., classes which are over-represented in the training data set tend to dominate the prediction of new vectors. To counter this phenomenon, there are correcting strategies that take this bias effect into account for better classification.

Boosting. Boosting is the process by which a powerful classifier is built by the incremental use of weak classifiers. The underlying idea is that, even if none of the weak classifiers performs particularly well by itself, when they are taken together (perhaps weighted with different weights), they yield quite accurate classifiers. Each time a new weak classifier is added, the training data is re-weighted. In particular, objects that are misclassified are given more importance (higher weight) and object correctly classified see their weight decreased. This way, each next weak classifier must focus more on the objects that the preceding weak classifiers found more difficult to classify.

A popular boosting algorithm, which has found also many applications in bioinformatics, is AdaBoost ([28]).

The Extraction of Classifying Logic Formulas. The learning of propositional formulas able to execute a correct classification task is performed by several methods presented in the literature. For instance, the already discussed decision trees may be viewed as propositional formulas whose clauses are organized in a hierarchy, and indeed one of their first implementation was designed to deal with data express in binary or qualitative form. A more recent alternative is the more sophisticate LAD system, originally proposed in [14], and the greedy approach originally proposed in [79]. In this category belongs also *Lsquare*, described in ([26]). The basic idea of this method is that the rules are determined using a particular problem formulation that turns out to be a well know and hard combinatorial optimization problem, the *minimum cost satisfiability problem*, or MINSAT. The DNF formulas identified have the property of being created by conjunctive clauses that are searched for following the order with which they cover the training set. Therefore, they are formed by few clauses with large coverage (the interpretation of the trends present in the data) and several clauses with smaller coverage (the interpretation of the outliers). *Lsquare* has been applied with success to different bioengineering problems and further references to these applications will be given in the next sections of this chapter.

4. Microarray analysis

Microarrays (also called DNA arrays) are semiconductor devices used to measure the expression level of thousands of genes within a single, massively parallel, experiment. A microarray consists of a grid with several rows and columns. Each grid cell contains some ad-hoc probe DNA sequence, chosen so as to hybridize, by Watson-Crick complementarity, to the DNA (in fact, the mRNA) of a target gene. In the experiment, the mRNA sequences carrying the gene message, i.e., the instructions on which amino acids compose a particular protein, are amplified, then fluorescently tagged, and poured on the array. After hybridization, the array is scanned by a laser to quantify the amount of fluorescent light in each grid cell. This quantity measures the expression level of the particular gene selected by the cell probe.

Each microarray experiment yields a large amount of information as its output. Typically, a microarray is organized in a rectangular grid, where each row is associated to a sample, e.g., tissue cells, and each column is associated to a target gene. Usually, the the number of rows is in the order of hundreds, and is much smaller than the number of columns that can be as large as a hundred thousands. It follows that the experiment output can be an array of as many as a billion numbers (or bits, if some binarization has been applied). This type of data size naturally calls for feature selection and classification algorithms, if one wants to make any meaningful use of the experiment's output.

One of the main uses of microarray experiments is for *tissue classification* with respect to some tumor. Different tissue cells from both healthy individuals and diseased patients are examined to see their expression profiles with respect to several genes, some of which may be related to the disease under study, while the others may be totally unrelated. The analysis of the expression profiles should hint as to which genes correlate the most with the disease (maybe at different stages). The tissue classification should also be able to allow to classify a new sample as healthy or diseased (and in the latter case, at which stage). All of the works that we survey in this section have chosen cases of tissue classification for the experimental results proving the effectiveness of the proposed methods.

Microarrays are also used in the pipeline to drug discovery and development [32, 54]. They can be used to identify drug targets (the proteins with which drugs actually interact) and can also help to select individuals with similar biological patterns. This way, drug companies can choose the most appropriate candidates for participating in clinical trials of new drugs. Finally, microarrays could help in finding protein or metabolic patterns through the analysis of coregolated genes.

The data sets. The recent years have seen an explosion of interest in classification approaches and applications to microarray experiments. The number of published papers on this topic probably exceeds now a hundred. These papers are, generally speaking, very similar to each other, both in the techniques employed and in the data sets which are utilized.

One of the first such data sets concerns the expression of 2,000 genes measured in 62 epithelial colon samples, both tumoral and healthy ([4]). Another popular data sets is the Lymphoma data set (Alizadeh et al.), which concerns tumor and healthy samples in 47 patients, measured across 4,026 genes. The table below reports the characteristics of the most common data sets used by the majority of papers on classification and feature selection problems in gene expression analysis. For all these data sets, the samples are classified in just two classes, tumoral and normal:

Data set	Citation	# samples	# genes
1. Breast cancer	(Veer et al. [71])	97	24,481
2. Colon	(Alon et al. [4])	62	2,000
3. Leukemia	(Golub et al. [32])	72	7,129
4. Lung cancer	(Gordon et al. [33])	181	12,533
5. Lymphoma	(Alizadeh et al. [1])	47	4,026
6. Ovarian	(Petricoin et al. [61])	253	15,154
7. Ovarian	(Schummer et al. [66])	32	$\simeq 100,000$
8. Prostate	(Singh et al. [68])	21	12,600

The papers. As we just mentioned, the number of papers on classification approaches for microarray data is very large. Almost invariably, all these papers employ some of the classification procedures that we described in Section 3 (in particular, the use of Support Vector Machines is very popular), and then proceed to propose some variants of these procedures or some new *ad hoc* methods.

In [44], Hu et al. consider the standard classification methods and run a comparative study of their effectiveness over gene expression data. They consider classification approaches such as SVMs, Decision Trees (C4.5), and Boosting (AdaBoost), and compare their performances over the data sets reported in Table 4 (with the exception of data set # 7). Generally speaking, boosting methods perform better than the other procedures on these data sets. By using the raw original data, the rate of classification success for C4.5 (decision trees) is 75.3%, for AdaBoost it is 76.7% and for SVMs it is 67.1%. The paper also shows how feature selection preprocessing, i.e., extracting a certain number of genes and using only them for classification purposes, can greatly improve the performance of the classifiers. The approach used for feature selection here is *information gain ratio*, an entropy-based measure of the discrimination power of each gene. In preprocessing, the authors select 50 genes with the highest information gain, and then classification is performed over the data restricted to these genes. The results show the effectiveness of feature selection. In particular, the rate of classification success improves to 89.6% for C4.5, to 94.1% for AdaBoost, and to 88.3% for SVMs.

The work by Ben-Dor et al. [6] study some computational problems related to tissue classification by gene expression profiles. The authors investigate some scoring procedures and clustering algorithms used to classify tissues with respect to several tumors, using data sets 2, 3, and 7, of table 4. The data sets are preprocessed via a simple feature selection scheme, whose goal is to identify a subset of genes relevant for the tumors under study. The authors use a measure of "relevance" to score each gene. The intuition is that an informative gene should have quite different values in the two classes (normal and tumor), and the relevance quantifies the gene's discriminating power. Classification is done via SVMs, boosting, KNN, and via a clustering-based classifier proposed by the authors. The main results of the paper is that FS is extremely important in the classification of tumor samples, as some of the original features "can have a negative impact on predictive performance". The paper demonstrates success rate of at least 90% in tumor versus normal classification, using sets of selected genes.

SVMs are used for classification of gene expression data in [16, 35, 29], DSVMs are used in [78], while a variant of KNN is used in [72] for cancer classification using microarray data. In this work, the authors propose to use a specific distance function, learned from the data, instead of the traditional Euclidean distance for KNN classification. The experiments show that the performance of the proposed KNN scheme is competitive to those of more sophisticated classifiers such as SVMs and the uncorrelated linear discriminant analysis (ULDA) in classifying gene expression data.

In [50], Li et al. perform yet another comparative study of feature selection and classification methods for gene expression data. Among the classification methods that are employed, there are SVMs, KNN, Decision Trees and Error Correcting Output Codes. The results are inconclusive, in that no method clearly emerges as a winner from this comparison. The paper deals mainly with multiclass classification problems, which are shown to be much more difficult than binary classification for gene expression datasets. The paper also employs several different Feature Selection Methods. One of the data sets used is the ALL/AML Leukemia data set (number 3 in table 4) which encompasses gene expression profiles of two acute cases of leukemia: acute lymphoblastic leukemia (ALL) and acute myeloblastic leukemia (AML). As the ALL part of the data comes from two types of cells (B-cell and T-cell), and the AML part is split into bone marrow samples and peripheral blood samples, the data can be seen as a four-class data set. Among the conclusion drawn in the paper, there are (i) it is difficult to ascern a clear winner for the feature selection method; (ii) the interaction of FS and classification methods seems quite complex; (iii) the final accuracy of classification seems to depend more on the classification method employed than on the FS method; (iv) although the results are good for datasets with few classes, they degrade quickly as the number of classes increase.

Another work ([73]) compares the performance of several feature selection and classification methods for gene expression data. The paper also proposes a somewhat innovative approach for feature selection based on a variant of Linear Discriminant Analysis. This variant is particularly suited for "undersampled" problems, i.e., problems in which the number of data points is much smaller than the data dimension (the number of features). Notice that microarray data are naturally undersampled.

The work of Bertolazzi et al. [7] uses Logical Classification and a reduction to the Set Covering problem in order to extract the most relevant features from the Leukemia data set (set number 3 in Table 4). The FS problem is solved by using an ad-hoc metaheuristic of the GRASP type. The data is preprocessed in order to become discrete/binary, so that it can be used by the logical classifier *Lsquare*. This classifier produces a set of boolean formulas (rules) according to which an object can be classified as belonging to one of two classes. Overall, the method achieves a performance of $\simeq 92\%$ success rate in classification on this data set. One important result of the paper is that it shows that the knowledge of as few as two or three features (levels of expression of two or three genes) is sufficient to define extremely accurate rules.

We finally mention the work of Umpai and Aitken [45], who utilize evolutionary algorithms for the selection of the most relevant genes in microarray data analysis. The authors remark on the importance of feature selection for classification, and propose a genetic algorithm for FS whose performance is evaluated using a low variance estimation technique. The computational methods developed perform robustly and accurately (with 95% classification success rate on the Leukemia data set), and yield results in accord with clinical knowledge. The study also confirms that significantly different sets of genes are found to be most discriminatory as the sample classes are refined.

5. Species discrimination through DNA Barcode

5.1. Problem definition and setting

Species identification based on morphological keys is showing its limitations for several reasons. Among these reasons, we find the lack of a sufficient number of taxonomists –and consequently of the expertise required to recognize the morphological keys–, and the fact that the chosen observable keys could be absent in a given individual, since they are effective only for a particular life-stage or gender. However the main reason of all is that the small size of the organisms precludes easy visual identification, since much of their important morphology may be at scales beyond the resolution of light microscopy. According to [10] species identification based on morphological keys have brought in the past to the description of about 1.5 million of unique taxa to the species level, but the total number of species is likely to be in the region of 10 million (May 1988). Most of the relative large organisms (body sizes greater than 10 mm) have been described. However, the vast majority of organisms on the Earth have body sizes less than 1 mm ([10]) hence for these groups the taxonomic deficit is much higher.

The use of a specified DNA sequence to provide taxonomic identification (fingerprinting) for a specimen is a technique that should be applicable to all cellular (and much viral) life (see [11, 13, 12] for a systematic setting of the subject).

These DNA sequences were also called *Molecular Operational Taxonomic Units* (M-OTU for short), where an M-OTU is a terminal node (an organism) in coalescent trees obtained by sequencing an informative sequence of DNA. To be informative, the segment of DNA must be known to be orthologous between species (as paralogues will define gene- rather than organism- trees), and the segment must encompass sufficient variability to allow discrimination between M-OTU. The comparison between the between-taxon difference rate and the within-taxon variation and error rates will define the accuracy and specificity of the M-OTU measurement.

Two problems are generally addressed, i.e., identification of specimens (classification), given a training set of species, and identification of new species. In both cases a set S of M-OTU for which the species is known and a new M-OTU x are given. In the classification setting one would know the most likely species of x. In the species identification case the problem is to find the most likely species of x or determine that x is likely to belong to a new species.

A crucial parameter in this approach is the length of the DNA sequence. In [43] it is shown that while long sequences are needed to obtain correct phylogenetic trees and to identify new species, smaller sequences are sufficient to classify specimens.

Identification methods based on small DNA subsequences are first proposed for least morphologically distinguished species like bacteria, protists and viruses [55, 57] and then extended to higher organisms [15, 17].

More recently, in his first paper on this topic [40] Hebert proposes a new technique, *DNA barcoding*, that uses a specific short DNA sequence from a *standardized and agreed-upon* position in the genome as a molecular diagnostic for species-level identification. He identifies a small portion of the mitochondrial DNA (mt-DNA), the gene cytochrome c oxidase I (COI), to be used as a taxon "barcode", that differs by several percent, even among closely related species, and collects enough information to identify the species of an individual. This molecule, previously identified by [63] as a good target for analysis, is easy to isolate and analyze and it has been shown [53] that resumes many properties of the entire mt-DNA sequence. Since 2003, COI has been used by Hebert to study fish, birds, and other species [41]; one of the most significant results concerns the identification of criptic species among insect parasitoids [69]. For sake of completeness we remind that another mt-DNA subsequence (gene), Cytochrome b, was proposed as a common species-level marker, while COI is specific for animal species [39].

On the basis of these results the Consortium of Barcode of Life (CBOL)[1] was established in 2003. CBOL is an international initiative devoted to developing DNA barcoding as a global standard for the

[1] http://www.barcoding.si.edu/

identification of biological species, and has identified data analysis issue as one of the central objectives of the initiative. In particular:

(a) optimize sample sizes and geographic sampling schemes, as barcodes are not easy to measure, and large samples are very expensive;

(b) consider various statistical techniques for assigning unidentified specimens to known species, and for discovering new species;

(c) stating similarity among species using character-based barcodes and identify what are the character-based patterns of nucleotide variation within the sequenced region;

(d) identify small portion of the barcode that are relevant for specie classification, as sequencing long molecules is expensive (shrinking the barcode).

The last three topics deal mainly with data mining: problems (b) and (c) with classification and (d) with feature selection.

A last observation must be made; the word “barcode” has been used not only to indicate a specific DNA sequence, as in Hebert and CBOL approach, but it could identify other “strings” related to the DNA sequence. In some approaches arrays of gene expression derived through DNA microarrays experiments are used as fingerprinting, in other approaches the aminoacid chains derived from given DNA sequences are considered, while in other approaches the barcode is associated to a set of particular substrings of the DNA chain.

5.2. Methods

A few approaches for barcode analysis have been proposed till now. They are either based on the concept of *distance* between M-OTUs or *character* based. Distance-based methods solve the Classification and Identification problems but do not fully belong to the category described in section 3; they don’t have a training phase and are mainly based on the construction of taxonomies. The character-based methods always comprise a feature selection phase and follow the schema presented in sections 2 and 3.

Distance-based methods. In distance-based methods, the assumption is made that all the DNA sequences are aligned. For these methods different type of distance measures are used. The *Hamming distance* between two aligned barcodes is defined as the number of positions where the two sequences have different nucleotides. In these case the new sequence is assigned to the species of the closest sequence or to the species with minimum average distance. The *convex-score similarity* between two aligned barcode sequences is determined from the positions where the two sequences have matching nucleotides by summing the contributions of consecutive runs of matches, where the contribution of a run is convexly increasing with its length. A new sequence is assigned to the species containing the highest scoring sequence. Another type of distance is the one defined on a coalescent tree. In this case, M-OTU are analyzed by first creating M-OTU profiles (i.e. identifying those loci variable enough that two unrelated individuals are unlikely to have the same alleles) and then using the Neighbor Joining (NJ) method [64] to obtain a phylogenetic tree (the NJ tree), so that each species is identified as represented by a distinct, non overlapping cluster of sequences in this tree: no feature selection is done. The principle of the NJ tree is to find pairs of M-OTUs that minimize the total branch length at each stage of clustering of OTUs starting with a star-like tree.

Character-based methods. A first set of character based methods are those which solve the so called *string barcoding problem*, that allow, given a data base of “known” strings (where known in our case means belonging to a class) to extract which of the strings in the database the unknown one is most similar to. These methods require the alignment of the input DNA sequences. In this case one could simply use similarity searching programs such as BLAST[2] to identify the unknown string, but it is not this simple both for computational complexity and experimental costs. Instead, we can only test for the presence of some particular substring in the unknown string (*substring test*). What we need is a set of substring tests such that on every string in the known set, the set of answers (yes or no) that we receive is unique with respect to any other string in the known set. Then we perform the entire set of tests on the unknown string and compare the answers we receive with the answers for every known string. Since in biology each test is quite expensive, the number of tests must be minimized. So the string barcoding problem is solved with an approach that starts with a set of known strings and builds a minimum cardinality set of substrings that allow us to identify an unknown string by using substring tests.

This problem is introduced in [34] where they use suffix trees to identify the critical substrings, integer-linear programming (ILP) to express the minimization problem, and a simple idea that reduces the size of the ILP, allowing it to be solved efficiently by the commercial ILP solver CPLEX for sequences of about 100,000 base pairs. A more efficient highly scalable algorithm, based on the same approach, is due to [22], that deals with DNA sequence much larger than those of [34]. It uses a greedy setcovering algorithm for a problem instance with $\mathcal{O}n^2$ elements corresponding to the pairs of sequences. In ([8]) the substrings have unit length and the problem is solved through feature selection methods based on integer programming and the logic classification tool *Lsquare* (both already introduced in the initial sections of this chapter). The limited number of features of this application (the loci of the COI barcode sequence are less that 700) makes it possible to solve the feature selection model at optimality with a commercial integer programming code, and then extract the logic rules from the small set (20 to 30 loci) of selected features. A one-against-all strategy is used to solve multi-class problems, and the results obtained show a high degree of precision with combinations of very compact logic formulas.

Other character-based methods methods have been proposed by the Data Analysis Working Group of CBOL, in particular in ([47]) string kernel methods for sequence analysis are applied to the the problem of species-level identification based on short DNA barcodes. This method does not require DNA sequences to be aligned; sorting-based algorithms for exact string k-mer kernels and a divide-and-conquer technique for kernels with mismatches are proposed. Similarity kernel representation opens possibility for building highly accurate predictors (they propose support vector machine classifier), clustering of unknown sequences, alternative tree representations with NO alignment and alternative visualization of data. This method is applied on three datasets, (Astraptes fulgerator, Hesperiidae, and fish larvae, and shows a very high performance w.r.t. traditional suffix tree-based approaches).

Computational performance. Not many computational results and performance analysis are reported in the literature. In [58] an approach based on combining several distance- and character-based classifiers is proposed. The comparison among the performance of this approach and the behavior of all the selected methods run alone is presented. The proposed approach maintains a good classification accuracy (98%) even if the percentage of barcodes removed from each species and used for testing is equal to 50% (the accuracy ranges between 97.4% and 92.4% in the other methods). In order to test the accuracy of new species detection and classification a regular leave-one-out procedure is devised. More precisely a whole species is deleted and from each remaining species 0 to 50 percent of the barcodes are randomly deleted, to obtain a test set with deleted sequences and a training set with the remaining sequences. Also in this case the combined approach achieves an accuracy of 97% even if the percentage of barcodes removed from each species and used for testing is equal to 50%, while the other methods accuracy ranges between 63% and 89.7%.

The Data Analysis Group of CBOL is building a web portal that will allow users to widely analyze the performance of several classification methods.

6. Tag SNP selection and genotype reconstruction

6.1. Problem definition and setting

It is well known that genetic variation among different individuals is limited to a small percentage of positions in DNA sequences (99% of two DNA molecules being identical). These positions are called Single Nucleotide Polymorphisms (SNPs) and are characterized by the fact that two possible values (alleles) of the four bases (T, A, C, G) are observed across a population at such sites, and that the minor allele frequency is at least 5%. The knowledge of such polymorphisms is considered crucial in

h_1	C	C	T	A	T	G	C
h_2	A	C	T	A	G	G	A
g_{12}	C/A	C	T	A	T/G	G	C/A

Figure 2: Examples of Genotypes.

disease association studies over a population of individuals, and is the target of the HapMap project, that has already released a public data base of one million SNPs *genotyped* from four populations of three geographical areas (Africa, East Asia, and Europe). Genome-wide association studies aim at identifying common genetic factors that influence health and disease. A genome-wide association study is defined as any study of genetic variation across the entire human genome that is designed to identify genetic associations with observable traits (such as blood pressure or weight), or the presence or absence of a disease or condition. Even if the number of SNPs identified in human genome is low (about seven millions of sites) w.r.t. the complete sequence, the costs for extracting this knowledge is prohibitive and one of the major research challenges arisen in the last years has been to find a selected number of SNPs (*Tag SNPs*) that are representative of all the other SNPs. This approach is supported by the observation that DNA molecules have a *block structure* [60, 74]. Blocks are subsequences of DNA that have been transmitted during the evolution without splits in the sequence. A result of block transmission is *Linkage Disequilibrium* (LD), a parameter related to some combinations of alleles or genetic markers that in a population can occur more or less frequently than would be expected from a random formation of haplotypes from alleles based on their frequencies. In case of two SNPs a measure of LD is given by $\delta = p_1p_2 - h_{12}$, where p_1p_2 denote the marginal allele frequencies at the two loci and h_{12} denotes the haplotype frequency in the joint distribution of both alleles. A block is a region of the DNA where for each pair of SNPs $\delta \neq 0$; this means that the information contained in a block is redundant and suggests that it is possible to find a small set of SNPs (one SNP for each block for instance) able to predict all the others. Such SNPs are commonly called Tag SNPs and the problem is called *Tag SNP Selection* (TSS for short).

In the following we illustrate the two main variants of the problem: the first one (called *Tag SNP selection*) aims to select a minimum set of Tag SNPs able *to represent* all the other SNPs or, in other words, to maintain enough statistical power to identify phenotype-genotype association. The second one, *SNP reconstruction*, developed in the last years (after 2006) mainly focuses on the reconstruction problem, i.e., the problem of computing the allelic values of all the other SNPs from the set of Tag SNPs.

Before introducing the two problems we need some additional notation.

In this context, it is usual to view DNA molecules as structured into subsequences called chromosomes. A *genotype* is a combination of alleles located in homologous chromosomes of a DNA molecule of a given individual; in case of diploid organisms, characterized by two chromosomes (the maternal one and the paternal one), the genotype is a sequence of pairs of alleles located in certain loci, corresponding to SNPs, of the DNA sequence. If the two alleles are identical, the locus is called *homozygous*, if the two alleles are different, it is called *heterozygous*. In case of heterozygous loci, *the phase* (i.e. the value of the locus associated to maternal and paternal chromosomes) may or may not be known. When the phase is known, then the genotype is split in two sequences, called *haplotypes* (see Figure 2). A genotype of length m is usually represented by a $\{0, 1, 2\}$ sequence where 0 and 1 stand for homozygous types $\{0, 0\}$, $\{1, 1\}$ (0 is usually associated to the most frequent allele) and 2 stands for a heterozygous type.

Let $G = \{g_1, \ldots, g_n\}$ be the set of input genotypes, where each of the n elements is an m-dimensional vector (m is the number of SNPs). We use $g_{i,j}$ to denote the j-th component $(0, 1$ or $2)$ of the genotype g_i.

A *phasing* of a genotype g_i is a pair of haplotypes $h_i^1, h_i^2 \in \{0, 1\}^m$ such that $h_{i,k}^1 \neq h_{i,k}^2$, if $g_{i,k} = 2$ and $h_{i,k}^1 = h_{i,k}^2 = g_{i,k}$ if $g_{i,k} \in \{0, 1\}$.

Given the genotype matrix G, the aim of Tag SNPs selection can be declared as follows: define a partition of the SNPs set in two sets, TAG and $non - TAG$, so that a SNP in $non - TAG$ can be computed from one or more SNPs in TAG.

6.2. Tag SNP selection

A good review of Tag selection methods can be found in [37]. The authors, after a discussion of basic assumptions, describe selection algorithms utilizing a 3-step unifying framework: (1) determining predictive neighborhoods; (2) defining a quality measure describing how well Tag SNPs captures the variance of the full set; (3) minimizing the number of Tag SNPs. The three steps are quite independent of each other, and in fact they could be combined in different ways.

Determining predictive neighborhoods. One of the crucial parameters in determining predictive neighborhoods is the dimension of the genomic region. If the region is large, an interval must be defined where tagging SNPs can be be selected to predict a given SNP. In fact the problem of Tag SNPs selection is NP-hard in general case but becomes easy when a finite number of neighborhoods is used to predict a given SNP. Moreover, when the region is large, long range correlations may be seen by random chance, and hence the region must be partitioned into intervals, to avoid taking into account these correlations. If the genomic region is small, i.e., a single gene, it is not necessary to select an interval.

In [37] different approaches to the problem are presented. One effective method is based on partitioning a chromosome into blocks of SNPs exhibiting low haplotype diversity (high LD) and select within each block a set of SNPs that represent all the other SNPs within that block. However in [37] the authors observe that it is not easy to define block boundaries, there are many methods that produce different blocks and there is no metric to measure the quality of different decompositions. Another method uses a sliding window of a fixed number of positions to define neighborhoods of a given SNP. Finally one of the best method uses the metric LD maps of Maniatis et al. [52] to evaluate distances between two SNPs: only those SNPs whose distance is below a certain threshold in this metric are considered correlated.

Defining a quality measure. Defining a quality measure aims at evaluating how well a set of Tag SNPs captures the variation in the complete set. In [37] a simple metric to evaluate the correlation between two SNPs is proposed whose description demands the introduction of additional notation. Let $H = \{h_1, \ldots, h_n\}$ be the matrix of all haplotypes and let $S = \{s_1, \ldots, s_m\}$ be the set of all SNPS, denote by T the set of all Tag SNPs, by t a Tag SNP, by h_i^T the reduced haplotype h_i corresponding to the SNPs in T and by H^T the set of all these reduced haplotypes. Finally, define ϕ_h (ϕ_g)as the frequency of haplotype h in H (genotype g in G) respectively.

The proposed pairwise metric correlates one single SNP with any other SNP and is based on LD between two SNPs.

$$r_h^2(s,t) = 1 - \frac{\frac{phi_{s_1,t_1}}{\phi_{s1}}(1 - \frac{\phi_{s_1,t_1}}{\phi_{s_1}})}{2\phi_{t_1}(1-\phi_{t_1})} \quad if \quad t \neq s$$

$$r_h^2(s,t) = 1 \quad if \quad t = s$$

where s is the predicted SNP, t is the tag SNP, ϕ_{s1} and ϕ_{t1} are the frequencies of allele 1 in s and t, and ϕ_{s_1,t_1} is the frequency of haplotypes having 1 in both s and t.

If $r^2 = 1$ then one of the two SNPs can fully predict the other (no loss of information). If $r^2 \neq 1$ then a threshold is defined; if this threshold is low, less tag SNPs are required with a significant loss of predicting power.

To maintain the predicting power we have to augment the sample dimension. Additional details on this topic can be found also in [5]. The same authors propose also a more sophisticated multivariate metric, based on the correlation of a single SNP with a set of SNPs (for details see [37]).

Minimizing the number of Tag SNPs. Greedy methods are the most used for selecting a reduced number of tag SNPs. These methods add to a set of tag SNPs a new SNP for which the quality function has the largest increase. Obtaining the optimal value of the quality function is a NP-hard problem in most cases. Branch and bound techniques are often used in case of pairwise metrics and can find optimal solution even with hundreds of SNPs. The same is true when the problem is restricted to blocks. They become very inefficient for large set of SNPs.

Clustering is another possibility for pairwise metrics, even if it does not guarantee to find the minimum set of tag SNPs. Dynamic programming is used to partition haplotypes into blocks, thus reducing the search space. The same technique is also used to predict tag SNPs without partitioning haplotypes into blocks.

6.3. The Reconstruction Problem

A crucial aspect in association studies is that of finding a set of Tag SNPs and to design a *reconstruction method* such that all the other SNPs can be predicted from the Tag SNPs through the proposed method. This topic has been recently addressed by a number of studies (see [59] for a complete bibliography). In the following we present the approaches proposed in [38, 42, 59, 9]

First we introduce some more definitions and notation, as proposed in [38]. In a genotype setting, a *prediction function* is a function $f : \{0, 1, 2\}^T \to \{0, 1, 2\}^m$. Denote by f_j the j-th component of a predicted genotype. For a given vector $q \in \{0, 1, 2\}^T$ of tag SNPs values let $f_j(q)$ denote the j-th component of that vector. Observe that $f_j = f_j(q), \quad \forall j \in T$.

To better qualify the objective of reconstruction, the notion of *prediction error* associated with a pair of TAG and non-TAG sets of a given set of individuals is introduced. The *prediction error* is the proportion of the number of alleles that are wrongly reconstructed on the total number of alleles in SNPs of the non-TAG set. Once defined the prediction error the number of tags could be minimized subject to upper bounds on prediction error measured in leave-one-out cross-validation experiments [37, 38]. Some other approaches search for a partition of the SNPs set and a reconstruction function for which the prediction η error is minimized. Since the frequencies of the genotypes are not known, a learning problem is formulated, where the available data is split into a training set and a testing set. The training set is used to learn the distribution of haplotypes. At this point the problem becomes that one of finding a set of Tag SNPs T of size t such that η is minimized when the haplotype is randomly chosen from the training set. In this way the training set is used to search the partition of the SNPs set and the reconstruction function, while the test set is used to compute the *prediction error*. Most of the methods presented follow this scheme. A fundamental role is played by the reconstruction function, that characterizes the different approaches.

In [38] it is described a dynamic programming algorithm, STAMPA, that takes as training set a given set of DNA sequences and basest he prediction function on the biological hypothesis that each SNP is strongly related only with the two neighboring Tag SNPs (hypothesis strictly related to the Linkage Disequilibrium structure of the DNA molecules). This results in an algorithm whose computational complexity is sufficiently small. The prediction function uses a majority vote (described in the following) in order to determine which value is more likely to appear in the unknown position. The dynamic procedure is based on the following In the paper, a random algorithm to find the Tag SNPs is also proposed.

In [42] the prediction method is based on rounding of Multivariate Linear Regression (MLR) analysis in sigma-restricted coding and a dependence is shown between the reconstruction method and the tag SNP selection. We illustrate the method in the haplotype setting, using the notation introduced in the above sections. Let S be the sample population and let h be the haplotype restricted to the tag SNP set. We want to predict a non-tag SNP s from the value of the Tag SNPs of individuals in S. S and x are represented as a matrix M with $n + 1$ rows (the sample individuals and the unknown individual) and $t + 1 columns$ corresponding to the t tag SNPs and a single non tag SNP. All the values in M are known, except the value of s in x. The MLR SNP prediction method consider all possible resolution of s together with the set of tag SNPs T and choose the closest to T.

In [59] an algorithm based on linear algebra is proposed, that identifies a set of Tag SNPs and a set of linear formulas on these Tag SNPs able to predict all the other SNPs, for genotype data sets. The reconstruction function is a linear function of the set of tag SNPs. SNPs genotype data are converted to numerical dataThe algorithm is tested on a large set of data from public data bases and from association studies. In this approach, since each predicted SNP is a function of all the Tag SNPs, it is possible to keep into account possible LD between distant SNPs.

In [9] the selection of TAG SNPs is accomplished via a direct feature selection approach: a combinatorial model like the one described in section 2 (see also [7]) is used in its unsupervised version, where the role of the features is played by the SNPs and the subset selected by the integer programming model are exactly the TAG SNPs. The assumption behind this approach is that the TAG SNPs must retain the most information contained in the data in order to predict the values of the other SNPs. This assumption is then tested using two different reconstruction techniques. The first one is the *majority vote*, proposed by Shamir in [38]. The general principle of that method is that genotypes that are similar on the Tag SNPs are also likely to be similar on the remaining SNPs. Therefore, the non-TAG SNPs of an new individual are assigned their value according to the most frequent values that are present in the individuals of the training set that mostly agree with that individual over the TAG SNPs. The second method adopted uses a classification approach: each non-TAG SNPs plays in turn the role of the class variables, and a classification formula is learned from the training individuals; such formula is constructed to predict with the highest possible precision the value of the class (e.g., the non-TAG SNP) from the values of the TAG SNPs. This way, each non-TAG gets its own prediction model that is used for its reconstruction in new individuals. The method used to perform the classification task is *Lsquare*(see [26]), already described in Section 3.

The second reconstruction method appears to be more precise but requires a higher computational cost, as a classification problem has to be solved for each non-TAG SNPs.

7. Conclusions

This chapter is devoted to the analysis of feature selection and classification algorithms for computational molecular biology. Its main purpose is to bring to evidence the difficult challenges of data analysis in this application context, mainly due to the large dimension of the data sets that need to be analyzed and to the lack of established methods in a field where technology, knowledge and problems change at a very fast pace.

We have first defined three relevant problems that arise in computational biology: Classification, Representation, and Reconstruction. In the rest of the chapter, some methods and algorithms from the literature are put in the perspective of these three problems. This part is divided into two main blocks: the first devoted to feature selection and classification (the methods of analysis), the second to the description of three different applications where the methods are applied (Microarray analysis, Barcode analysis, TAG SNPs selection).

From the material we presented, three main conclusions may be drawn:

(i) It appears that the current state of the technology in molecular biology strongly demands the deployment of sophisticated analysis tools that rely heavily on computational power and on good-quality algorithms.

(ii) Different methods with different characteristics are available and no simple rule can orient the scientist in the selection of the proper method for a given problem; rather, such choice requires a good knowledge and comprehension of both the method and the problem.

(iii) Finally, when the proper knowledge is available (as is the case for the applications considered and discussed in the second part of this chapter), then a significant contribution to the advance of science in molecular biology can be provided.

22.

References

[1] Alizadeh AA, Eisen MB, Davis RE, Ma C, Lossos IS, Rosenwald A, Boldrick JC, Sabet H, Tran T, Yu X, Powell JI, Yang L, Marti GE, Moore T, Hudson J Jr, Lu L, Lewis DB, Tibshirani R, Sherlock G, Chan WC, Greiner TC, Weisenburger DD, Armitage JO, Warnke R, Levy R, Wilson W, Grever MR, Byrd JC, Botstein D, Brown PO, Staudt LM. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769):503-511, 2000.

[2] Allwein E., Schapire R.and Singer Y., Reducing multiclass to binary, *Journal of Machine Learning Research*, 1, 113–141, 2000.

[3] Almuallim H., and Dietterich T.G. Learning with many irrelevant features. In *Proceedings of the* 9^{th} *National Conference on Artificial Intelligence.* MIT Press, Cambridge, Mass., 1991.

[4] Alon U., Barkai N., Notterman D.A., Gish K., Ybarra S., Mack D and Levine A.J., Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci.*. 96(12):6745-6750, 1999.

[5] Bafna V., Halldórsson B. V., Schwartz R., Clark A. G.and Istrail S., Haplotypes and Informative SNP Selection Algorithms: Dont Block Out Information, *Proceedings of the seventh annual international conference on Research in computational molecular biology 2003, Berlin, Germany RECOMB03*, 19–27, 2003.

[6] Ben-Dor A., Bruhn L., Friedman N., Nachman I., Schummer M.and Yakhini Z., Tissue classification with gene expression profiles. *Journal of Computational Biology*, 7(3-4), 559–83, 2000.

[7] Bertolazzi P., Felici G., Festa P., Lancia G., Logic classification and feature selection for biomedical data, *Computer and Mathematics with Applications* 55(5), 889–899,2008.

[8] Bertolazzi P. and Felici, G., Learning to Classify Species with Barcodes, IASI Tech. Rep. 665, 2007.

[9] Bertolazzi P., Felici G., Festa P., Logic Based Methods for SNPs Tagging and Reconstruction, IASI Tech. Rep. 667, submitted to Computer and Operation Research, 2007.

[10] Blaxter M., Mann J., Chapman T., Thomas F., Whitton C., Floyd R. and Abebe E., Defining operational taxonomic units using DNA barcode data, *Phil. Trans. R. Soc. B*, 360(1462), 1935–1943, 2005.

[11] Blaxter M., Molecular systematics: counting angels with DNA *Nature* 421, 122-124, 2003.

[12] Blaxter M., The promise of a molecular taxonomy, *Phil. Trans. R. Soc. B* 359, 669-679, 2004.

[13] Blaxter M. and Floyd R., Molecular taxonomics forbiodiversity surveys: already a reality, *Trends Ecol. Evol.* 18, 268-269, 2003.

[14] Boros E., Ibaraki T., and Makino K., Logical analysis of binary data with missing bits, *Artificial Intelligence* 1999; 107:219-263.

[15] Brown B., Emberson R.M. and Paterson A.M., Mitochondrial COI and II provide useful markers for Weiseana (Lepidoptera, Hepialidae) species identification, *Bull. Entomol.*, 89, 287-294,1999.

[16] Brown M., Grundy W.N., Lin D., Christianini N., Sugnet C.W., Furey T.S., Ares M. Jr and Haussler D., Knowledge-based analysis of microarray gene expression data by using support vector machines, *Proc. Natl. Acad. Sci.* 97(1), 262–267, 2000.

[17] Bucklin A., Guarnieri M., Hill R. S.,Bentley A. M. and Kaartvedt S., Taxonomic and systematic assessment of planktonic copepods using mitochondrial COI sequence variation and competitive, species-specific PCR, *Hydrobiology*, 401, 239-254, 1999.

[18] Chang C-J., Huang Y-T., Chao K-M., A greedier approach for finding tag SNPs, *Bioinformatics*, 22(6), pages 685-691, 2006.

[19] Charikar M., Guruswami V., Kumar R., Rajagopalan S. and Sahai A., Combinatorial Feature Selection Problems. In *Proceedings of FOCS*, 2000.

[20] Cristianini N. and Shawe-Taylor J. , An Introduction to Support Vector Machines and other kernel-based learning methods, Cambridge University Press, 2000.

[21] Dasarathy B. V. (editor), Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques, IEEE Computer Society, Los Alamitos, 1991.

[22] DasGupta B., Konwar K. M., Mandoiu I. I. and Shvartsman A. A. Highly Scalable Algorithms for Robust String Barcoding, *International Conference on Computational Science* (2), 1020–1028, 2005.

[23] Dash M. and Liu H., Feature Selection for Classification. *Intelligent Data Analysis*, I(3), 1997.

[24] Dietterich T. G. and Bakiri G., Solving Multiclass Learning Problems via Error-Correcting Output Codes, *Journal of Artificial Intelligence Research*, 2, 263–286, 1995.

[25] Felici G., de Angelis V., Mancinelli G., Feature Selection for Data Mining, in *Data Mining and Knowledge Discovery Approaches Based on Rule Induction Techniques*, G. Felici and E. Triantaphyllou eds., Springer, 227–252, 2006.

[26] Felici G. and Truemper K., A minsat approach for learning in logic domains, *INFORMS Journal on Computing*, 13(3), 1–17, 2001.

[27] Felici G. and Truemper K. The Lsquare System for Mining Logic Data, *Encyclopedia of Data Warehousing and Mining*, (J. Wang ed.), vol. 2, Idea Group Inc., 693–697, 2006.

[28] Freund Y. and Schapire R. E., A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1), 119–139, 1997.

[29] Furey T. S., Christianini N., Duffy N., Bednarski D. W., Schummer M.and Haussler D., Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10), 906–914, 2000.

[30] Garey M.R. and Johnson D.S., Computer and Intractability: a guide to the theory of NP-completeness, Freeman, San Francisco, 1979.

[31] Gennari J. H., Langley P., and Fisher D., Models of incremental concept formation. *Artificial Intelligence* 40, 11–61, 1989.

[32] Golub T.R., Slonim D.K., Tamayo P., Huard C., Gaasenbeek M., Mesirov J.P., Coller H., Loh M.L., Downing J.R., Caligiuri M.A., Bloomfield C.D., Lander E.S., Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science.* 286(5439), 531–7, 1999.

[33] Gordon G. J., Roderick V. Jensen, Li-Li Hsiao, Gullans S. R. , Blumenstock J, E., Ramaswamy S., Richards W. G., Sugarbaker D. J. and Bueno R., Translation of Microarray Data into Clinically Relevant Cancer Diagnostic Tests Using Gene Expression Ratios in Lung Cancer and Mesothelioma. *Cancer Research* 62, 4963-4967, 2002.

[34] Rash, S.and Gusfield, D. String barcoding: Uncovering optimal virus signatures, *Proc. 6th Annual International Conference on Computational Biology*, pp. 254–261, 2002.

[35] Guyon I., J. Weston, S. Barnhill and V. Vapnik, Gene Selection for Cancer Classification using Support Vector Machines, *Machine Learning*, 46(1-3), 389–422, 2002.

[36] Hall M. A., Correlation-based Feature Selection for Machine Learning, in *Proceedings of the* 17th *International Conference on Machine Learning*, Stanford University, C.A. Morgan Kaufmann Publishers, 2000.

[37] Halldrsson B.V., Istrail S. and De La Vega F.M., Optimal Selection of SNP Markers for Disease Association Studies, *Hum. Hered*, 58,190–202, 2004.

[38] Halperin E., Kimme G., and Shamir R., Tag SNP selection in genotype data for maximizing SNP prediction accuracy, *Bioinformatics*, 21, 195–203, 2005.

[39] Hajibabaei M., Singer G. A. C. , Clare E. L., Paul D. N. and Hebert P. D. N., Design and applicability of DNA arrays and DNA barcodes in biodiversity monitoring, *BMC Biology* , 5(24), 2007.

[40] Hebert P.D.N., Cywinska A, Ball S.L. and deWaard J.R., Biological identifications through DNA barcodes, *Proc. R. Soc. Lond. B (2003)*, 270, 313-321, 2003.

[41] Hebert P.D.N., Penton E.H, Burns J.M, Janzen D.H and Hallwachs W., Ten species in one: DNA barcoding reveals cryptic species in the Neotropical skipper butterfly Astraptes fulgerator, *Proc. Natl Acad. Sci. USA.*, 101, 14812-14817, 2004.

[42] He J. and Zelikovsky A., Tag SNP Selection Based on Multivariate Linear Regression, International Conference on Computational Science (2), *Lecture Notes in Computer Science* 3992,750–757, 2006.

[43] Jia Min X. and Hickey D.A., Assessing the effect of varying sequence length on DNA barcoding of fungi, *Mol Ecol Notes* 1, 7(3), 365-373, 2007.

[44] Hu H., Li J., Plank A., Wang H. and Daggard G., A Comparative Study of Classification Methods for Microarray Data Analysis, *Proceedings of the fifth Australasian conference on Data mining and analystics*, Sydney, Australia 61, 33–37, 2006.

[45] Jirapech-Umpai T. and Aitken S., Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes, *BMC Bioinformatics.* 6, 2005.

[46] Koller D. and Sahami M., Hierachically classifying documents using very few words. In *Machine learning: Proceedings of the* 14th *International Conference*, 1997.

[47] Kuksa P., Pavlovic V., Kernel methods for DNA barcoding, *Snowbird Learning Workshop*, San Juan, Puerto Rico, March 19-22, 2007.

[48] Kwok Pui-Yan(ed.), Single Nucleotide Polymorphism: Methods and Protocols, *Methods in Molecular Biology.* Human Press Inc, Totowa, New Jersey, 2003.

[49] Langley P., Selection of relevant features in machine learning, in *Proceedings of the AAAI Fall Symposium on Relevance*, AAAI Press, 1994.

[50] Chengliang Zhang Li T. and Mitsunori Ogihara A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression, *Bioinformatics*, 20(15), 2429–2437,2004.

[51] Liu H. and Setiono R., A probabilistic approach to feature selection: A filter solution. In *Machine learning: Proceedings of the* 13th *International Conference on Machine Learning.*

[52] Maniatis N., Collins A., Xu C.F., Mcfhy L.C., Hewett D.R., Tapper W., Ennis S., Ke X., Morton N.E., The first linkage disequilibrium (LD) maps: delineation of hot and cold blocks by diplotype analysis, *Proc Natl Acad Sci USA*, 99, 2228-2233, 2002.

[53] Min X. J. and Hickey D. A., DNA barcodes provide a quick preview of mitochondrial genome composition, *PLoS ONE* , 2(3), e325, 2007.

[54] Montgomery D. and Undem B. L., Drug Discovery. CombiMatrix' customizable DNA Microarrays, *Genetic Engineering News* 22(7), 2002.

[55] Nanney, D. L., Genes and phenes in Tetrahymena, *Bioscience* , 32, 783-740, 1982.

[56] Oliveira A. L. and VincetelliA. S., Constructive induction using a non-greedy strategy for feature selection, in *Proceedings of the* 9^{th} *International Conference on Machine Learning*, 355–360, Morgan Kaufmann, Aberdeen, Scotland, 1992.

[57] Pace N. R., A molecular view of microbial diversity and the biosphere, *Science*, 276, 734-740, 1997.

[58] Pasaniuc B., Kentros S. and Mandoiu I.I., Boosting Assignment Accuracy by Combining Distance- and Character-Based Classifiers, *The DNA Barcode Data Analysis Initiative: Developing Tools for a New Generation of Biodiversity Data*, Paris, France, 2006.

[59] Paschou P., Mahoney M.W., Javed A., KiddJ.R., Pakstis A.J., Gu S., Kidd K.K. and Drineas P., Intra- and interpopulation genotype reconstruction from tagging SNPs, *Genome Res.*, 17, 96-107, 2007.

[60] Patil N., Blocks of Limited Haplotype Diversity Revealed by High-Resolution Scanning of Human Chromosome 21, *Science*, 294, 1719–1723, 2001.

[61] Petricoin E.F., Ardekani A.M., Hitt B.A., Levine P.J., Fusaro V.A., Steinberg S.M., Mills G.B., Simone C., Fishman D.A., Kohn E.C., Liotta L.A., Use of proteomic patterns in serum to identify ovarian cancer, *Lancet*, 359(9306):572-577, 2002.

[62] Quinlan, J. R., *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers, 1993.

[63] Saccone C., DeCarla G., Gissi C., Pesole G. and Reynes A., Evolutionary genomics in the Metazoa: the mitochondrial DNA as a model system, *Gene*, 238, 195-210, 1999.

[64] Saitou N., Nei M., The Neighbour-joining method: a new method for reconstructing phylogenetic trees, *Mol. Biol. Evol.*, 4(4), 406–425, 1987 .

[65] Schlimmer J. C., Efficiently inducing determinations: A complete and systematic search algorithm that uses optimal pruning, in*Proceedings of the* 10^{th} *International Conference on Machine Learning*, 284–290, Amherst, MA:Morgan Kaufmann, (1993).

[66] Schummer M., Ng W.V., Bumgarner R.E., Nelson P.S., Schummer B., Bednarski D.W., Hassell L., Baldwin R.L., Karlan B.Y., Hood L., Comparative hybridization of an array of 21,500 ovarian cDNAs for the discovery of genes overexpressed in ovarian carcinomas. *Gene*, 238(2):375-385, 1999.

[67] Sheinvald J., Dom B. and Niblack W., Unsupervised image segmentation using the minimum description length principle, in *Proceedings of the* 10^{th} *International Conference on Pattern Recognition*, 1992.

[68] Singh D., Febbo P.G., Ross K., Jackson D.G., Manola J., Ladd C., Tamayo P., Renshaw A.A., D'Amico A.V., Richie J.P., Lander E.S., Loda M., Kantoff P.W., Golub T.R.and Sellers W.R., Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1(2):203-209, 2002.

[69] Smith M. A., Woodley N. E., Janzen D. H., Hallwachs W., and Hebert P. D. N., DNA barcodes reveal cryptic host-specificity within the presumed polyphagous members of a genus of parasitoid flies (Diptera: Tachinidae), *PNAS* , 103, 3657-3662, 2006.

[70] Vapnik V. N., *Statistical learning theory.* Wiley, New York, 1998.

[71] Veer L.J., Dai H., van de Vijver M.J., He Y.D., Hart A.A., Mao M., Peterse H.L., van der Kooy K., Marton M.J., Witteveen A.T., Schreiber G.J., Kerkhoven R.M., Roberts C., Linsley P.S., Bernards R. and Friend S.H., Gene expression profiling predicts clinical outcome of breast cancer. *Nature.*415(6871), 530–6, 2002.

[72] Xiong H. and Chen X., Kernel-based distance metric learning for microarray data classification. *BMC Bioinformatics.* 7, 2006

[73] Ye J., Li T., Xiong T and Janardan R., Using Uncorrelated Discriminant Analysis for Tissue Classification with Gene Expression Data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1(4),181–190, 2004.

[74] Zhang K., Qin Z. S., Liu J. S., Chen T., Waterman M. S. and Sun F. Haplotype Block Partitioning and Tag SNP Selection Using Genotype Data and Their Applications to Association Studies, *Genome Res.*,14, 908–916, 2004.

[75] Zhang K., Qin Z. S., Liu J. S., Chen T., Waterman M. S. and Sun F. HapBlock: haplotype block partitioning and Tag SNP selection software using a set of dynamic programming algorithms, *Bioinformatics*, 21, 131–134, 2005.

[76] Orsenigo C. and Vercellis C., Discrete support vector decision trees via tabu-search, *Journal of Computational Statistics and Data Analysis*, 47, 311–322, 2004.

[77] Orsenigo C. and Vercellis C., Accurately learning from few examples with a polyhedral classifier, *Computational Optimization and Applications*, 38, 235–247, 2007.

[78] Orsenigo C., Gene selection and cancer microarray data classification via mixed-integer optimization, In: Evolutionary computation, machine learning, data mining in bionformatics. *Lecture Notes in Computer Science* 4973, 141–152, 2008.

[79] Triantaphyllou E., The OCAT approach for data mining and knowledge discovery. *Working Paper*, IMSE Department, Louisiana State University, Baton Rouge, LA 70803-6409, USA, 2001.