Chapter 2

Machine Learning: An Indispensable Tool in Bioinformatics

Iñaki Inza, Borja Calvo, Rubén Armañanzas, Endika Bengoetxea, Pedro Larrañaga, and José A. Lozano

Abstract

The increase in the number and complexity of biological databases has raised the need for modern and powerful data analysis tools and techniques. In order to fulfill these requirements, the machine learning discipline has become an everyday tool in bio-laboratories. The use of machine learning techniques has been extended to a wide spectrum of bioinformatics applications. It is broadly used to investigate the underlying mechanisms and interactions between biological molecules in many diseases, and it is an essential tool in any biomarker discovery process.

In this chapter, we provide a basic taxonomy of machine learning algorithms, and the characteristics of main data preprocessing, supervised classification, and clustering techniques are shown. Feature selection, classifier evaluation, and two supervised classification topics that have a deep impact on current bioinformatics are presented. We make the interested reader aware of a set of popular web resources, open source software tools, and benchmarking data repositories that are frequently used by the machine learning community.

Key words: Machine learning, data mining, bioinformatics, data preprocessing, supervised classification, clustering, classifier evaluation, feature selection, gene expression data analysis, mass spectrometry data analysis.

1. Introduction

The development of high-throughput data acquisition technologies in biological sciences in the last 5 to 10 years, together with advances in digital storage, computing, and information and communication technologies in the 1990s, has begun to transform biology from a data-poor into a data-rich science. While previous lab technologies that monitored different molecules could

R. Matthiesen (ed.), *Bioinformatics Methods in Clinical Research*, Methods in Molecular Biology 593,
DOI 10.1007/978-1-60327-194-3_2,

quantify a limited number of measurements, current devices are able to screen an amount of molecules nonenvisaged by biologists 20 years ago. This phenomenon is gradually transforming biology from classic hypothesis-driven approaches, in which a single answer to a single question is provided, to a data-driven research, in which many answers are given at a time and we have to seek the hypothesis that best explains them.

As a reaction to the exponential growth in the amount of biological data to handle, the incipient discipline of *bioinformatics* stores, retrieves, analyzes and assists in understanding biological information. The development of methods for the analysis of this massive (and constantly increasing) amount of information is one of the key challenges in bioinformatics. This analysis step – also known as *computational biology* – faces the challenge of extracting biological knowledge from all the in-house and publicly available data. Furthermore, the knowledge should be formulated in a transparent and coherent way if it is to be understood and studied by bio-experts.

In order to fulfill the requirements of the analysis of the biodata available, bioinformatics has found an excellent and mature ally in the *data mining* field. Thanks to the advances in computational power and storage of the previous decade, the data mining field achieved a notable degree of maturity in the late 1990s, and its usefulness has largely been proven in different application areas such as banking, weather forecasting, and marketing. Data mining has also demonstrated its usefulness in different medical applications, resulting in the well-known *evidence-based medicine* and *medical informatics* fields. At present, the time has come for its application in biology. The participation of data mining specialists or statisticians is broadly accepted in multidisciplinary groups working in the field of bioinformatics. Although the term *data mining* can be interpreted as having a number of different meanings within a wide range of contexts, when related to bioinformatics, it refers to the set of techniques and working trends aimed at discovering useful relationships and patterns in biological data that were previously undetected. **Figure 2.1** illustrates all the different steps that are included in a classical data mining approach that is fully valid too for the analysis of biodata, which combines techniques from the domains of *statistics, computer science,* and *artificial intelligence.*

Due to the nature and characteristics of the diverse techniques that are applied for biological data acquisition, and depending on the specificity of the domain, the biodata might require a number of preparative steps prior to its analysis. These steps are illustrated in the first three steps in **Fig. 2.1**. They are usually related to the selection and cleaning, preprocessing, and transformation of the original data. Once data have been prepared for analysis, the *machine learning* field offers a range of modelization techniques

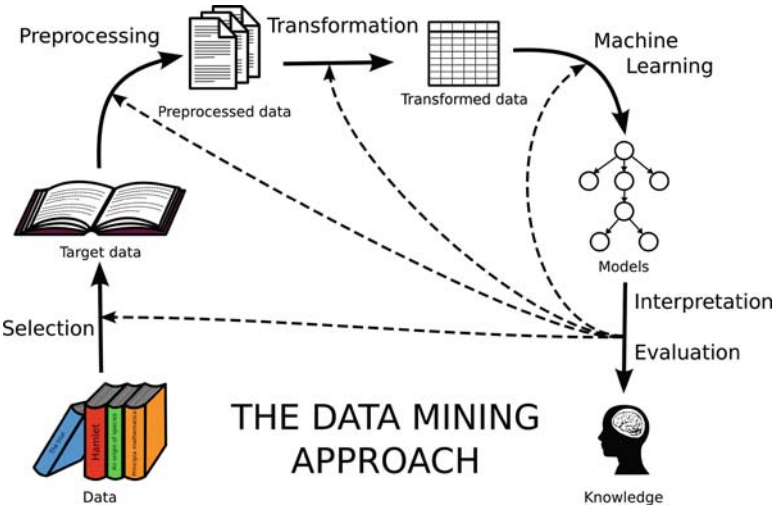

Fig. 2.1. The general chain of work of a common data mining task.

and algorithms for the automatic recognition of patterns in data, which have to be applied differently depending on the goals of the study and the nature of the available data.

Data mining techniques provide a robust means to evaluate the generalization power of extracted patterns on unseen data, although these must be further validated and interpreted by the domain expert. The implication of the bio-expert in the inspection and validation of extracted patterns is essential for a useful outcome of data mining since these patterns provide the possibility to formulate novel hypotheses to be further tested and new research trends to be opened. After all, the discovery of new *knowledge* is regarded as the ultimate desired result of the data mining chain of work shown in **Fig. 2.1**.

From now on, this chapter will focus on the machine learning discipline, which is the most representative task of many data mining applications. Machine learning methods are essentially computer programs that make use of sampled data or past experience information to provide solutions to a given problem. A wide spectrum of algorithms, commonly based on the artificial intelligence and statistics fields, have been proposed by the machine learning community in the last decades.

Prompramote et al. (1) point out a set of reasons to clear up the wide use of machine learning in several application domains, especially in bioinformatics:

- Experts are not always able to describe the factors they take into account when assessing a situation or when explaining the rules they apply in normal practice. Machine learning can serve as a valuable aid to extract the description of the hidden situation in terms of those factors and then propose the rules that better describe the expert's behavior.

- Due to the inherent complexity of biological organisms, experts are very often confronted with finding nondesired results. Unknown properties could be the cause of these results. The dynamic improvement of machine learning can cope with this problem and provide hints to further describe the properties or characteristics that are hidden to the expert.
- As new data and novel concept types are generated every day in molecular biology research, it is essential to apply techniques able to fit this fast-evolving nature. Machine learning can be adapted efficiently to these changing environments.
- Machine learning is able to deal with the abundance of missing and noisy data from many biological scenarios.
- Machine learning is able to deal with the huge volumes of data generated by novel high-throughput devices, in order to extract hidden relationships that exist and that are not noticeable to experts.
- In several biological scenarios, experts can only specify input–output data pairs, and they are not able to describe the general relationships between the different features that could serve to further describe how they interrelate. Machine learning is able to adjust its internal structure to the existing data, producing approximate models and results.

Machine learning methods are used to investigate the underlying mechanisms and the interactions between biological molecules in many diseases. They are also essential for the biomarker discovery process. The use of machine learning techniques has been broadly extended in the bioinformatics community, and successful applications in a wide spectrum of areas can be found. Mainly due to the availability of novel types of biology throughput data, the set of biology problems on which machine learning is applied is constantly growing. Two practical realities severely condition many bioinformatics applications (2): a limited number of samples (*curse of data set sparsity*) and several thousands of features characterizing each sample (*curse of dimensionality*). The development of machine learning techniques capable of dealing with these *curses* is currently a challenge for the bioinformatics community. **Figure 2.2**, which has been adapted and updated from the work of Larrañaga et al. (3), shows a general scheme of the current applications of machine learning techniques in bioinformatics.

According to the objectives of the study and the characteristics of the available data, machine learning algorithms can be roughly taxonomized in the following way:

- Supervised learning: Starting from a database of training data that consists of pairs of input cases and desired outputs, its goal is to construct a function (or model) to accurately

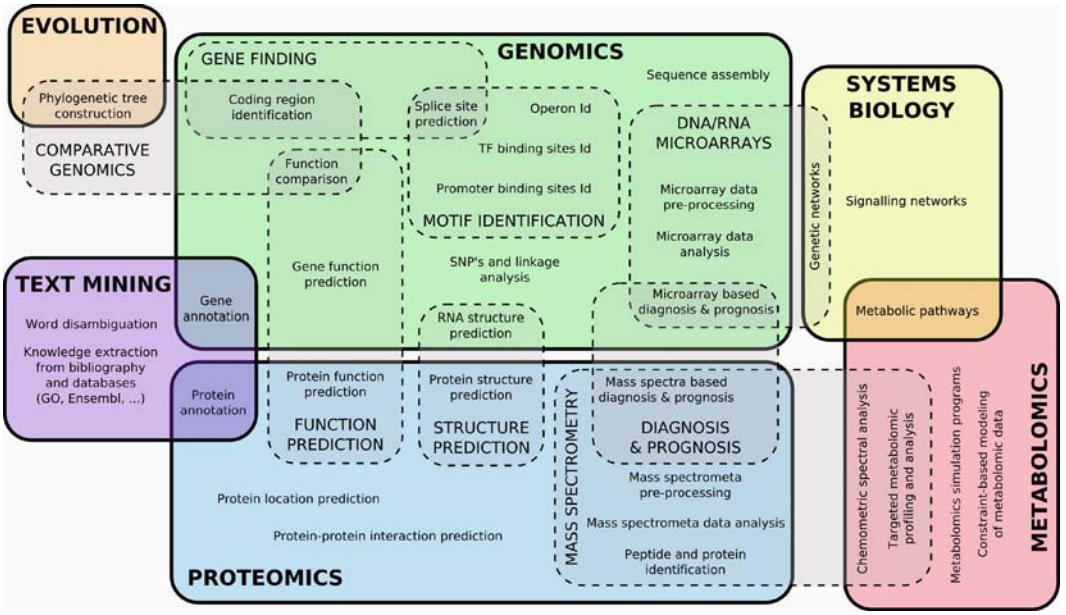

Fig. 2.2. General scheme of the current applications of machine learning techniques in bioinformatics.

predict the target output of future cases whose output value is unknown. When the target output is a continuous-value variable, the task is known as *regression*. Otherwise, when the output (or label) is defined as a finite set of discrete values, the task is known as *classification*.

- Unsupervised learning or clustering: Starting from a database of training data that consists of input cases, its goal is to partition the training samples into subsets (clusters) so that the data in each cluster show a high level of proximity. In contrast to supervised learning, the labels for the data are not used or are not available in clustering.
- Semisupervised learning: Starting from a database of training data that combines both labeled and unlabeled examples, the goal is to construct a model able to accurately predict the target output of future cases for which its output value is unknown. Typically, this database contains a small amount of labeled data together with a large amount of unlabeled data.
- Reinforcement learning: These algorithms are aimed at finding a policy that maps states of the world to actions. The actions are chosen among the options that an agent ought to take under those states, with the aim of maximizing some notion of long-term reward. Its main difference regarding the previous types of machine learning techniques is that input–output pairs are not present in a database, and its goal resides in online performance.

- Optimization: This can be defined as the task of searching for an optimal solution in a space of multiple possible solutions. As the process of learning from data can be regarded as searching for the model that best fits the data, optimization methods can be considered an ingredient in modeling. A broad collection of exact and heuristic optimization algorithms has been proposed in the last decade.

The first two items just listed, supervised and unsupervised classification, are the most broadly applied machine learning types in most application areas, including bioinformatics. Even if both topics have a solid and well-known tradition, the 1990s constituted a fruitful development period of different techniques on both topics, and they fulfill the requirements of the majority of classification experts and studies. That is why this chapter focuses on these two well-known classification approaches, leaving the rest of the topics out of its scope. The interested reader can find qualified reviews on semisupervised learning, reinforcement learning, and optimization in classical books of the machine learning literature (4, 5).

The rest of the chapter is organized as follows. The next section addresses the main techniques applied for data preparation and preprocessing. **Sections 3** and **4** provide an overview of supervised and unsupervised classification topics, respectively, highlighting the principal techniques of each approach. Finally, the interested reader is also directed to a set of web resources, open source software tools, and benchmarking data repositories that are frequently used by the machine learning community. Due to the authors' area of expertise, a special emphasis will be put on the application of the introduced techniques to the analysis of gene expression and mass spectrometry data throughout the chapter. The following references cover extensive reviews on the use of different machine learning techniques in gene expression (6, 7) and mass spectrometry (8, 9).

2. Engineering the Input; the First Analysis Step: Data Preprocessing

Machine learning involves far more than choosing a learning algorithm and running it over the data. Prior to any direct application of machine learning algorithms, it is essential to be conscious of the quality of the initial raw data available, and accordingly, we must discard the machine learning techniques that are not eligible or suitable. The lack of data quality will lead to poor quality in the mined results. As a result, the need to ensure a minimum quality of the data – which might require among other decisions, to discard a part of the original data – is critical, especially in the field

of bioinformatics for several biological high-throughput devices such as DNA microarray or mass spectrometry–based studies, in which the preparation of the raw data could demand the majority of the data mining work.

The *data preprocessing* task is subdivided as a set of relevant steps that could improve the quality – success – when applying machine learning modelization techniques. These procedures are considered "engineering" the input data: They refine/depurate the data to make it more tractable for machine learning schemes. The human attention and time needed by these procedures are not negligible, and the data preprocessing step could be the most time-consuming task for certain data mining applications.

This section briefly describes the main properties and advantages of three well-known data preprocessing topics that are among the most usually applied. These are missing value imputation, data normalization, and discretization. Although several authors consider that the feature selection process belongs to the data preprocessing category, we will revise it as part of the basic supervised modelization scheme.

2.1. Missing Value Imputation

Multiple events can cause the loss of data for a particular problem: malfunctioning measurement equipment, deletion of the data due to inconsistencies with other recorded data, data not entered due to misunderstandings, etc. The first factor is especially critical in modern biological devices, and large amounts of missing data can occur in several biological domains.

Regardless of the reason for data loss, it is important to have a consistent criterion for dealing with the missing data. A simple choice could be the exclusion of the complete sample having any missing value, although this is not an advisable solution since it increases the risk of reaching invalid and nonsignificant conclusions. As an example, let us consider the case of the personal and economical effort required to obtain a DNA microarray sample. Another reason to apply an imputation method is that several classification algorithms cannot be applied on the event of missing values happening.

As the manual imputation of missing values is a tedious and commonly unfeasible approach, the machine learning community has proposed a number of alternatives to handle this situation. The most common approach is to use attribute mean/mode to fill in the missing value: This approach can be improved by imputing the mean/mode conditioned to the class label. More advanced approaches such as decision tree or Bayesian inference and imputation based on the expectation-maximization (EM) algorithm are also proposed in the related literature.

Due to the specificities of biodata, the bioinformatics community has proposed interesting imputation methods that are

most suited according to the different data acquisition methods and nature of the data. For instance, the amount of missing data could be huge in DNA microarray data due to technical failure, low signal-to-noise ratio, and measurement error. That is why the gene expression researchers' community has focused its attention on the proposal of specific imputation methods for DNA microarray data (6).

2.2. Data Normalization

This type of data transformation consists of the process of removing statistical errors in repeated measured data. Data are scaled to fall within a small, specified range, thus allowing a fair comparison between different data samples. The normalization methods identify and remove the systematic effects and variations that usually occur due to the measurement procedure. In this way, a fair integration and comparison of different data samples are guaranteed. Common statistical normalization techniques include min-max normalization, *z*-score normalization, and normalization by decimal scaling (6). Both the DNA microarray and mass spectrometry bioinformatics communities have developed a broad spectrum of interesting normalization methods that are specially suited for the specificities of these domains.

2.3. Data Discretization

Some classification algorithms (e.g., general Bayesian networks) cannot handle attributes measured on a numerical scale. Therefore, if these techniques are to be applied, continuous data must be transformed. This demands the discretization of continuous-range attributes into a small number of distinct states. Although many authors argue that discretization brings about "loss of information" from the original data matrix, other researchers minimize this effect and encourage its use.

Nowadays, a broad range of discretization methods is available for data analysts. Since there are many ways to taxonomize discretization techniques, these can be categorized based on their use of the class label.

Unsupervised discretization methods quantize each attribute in the absence of any knowledge of the classes of the samples. The two most well-known unsupervised techniques are equal-width binning – based on dividing the range of the attribute into a predetermined number of equal-width intervals – and equal-frequency binning – based on dividing the range of the attribute into a predetermined number of intervals with an equal amount of instances.

On the other hand, *supervised discretization* methods take the class label into account for the discretization process. The most widely used algorithm of this category is "entropy-based discretization" (10), which has proven to obtain positive results in a broad range of problems. The goal of this algorithm is to find splits that minimize the class entropy over all possible boundaries,

thus creating intervals with a majority of samples of a single class and a reduced number of samples of the rest of the classes.

Many DNA microarray researchers feel comfortable discretizing original continuous values in three intervals and interpreting them as "underexpression" (with respect to the reference sample), ″baseline,″ and ″overexpression.″

3. Supervised Classification: The Class Prediction Approach

Supervised classification, also known as class prediction, is a key topic in the machine learning discipline. Its starting point is a training database formed by a set of N independent samples $D_N = \{(\boldsymbol{x}^1, c^1), \ldots, (\boldsymbol{x}^N, c^N)\}$ drawn from a joint, unknown probability distribution $p(\boldsymbol{x}, c)$. Each sample $(\boldsymbol{x}^i, c^i)$ is characterized by a group of d predictive variables or features $\{X_1, \ldots, X_d\}$ and a label or class variable of interest C, which "supervises" the whole ongoing process. We will limit our study to the case where the class variable is defined for a finite set of discrete values. Once the needed *preprocessing* steps are performed over the available data, a supervised classification algorithm uses the training database to induce a classifier whose aim is to predict the class value of future examples with an unknown class value.

Supervised classification is broadly used to solve very different bioinformatics problems such as protein secondary structure prediction, gene expression–based diagnosis, or splice site prediction. Current supervised classification techniques have been shown capable of obtaining satisfactory results.

Although the application of an algorithm to induce a classifier is the main step of the supervised classification discipline, two other aspects are vital in this overall process:

- The need to fairly estimate the predictive accuracy of the built model.
- The need for a dimensionality reduction process (e.g., feature selection), in order to improve the prediction accuracy or to handle a manageable number of attributes.

These two concepts are introduced in this section, together with an overview of the main supervised classification algorithms.

3.1. Main Classification Models

Motivated by the "no free lunch" assumption which ensures that there is not a single classification method that will be the best for all classification problems, a notable battery of supervised classification algorithms was proposed by the machine learning and statistics communities in the 1980s and 1990s. Among these, classification models of very diverse characteristics can be found, each

defining a different decision surface to discriminate the classes of the problem. When the only objective is to optimize the predictive accuracy, the common methodology is to evaluate and compare the accuracy of a group of classifiers. However, other factors such as the classifier's transparency, simplicity, or interpretability could be crucial to selecting a final model. Since a description of all the available classification algorithms is beyond the scope of this chapter, we briefly present the main characteristics of four representative models with such different biases: classification trees, Bayesian classifiers, nearest neighbor, and support vector machines.

3.1.1. Classification Trees

Due to its simplicity, speed of classifying unlabeled samples, and intuitive graphical representation, classification trees is one of the most used and popular classification paradigms. The predictive model can be easily checked and understood by domain experts, and it is induced by a recursive top-down procedure. Each decision tree starts with a root node that gathers all training samples. The rest of the nodes are displayed in a sequence of internal nodes (or questions) that recursively divide the set of samples, until a terminal node (or leaf) that does the final prediction is accessed. **Figure 2.3** shows an example of a classification tree.

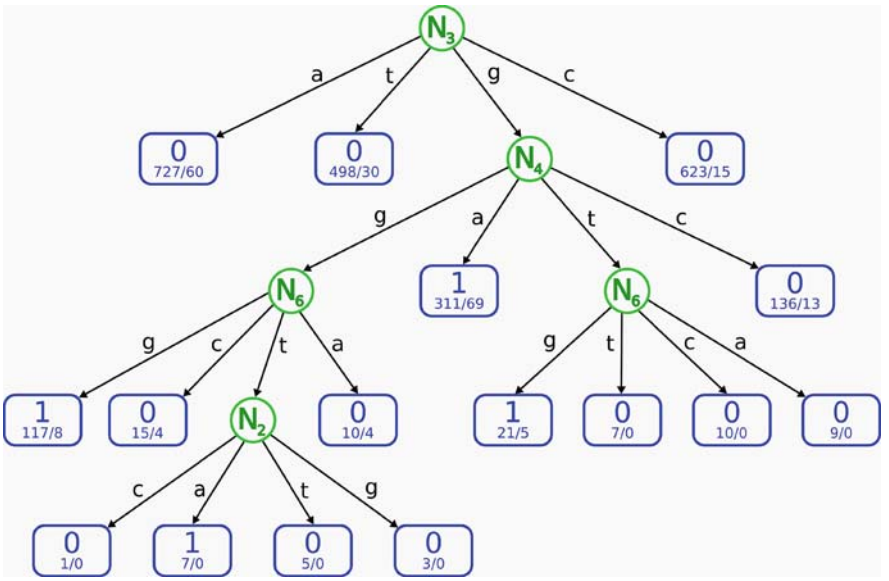

Fig. 2.3. Example of a decision tree constructed to identify donor splice sites. The model was generated from a data set where the class labels are true (1) and false (0) donor sites. The predictive variables represent the nucleotides around the 2-bp constant donor site (from N_1 to N_7). The circles represent the internal nodes and the rounded squares the terminal nodes. In each terminal node, the label of the majority class is indicated (1 or 0). Below this class label, each terminal node shows the number of donor sites in the training set that end up in the node (*left* figure), together with the number of samples that do not belong to the majority class (*right* figure).

Each internal node divides the instances based on the values of a specific informative variable that shows the highest correlation degree with the class label. The related literature proposes a broad range of metrics to measure this correlation degree, mainly based on information theory. Terminal nodes will ideally have samples of only one of the classes, although a mixture of classes is usually found. In order to avoid trees that are too specific and deep, after the tree passes through an initial growing phase, a pruning mechanism is applied in order to delete unrepresentative parts of the tree and to limit the effect of overfitting.

In spite of its popularity in many data analysis areas, in the case of bioinformatics problems – which usually have a limited number of samples per study – its use is not so extended. This could be explained due to its tendency to induce too simple and small trees when a small number of samples are provided.

Due to the instability of the basic formulation of this algorithm – small changes on the training set lead to very different trees – averaging processes are used to obtain more robust classifiers. Random forests average the prediction of a "forest" of decision trees built from resampled training sets of the original data set.

3.1.2. Bayesian Classifiers

This family of classifiers offers a broad range of possibilities to model $p(c|\ x_1,\ x_{2,\ldots,}\ x_d)$, which is the class distribution probability term conditioned to each possible value of the predictive variables. This term, in conjunction with the a priori probability of the class $p(c)$ and by means of Bayes' rule, is used to assign the most probable a posteriori class to a new unseen sample:

$$\gamma(x) = \arg\max_c\ p(c|x_1, x_2,\ldots, x_d) = \arg\max_c\ p(c)\ p(x_1,x_2,\ldots,x_d \mid c).$$

All the statistical parameters are computed from training data, commonly by their maximum-likelihood estimators.

Depending on the degree of complexity of the relationships between the variables of the problem to be modeled, an interesting battery of Bayesian classifiers can be found in the literature. N*aïve Bayes* is the most popular member of Bayesian classifiers. It assumes that all domain variables are independent when the class value is known. This assumption dramatically simplifies the exposed statistics, and only the univariate class-conditioned terms $p(x_i|c)$ are needed. Although this assumption is clearly violated in many situations (especially in many real problems with inherent complexity), the naïve Bayes classifier is able to obtain accurate enough results in many cases.

The *tree-augmented* classifier (11) goes one step further by learning a tree structure of dependences between domain variables. Besides the class label, each variable – except the tree root attribute – is conditioned by another predictor, and statistics of the form $p(x_i|c,x_j)$ have to be computed. This restriction in the

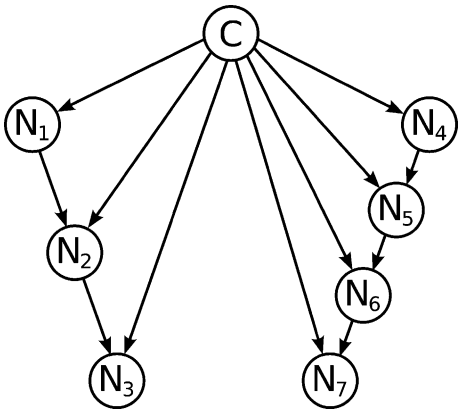

Fig. 2.4. Example of a Bayesian network constructed to identify donor splice sites. The model was generated from a data set where the class labels are true and false donor sites. The predictive variables represent the nucleotides around the 2-bp constant donor site (from N_1 to N_7). During the model-building process, a maximum number of two parents was used as a constraint.

number of parents is overcome by the *k-dependence Bayesian classifier*, which allows each predictive variable to be conditioned by up to *k* parent attributes.

The expressiveness of the graphical structure of a Bayesian classifier (*see* **Fig. 2.4** for an example), which is able to depict the conditional dependence relationships between the variables, is highly appreciated by domain experts, who are able to visually perceive the way in which the model operates. This property of Bayesian classifiers is increasing in popularity in the bioinformatics area. However, due to the nature of data from some bioinformatics tasks with a small number of samples and a large number of variables (e.g., gene expression domains), their application is severely restricted because the impossibility to compute reliable and robust statistics when complex relationships need to be learned from the scarce data. Because of its simplicity, and regardless of its lack of ability to represent too complex relationships among predictor variables of a problem, the naïve Bayes classifier is the most appropriate alternative in such scenarios.

3.1.3. The k-Nearest-Neighbor Paradigm

The basic formulation of the *k–nearest-neighbor* algorithm classifies an unlabeled sample by assigning it to the most frequent class among its *k* nearest samples. While a large battery of variations to follow this aim has been proposed, the majority-voting scheme among the *k* nearest samples for class prediction is the most commonly used. Other variants include the "distance-weighted nearest-neighbor" and the "nearest-hyperrectangle" methods. Implementations commonly use the Euclidean distance for numeric attributes and nominal-overlap for symbolic features.

More distance schemes are the Mahalanobis and the "modified value difference" metrics for numeric and symbolic features, respectively. See **Sections 4.7** and **6.4** in Witten and Frank (12) for a description of these alternatives and other variants. Also known as, ″instance-based learning,″ or ″lazy learning,″ this technique does not induce an explicit expression of the predictive model. Although able to obtain competitive predictive accuracies in many problems, it is discarded in many real situations where a descriptive knowledge discovery output is needed. This is due to the absence of an explicit model to be checked and observed by domain experts. The effect of the *k* parameter can be seen in **Fig. 2.5.**

3.1.4. Support Vector Machines

Support vector machines (*SVMs*) are one of the most popular classification techniques in use today. Its robust mathematical basis and

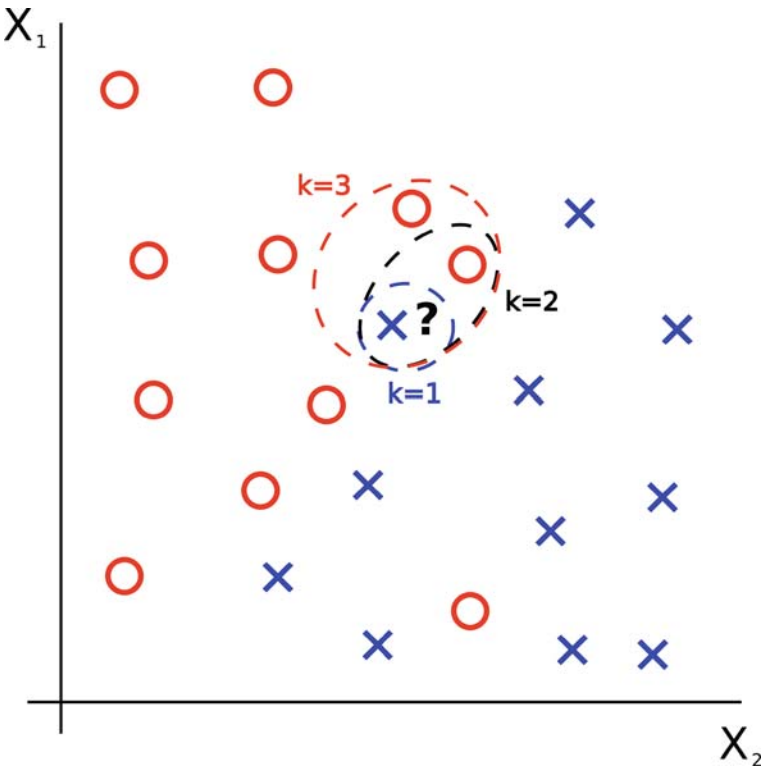

Fig. 2.5. Example of a *k*-nearest-neighbor classification. The problem consists of two variables, X_1 and X_2, and two classes, circle and cross. The circles and crosses represent the known examples, and the question mark a new instance that we need to classify. A 1-nearest neighbor classifies an unlabeled instance as the class of the known instance closest to the instance. In this case, a 1-nearest neighbor would classify the question mark as a cross. A 2-nearest neighbor looks at the two closest examples. In our example, we have a circle and a cross and thus have to choose a way to break the ties. A 3-nearest neighbor would classify the question mark as a circle (we have two circles and a cross). Setting the *k* at an odd value allows us to avoid ties in the class assignment.

the good accuracies that it demonstrates in many real tasks have placed it among practitioners' favorites. SVMs map input samples into a higher-dimensional space where a maximal separating hyperplane among the instances of different classes is constructed. The method works by constructing another two parallel hyperplanes on each side of this hyperplane. The SVM method tries to find the separating hyperplane that maximizes the area of separation between the two parallel hyperplanes. It is assumed that a larger separation between these parallel hyperplanes will imply a better predictive accuracy of the classifier. As the widest area of separation is, in fact, determined by a few samples that are close to both parallel hyperplanes, these samples are called *support vectors*. They are also the most difficult samples to be correctly classified. As in many situations, it is not possible to perfectly separate all the training points of different classes; the permitted distance between these misclassified points and the far side of the separation area is limited. Although SVM classifiers are popular due to the notable accuracy levels achieved in many bioinformatics problems, they are also criticized for the lack of expressiveness and comprehensibility of their mathematical concepts.

3.1.5. Ensemble Approaches

Although the most common approach is to use a single model for class prediction, the *combination of classifiers* with different biases is gaining popularity in the machine learning community. As each classifier defines its own decision surface to discriminate between problem classes, the combination could construct a more flexible and accurate decision surface. While the first approaches proposed in the literature were based on simple combinative models (majority vote, unanimity vote), more complex approaches are now demonstrating notable predictive accuracies. Among these we can cite the bagging, boosting, stacked generalization, random forest, or Bayesian combinative approaches. Due to the negative effect of small sample sizes on bioinformatics problems, model combination approaches are broadly used due to their ability to enhance the robustness of the final classifier (also known as the meta-classifier). On the other hand, the expressiveness and transparency of the induced final models are diminished.

3.2. Evaluation and Comparison of the Model Predictive Power

Since the assessment of the predictive accuracy of a classification model is a key issue in supervised classification, it is essential to measure the predictive power of our model over future unseen samples. This has been subject of deep research in the data analysis field during the last decades, resulting in an abundance of mature and robust concepts and techniques for model evaluation (13). Next, we review the most essential ones.

Given a two-class (positive and negative) problem, a *confusion matrix* such as the one presented in **Table 2.1** applies. This table gathers the basic statistics to assess the accuracy of a predictive

Table 2.1
Confusion matrix for a two-class problem

		Predicted class	
		+	−
Acutal class	+	a	b
	−	c	d

model, showing from qualitative and quantitative points of view a "photograph" of the hits and errors obtained by our model in an accuracy estimation procedure. Considering the counters *a, b, c,* and *d* is enough to compute the following key measures in model evaluation:

- Error rate, the portion of samples the model predicts incorrectly: $(b + c)/(a + b + c + d)$;
- True-positive rate or sensitivity, the portion of the positive samples the model predicts correctly: $a/(a + b)$;
- True-negative rate or specificity, the portion of the negative samples the model predicts correctly: $d/(c + d)$;
- False-negative rate or miss rate, the portion of the positive samples the classifier predicts falsely as negative: $b/(a + b)$;
- False-positive rate or false-alarm rate, the portion of the negative samples the classifier predicts falsely as positive: $c/(c + d)$.

These statistics are computed via an accuracy estimation technique. Since our working data set has a finite set of samples, evaluation involves splitting the available samples into several training and test sets. Since we know the class labels of the samples in the test sets, it is possible to evaluate the models induced by applying a particular classification algorithm by comparing the predictions that the model provides for the test cases. This computes the different accuracy scores. Obviously, the simplest way to estimate the predictive accuracy is to train the model over the whole data set and test it over the same instances. However, within the machine learning community, it is broadly accepted that this procedure, known as resubstitution error, leads to an optimistic bias. That is why machine learning researchers suggest a number of "honest" evaluation schemes, the most popular of which are the following:

- The *hold-out* method randomly divides the data set into a training set and a test set. The classification algorithm is induced in the training set and evaluated in the test set. This

technique can be improved by applying different random train-test partitions. The latter is known as *repeated hold-out*.

- The *k-fold cross-validation* method involves partitioning the examples randomly into k folds or partitions. One partition is used as a test set and the remaining partitions form the training set. The process is repeated k times using each of the partitions as the test set. In *leave-one-out cross-validation,* a single observation is left out each time; i.e., it implies an *N-fold cross-validation* process, where N is the number of instances. *Stratified cross-validation* involves creating partitions so that the ratio of samples of each class in the folds is the same as in the whole data set. **Figure 2.6** shows a scheme of a fivefold cross-validation process.
- The *bootstrap* methodology has been adapted for accuracy estimation. This resampling technique involves sampling with replacement from the original data set to produce a group of *bootstrap data sets* of N instances each. Several variants of the bootstrap estimation can be used for accuracy assessment.

Receiver operating characteristic (ROC) curves are an interesting tool for representing the accuracy of a classifier. The ROC analysis evaluates the accuracy of an algorithm over a range of possible operating (or tuning) scenarios. A ROC curve is a plot of a model's true-positive rate against its false-positive rate: sensitivity versus 1-specifity. The ROC curve represents a plot of these two concepts for a number of values of a parameter (operating

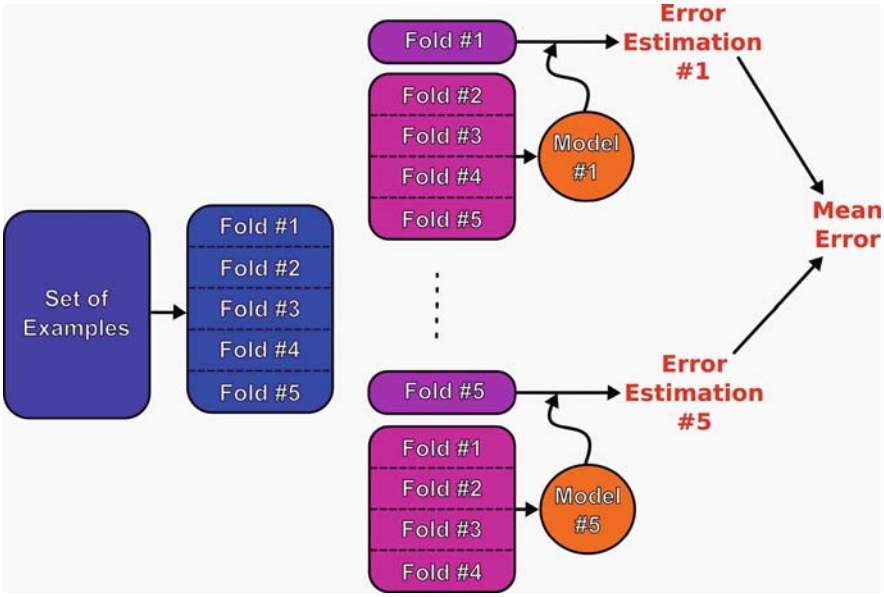

Fig. 2.6. Example of a 5-fold cross-validation process.

scenarios) of the classification algorithm. Examples of this free parameter are the different class misclassification costs or the variation in the class decision threshold of a probabilistic classifier. The area under the ROC curve can also be used for predictive accuracy estimation.

Due to the specificities of many bioinformatics problems that have an extremely low number of samples, the bioinformatics community has proposed novel predictive accuracy estimation methods with promising characteristics, such as bolstered error estimation (14).

Once the predictive accuracy of a group of classifiers in a specific domain has been estimated, an essential question is to perform a comparison between their accuracies. The statistics community has proposed (15) a varied and solid battery of parametric and nonparametric hypothesis tests to assess the degree of significance of the accuracy difference between compared algorithms. Several pioneering papers have alerted (13) the machine learning community about the need to apply statistical tests in order to complete a solid and reliable comparison between classification models. Going further than the classic comparison of classifiers in a single data set, novel conclusive references (16) establish the guidelines to perform a statistical comparison of classifiers in a group of data sets. The use of statistical tests has been extended in the bioinformatics community during recent years.

3.3. Feature Selection

It is well known by the machine learning community that the addition of variables to the classification model is not monotonic with respect to the predictive accuracy. Depending on the characteristics of the classification model, irrelevant and redundant features could worsen the prediction rate. As a natural answer to this problem, the feature selection (FS) problem can be defined as follows: Given a set of initial candidate features in a classification problem, select a subset of relevant features to build a robust model. Together with the improvement in computational and storage resources, a broad and varied range of interesting FS techniques has been proposed in the last 10–15 years, which has brought the FS topic to a high level of maturity and protagonism in many data analysis areas (17).

In contrast to other dimensionality reduction techniques such as those based on projection (e.g., principal component analysis) or compression (e.g., using information theory), FS techniques do not alter the original representation of the variables; they merely select a subset of them. Thus, they preserve the original semantics of the variables, hence offering the advantage of interpretability by a domain expert.

Besides the increase in accuracy, an FS procedure can bring several advantages to a supervised classification system such as decreasing the cost of data acquisition, improving the simplicity

and understanding of the final classification model, and gaining deeper insight into the underlying processes that generated the data.

Although there are many ways to taxonomize FS techniques, these can be divided into three categories depending on how the FS search process interacts with the classification model. We thus have the filter, wrapper, and embedded approaches.

Filter techniques assess the relevance of features by looking only at the intrinsic characteristics of the data, and the interaction with the classification model is ignored. Most filter techniques are based on univariate feature relevance scores, which measure the correlation degree of each attribute with the class label. By means of a univariate metric, a ranking of features is established and low-scoring features are removed. Afterwards, this subset of high-ranked features is used to construct the final classification model. Although univariate metrics are computationally simple and fast, they ignore feature dependencies. Thus, a set of interesting multivariate filter techniques that take into consideration feature dependencies and redundancies has been proposed in the last years.

Wrapper techniques perform a search in the space of feature subsets by incorporating the classification algorithm within the process. The goodness of each subset is obtained by evaluating the predictive power of the classification algorithm when it is trained with the features included in the subset. As the cardinality of possible feature subsets is 2^n (where n is the number of initial attributes), a set of heuristic procedures has been proposed to conduct the search: sequential local techniques, genetic algorithms, ant-colony optimization approaches, etc. The main weaknesses of these techniques are that they have a higher risk of overfitting than filter techniques and they are very computationally intensive, especially if the classifier-building algorithm has a high computational cost.

Several classifier types (e.g., decision trees, decision rules) incorporate (embed) their own FS procedure in the model induction phase, and they do not make use of all initial variables to construct the final classifier. This FS modality is known as *embedded*. These techniques include the interaction with the classification model, and they have a lower computational cost than wrapper procedures.

As modern high-throughput biological devices are capable of monitoring a large number of features for each sample, the application of feature selection techniques in bioinformatics is an essential prerequisite for model building (18). As the magnitude of screened features is of several thousands in many problems, the direct application of any supervised modeling technique is unfeasible. This computational problem is worsened by the small sample sizes available for many bio-scenarios. While many

feature selection techniques developed by the machine learning community are being used with success in bioinformatics research, the bio-community has also proposed during the last years an interesting set of techniques that fit the specificities of their data. The use of feature selection techniques is mandatory in any biomarker discovery process. The protagonism of feature selection is crucial in domains such as DNA microarray studies, sequence analysis, mass spectra, SNP analysis, or literature text mining (18).

4. Unsupervised Classification or Clustering: The Class Discovery Approach

Unsupervised classification – or clustering – is a key topic in the machine learning discipline. Its starting point is a training database formed by a set of N independent samples $D_N = (x^1, \ldots, x^N)$ drawn from a joint and unknown probability distribution $p(\boldsymbol{x}, c)$. Each sample is characterized by a group of d predictive variables or features $\{X_1, \ldots, X_d\}$ and C is a hidden variable that represents the cluster membership of each instance. In contrast to supervised classification, there is no label that denotes the class membership of an instance, and no information is available about the annotation of the database samples in the analysis. Clustering, which is also informally known as "class discovery," is applied when there is no class to be predicted, but rather when the instances are to be divided into natural groups. Once the appropriate *preprocessing* steps are performed over the available data, clustering techniques partition the set of samples into subsets according to the differences/similarities between them. The different objects are organized/taxonomized into groups such that the degree of association between two objects is maximal if they belong to the same group and minimal otherwise. Clustering reflects an attempt to discover the underlying mechanism from which instances originated.

A key concept in clustering is the type of distance measure that determines the similarity degree between samples. This will dramatically influence the shape and configuration of the induced clusters, and its election should be carefully studied. Usual distance functions are the Euclidean, Manhattan, Chebychev, or Mahalanobis.

The validation of a clustering structure, both from statistical and biological points of view, is a crucial task. Statistical validation can be performed by assessing the cluster coherence or by checking the robustness against the addition of noise. An intuitive criterion to be taken into account by any clustering algorithm is the minimization of dissimilarities of samples belonging to the

same cluster (intracluster homogeneity), together with the maximization of the dissimilarities between the samples of different clusters (intercluster heterogeneity). Nevertheless, the problem of biological cluster validation is a highly demanded task by bio-experts that still remains an open challenge. Since a common characteristic of biological systems is the fact that they are not completely characterized, the election of the best cluster configuration is regarded as a difficult task for biologists. However, there are examples of recent methodologies (19) thought to validate clustering structures in different bioinformatics scenarios.

In many bio-scenarios, available samples are not annotated, which has led clustering to have been broadly used to solve different bioinformatics problems such as grouping homologous sequences into gene families, joining peaks that arise from the same peptide or protein in mass spectra experiments, or grouping similar gene expression profiles in DNA microarray experiments.

Clustering techniques play a central role in several bioinformatics problems, especially in the clustering of genes based on their expression profiles in a set of hybridizations. Based on the assumption that expressional similarity (i.e., co-expression) implies some kind of relationship, clustering techniques have opened a way for the study and annotation of sequences. As a natural extension to clustering, the recently revitalized *biclustering* topic has become a promising research area in bioinformatics (20). As it is known that not all the genes of a specific cluster have to be grouped into the same conditions, it seems natural to assume that several genes can only change their expression levels within a specified subset of conditions. This fact has motivated the development of specific biclustering algorithms for gene expression data.

In the following subsections, we briefly present the two principal families of clustering algorithms.

4.1. Partitional Clustering

Clustering algorithms that belong to this family assign each sample to a unique cluster, thus providing a *partition* of the set of points. In order to apply a partitional clustering algorithm, the user has to fix in advance the number of clusters in the partition. Although there are several heuristic methods for supporting the decision on the number of clusters (e.g., the Elbow method), this problem still remains open.

The *k-means algorithm* is the prototypical and best-known partitional clustering method. Its objective is to partition the set of samples into K clusters so that the within-group sum of squares is minimized. In its basic form, the algorithm is based on the alternation of two intuitive and fast steps. Before the iteration of these two steps starts, a random assignment of samples to K initial clusters is performed. In the first step, the samples are assigned to

clusters, commonly to the cluster whose centroid is the closest by the Euclidean distance. In the second step, new cluster centroids are recalculated. The iteration of both steps is halted when no movement of an object to a different group will reduce the within-group sum of squares. The literature provides a high diversity of variations of the *K-means algorithm,* especially focused on improving the computing times. Its main drawback is that it does not return the same results in two different runs, since the final configuration of clusters depends on the initial random assignments of points to *K* initial clusters.

In *fuzzy* and *probabilistic* clustering, the samples are not forced to belong completely to one cluster. Via these approaches, each point has a degree of belonging to each of the clusters. Guided by the minimization of intracluster variance, the literature shows interesting fuzzy and probabilistic clustering methods, and the field is still open for further publication opportunities.

4.2. Hierarchical Clustering

This is the most broadly used clustering paradigm in bioinformatics. The output of a hierarchical clustering algorithm is a nested and hierarchical set of partitions/clusters represented by a tree diagram or *dendrogram,* with individual samples at one end (bottom) and a single cluster containing every element at the other (top). Agglomerative algorithms begin at the bottom of the tree, whereas divisive algorithms begin at the top. Agglomerative methods build the dendrogram from the individual samples by iteratively merging pairs of clusters. Divisive methods rarely are applied due to their inefficiency. Because of the transparency and high intuitive degree of the dendrogram, the expert can produce a partition into a desired number of disjoint groups by cutting the dendrogram at a given level. This capacity to decide the number of final clusters to be studied has popularized the use of hierarchical clustering among bio-experts.

A dissimilarity matrix with the distance between pairs of clusters is used to guide each step of the agglomerative merging process. A variety of distance measures between clusters is available in the literature. The most common measures are single-linkage (the distance between two groups is the distance between their closest members), complete-linkage (defined as the distance between the two farthest points), Ward′s hierarchical clustering method (at each stage of the algorithm, the two groups that produce the smallest increase in the total within-group sum of squares are amalgamated), centroid distance (defined as the distance between the cluster means or centroids), median distance (distance between the medians of the clusters), and group average linkage (average of the dissimilarities between all pairs of individuals, one from each group).

5. Machine Learning Tools and Resources

Together with the improvement in computer storage and computation capacities, the machine learning community has developed a large number of interesting resources during the last decade. These common resources have crucially helped in the development of the field, and they have served as a useful basis to share experiences and results among different research groups.

Due to specific requirements of bioinformatics problems, the bioinformatics community has also contributed to this trend by developing a large number of applications and resources during the last five years. Three popular websites that collect a large amount of varied machine learning resources are Kdnuggets (21), Kmining (22), and the Google Group on Machine Learning (23). The interested practitioner can find in those references the latest data mining news, job offers, software, courses, etc.

We will limit this section to a set of useful and popular resources that have been proposed by the machine learning and data mining communities and that are being used by the bioinformatics community.

5.1. Open Source Software Tools

The MLC++ software (Machine Learning Library in C++) (24) was a pioneering initiative in the 1990s, providing free access to a battery of supervised classification models and performance evaluation techniques. This resulted in a dynamic initiative of the field, offering a base library to develop a large variety of machine learning techniques that appeared in different international publications during the last decade.

MLC++ served as an inspiration for more advanced and user-friendly initiatives during the last decade. Among these, we consider that WEKA (Waikato Environment for Knowledge Analysis) (16) and R-project (25) are nowadays the most influential and popular open source tools: Both offer a huge battery of techniques to cover a complete data mining process. While the algorithms covered by WEKA tend to have a heuristic bias, the R-project is more statistically oriented. As an essential component of the R-project, it is mandatory to reference the popular Bioconductor-project (26), which offers a powerful platform for the analysis and comprehension of genomic data.

Although there are recent initiatives to develop a more user-friendly interface for the powerful tools of the R-project, the intuitive and ease of use of the working environment offered by WEKA is highly appreciated by practitioners not familiarized with current data mining tools.

Other powerful and well-known machine learning free software tools developed by prestigious data mining research laboratories include RapidMiner (27) and Orange (28).

5.2. Benchmarking Data Sets

A common procedure among the developers of machine learning algorithms is to test and compare novel and original classifiers in established data sets. The UCI Machine Learning Repository (29) gathers a varied collection of classification data sets that have become a benchmark repository for machine learning practitioners. The UCI Knowledge Discovery in Databases Archive (30)is an online repository of large and complex data sets that proposes a set of varied, nontrivial data analysis challenges.

A novel and interesting initiative is the Swivel project (31), which is also known as the, ″YouTube of data.″ Any registered user can upload his or her own data collection and correlate it with other data sets. The amount and variety of the collected data sets will surpass the expectations of any interested practitioner.

The interested researcher can find online repositories that collect preprocessed biological data sets ready to be loaded by machine learning software tools. The Kent Ridge Biomedical Data Set Repository (32) gathers a collection of benchmark gene expression and mass spectrometry databases to be mined by supervised classification techniques.

Acknowledgments

This work has been partially supported by the Etortek, Saiotek, and Research Groups 2007–2012 (IT-242-07) programs (Basque Government), the TIN2005-03824 and Consolider Ingenio 2010 – CSD2007-00018 projects (Spanish Ministry of Education and Science), and the COMBIOMED network in computational biomedicine (Carlos III Health Institute).

References

1. Prompramote S, Chen Y, Chen Y-PP. (2005) Machine learning in bioinformatics. In *Bioinformatics Technologies* (Chen Y-PP., ed.), Springer, Heidelberg, Germany, pp. 117–153.
2. Somorjai RL, Dolenko B, Baumgartner R. (2003) Class prediction and discovery using gene microarray and proteomics mass spectroscopy data: curses, caveats, cautions. *Bioinformatics* 19:1484–1491.
3. Larrañaga P, Calvo B, Santana R, Bielza C, Galdiano J, Inza I, Lozano JA, Armañanzas R, Santafé G, Pérez A, Robles V. (2006) Machine learning in bioinformatics. *Briefings in Bioinformatics* 7: 86–112.
4. Alpaydin E. (2004) *Introduction to Machine Learning,* MIT Press, Cambridge, MA.
5. Mitchell T. (1997) *Machine Learning,* McGraw Hill, New York.
6. Causton HC, Quackenbush J, Brazma A. (2003) *A Beginner's Guide. Microarray Gene Expression Data Analysis,* Blackwell Publishing, Oxford.

7. Parmigiani G, Garett ES, Irizarry RA, Zeger SL. (2003) *The Analysis of Gene Expression Data*, Springer-Verlag, New York.
8. Hilario M, Kalousis A, Pellegrini C, Muller M. (2006) Processing and classification of protein mass spectra. *Mass Spectrometry Rev* 25:409–449.
9. Shin H, Markey M. (2006) A machine learning perspective on the development of clinical decision support systems utilizing mass spectra of blood samples. *J Biomed Inform* 39:227–248.
10. Fayyad UM, Irani KB. (1993) Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence,* pp. 1022–1029.
11. Friedman N, Geiger D, Goldszmidt M. (1997) Bayesian network classifiers. *Mach Learn* 29:131–163.
12. Witten IH, Frank E. (2005) *Data Mining. Practical Machine Learning Tools and Techniques (2nd ed.),* Morgan Kaufmann, San Francisco.
13. Dietterich TG. (1998) Approximate statistical test for comparing supervised classification learning algorithms. *Neural Comp* 10:1895–1923.
14. Sima C, Braga-Neto U, Dougherty E. (2005) Superior feature-set ranking for small samples using bolstered error estimation. *Bioinformatics* 21:1046–1054.
15. Kanji GK. (2006) *100 Statistical Tests,* SAGE Publications, Thousand Oaks, CA.
16. Demsar J. (2006) Statistical comparisons of classifiers over multiple data sets. *J Mach Learn Res* 7:1–30.
17. Liu H, Motoda H. (2007) *Computational Methods of Feature Selection,* Chapman and Hall–CRC Press, Boca Raton, FL.
18. Saeys Y, Inza I, Larrañaga P. (2007) A review of feature selection methods in bioinformatics. *Bioinformatics* 23:2507–2517.
19. Sheng Q, Moreau Y, De Smet F, Marchal K, De Moor B. (2005) Advances in cluster analysis of microarray data. In *Data Analysis and Visualization in Genomics and Proteomics* (Azuaje F, Dopazo J, Eds.), Wiley, New York, pp. 153–173.
20. Cheng Y, Church GM. (2000) Biclustering of expression data. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, pp. 93–103.
21. Kdnuggets: Data Mining, Web Mining and Knowledge Discovery (2008) http://www.kdnuggets.com
22. Kmining: Business Intelligence, Knowledge Discovery in Databases and Data Mining News (2008) http://www.kmining.com
23. Google Group – Machine Learning News (2008) http://groups.google.com/group/ML-news/
24. Kohavi R, Sommerfield D, Dougherty J. (1997) Data mining using MLC++, a machine learning library in C++. *Int J Artif Intell Tools* 6:537–566.
25. Dalgaard R. (2002) *Introductory Statistics with R,* Springer, New York.
26. Gentleman R, Carey VJ, Huber W, Irizarry RA, Dudoit S. (2005) *Bioinformatics and Computational Biology Solutions Using R and Bioconductor,* Springer, New York.
27. Mierswa I, Wurst M, Klinkerberg R, Scholz M, Euler T. (2006) YALE: Rapid prototyping for complex data mining tasks. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 935–940.
28. Demsar J, Zupan B, Leban G. (2004) *Orange: From Experimental Machine Learning to Interactive Data Mining,* White Paper, Faculty of Computer and Information Science, University of Ljubljana, Slovenia.
29. Asunción A, Newman DJ. (2008) *UCI Machine Learning Repository,* University of California, Irvine, School of Information and Computer Sciences. http://archive.ics.uci.edu/ml/
30. Hettich S, Bay SD. (1999) *The UCI KDD Archive,* University of California, Irvine, School of Information and Computer Sciences. http://kdd.ics.uci.edu
31. Swivel project – Tasty Data Goodies (2008) http://www.swivel.com
32. Kent Ridge Biomedical Data Set Repository (2008) http://research.i2r.a-star.edu.sg/rp/

http://www.springer.com/978-1-60327-193-6

Bioinformatics Methods in Clinical Research
Matthiesen, R. (Ed.)
2010, X, 390 p. 63 illus., Hardcover
ISBN: 978-1-60327-193-6
A product of Humana Press