28

Novel Machine Learning Techniques for Micro-Array Data Classification

Neamat El Gayar, Eman Ahmed and Iman El Azab
Faculty of Computers and Information,
Cairo University,
Egypt

1. Introduction

Machine learning, data mining and pattern recognition have been quite often used in various contexts of medical and bioinformatics applications. Currently computational methods and tools available for that purpose are quite abundant. The main aim of this chapter is to outline to the practitioners the basic concepts of the fields focusing on essential machine learning tools and highlighting their best practices to be successfully used in the medical domain. We present a case study for DNA microarray classification using ensemble methods and feature subset selection techniques.

The background section will begin by introducing the reader to the fields of pattern recognition, machine learning and data mining. It will then focus on some of the most important concepts related to machine learning.

In particular in section 2 we review the most popular machine learning models for classification used in the context of the medical domain. We then describe one of the most powerful and widely used classifiers for high dimensional feature spaces; the support vector machines (SVM). We cover the area of classifier evaluation and comparison to provide practitioners with essential understanding of how to test, validate and select the appropriate models for their applications. Finally, we summarize the main advances in the field of ensemble learning, feature subset selection and feature subset ensembles.

Section 3 presents a review of using machine learning in various fields of bioinformatics.

In section 4, a recent case study on DNA microarray data that uses an ensemble of SVMs coupled with feature subset selection methods is presented. We show how the proposed model can alleviate the curse of dimensionality associated with expression-based classification of DNA data in order to achieve stable and reliable results.

Section 5 describes the data used and the experiments conducted, while section 6 presents results and a comparative analysis for the proposed models.

Finally in section 7 we summarize the main contributions of this chapter and review the main guidelines to effectively use machine learning tools. We end this section by highlighting a set of challenges that need to be addressed and propose some future research directions in the field.

2. Background

2.1 Pattern recognition, machine learning and data mining

Pattern recognition can be defined as the categorization of the input data into identifiable classes via the extraction of significant features or attributes of the data from a background of irrelevant detail (Duda et al, 2000). The task of pattern recognition is also viewed as the transformation from the measurement space to the feature space and finally to a decision space.

Machine learning techniques aim at producing a system that can learn and adapt from the environment and hence exhibits a kind of intelligence essential for applications that lack known solutions (Alpydin, 2004). Machine learning models very often attempt to optimize a criterion function through exploiting information from training examples.

Data mining, on the other hand, can be thought of as a collection of statistical, machine learning, pattern recognition and artificial intelligence tools that help uncover and extract 'hidden' knowledge from data. Particularly in the medical domain data mining refers often to techniques and methods that analyze large amounts of data. These techniques include among many others classification, clustering, association rule mining and regression or prediction.

Cluster analysis usually addresses segmentation problems. The objective of this analysis is to separate data with similar characteristics from the dissimilar ones. Cluster analysis is frequently the first required task of the mining process. Cluster analysis can also be used for outlier detection to identify samples with peculiar behavior. Among the most simple and efficient clustering techniques are K-means, fuzzy K-means, Self Organizing maps; in addition to more advanced clustering methods like evolving clustering techniques and distributed clustering.

The purpose of *association rule mining,* on the other hand, is to search for the most significant relationship across large number of variables or attributes. Sometimes, association is viewed as one type of dependencies where affinities of data items are described (e.g., describing data items or events that frequently occur together or in sequence). Some techniques for association analysis are nonlinear regression, rule induction, Apriori algorithm and Bayesian networks.

Time Series prediction is also an important aspect in data mining whereby the temporal structure and ordering of the data is utilized to estimate some future value based on current and past data samples. Time-series prediction encompasses a wide variety of applications.

As mentioned earlier, the purpose of this chapter is to provide a broad introduction to the fundamentals of machine learning suitable for bioinformatics. The rest of the chapter will mainly focus on the classification problem.

2.2 Machine learning models for classification

Classification is usually referred to as the process of devising models that can predict categorical (discrete, unordered) class labels. Often machine learning models are used for these purposes that learn the class functions using a set of given training examples.

Popular machine learning classification models are decision tree classifiers, Bayesian classifiers, Bayesian belief networks, rule based classifiers and Backpropagation- Multi layer neural network (Hand et. al, 2001). More recent approaches to classification include support vector machines and ensemble methods. In addition, other approaches are frequently encountered in the literature like *k*-nearest-neighbor classifiers, case-based reasoning, genetic algorithms, rough sets and fuzzy logic techniques.

According to a recent ranking (KDnuggets : Polls, 2006) common classification models used in the data mining community are decision trees, decision rules, logistic regression, artificial

neural networks, support vector machines, the naïve Bayesian classifier and Bayesian networks.

It is worth mentioning at this point that particularly in medical applications sometimes models are preferred that are more interpretable. Such models posses some characteristics like being able to make knowledge discovered from data explicit and communicable to domain experts, the provision of an explanation when deploying and using the knowledge with new cases, in addition to the ability to encode and use the domain knowledge in the data analysis process (Bellazi & Zupan 2008). Decision trees and Bayesian networks are among the models that are easily explainable. Decision Trees are sometimes preferred over more accurate classifier because of their descriptive power; i.e. the ability interpret classification rule produced by the model. This is particularly important for 'safety critically' medical applications where results are required to be understood by domain experts.

Moreover, the fact that medical data can often be imperfect is complemented in practice by exploiting domain knowledge. Building classification models using background knowledge is very useful in order to take into account information which is already known and should not be rediscovered from data. Background knowledge can be expressed using different models like Bayesian models, decision rules and fuzzy rules.

From another perspective, in the bioinformatics applications and in particular for the DNA microarray data classification; more powerful tools are needed to deal with the challenges posed by the low sample size, high dimensionality, noise and large biological variability present in the data.

We therefore devote the next subsections for reviewing Support Vector Machines (SVMs), ensembles methods and feature subset selection techniques. These techniques are known to be robust tools for classification in noisy, high-dimensional and complex domains.

2.3 Support vector machines

This section is devoted to review one of the most powerful and widely used classifiers for high dimensional feature spaces; the support vector machines (SVM).

SVMs are binary classifiers that aim to produce an optimal classifier that lies in midway between the nearest data points of the 2 classes of the problem at hand.

In case of linearly separable problem, SVMs discriminate between two classes by fitting an optimal separating hyper-plane in the midway between the closest training samples of the opposite classes in a multi-dimensional feature space. This is done by maximizing the margin which is the distance between the closest training samples and the classifier.

Given Z a training dataset with N samples in d-dimensional feature space R^d.

Each x_i has class $y_i = \pm 1$.

The objective is to find the linear hyper-plane represented by:

$$f(x) = wx + b \tag{1}$$

Where $\boldsymbol{w}$ is the weight vector and b is the bias that maximizes the margin under the constraint of correct classification. It was found that minimizing $\boldsymbol{w}$ maximizes the margin. This forms the following optimization problem:

$$\min\left(\frac{w^2}{2} + C\sum_{i=1}^{N}\theta_i\right) \tag{2}$$

With *C* as regularization parameter and θ as slack variables.
In case of non-linearly separable classes, the input samples are mapped to a high dimensional feature space using a kernel function. Thanks to kernel trick, it is possible to work within the newly transformed feature space without having to map every sample explicitly.
The training of the SVM requires getting optimal parameter values for the regularization parameter *C*.
The final SVM function for non-linearly separable case is represented by:

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i k\langle x, x_i \rangle + b \tag{3}$$

Where α_i are Lagrange multipliers.
Further detailed explanation can be found in (Abe, 2005).

2.4 Ensemble learning

Ensemble classifiers - also called Multiple Classifier Systems (MCS) - are based on the design of several classifiers separately then joining the final classification decision. MCS are a preferred solution to recognition problems because it allows simultaneous use of different feature descriptors of many types, corresponding measures of similarity and many classification procedures. Examples of these techniques include bagging, boosting, and mixtures of experts and others. Refer to (Roli & Giacinto 2002) (Kuncheva, 2004) (MCS series) for a good review on methods and research in that area.
Perhaps the most obvious motivation for classifier ensembles is the possibility to boost the classification accuracy by combining classifiers that make different errors or by combining local experts. The fact that the best individual classifier for the classification task at hand is very difficult to identify unless deep prior knowledge is available is also a motivation for using multiple classifiers. Another reason is when the features of a sample may be presented in very diverse forms, making it impossible to use them as input for one single classifier. Another rationale is the desire to boost efficiency by using simple and cheap classifiers that operate only on a small set of features. These are all cases that can be found in medical and bioinformatics data.
Classifier combination can fall under one of the following taxonomies according to the type of outputs produced by the classifiers (Kittler et al. 1998). Classifier outputs can be crisp outputs (also called abstract level), ranked list of data classes or measurement level outputs. For abstract level, a classifier outputs a unique label for every pattern to be classified. The combination of such classifiers is usually done by voting strategies, such as majority vote, weighted majority vote or by trained fusion rules such as Behavioural Knowledge Space (Kuncheva, 2004).
For rank level classifiers, the output is a ranked list of labels for every pattern. Borda Count is the common technique to combine these rankings. The rankings from all classifiers are combined by ranking functions assigning votes to the classes based on their positions in the classifiers' rankings. The final decision is taken as the minimum of the sum of these rankings. Finally, at the measurement level, the classifier output represents the degree of belongingness in each class. For this type of output various combination rules can be applied like product, sum, mean, etc. These combination rules are derived mainly from Bayesian decision rule. Non-Bayesian combinations can also be applied such that a weighted linear combination of classifiers is learnt using optimization techniques.

In our model described in section 4, we present a combiner based on a SVM trainable classifier that works on measurement level outputs of the base classifiers.

2.5 Feature subset selection and feature subset ensembles

A common way to build base classifiers for further combination is by randomly selecting different subsets of features and training classifiers on those subsets. Feature subset selection should enforce diversity among classifiers created and hence lead to more robust ensembles.
In applications that are characterized by having a huge number of features, feature subset ensembles can be used in order to make use of all the features.
The way this method works is by sub-sampling the features such that the base classifiers in the ensemble can be built on different subsets of features, either disjoint or overlapping. So instead of over-whelming a single classifier with all the features, individual classifiers can be built on groups of feature then their decisions are combined to get the final decision.
The choice of features for each subset depends on the problem at hand. The features may be naturally grouped forming the feature subsets. They can also be selected by any available feature selection method. The random subspace methods (Ho, 1998) work well when there is redundant information dispersed across all the features (Kuncheva, 2004).
Also, various heuristic search techniques such as genetic algorithms, tabu search and simulated annealing are used for feature subset selection. The feature subsets can be selected one at a time or all at the same time in one run of the algorithm by optimizing some ensemble performance criterion function (Kuncheva, 2004).
Random selection is the intuitive way for selecting samples and is the simplest method available. It assumes a uniform distribution for all the samples.
There are two types of random selection: *Random selection without replacement* in which the samples are randomly selected then removed so that they cannot be chosen again and *Random selection with replacement* where the samples are randomly selected then placed back so that they can be chosen again.
In our case study presented later, we use *Random selection without replacement*. We also propose a feature subset selection method based on the *K*-means clustering algorithm. K-means is a typical partition-based clustering method. Given a pre-specified number *K*, the algorithm iteratively partitions the data set till it gets *K* disjoint subsets. In these iterations, *K*-means tries to minimize the sum of the squared distances of the samples from their cluster centres. It is a simple and fast algorithm. In our proposed approach the genes are the objects of interest to be clustered and they are characterized by their expression values among the samples in the microarray dataset.

2.6 Classifier testing and evaluation

As follows we present main concepts for classifier evaluation and comparison. We start by reviewing cross validation and then discuss main performance measures that can be used to evaluate classification results.

2.6.1 Cross validation

A classifier usually learns from the available data. The problem is that the resulting classifier may fit on the training data, but might fail to predict unseen data.
Cross validation is a technique for assessing the generalization performance of a given classifier. It can be used for estimating the performance of a given classifier as well as for tuning the model parameters.

Methods of cross validation include *Re-substitution Validation*, *Hold-Out Validation*, *K-Fold Cross Validation*, and *Leave-One-Out Cross Validation* as will be described next.
In *Re-substitution Validation* all the available dataset is used for training the classifier. Then, it is tested on the same dataset. This makes it liable to overfitting. Thus, the classifier might perform well on the available data yet poorly on future unseen test data. However in the *Hold-Out Validation* the available dataset is split into 2 sets: one for training and the other for testing the model, such that the model can be tested on unseen data. For this approach the results are highly dependent on the choice for the training / test split. The instances in the test set may be too easy or too difficult to classify and this can skew the results. On the other hand, the instances in the test set may be valuable for training and when they are held out, the prediction performance may suffer leading to skewed results.
To overcome this drawback in *K-Fold Cross Validation*, the available data is divided into k equally sized folds. Subsequently, k iterations of training and validation are performed such that, within each iteration a different fold of the data is held-out for validation while the remaining (k-1) folds are used for training the classification model. Data is usually stratified prior to being split into k folds i.e. data is rearranged to ensure that each fold contains instances of all the classes in the problem at hand.
Leave-One-Out Cross Validation (LOOCV) is a special case of k-fold cross validation where k equals the number of instances in the data. In other words, in each iteration, all the data except for a single instance are used for training the model and the model is tested on that single instance. An accuracy estimate obtained using LOOCV is known to be almost unbiased but it has high variance.
To obtain more reliable performance estimates, multiple runs of k-fold cross validation can be applied. The data is reshuffled and re-stratified before each round. This is referred to as *Repeated K-Fold Cross Validation.*

2.6.2 Performance measures

In this section we review some of the most important measures to calculate classifier performance. In particular we discuss the accuracy, sensitivity, specificity and precision measures.
A classifier is tested by applying it to unseen test data with known classes and comparing the predicted classes resulted from the classifier with the target classes.
The confusion matrix summarizes the correct and incorrect classifications resulted from a given classifier. It displays both the actual target classes and the predicted classes. The matrix dimension is $M \times M$, where M is the number of classes of the problem at hand. The entry m_{ij} of such a matrix denotes the number of samples whose actual class is w_i, and which are assigned by the classifier to class w_j.
Usually, in medical diagnosis, there are two classes, the positive class that indicates infection/sickness and the negative class that indicates being healthy. For assessing the performance of a given classifier, there are other important measures that need to be considered in addition to accuracy. Among them are the sensitivity and the specificity. Sensitivity is the proportion of correctly classified samples for being positive of all the samples that are actually positive, while specificity is the proportion of correctly classified samples for being negative of all the samples that are actually negative.
In the confusion matrix, in figure 1, 'A' represents the number of samples that actually belong to the positive class and are predicted to belong to the positive class. This is also

Confusion Matrix		Predicted Classes	
		+ve	-ve
Actual Target Classes	+ve Class	A	B
	-ve Class	C	D

Fig. 1. Confusion matrix.

quite often referred to as the *true positive (TP)*. For medical application this would indicate the number of patterns found to be sick; while they are really sick. '*B*' on the other hand represents the *false negatives (FN)*; i.e the number of sick people (positive samples) who have been falsely classified to be healthy (or negative). Similarly 'C' is referred o as the *false positive (FP)* while 'D' indicates the *true negative (TN)*.

Accuracy is the percentage of correctly classified samples. It can also be formulated as in equation 4 using TP, TN, FP and FN.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \times 100\% \tag{4}$$

Sensitivity measures the ability of a classifier to recognize the positive class (in our application to detect sick people). It is also known as *True Positive Rate (TPR)* or *recall*.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\% \tag{5}$$

On the other hand **Specificity** measures the ability of a classifier in detecting the negative class (i.e healthy samples). This is also known as *True Negative Rate (TNR)*.

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \times 100\% \tag{6}$$

The relationship between sensitivity and specificity, as well as the performance of the classifier, can be visualized and studied using a receiver operating characteristic (*ROC)* Curve. It is a graphical plot of the *sensitivity (TPR)* versus *false positive rate (FPR)* which is (1 − *specificity*), for a binary classifier as its discrimination threshold is varied.

The ROC space is defined by two axes which are *FPR* and *TPR* representing the *x*-axis and the *y*-axis, respectively. This depicts relative trade-offs between true positive representing benefits and false positive representing costs. A point in the ROC space represents a prediction of the classifier.

A prefect classification would yield a point in the upper left corner or coordinate (0, 1) of the ROC space where there is 100% sensitivity (no false negatives) and 100% specificity (no false positives). The point (0, 0) represents a classifier that predicts all cases to be negative, while the point (1, 1) corresponds to a classifier that predicts every case to be positive. Point (1, 0) is the classifier that is incorrect for all classifications as it means 100% false negatives and 100% false positives. The diagonal divides the ROC space. Generally, the points above the diagonal represent good classification results while the points below the line poor results.

Precision is the proportion of the true positives against all the positive results.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100\% \tag{7}$$

F_1 score (also **F-score** or **F-measure**) is a measure that considers both the precision and the recall of the test to compute the score.

$$\text{F} = 2.\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

This is also known as the F_1 measure, because recall and precision are evenly weighted. It is a special case of the general F_β measure (for non-negative real values of β).

$$\text{F}_\beta = \left(1 + \beta^2\right).\frac{\text{Precision} \times \text{Recall}}{\beta^2.(\text{Precision} + \text{Recall})} \tag{9}$$

Two other commonly used *F* measures are the F_2 measure, which weights recall higher than precision, and the $F_{0.5}$ measure, which puts more emphasis on precision than recall. Usually the measure chosen for performance evaluation is application dependent. In medical applications usually precision and recall – in addition to accuracy- are very important. In our case study presented in section 4 we use accuracy, sensitivity, specificity and precision to evaluate the proposed model for DNA microarray data classification.

3. Machine learning techniques in bioinformatics

Due to the availability of huge amounts of data delivered by high-throughput biotechnologies, data management procedures are required to provide the ability to store and retrieve biological information efficiently (Valentini, 2008)(Goble & Stevens,2008); this is in addition to the need of methods to extract and model biological knowledge from the data (Baldi & Brunak, 2001).

Machine learning techniques deal with a wide range of bioinformatics problems in genomics, proteomics, gene expression analysis, biological evolution, systems biology, and other relevant bioinformatics domains (Valentini, 2008) (Larranaga et.al., 2005). As follows we briefly review the use of machine learning in each of the previously mentioned fields of bioinformatics. However, the rest of the chapter will focus on micro-array data classification.

The state of a cell consists of all those variables-both internal and external-which determine its behaviour. According to the Central Dogma of molecular biology, the activity of a cell is determined by which of its genes are expressed i.e., which genes are ``turned on'', resulting in the active production of the respective proteins. When a particular gene is expressed, its DNA is first transcribed into the complementary messenger RNA (mRNA), which is then translated into the specific protein this gene codes for. We can measure the level of expression of each gene (i.e. how much each gene is ``turned on'') by measuring how many mRNA copies are present in the cell (Lander, 1996).

Genomics is one of the most important domains in bioinformatics. It studies biological sequences at genome level such as DNA and RNA. (Mathe´ et al., 2002) provide a review on some important applications which are locating the genes in a genome and identifying its function. Ensemble methods have been applied to predict gene function in comparison with

single classifier as in (Re & Valentini, 2010), where several data sources are integrated then input to SVM base classifiers and combined using weighted average and decision templates. The ensembles outperform the single SVM classifier. Sequence information is also used for gene function and RNA structure prediction (Freyhult, 2007) as well as many other relevant genomics problems.

Gene expression data analysis is a well-established bioinformatics domain where Machine Learning methods for classification and clustering have been widely applied. *DNA gene expression microarrays* allow biologists to study genome-wide patterns of gene expression in any given cell type, at any given time, and under any given set of conditions (Baldi & Brunak, 2001). Gene expression data is arranged into a matrix where, columns represent genes and rows represent the samples. Each element in the matrix represents the expression level of a gene under a specific condition and it is represented by a real number.

The use of these arrays produces large amounts of data, potentially capable of providing fundamental insights into biological processes ranging from gene function to development, cancer, aging and pharmacology (Baldi & Brunak, 2001). However the data needs to be pre-processed first, i.e. modified to be suitably used by machine learning algorithms. Then the data is analyzed to look for useful information.

Clustering techniques such as *k*-means, hierarchical clustering (Eisen et al., 1998) and self-organizing maps (SOMs) (Tamayo et al., 1999) have been applied to identify genes according to their function similarities. These methods assume that related genes have similar expression patterns across all samples and hence divide the set of genes into disjoint groups. Accordingly, identifying local patterns with subset of genes that are similarly expressed over a subset of samples is difficult using traditional clustering techniques. (AboHamad et al., 2010) propose a bi-clustering technique which is based on clustering similarly expressed genes set over a subset of samples simultaneously. On the other hand, many classification techniques are used. The majority of papers published in the area of machine learning for genomic medicine deal with analyzing gene expression data coming from DNA microarrays, consisting of thousands of genes for each patient, with the aim to diagnose (sub) types of diseases and to obtain a prognosis which may lead to individualized therapeutic decisions (Bellazi & Zupan, 2008). The published papers are mainly related to oncology, where there is a strong need for defining individualized therapeutic strategies (Mischel & Cloughesy, 2006). A seminal paper from this area is that of (Golub et al., 1999) and focuses on the problem of the early differential diagnosis of acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL). Several classification techniques have been applied on different benchmark datasets, among these are decision trees, naïve bayes classifier, multilayer perceptron and SVMs which have proved to be very effective in such applications. The mentioned classification approaches are usually coupled with feature (gene) selection methods to improve the performance. To avoid removing some features, the use of ensembles has emerged with different ways of distributing features among subsets as explained in section 2.

Proteomics is the field that studies proteins. Proteins transform the genetic information into actions performed in life. The prediction of the secondary and tertiary structure of proteins represents one of the main challenges for Machine Learning methods in bioinformatics. Neural networks have been applied to predict protein secondary structure (Baldi & Brunak, 2001). This is due to the fact that proteins are very complex macromolecules with thousands of atoms and bounds so there are huge number of possible structures. This makes protein structure prediction a very complicated combinatorial problem where optimization

techniques are required. Machine learning is also applied for protein function prediction, fold recognition as well as other relevant proteomics problems (Valentini, 2008).
Systems biology is an emerging bioinformatics area where Machine Learning techniques play a central role (Kitano, 2002). It is concerned with modelling biological processes inside the cell. Mathematical models and learning methods are required to model the biological networks ranging from genetic networks to signal transduction networks to metabolic pathways (Bower et al., 2004).
Phylogenetic trees are schematic representations of organisms' evolution. Machine Learning is applied for phylogenetic tree construction by comparisons made by multiple sequence alignment where many optimization techniques are used (Larranaga et al., 2005).
As follows we present a novel machine learning model for micro-array data classification. Experiments and comparative results demonstrate the efficiency of these models to deal with high-dimensional DNA data.

4. A case study of an SVM ensemble using feature subset selection for DNA classification

In this section, we present a case study of a SVM ensemble that uses SVM base classifiers and another SVM classifier for combining the results of the base classifiers to get the final classification. The proposed ensemble uses *k*-means clustering for grouping the features into subsets. The ensemble is referred to as *k*-means-SVM fusion throughout the rest of this chapter.
The flow charts in figures 2, 3 and 4 illustrate the main phases used for building the ensemble. A dataset consisting of a set of labelled examples is initially given. Each example is characterized by a set of features and a label indicating its class. The dataset is divided into a training set, a validation set and a test set. For clustering, the features are grouped into *k* feature subsets and *k*-means clustering is applied to the training set. Then, each of the SVM base classifiers is trained using the training set characterized by features of a single feature subset. The SVM classifier responsible for fusion is trained using the validation set and then the ensemble is ready to be tested on the test set.
Figure 5 presents the steps of the algorithm for building the ensembles that use an SVM classifier for fusion in more details. Initially, the available data set Z containing all features is divided into training Z_{Train}, validation Z_{Valid} and testing set Z_{Test}. The training set Z_{Train} is used for building the base SVM classifiers and determining their parameters through cross validation by further splitting the training set into training part and validation part, applying grid search using range of values for the parameters and selecting the parameter values that resulted in the best accuracy among the validation set. The validation set Z_{Valid} is used to train the combiner SVM, while the test set Z_{Test} is used to evaluate the overall ensemble.
Before any of the data parts are applied, we first perform a feature subset selection procedure to choose *k* subsets to be used as an input for each base classifier. The input to any base SVM classifier *i* is hence the training samples with features in the cluster *i*. After the base classifiers are trained using the training portion Z_{Train}, the combiner is trained using the validation set Z_{Valid} as follows:
The outputs of the *k* SVMs are collected to form a new feature-sample training matrix for the SVM combiner where each sample is characterized by the base SVM outputs as features and its label is the same as its original labels.
In the test phase, the overall accuracy of the ensemble is tested using the remaining samples of Z_{Test} reserved.

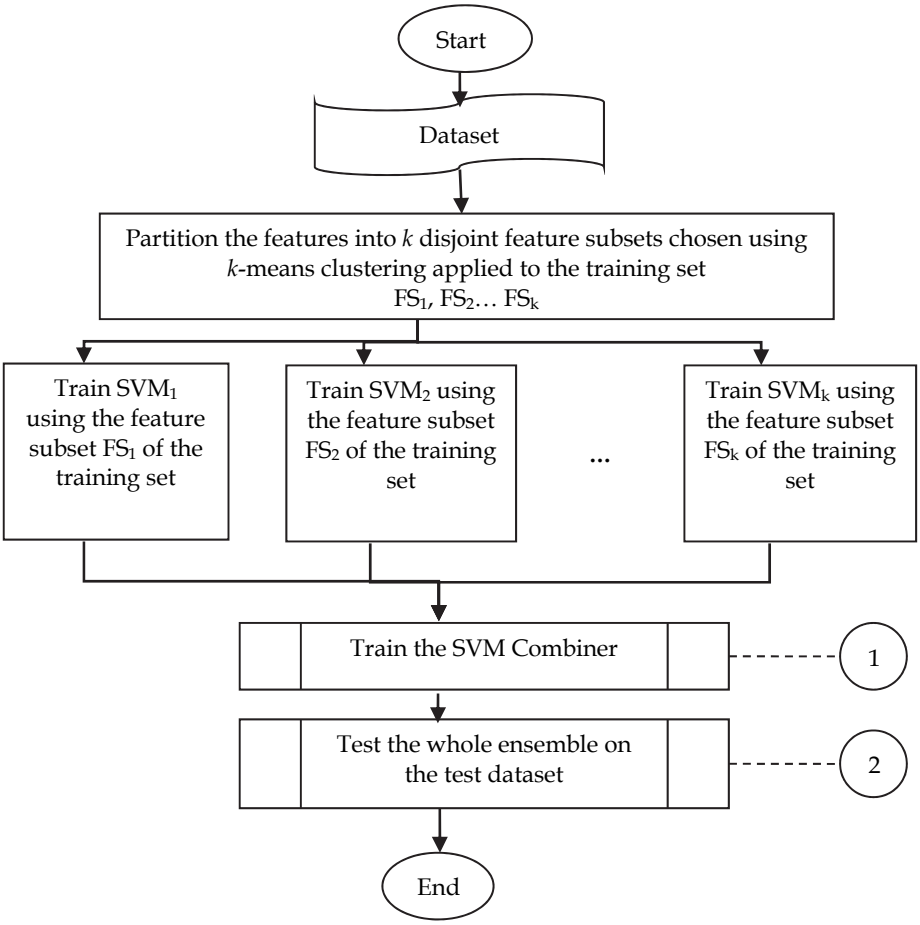


Fig. 2. Ensembles using SVM fusion.

Cross validation is used to evaluate the proposed model using different partitions of Z for training and validation.

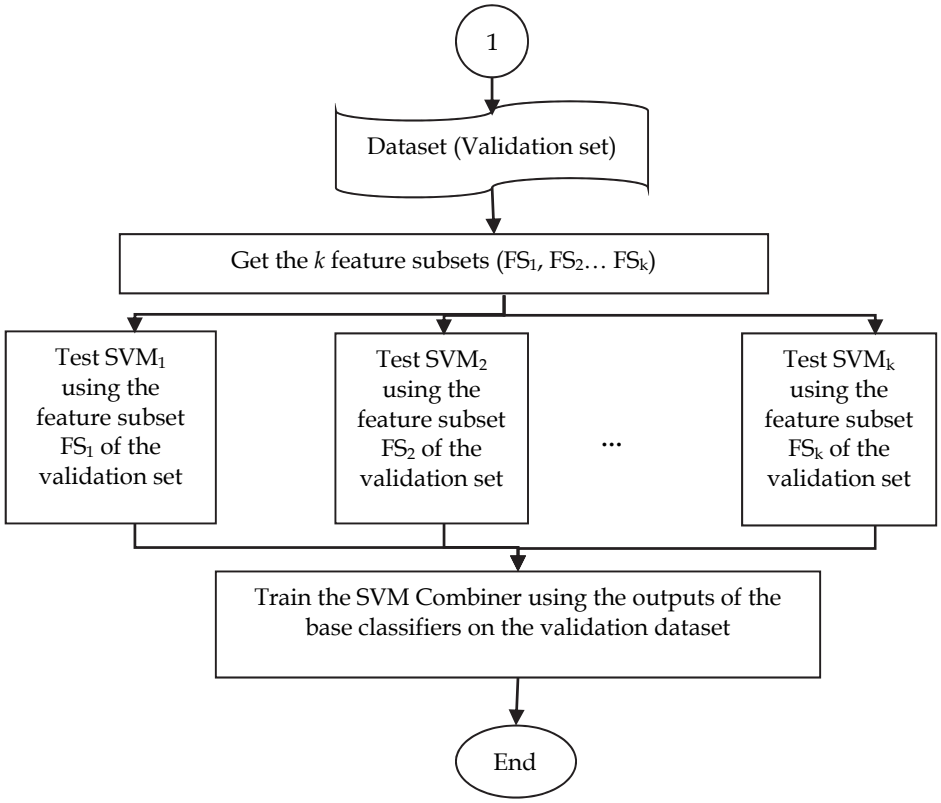


Fig. 3. Training the SVM Combiner in the Ensemble.

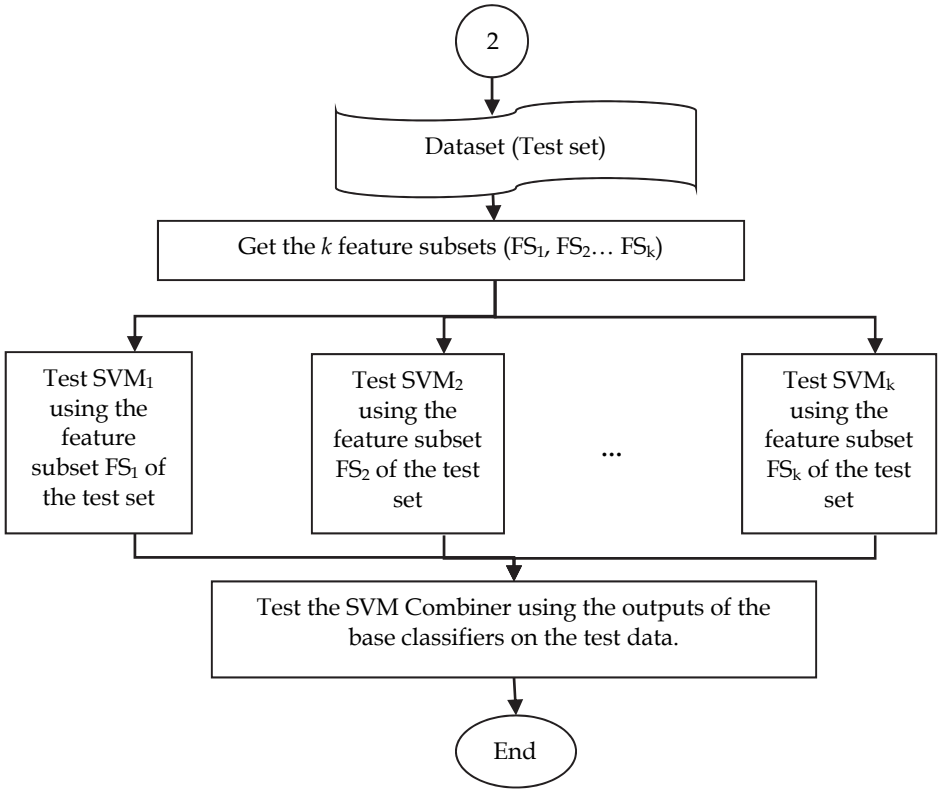


Fig. 4. Testing the ensemble with SVM fusion.

5. Data and experimental setup

In this section, we describe the experiments conducted to test and evaluate the proposed the *k*-means-SVM fusion ensemble on the Leukaemia data set. Refer to (Ahmed et al., 2010) for more experiments on other data sets.

The Leukemia dataset is a benchmark micro-array dataset which consists of 72 samples, 7129 features and 2 classes (AML and ALL). The 72 samples consist of 47 samples of Acute Lymphoblastic Leukemia (ALL) and 25 samples of Acute Myeloblastic Leukemia (AML). The training and test samples in (Chang & Lin, 2001), (Golub et al., 1999) are merged then normalized as indicated in (Shevade & Keerthi, 2003).

The proposed *k*-means-SVM fusion ensemble is compared to a single SVM classifier as well as three different SVM ensembles: the *random-majority vote* ensemble, the *k-means-majority vote* ensemble and the *random-SVM fusion* ensemble. The *random-majority vote* ensemble uses the random subspace method to select feature subsets and distribute them among base classifier in the ensembles. A fixed fusing rule –majority vote- is used to combine the output of the base classifiers. In contrast, the *k-means-majority vote* ensemble uses feature subsets resulting from k-means clustering as described in section 4 but still uses majority voting for

classifier fusion. Alternatively, the *random-SVM fusion* ensemble uses Random subspace method for feature subset selection and a trainable SVM classifier as a combination rule.

Given:
- Z, a set of N crisp labelled samples x.
- FS: $(1 \rightarrow n)$, a set containing all n features.

Step 1: Choose feature subsets using *k*-means clustering
- k-means on dataset Z_{Train}
  - Initialize k, the number of clusters and the number of the base classifiers.
  - Cluster analysis on features using Z_{Train} using k-means algorithm.
  - Get \$k\$ disjoint feature subsets FS_1, FS_2, … , FS_K.

Step 2: Train the Base Classifiers
- Train the base classifiers using Z_{Train} such that each base classifier SVM_i is trained with feature subset FS_i.

Step 3: Train the Combiner
- Test every $SVM\ (1 \rightarrow k)$ base classifiers using Z_{Valid} with the corresponding feature subset $FS(1 \rightarrow k)$.
- Train the SVM(combiner) using the outputs of the $SVM(1 \rightarrow k)$ on the validation data set Z_{Valid}.

Step 4: Test the ensemble
- Test the ensemble using the Test data set Z_{Test} as follows:
  - Z_{Test} is passed through the $SVM\ (1 \rightarrow k)$ base classifiers.
  - Z_{Test} with the outputs of the SVM base classifiers as features is given to the SVM(combiner).
  - SVM(combiner) classifies the samples of Z_{Test}.

Fig. 5. Ensembles using SVM fusion.

The suggested ensembles are tested for different number of feature subsets, i.e different number of base classifiers. Experiments are repeated for number of feature subsets $k = 2^n\text{-}1$ where n = 2, 3, 4, 5, 6. Results are compared using different measures of performance including accuracy, sensitivity, specificity and precision. For the sake of brevity we only present results based on accuracy and sensitivity.

Cross validation is used to obtain different training, test and validation sets. Since DNA microarrays are characterized by having a very small number of samples, the training and validation sets are overlapped with 1/3 of the samples.

The usage of cross validation differs according to the ensemble model. For the ensembles that use majority vote combiner, the dataset is divided into a training set and a test set. The training set is used to train the base classifiers while the test set is used to test the base classifiers then the majority vote is applied to their outputs. For tuning the parameters of the base classifiers, the training set is further split into a training set and a validation set on which the classifier is validated. Grid search using range of values for the parameters is applied and the parameter values that get the best performance on the validation set are chosen for the base classifiers.

For the ensembles that use a SVM for fusion, the dataset is divided into a training set, validation set and a test set using k-fold cross validation. The training set is used to train the base classifiers then the validation set is used to test the base classifiers. The outputs of the

base classifiers are then used in the training of the SVM combiner. The test set is then used to test the whole ensemble.

All experiments are performed using LibSVM (Chang & Lin, 2001) using 5 fold cross validation for training the base classifiers and the combiner. *K*-means is applied with values for *k* ranging from 3 to 63 with increment of 2. Linear kernels are chosen for the SVM base classifiers as well as for the SVM combiner as it was found in literature that they are suitable for the high dimensional microarray datasets (Chang & Lin, 2001), (Bertoni et al., 2005). For each SVM with linear kernel, parameter *C* requires to be optimized. This is done using grid search by 5 fold cross validation.

We experimented with exponentially growing sequences of C in the rage of -15 to 15 to identify a good value for the parameter.

6. Results and discussions

Figure 6 compares the accuracy of the *k-means-SVM fusion* ensemble to the *random-majority vote* ensemble, the *k-means-majority vote* ensemble and the *random-SVM fusion* ensemble for different number of feature subsets (i.e. different number of base classifier or ensemble sizes).

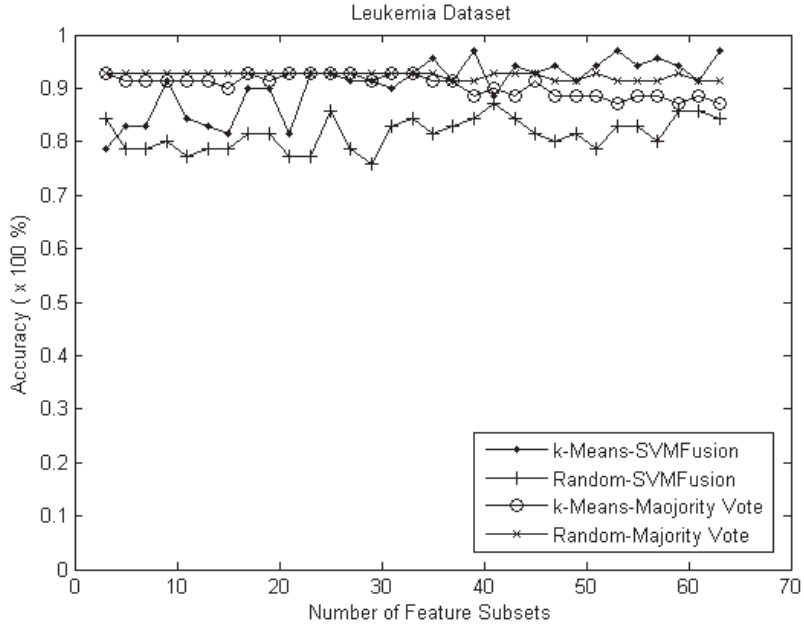


Fig. 6. Test accuracies of the four ensembles with respect to the number of feature subsets.

The accuracy of the *K*-Means-SVM fusion ensemble seems to outperform the other models with growing number of feature subsets (i.e. with increased number of base classifiers). Also, the accuracy of random-SVM fusion increases with higher number of feature subsets. However, it still has the lowest accuracy among the four ensembles.

On the other hand, for the *k-Means-majority vote* ensemble, increasing the number of feature subsets results in a drop of the accuracy; while for *random-majority vote* ensemble results are not affected by the change of the number of feature subsets.
Table 1 summarizes the best results obtained for each ensemble across the different number of feature subsets in addition to those obtained using a single SVM classifier. The number of feature subsets at which the best results are obtained is mentioned for each ensemble. It can be noticed that the ensembles that use majority vote combiner work well only with small number of feature subsets while those that use an SVM classifier for fusion need a large number of feature subsets.
Results reveal that the *k-means-SVM fusion* ensemble outperforms *k-means-majority vote* ensemble as well as *random-SVM fusion* and *random-majority vote* ensembles. *K-means-SVM fusion ensemble* also shows to have a high sensitivity with respect to the other ensembles in the comparison.
Since for the leukemia dataset, both classes are patients, there is no *positive* or *negative* class. Accordingly, the sensitivity is calculated twice; at first, considering AML as the *positive* class and then considering ALL as the *positive* class. The average of both is then calculated.

Classifier/Ensemble	Accuracy	Sensitivity (Average)
Single SVM classifier	92.86 ± 15.97	81.68
Random-Majority Vote (3)	92.86 ± 15.97	90.00
Random-SVM Fusion (41)	87.14 ± 19.17	84.67
K-Means-Majority Vote (3)	92.86 ± 15.97	90.00
***K*-Means-SVM Fusion (63)**	**97.14 ± 3.92**	**96.89**

Table 1. Best classification accuracy and sensitivity measures obtained by applying the ensembles and the single SVM classifier to the leukemia dataset. The number of feature subsets at which the best results are obtained are mentioned between brackets.

Figures 7-10 illustrate for each ensemble the improvement of the combined model over the average performance of the base classifiers. The figures show the average accuracies of the base classifiers of each ensemble compared to the ensemble accuracies. In addition the ratio of the ensemble accuracies to those of the base classifiers are depicted. It is obvious that the *k-means-SVM fusion* has the best ratio among the four ensembles.
Figure 7 shows the results for the *k-means-majority vote* ensemble. It can be noticed that the ensemble improves the performance of the base classifiers but its accuracy drops with higher number of the feature subsets. So, it works better with small number of feature subsets. Figure 8 demonstrates the performance of the *k-means-SVM fusion ensemble*. Clearly the ensemble enhances the performance of the base classifiers except when using 3 feature subsets. Unlike the *k*-means-majority vote ensemble, its performance does not drop with increased number of feature subsets. *K-means-SVM fusion* ensemble achieves the best accuracy among the four ensembles when using 63 feature subsets. Figure 9 summarizes the performance of the *random-majority vote* ensemble. It is noticed that it has a slight improvement over the average performance of the base classifiers resulting in a nearly constant behaviour. Figure 10 demonstrates that *random-SVM fusion* ensemble does not

work well on using a small number of feature subsets but as the number of feature subsets increase, it improves the performance and become better than the average performance of the base classifiers.

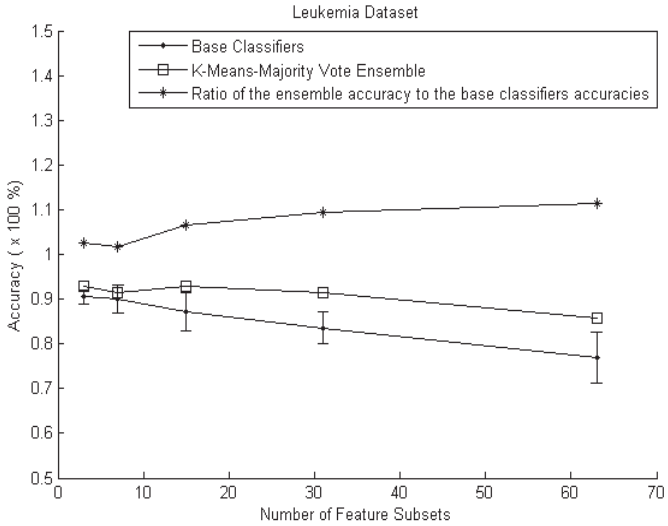


Fig. 7. Results of the k-means-majority vote ensemble.

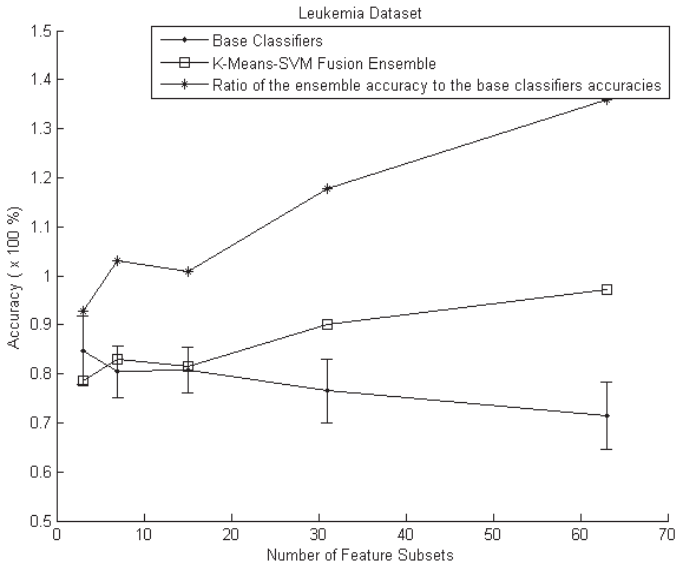


Fig. 8. Results of the k-means-SVM fusion ensemble.

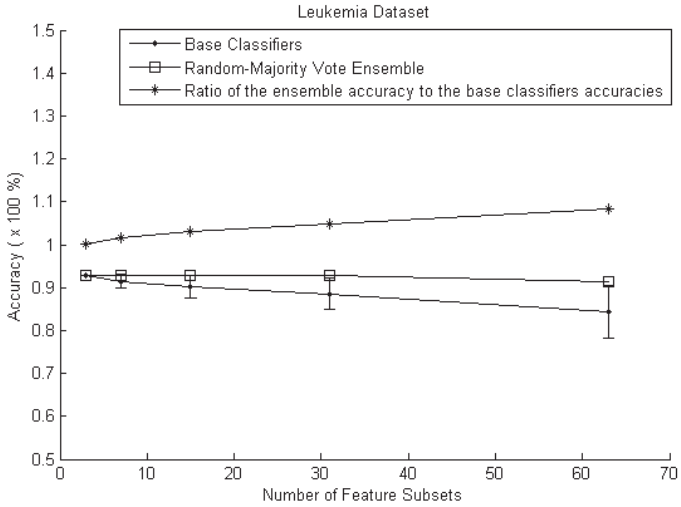


Fig. 9. Results of the random-majority vote ensemble.

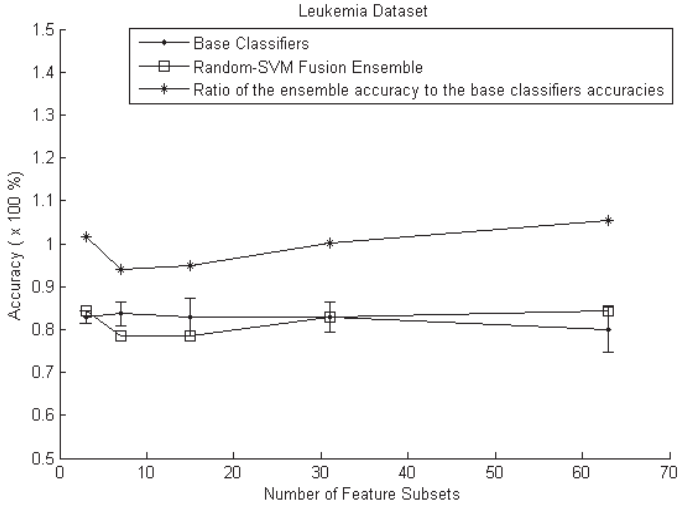


Fig. 10. Results of random-SVM fusion ensemble.

As a general conclusion of the previous experiments we can state that the ensembles with SVM classifier as base classifiers generally improve the classification accuracy over single classifiers. Ensembles that use an SVM classifier for fusion outperform those that use majority vote as a combiner when using a reasonably large number of feature subsets and base classifiers.

According to the study on the leukemia dataset, *k*-means-SVM fusion ensemble performs the best among the four ensembles with regards to both accuracy and sensitivity. More results to confirm this conclusion are reported in (Ahmed et al., 2010).

7. Conclusions and future directions

This chapter presents a broad introduction to machine learning and focuses on the classification problem in bioinformatics. In particular we cover main terminologies from the pattern recognition, machine learning and data mining fields. We try to review main models used for classification and to elaborate on classifier testing and evaluation techniques. We devote a special attention to SVM, ensemble techniques and feature subset ensembles as they are the base of our proposed DNA micro-array data classification model. The proposed classification model exploits the use of powerful machine learning models such as SVMs and ensemble methods coupled with feature subset selection. The proposed approach proves to be able to deal with data challenges that are imposed by this application which is mainly the huge number of features and the small samples size.

Results are shown on the leukemia dataset and compared to four different models. The study concludes that the use of ensembles is very fruitful in such applications. The way of distributing the features among subsets affects the performance of the ensemble. *K*-means is a systematic way that proved to be suitable for clustering the features into subsets especially when used with a SVM classifier for combination. For the leukemia dataset, *k*-means-SVM fusion ensemble performed the best with respect to accuracy and sensitivity. The study confirms the importance of ensembles in bioinformatics applications and highlights that the coupling between the method of distributing the features among subsets and the combination method is crucial for obtaining good results.

Different method can be investigated for distributing the features among subsets. Higher numbers of base classifiers / numbers of feature subsets can be experimented with. Time complexity of the proposed models need to be calculated and accessed. The use of other combiners especially classifiers are worth investigating. In addition, the use of the proposed models can be extended to other data sets and other domains in the bioinformatics field.

8. Acknowledgment

This work was supported by DFG (German Research Society) grants SCHW 623/3-2 and SCHW 623/4-2.

9. References

Abe,S. (2005). Support Vector Machines for Pattern Classification, *Springer*, ISBN 1-85233-929-9.

Abohamad, W., Korayem, M. & Moustafa,K. (2010), Biclustering of DNA Microarray Data Using Artificial Immune System, *Proceedings of International Conference on Intelligent Systems Design and Applications (ISDA),* pp. 1223-1228, Cairo, Egypt.

Ahmed, E., El-Gayar, N. & El-Azab, I.A. (2010). Support Vector Machine Ensembles Using Features Distribution among Subsets for Enhancing Microarray Data Classification, *Proceedings of International Conference of Systems and Design (ISDA)*, Cairo, Egypt, December, 2010.

Alpydin , E. (2004). *Introduction to Machine Learning*. The MIT Press, ISBN 0-262-01211-1.

Baldi,P.& Brunak,S. (2 ed). (2001). *Bioinformatics The Machine Learning Approach*. MIT Press, ISBN 0 – 262 – 02506 – X.

Bellazi, R. & Zupan, B. (2008). Predictive data mining in clinical medicine: Current issues and guidelines, *INTERNATIONAL JOURNAL OF MEDICAL INFORMATICS*, Vol. 7, pp. (81 – 97).

Bernal, A., Crammer, K., Hatzigeorgiou, A. & Pereira, F., (2007). Global discriminative learning for higher-accuracy computational gene prediction, *PLoS Computational Biology*, Vol. 3, No. 3.

Bertoni, A., Folgieri, R. & Valentini, G. (2005). Bio-molecular cancer prediction with random subspace ensembles of support vector machines, *Neurocomputing*.

Bower, J. & Bolouri, H. (2004). Computational Modeling of Genetic and Biochemical Networks, *MIT Press*.

Brent, M. & Guigo, R. (2004). Recent advances in gene structure prediction, *Current Opinion in Structural Biology*. Vol. 14, No. 3, pp.(264–272).

Chang, C.-C. & Lin, C.-J. (2001). *LIBSVM: a library for support vector machines*. Available from http://www.csie.ntu.edu.tw/_cjlin/libsvm.

Duda, R. O., Hart, P. E. & Stork, D. G. (2nd ed). (2000). *Pattern Classification*, John Wiley & Sons.

Eisen, M. B., Spellman, P. T., Brown, P. O. & Botstein, D.(1998).Cluster analysis and display of genome-wide expression patterns, *in Proc. Natl. Acad. Sci, National Acad Sciences*, Vol. 95, No. 25., pp. (14 863–14 868), USA.

Freyhult, E. (2007). *A Study in RNA Bioinformatics, Identication, Prediction and Analysis*. PhD thesis, ACTA Universitatis Upsaliensis Uppsala.

Goble, C. & Stevens, R. (5 August 2008). State of the nation in data integration for bioinformatics. *Journal of Biomedical Informatics (in press)*, available on line at http://www.sciencedirect.com

Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C. Gaasenbeek, M., Mesirov, JP., Coller, H., Loh, M.L., Downing, J.R.,Caligiuri, M.A., Bloomfield, C.D. & Lander, E. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, *Science*, pp. (531 – 537).

Hand, D., Mannila, H. & Smyth, P. (2001). *Principles of Data Mining*, MIT Press.

Handl, J., Kell, D. & Knowles, J. (2007). Multiobjective optimization in bioinformatics and computational biology, *IEEE/ACM Trans. Comput. Biol. Bioinformatics,* Vol. 4, No. 2, pp. (279–292).

Ho, T. K. (1998).The random space method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 8, pp. (832–844).

Holloway, D., Kon, M. & DeLisi, C. (2007). Machine learning for regulatory analysis and transcription factor target prediction in yeast, *Systems and Synthetic Biology*, Vol. 1, No. 1, pp. (25–46).

KDnuggets , Polls, *Data Mining Methods* (Apr 2006) Available from: http://www.kdnuggets.com/polls/2006/data_mining_methods.htm

Kitano, H. (2002). Systems biology: A brief overview, *Science*. Vol. 295, No. 5560, pp.(1662 – 1664).

Kittler, J., Hatef, M., Duin, R.P.W. & Matas, J. (1998). On Combining Classifiers. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 3, pp. (226-239).

Krallinger, M., Erhardt, R.A. & Valencia, A. (2005). Text-mining approaches in molecular biology and biomedicine. *Drug Discovery Today*, Vol. 10, No.6, pp. (439–45).

Kuncheva, L. I. (2004). *Combining Pattern Classifiers: Methods and Algorithms*. John Wiley Sons, Inc, ISBN 0-471-21078-1.

Lander, E.S. (1996). The new genomics global views of biology. *Science*, Vol. 274, No. (5287), pp.(536 – 539), (October 1996).

Larranaga,P., Calvo, B., Santana, R., Bielza, C., Galdiano, J., Inza, I., Lozano, J. A., Armananzas, R., Santafe, G., Perez, A. & Robles, V. (2005). Machine learning in bioinformatics, *Briefings in bioinformatics*, Vol. 7, No. 1, pp. (86-112).

Lopez-Bigas, N. & Ouzounis, C. (2004). Genome-wide identification of genes likely to beinvolved in human genetic diseases, *Nucleic Acid Research*, Vol. 32, No. 10, pp. (3108 – 3114).

Mathe´, C., Sagot, M-F and Schlex, T. (2002). Current methods of gene prediction, their strengths and weaknesses. *Nucleic Acids Research*. Vol. 30, No. 19, spp. (4103–4117).

(MCS series) Multiple Classifier Systems. Lecture Notes in Computer Science, *Springer Verlag*, Vols. 1857 (2000), 2096 (2001), 2364 (2002), 2709 (2003), 3077 (2004), 3541 (2005), 4472 (2007), 5519 (2009), 5997 (2010), 6713 (2011).

Mischel, P.S., Cloughesy, T. (2006). Using molecular information to guide brain tumor therapy, *Nat. Clin. Pract. Neurol.* Vol.2, pp. (232–233).

Pomeroy, S.L., Tamayo, P., Gaasenbeek, M., Sturla, L.M., Angelo, M., McLaughlin, M.E., Kim, J.Y., Goumnerova, L.C., & et al. (2002). Prediction of central nervous system embryonal tumour outcome based on gene expression, *Nature*, Vol. 415, pp. (436 - 442).

Ratsch, G., Sonnenburg, S., Srinivasan, J., Witte, H., Mller, K.-R., Sommer, R.-J. & Scholkopf, B. (2007). Improving the Caenorhabditis elegans genome annotation using machine learning, *PLoS Computational Biology* , Vol. 3, No. 2.

Re, M. & Valentini, G. (2010). Prediction of Gene Function Using Ensembles of SVMs and Heterogeneous Data Sources. Applications of supervised and unsupervised ensemble methods, *Computational Intelligence Series*, Springer, Vol.245, pp. (79-91).

Ritchie, M., White, B.C., Parker, J.S., Hahn, L.W. & Moore, J.H. (2003). Optimization of neural network architecture using genetic programming improves detection and modelling of gene-gene interactions in studies of human diseases, *BMC Bioinformatics,* Vol. 4, No. 28.

Roli, F. & Giacinto, G. (2002). *Design of Multiple Classifier Systems, HYBRID METHODS IN PATTERN RECOGNITION* , H Bunke and A Kandel (Eds.) , World scientific.

Shevade, S. K. & Keerthi, S. S. (2003). A simple and efficient algorithm for gene selection using sparse logistic regression, *Bioinformatics*, Vol. 19, No.17, pp. (2246 – 2253).

Shipp, M.A., Ross, K.N., Tamayo, P., Weng, A.P., Kutok, J.L., Aguiar, R.C., Gaasenbeek, M., Angelo, M. & et al. (2002). Diffuse large B-cell lymphoma outcome prediction by gene expression profiling and supervised machine learning, *Nat. Med.* Vol. 8, pp. (68–74).

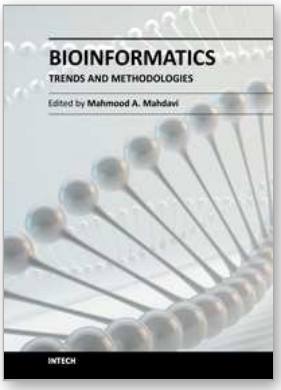
Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E. , Lander, E. S. & Golub, T. R. (1999). Interpreting patterns of gene expression with self-organizing

maps: Methods and application to hematopoietic differentiation, *in Proc. Natl. Acad. Sci. USA, National Acad Sciences*, Vol. 96, No. 6, pp. (2907–2912).

Valentini, G. (August 2008). Guest editorial computational intelligence and machine learning in bioinformatics, *Preprint submitted to Elsevier*.

Van't 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., et al., (2002). Gene expression profiling predicts clinical outcome of breast cancer, Nature, Vol. 415, pp. (530–536).

Won K. -J, Pru¨gel-Bennet A. & Krogh A. (2004). Training HMM structure with genetic algorithm for biological sequence analysis. *Bioinformatics*;Vol. 20, No. 18, pp.(3613–3619).



Bioinformatics - Trends and Methodologies
Edited by Dr. Mahmood A. Mahdavi

ISBN 978-953-307-282-1
Hard cover, 722 pages
Publisher InTech
Published online 02, November, 2011
Published in print edition November, 2011

Bioinformatics - Trends and Methodologies is a collection of different views on most recent topics and basic concepts in bioinformatics. This book suits young researchers who seek basic fundamentals of bioinformatic skills such as data mining, data integration, sequence analysis and gene expression analysis as well as scientists who are interested in current research in computational biology and bioinformatics including next generation sequencing, transcriptional analysis and drug design. Because of the rapid development of new technologies in molecular biology, new bioinformatic techniques emerge accordingly to keep the pace of in silico development of life science. This book focuses partly on such new techniques and their applications in biomedical science. These techniques maybe useful in identification of some diseases and cellular disorders and narrow down the number of experiments required for medical diagnostic.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821