7

Feature Selection and Dimensionality Reduction in Genomics and Proteomics

Milos Hauskrecht[1,2,3], Richard Pelikan[2], Michal Valko[1], and James Lyons-Weiler[3]

[1] Department of Computer Science, University of Pittsburgh, Pittsburgh, PA 15260, USA.
[2] Intelligent Systems Program, University of Pittsburgh, Pittsburgh, PA 15260, USA.
[3] Department of Biomedical Informatics, University of Pittsburgh, Pittsburgh, PA 15260, USA.
milos@cs.pitt.edu, pelikan@cs.pitt.edu, michal@cs.pitt.edu, lyonsweilerj@upmc.edu

7.1 Introduction

As technology improves, the amount of information we collect about the world increases. Sensor networks collect traffic or weather information in real-time, documents and news articles are distributed and searched on-line, information in medical records is collected and stored in electronic form. All of this information can be mined so that the relations among components of the underlying systems are better understood and their models can be built. Microarray and mass spectrometry (MS) technologies are producing large quantities of genomic and proteomic data relevant for our understanding of the behavior and function of an organism, or characteristics of disease and its dynamics. Thousands of genes are measured in a typical microarray assay; tens of thousands of measurements comprise a mass spectrometry proteomic profile. The high-dimensional nature of the data demands the development of special data analysis procedures that are able to adequately handle such data. The central question of this process becomes the identification of those *features* (measurements, attributes) that are most relevant for characterizing the system and its behavior. We study this problem in the context of classification tasks where our goal is to find features that discriminate well among classes of samples, such as samples from people with and without a certain disease.

Feature selection is a process that aims to identify a small subset of features from a large number of features collected in the data set. Two closely-related objectives may drive the feature selection process: (1) Building a reliable classification model which discriminates disease from control samples with high accuracy. The model is then applied to early detection and diagnosis of the

disease. (2) Biomarker discovery task where a small set of features (genes in DNA microarrays, or peaks in proteomic spectra) that discriminate well between disease and control groups is identified so that the responsible features can be subjected to further laboratory exploration.

In principle, building a good classification model does not require feature selection. However, when the sample size is small in comparison to the number of features, feature selection may be necessary before a classification model can be reliably learned. With a small sample size, the estimates of parameters of the model may become unreliable and may cause *overfitting*, a phenomenon in which each datum is fit so rigidly that the model lacks flexibility for future data. To avoid overfitting, feature selection is applied to balance the number of features in proportion to the sample size. On the other hand, identification of a small panel of features for biomarker discovery purposes requires a classification model so that the discriminative behavior of the panel can be assessed.

The dimensionality of typical genomic and proteomic data sets one has to analyze surpasses the number of samples collected in typical studies by a large margin. For example, a typical microarray study can consist of up to a hundred samples with thousands of gene-expression measurements. Mass spectrometry (MS) proteomic profiling is less expensive and as a result one can often see data sets with two to three hundred profiles. MS profiles consist of thousands of measurements. Typically, "peaks" are selected among those measurements, and number in the hundreds. In either case, feature selection becomes important for both the biomarker discovery and interpretive analysis tasks; one has to seek a robust combination of feature selection methods and classification models to assure their reliability and success. Finally, feature selection may be a one-shot process, but typically, it is a search problem where more than one feature subset is evaluated and compared. Since the number of possible feature subsets is exponential in the number of constituent features, efficient feature selection methods are typically sought.

Feature selection methods are typically divided into three main groups: *Filter*, *wrapper* and *embedded methods*. Filter methods rank each feature according to some univariate metric, and only the highest ranking features are used; the remaining features are eliminated. Wrapper algorithms (Kohavi and John, 1998) search for the best subset of features. To assess the quality of a feature set, these methods rely on and interact with a classification algorithm and its ability to discriminate among the classes. The wrapper algorithm treats a classification algorithm as a black box, so any classification method can be combined with the wrapper. Standard optimization techniques (hill climbing, simulated annealing or genetic algorithms) can be used. Embedded methods search among different feature subsets, but unlike wrappers, the process is tied closely to a certain classification model and takes advantage of its characteristics and structure. In addition to feature selection approaches, in which a subset of original features is searched, the dimensionality problem can be often resolved via *feature construction*. The process of feature construc-

tion builds a new set of features by combining multiple existing features with the expectation that their combination improves our chance to discriminate among the classes as compared to the original feature space.

In this chapter, we first introduce the main ideas of four different methods for feature selection and dimensionality reduction and describe some of their representatives in greater depth. Later, we apply the methods to the analysis of one MS proteomic cancer data set. We analyze each method with respect to the quality of features selected and stress differences among the methods. Since our measuring criterion for feature effectiveness is how well it allows us to classify our samples, we compare the methods and their classification accuracy by combining them with a fixed classification method — a linear support vector machine (Vapnik, 1995). In closing, we analyze the results and give recommendations on the methods.

7.2 Basic Concepts

7.2.1 Filter Methods

Filter methods perform feature selection in two steps. In the first step, the filter method assesses each feature individually for its potential in discriminating among classes in the data. In the second step, features falling beyond some thresholding criterion are eliminated, and the smaller set of remaining features is used. This score-and-filter approach has been used in many recent publications, due to its relative simplicity. Scoring methods generally focus on measuring the differences between distributions of features. The resulting score is intended to reflect the quality of each feature in terms of its discriminative power. Many scoring criteria exist. For example, in the Fisher score (Pavlidis et al., 2001),

$$V(i) = \frac{(\mu_{(+)}(i) - \mu_{(-)}(i))^2}{\sigma^2_{(+)}(i) + \sigma^2_{(-)}(i)} \tag{7.1}$$

the quality of each feature is expressed in terms of the difference among the empirical means of two distributions, normalized by the sum of their variances. Table 7.1 displays examples of scoring criteria used in bioinformatics literature. Note that some of the scores can be applied directly to continuous quantities, while others require discretization. Scores can be limited to two classes, like the Fisher score, while others, such as the mutual information score, can be used in the presence of three or more classes. For the remainder of this chapter, we will assume our scoring metrics deal with binary decisions, where the data either belong to a positive (+) or negative (−) group.

7.2.1.1 Criteria Based on Hypothesis Testing

Some of the scoring criteria are related to statistical hypothesis testing and significance of their results. For example, the t-statistic is related to the null

Table 7.1. Examples of univariate scoring criteria for filter methods. See section Mathematical Details for definitions of these scores.

Criterion	References
Fisher score	(Golub et al., 1999; Furey et al., 2000; Pavlidis et al., 2001)
SAM scoring criterion	(Tusher et al., 2001; Storey and Tibshirani, 2003)
t-test	(Baldi and Long, 2001; Gosser, 1908)
Mutual information	(Tzannes and Noonan, 1973)
χ^2 (Chi square)	(Chernoff and Lehmann, 1954; Liu and Setiono, 1995)
AUC	(Hanley and McNeil, 1982)
$J5$ score	(Patel and Lyons-Weiler, 2004)

hypothesis H_0 under which the two class-conditional distributions $p(x|y = (+))$ and $p(x|y = (-))$ have the identical mean, that is $\mu_{(+)} = \mu_{(-)}$. The degree of violation of H_0 is captured by the p-value of the t-statistic with respect to the Student distribution. As a result, features can be ranked using the inverse of their p-value. Similarly, one can rank the features according to the inverse of the p-value of the Wilcoxon rank-sum test (Wilcoxon, 1945), a nonparametric method, testing the null hypothesis that the class-conditional densities of individual features are equal.

7.2.1.2 Permutation Tests

Any differential scoring metric (statistic) can be incorporated into and evaluated within the hypothesis testing framework via permutation tests. Permutation (or randomization) tests define a class of non-parametric techniques developed in the statistics literature (Kendall, 1945; Good, 1994), that are used to estimate the probability distribution of a statistic under the null (random) hypothesis from the available data. The estimate of the probability distribution of a scoring metric (Fisher score, J-measure, t-score, etc.) under the null condition allows us to estimate the p-value of the score observed in the data, similarly to the t-test or Wilcoxon rank-sum test. From the viewpoint of feature selection, the null hypothesis assumes that the conditional probability distributions for the two classes ($y = (+)$ or $(-)$) are identical under a feature x, that is, $p(x|y = (+)) = p(x|y = (-))$; or equivalently, that the data and the labels are independent, $p(x, y) = p(x)p(y)$. The distribution of data under the null hypothesis is generated through random permutations (of labels) in the data. The *permutation test algorithm* is shown below. The main cycle of the algorithm either scans through all possible permutations of

labels, or, if this set is too large, a large number B of permutations is generated randomly. With sufficient cycles, the distribution of the test statistic under the null hypothesis can be estimated reliably.

```
permutation_test
{
  Compute the test statistic T for the original data;
  For b = 1 to B do
        {Permute randomly the group labels in the data;
         Compute the test statistic T_b for the modified data;
        }
  Calculate the p-value of T with respect to the distribution defined by
        permutations b as: p = N_{T_b ≥ T}/B;  where N_{T_b ≥ T} is the number of
        permutations for which the test statistic T_b is better than T;
  Return p;
}
```

7.2.1.3 Choosing Features Based on the Score

Differential scores or their associated p-value scores allow us to rank all feature candidates. However, it is still not clear how many features should be filtered out. The task is easy if we always seek a fixed set of k features. In such a case, the top k features are selected with respect to the ordering imposed by ranking features by their score. However, the quality of these features may vary widely, so selecting the features based solely on the order may cause some poor features to be included in the set. An alternative method is to choose features by introducing a threshold on the value of the score. Unfortunately, not every scoring criterion has an interpretable meaning, so it is unclear how to select an appropriate threshold. The statistic typically used for this purpose is the p-value associated with the hypothesis test. For example, if the p-value threshold is 0.05 then there is a 5% chance the feature is not differentially expressed at the threshold value. Such a setting allows us to control the chance of *false positive* selections. These are features which appear discriminative by chance.

7.2.1.4 Feature Set Selection and Controlling False Positives

The high-dimensional nature of biological data sources necessitates that many features (genes or MS-profile peaks) be tested and evaluated simultaneously. Unfortunately, this increases the chance that false positives are selected. To illustrate this, assume we measure the expression of 10 000 independent genes and none of them are differentially expressed. Despite the fact that there is no differential expression, we might expect 100 features to have their p-value

smaller than 0.01. An individual feature with p-value 0.01 may appear good in isolation, but may become a suspect if it is selected from thousands of tested features. In such a case, the p-value of the combined set of the top 100 features selected out of 10 000 is quite different. Thus, adjustment of the p-value when performing multiple tests in parallel is necessary.

The *Bonferroni correction* adjusts the p-value for each individual test by dividing the target p-value for all findings by the number of findings. This assures that the probability of falsely rejecting any null hypotheses is less than or equal to the target p. The limitation of the Bonferroni correction is that it operates under the assumption of independence and as a result it is too conservative if features are correlated. Two alternatives to the Bonferroni correction are offered by: (1) the *family-wise error rate method* (FWER, (Westfall and Young, 1993)) and (2) methods for controlling the *false discovery rate* (FDR, (Benjamini and Hochberg, 1995; Tusher et al., 2001). FWER takes into account the dependence structure among features, which often translates to higher power. Benjamini and Hochberg (1995) suggest to control FDR instead of the p-value. The FDR is defined as the mean of the number of false rejections divided by the total number of rejections. The *significance analysis of microarrays* (SAM) method (Storey and Tibshirani, 2003) is used as an estimate of the FDR. Depending on the chosen threshold value for the test statistic T, it estimates the expected proportion of false positives on the feature list using a permutation scheme.

7.2.1.5 Correlation Filtering

To keep the feature set small, the objective is to diversify the features as much as possible. The selected features should be discriminative as well as independent from each other as much as possible. The rationale is that two or more independent features will be able to discriminate the two classes better than any of them individually. Each feature may differentiate different sets of data well, and independence between the features tends to reduce the overlap of the sets. Similarly, highly dependent features tend to favor the same data and thus are less likely to help when both are included in the panel. The extreme case is when the two features are exact duplicates, in which case one feature can be eliminated.

Correlation filters (Ross et al., 2000; Hauskrecht et al., 2005) try to remove highly correlated features since these are less likely to add new discriminative information (Guyon and Elisseeff, 2003). Various elimination schemes are used within these filters to reduce the chance of selected features being highly correlated. Typically, correlation filters are used in combination with other differential scoring methods. For example, features can be selected incrementally according to their p-value; the feature to be added next is checked for correlation with previously selected features. If the new feature exceeds some correlation threshold, it is eliminated (Hauskrecht et al., 2005).

7.2.2 Wrapper Methods

Wrapper methods (Kohavi and John, 1998) search for the best feature subset in combination with a fixed classification method. The goodness of a feature subset is determined using internal-validation methods, such as, k-fold or leave-one-out cross-validation (Krus and Fuller, 1982). Since the number of all combinations is exponential in the number of features, the efficiency of the search methods is often critical for its practical acceptance. Different heuristic optimization frameworks have been applied to search for the best subset. These include: *Forward selection*, *backward elimination* (Blum and Langley, 1997), *hill climbing*, *beam search* (Russel and Norvig, 1995), and randomized algorithms such as *genetic algorithms* (Koza, 1995) or *simulated annealing* (Kirkpatrick et al., 1983). In general, these methods explore the search space (subsets of all features) starting with no features, all features, or a random selection of features. For example, the forward selection approach builds a feature set by starting from an empty feature set and incrementally adding the feature that improves the current feature set the most. The procedure stops when no improvement in the feature set quality is possible.

7.2.3 Embedded Methods

Embedded methods incorporate variable selection as part of the model building process. A classic example of an embedded method is CART (Classification and Regression Trees, (Breiman et al., 1984)).

CART searches the range of each individual feature to find the split that optimally divides the observed data into a more homogeneous groups (with respect to the outcome variable). Beginning with the subsets of the variable that produces the most homogeneous split, each variable is again searched across its range to find the next optimal split. This process is continued within each new subset until all data are perfectly fit by the resulting tree, or the terminal nodes have a small sample size. The group constituting the majority of data points in each node determines the classification accuracy of the derived terminal nodes. Misclassification error from internal cross-validation can be used to backprune the decision tree and optimize its projected generalization performance on additional independent test examples.

7.2.3.1 Regularization/Shrinkage Methods

Regularization or shrinkage methods (Hastie et al., 2001; Xing et al., 2001) offer an alternative way to learn classifications for data sets with large number of features but small sample size. These methods trim the space of features directly during classification. In other words, regularization "effectively" shuts down (or zeros the influence of) unnecessary features.

Regularization can be incorporated either into the error criterion or directly into the model. Let $\mathbf{w}$ be a set of parameters defining a classification

model (e.g., the weights of a logistic regression model), and let Error($\mathbf{w}, \mathbf{D}$) be an error function reflecting the fit of the model to data (e.g., least-squares as likelihood-based error). A regularized error function is then defined as:

$$Error_{Reg}(\mathbf{w}, \mathbf{D}) = Error(\mathbf{w}, \mathbf{D}) + \lambda||\mathbf{w}||, \tag{7.2}$$

where $\lambda > 0$ is a regularization constant, and $\| \cdot \|$ is either the L_1 or L_2 norm. Intuitively, the regularization term penalizes the model for nonzero weights so the optimization of the new error function drives all unnecessary parameters to 0. Automatic relevance determination (ARD) (MacKay, 1992; Neal, 1998) achieves regularization effects in a slightly different way. The relevance of an individual feature is represented explicitly via model parameters and the values of these parameters are learned through Bayesian methods. In both cases, the output of the learning is a feature-restricted classification model, so features are selected in parallel with model learning.

7.2.3.2 Support Vector Machines

Regularization effects are at work also in one of the most popular classification frameworks these days: The support vector machine (SVM) (Burges, 1998; Schölkopf and Smola, 2002). The SVM defines a linear decision boundary (hyperplane) that separates case and control examples. The boundary maximizes the distance (also called *margin*) in between the two sample groups. The effects of margin optimization are twofold: Only a small set of data points (support vectors) are critical for the separation; the dimensions unnecessary for separation are penalized. Both of these processes help to fight the problem of model overfit. As a result, the SVM offers a robust classification framework that works very well for situations with a moderately large number of features and relatively small sample sizes.

7.2.4 Feature Construction

Better discriminatory performance can be often achieved using features constructed from the original input features. Building a new feature is an opportunity to incorporate domain specific knowledge into the process and hence to improve the quality of features. Nevertheless, a number of generic feature construction methods exist: Clustering; linear (affine) projections of the original feature space; as well as more sophisticated space transformations such as wavelet or kernel transforms. In the following, we briefly review three basic feature construction approaches: Clustering, PCA and linear discriminative projections.

7.2.4.1 Clustering

Clustering groups data components (data points or features) according to their similarity. Every data component is assigned to one of the groups (clusters); components falling into the same cluster are assigned the same value in

the new (reduced) representation. Clustering is typically used to identify distinguished sample groups in data (Ben-Dor et al., 2000; Slonim et al., 2000). In contrast to supervised learning techniques that rely heavily on class label information, clustering is unsupervised and the information about the target groups (classes) is not used. From the dimensionality reduction perspective, a data point is assigned a cluster label which is then used as its representation.

Clustering methods rely on the similarity matrix – a matrix of distances between data components. The similarity matrix can be built using one of the standard distance metrics such as Euclidean, Mahalanobis, Minkowski, etc., but more complex distances based on, for example, functional similarity of genes (Speer et al., 2005), are possible. Table 7.2 gives a list of some standard distance metrics one may use in clustering.

Table 7.2. Examples of distance metrics for clustering.

Metric	Formula
Euclidean distance	$d(r,s) = \sqrt{(\mathbf{x}_r - \mathbf{x}_s)(\mathbf{x}_r - \mathbf{x}_s)'}$
Standardized Euclidean distance	$d(r,s) = \sqrt{(\mathbf{x}_r - \mathbf{x}_s)D^{-1}(\mathbf{x}_r - \mathbf{x}_s)'}$
Mahalanobis distance	$d(r,s) = \sqrt{(\mathbf{x}_r - \mathbf{x}_s)\Sigma^{-1}(\mathbf{x}_r - \mathbf{x}_s)'}$
City Block (or Manhattan) metric	$d(r,s) = \sum_{j=1}^{n} \lvert\mathbf{x}_{rj} - \mathbf{x}_{sj}\rvert$
Minkowski metric	$d(r,s) = \sqrt[p]{\left(\sum_{j=1}^{n} \lvert\mathbf{x}_{rj} - \mathbf{x}_{sj}\rvert^p\right)}$
Cosine distance	$d(r,s) = \left(1 - \frac{\mathbf{x}_r\mathbf{x}_s'}{\sqrt{\mathbf{x}_r'.\mathbf{x}_r}\sqrt{\mathbf{x}_s'\mathbf{x}_s)}}\right)$
Correlation distance	$d(r,s) = 1 - \frac{(\mathbf{x}_r - \bar{\mathbf{x}}_r)(\mathbf{x}_s - \bar{\mathbf{x}}_s)'}{\sqrt{(\mathbf{x}_r - \bar{\mathbf{x}}_r)(\mathbf{x}_r - \bar{\mathbf{x}}_r)'}\sqrt{(\mathbf{x}_s - \bar{\mathbf{x}}_s)(\mathbf{x}_s - \bar{\mathbf{x}}_s)'}}$
Hamming distance	$d(r,s) = \frac{\#(x_{rj} \neq x_{sj})}{n}$
Jaccard distance	$d(r,s) = \frac{\#[(x_{rj} \neq x_{sj}) \wedge ((x_{rj} \neq 0) \vee (x_{sj} \neq 0))]}{\#[(x_{rj} \neq 0) \vee (x_{sj} \neq 0)]}$

$\mathbf{x}$ and $\mathbf{x}'$ denote a column vector and its transpose, respectively.
$\mathbf{x}_r$ and $\mathbf{x}_s$ indicate the r^{th} and s^{th} samples in the data set, respectively.
x_{rj} indicates the j^{th} feature of the r^{th} sample in the data set.
$\bar{\mathbf{x}}_r$ indicates the mean of all features in the r^{th} sample in the data set.
D is the diagonal matrix with diagonal elements given by v_i^2, which denotes the variance of i^{th} variable.
Σ is the sample covariance matrix.
The symbol # denotes counts; the number of instances satisfying the associated property.

7.2.4.2 Clustering Algorithms

The goal of clustering is to optimize intra- and inter-cluster distances among the components. Two basic clustering algorithms are: *k-means clustering* (McQueen, 1967; Ball and Hall, 1967), and *hierarchical agglomerative clustering* (Cormack, 1971; Eisen et al., 1998).

Briefly, the k-means algorithm clusters data into groups by iteratively optimizing positions of cluster centers (means) so that the sum of within-cluster distances (the distances between data points and their cluster centers) is minimized. Initial positions for cluster centers are generated randomly or by using heuristics. The algorithm is not guaranteed to converge to the optimal solution. On the other hand, hierarchical agglomerative methods work by combining pairs of data entities (features) or clusters into a hierarchical structure (called a dendrogram). The algorithm starts from unit clusters and merges them greedily (i.e., choosing the merge which most improves the fit of the clusters to the data) into larger clusters using an a priori selected similarity measure.

7.2.4.3 Probabilistic (Soft) Clustering

The k-means and agglomerative clustering methods assign every data point into a single cluster. However, sometimes it may be hard to decide what cluster the point belongs to. In *probabilistic (soft) clustering* methods, a data point belongs to all clusters, but the strength (weight) of its association with clusters differs by how well it fits cluster descriptions. Typically, the weight has probabilistic meaning and defines a probability with which a data point belongs to a cluster.

To calculate the probability, an underlying probabilistic model must be first fit to the data. Briefly, data are assumed to be generated from k different classes that correspond to clusters. Each class has its own distribution for generating data points. The parameters of these distributions as well as class (cluster) priors are fit (learned) using Expectation-Maximization techniques (Dempster et al., 1977). Once the model parameters are known, the probabilistic weights relating a data point and clusters are posterior probabilities of the point belonging to classes. A classic example of a probabilistic model often used in clustering is the Mixture of Gaussians model (McLachlan et al., 1997), where k clusters are modeled using k Gaussian distributions.

7.2.4.4 Clustering Features

Clustering methods can be applied to group either data points or features in the data. When clustering features, the dimensionality reduction is achieved by selecting a representative feature (typically the feature that is closest to the cluster center (Guyon and Elisseeff, 2003)), or by aggregating all features within the cluster via averaging to build a new (mean) feature. If we

assume k different feature clusters, the original feature space is reduced to a new k-dimensional space. An example method of feature clustering is to cluster features based on intra-correlation, and use the cluster center as a representative. Closely correlated features are not likely to help when separated, so grouping them away from more unrelated features will help diversify the resulting features.

7.2.4.5 Principal Component Analysis

Principal component analysis (PCA) (Jolliffe, 1986) is a widely used method for reducing the dimensionality of data. PCA finds projections of high-dimensional data into a lower dimensional subspace such that the variance retained in the projected data is maximized. Equivalently, PCA gives uncorrelated linear projections of data while minimizing their least square reconstruction error. Additionally, PCA works fully unsupervised; class labels are ignored. PCA can be extended to nonlinear projections using kernel methods (Bach and Jordan, 2001). Dimensionality reduction methods similar to PCA that let us project high dimensional features into a lower dimensional space include multidimensional scaling (MDS) (Cox and Cox, 1994) used often for data visualization purposes or independent component analysis (ICA) (Jutten and Herault, 1991).

7.2.4.6 Discriminative Projections

Principal component analysis identifies affine (linear) projections of data that maximize the variance observed in data. The method operates in a fully unsupervised manner; no knowledge of class labels is used to find the principal projections. The question is whether there is a way to identify linear projections of features such that they optimize the discriminability among the two classes. Techniques which try to achieve this goal include *Fisher's linear discriminant* (FLD) (Duda et al., 2000), *linear discriminant analysis* (Hastie et al., 2001) and more complex methods like *partial least squares* (PLS) (Denham, 1994; Dijkstra, 1983).

Take, for example, the linear discriminant analysis model. The model assumes that cases and controls are generated from two Gaussian distributions with means $\mu_{(-)}$, $\mu_{(+)}$ and the same covariance matrix Σ. The parameters of the two distributions are estimated from data using the maximum likelihood methods. The decision boundary that is defined by data points that give the same probability for both distributions is a line. The linear projection is defined as:

$$\mathbf{w} = \Sigma^{-1}(\mu_{(+)} - \mu_{(-)}), \tag{7.3}$$

where $\mu_{(-)}, \mu_{(+)}$ are the means of the two groups and Σ is the covariance for both groups, where $p(x|y) \sim N(\mu, \Sigma)$.

7.3 Advantages and Disadvantages

Each of the aforementioned methods comes with advantages and disadvantages. The following text briefly summarizes them.

Filter methods:

- Advantages: Univariate scores are very easy to calculate and thus, filter methods have a short running time. If our goal is a prediction, they often perform well in combinations with more robust classification methods such as the SVM.
- Disadvantages: Many differential scoring methods exist, it is unclear which one is best for the data set at hand. The features are analyzed independent of each other. This is a problem if our goal is to identify a small panel of discriminative features (biomarkers). Multivariate relations/dependencies must be incorporated through additional criteria, e.g., correlation filters.

Wrapper methods:

- Advantages: More comprehensive search of the feature set space. The feature set with the best discriminative potential on a fixed classification method is selected.
- Disadvantages: Running time is much longer than filter methods; many feature sets need to be analyzed and assessed. In addition, scoring of feature sets is based on internal cross-validation methods, which lengthens their running time. The reliability of the estimate of the internal cross-validation error needs to be considered. Low reliability of the internal validation error in combination with a large number of subsets examined can be lethal especially in various greedy search schemes.

Embedded methods:

- Advantages: Features and their selection are tuned to a specific model. Learning methods which incorporate aspects of regularization, like the SVM or regularized logistic regression, can learn very good predictive models even in the presence of high-dimensional data. We recommend trying SVM as a first step if the goal is only to build a predictive model.
- Disadvantages: Identification of a small set of features may be problematic. Backward feature elimination routines (Guyon and Elisseeff, 2003) can be used to reduce the feature panel to a more reasonable size.

Feature construction methods:

- Advantages: May incorporate the domain knowledge which may translate to improved feature sets.
- Disadvantages: If features are constructed using one of the out-of-box methods (e.g., PCA) the new features may be hard to interpret biologically. In addition, many feature construction techniques (e.g., clustering, PCA, ICA) work in an unsupervised mode, so high-quality features for discriminatory purposes are not guaranteed.

7.4 Case Study: Pancreatic Cancer

To illustrate some of the advantages and disadvantages of feature selection methods, we use a data set of MS proteomic profiles for pancreatic cancer collected at the University of Pittsburgh Cancer Institute (UPCI). Since full feature selection comparison is very hard to do without a full predictive model that combines both the feature selection and the classification stages we test feature selection methods in combination with one classification method – the linear support vector machine (SVM) (Vapnik, 1995). All classification results presented in the following text were obtained by using the repeated random subsampling strategy with 40 different train/test data splits using 70/30 train/test split ratio. The optimization criterion for the SVM method was a zero-one loss function, which focuses on improving classification error instead of sensitivity or specificity. The statistics reported are: Average test classification error (ACE), sensitivity (SN) and specificity (SP) and their standard deviations.

7.4.1 Data and Pre-Processing

The data set consists of 116 MS profiles, with 57 cancer cases (+ group) and 59 controls, matched according to their smoking history, age, and gender (− group). The data were generated using Ciphergen Biosystems Inc. SELDI-TOF (surface-enhanced laser desorption/ionization time-of-flight) mass spectrometry. Compounds such as proteins, peptides and nucleic acids for masses of up to 200 000 Daltons are recorded using this technology. Before applying feature selection techniques the data set was pre-processed using the *Proteomic Data Analysis Package* (PDAP) (Hauskrecht et al., 2005). The following pre-processing steps were applied: (1) Cuberoot variance stabilization, (2) local min-window baseline correction, (3) Gaussian kernel smoothing, (4) range-restricted intensity normalization, and (5) peak-based profile alignment. The quality of all profiles were tested beforehand on raw MS profile readings using total ion current (TIC). None of the profiles differed by more than two standard deviations from the mean TIC, which is our current quality-assurance/quality–control threshold for sample exclusion. After basic pre-processing, peaks in the range of $1\,500 - 1\,650$ Daltons were identified and their corresponding intensities were extracted.[4] This gave us a data set of 116 samples with 602 peak features.

[4] The region below 1 500 Daltons is unsuitable for analysis because of known signal reproducibility problems. The region is often referred to as the junk region. On the other hand, signals for higher mass-to-charge-ratios are of lower intensity which makes them hard to separate from the noise. An a priori upper limit is typically set to restrict the search for signal.

7.4.2 Filter Methods

7.4.2.1 Basic Filter Methods

Many univariate scoring metrics that assess the individual quality of features were proposed in the literature. An important question is how the rankings and subsequent feature selection induced by these metrics vary. Table 7.3 shows the number of overlapping features for the top 20 features selected according to four frequently used scoring criteria: Correlation, Fisher, t-statistic and Wilcoxon's p-value measures.

Table 7.3. Overlap of top 20 features for four different metrics.

	Correlation	*Fisher*	*t-statistics*	*Wilcoxon*
Correlation	–	18	12	18
Fisher	18	–	11	16
t-statistics	12	11	–	11
Wilcoxon	18	16	11	–

The table shows that different scoring metrics may induce rather different feature orders and as a result, different feature panels. It is very hard to argue that any one of them is the best. The quality depends strongly on the classification technique used in the next step, but even there the story is often unclear, and the best method tends to vary among the data sets. Table 7.4 illustrates the results obtained using top 20 choices of four scoring methods from Table 7.3 after we combine them with the linear SVM model. Standard deviations of performance statistics are also given. We see that the best classification error was obtained using the features selected based on the t-statistic score. While our experience is that the t-statistic score performs well on many proteomic data sets, other scoring metrics may often outperform it.

Table 7.4. Results for classifiers based on different feature filtering methods and the linear SVM. Standard deviations are given in parentheses.

	Correlation	*Fisher*	*t-statistics*	*Wilcoxon*
ACE	0.2500 (0.1178)	0.2188 (0.1075)	0.1743 (0.0684)	0.2611 (0.1091)
SN	0.8022 (0.0945)	0.8102 (0.1210)	0.8259 (0.0997)	0.7956 (0.1200)
SP	0.7142 (0.1249)	0.7628 (0.1423)	0.8327 (0.0852)	0.6961 (0.1607)

7.4.2.2 Controlling False Positive Selections

A problem with high-dimensional data is that some features may appear as good discriminators simply by chance. The problem of false positive identifications of features is critical for the biomarker discovery task. Clearly, a more

comprehensive analysis and validation of the feature in the lab may incur a significant monetary cost. While positive feature selections may influence also the generalizations of the predictive model and its classification accuracy, the classification methods are often more robust to handle them and the problem of false positive features is less pressing than for the biomarker discovery applications.

The false positive selection rate can be controlled via p-value on individual features, Bonferroni corrected p-value for the panel of features, or through false discovery rate. Table 7.5 shows the number of features out of 602 original features selected by each of these methods.

Table 7.5. P-value for t-statistics.

original number of features	$p < 0.05$	*Bonferroni* $p < 0.05$	*FDR* 0.2
602	13	0	5

Assuming that all features are independent and random, we expect to see about 30 false positive features under the simple p-value of 0.05 for each feature. Using this estimate and the fact that we see only 13 features for the p-value of 0.05 would lead us to the conclusion that all of these are likely obtained by chance. The caveat is that when features are dependent and correlated the expected numbers are very different. Indeed, features in this and other proteomic data sets exhibit a large amount of correlation among the features; so the result in the table is indicative of such a dependency. The Bonferroni correction typically leads to a very conservative bound that may be very hard to satisfy. For example, none of the features in our cancer data passed Bonferroni-corrected p-value of 0.05. FWER and FDR methods and their thresholds give better estimates of false positive selections and their rates for the real-world data and should be preferred over simple and Bonferroni-corrected p-value thresholding.

When selecting features, our objective is to strike the right balance between the number of features, the flexibility they may offer when building multivariate discriminators, and the risk of inclusion of false positive features. The FWER and FDR methods give better control over risks of false positives. However, choosing the optimal thresholds for these techniques is a matter of personal preference. For example, two different approaches can be taken. If the selected features are meant only for use with an automated classification routine, it may be more acceptable to risk selecting false positives, and thusly the threshold can be less stringent. On the other hand, if the selected features are to be investigated more thoroughly (e.g., to analyze them using wet lab techniques), it would be far less acceptable to suggest that false positives are informative features. In this case, the threshold should be set more aggressively.

7.4.2.3 Correlation Filters

Biological (genomic and proteomic) data sets often exhibit a relatively high number of correlations. The correlations can be introduced by the technology producing the data or they reflect true underlying dependencies among measured species. For example, a peak in a proteomic profile is formed by a collection of correlated measurements, triple or double charged ions cause the same signal to be replicated at different parts of the profiles, and finally some peaks are correlated because they share a common regulatory (or interaction) pathway.

Selecting two features that are near duplicates, even if they are highly discriminative, does not help the classification model and its accuracy. Correlation filtering alleviates the problem by removing features highly correlated with existing features in the panel. Table 7.6 illustrates the number of features one obtains by filtering out correlated features at different maximum allowed absolute correlation (MAC) thresholds from the original 602 features. We note that the amounts of correlates filtered out at higher thresholds are statistically significantly different (at $p = 0.01$) from what one would obtain for independent feature sets.

Table 7.6. Effect of correlation filtering.

Threshold	1	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1	0
Number of Features	602	460	247	119	52	22	12	9	6	3	1

Figure 7.1 illustrates the effect of correlation filtering when it is combined with the univariate feature scoring based on the t-statistic. We see that test errors for smaller feature sets (size 5) are improved if feature panels are decorrelated. However, for larger feature panels the effect of feature decorrelation may vanish since some good features that add some discriminative value to the panel are filtered out. For example, for 20 features in Figure 7.1 the effect of correlation filtering has disappeared and the SVM classifier based on the unrestricted t-statistic score performs better than classifiers with correlation thresholds of 0.75 and 0.5. This illustrates one of the problems of the method, identification of an appropriate MAC threshold. We must note that the effect as seen in Figure 7.1 may be less pronounced on other classification methods or on other data sets, while in some cases correlation thresholds may lead to superior performance. These differing outcomes are the results of tradeoffs of feature quality and overfit processes.

The plain correlation threshold filtering method suffers from a couple of problems. First, an identification of an appropriate correlation threshold in advance is hard. Moreover, for different feature sizes there appears to be a different threshold that works best so switching of thresholds may be appropriate. One solution to this problem is the parallel correlation filtering method

(Hauskrecht et al., 2005) that works at multiple correlation threshold levels in parallel and uses internal cross-validation methods to decide on what feature (correlation level) to select next. The performance of the method is compared to the unrestricted t-statistic filter and two correlation filtering methods based on simple MAC thresholds in Figure 7.1.

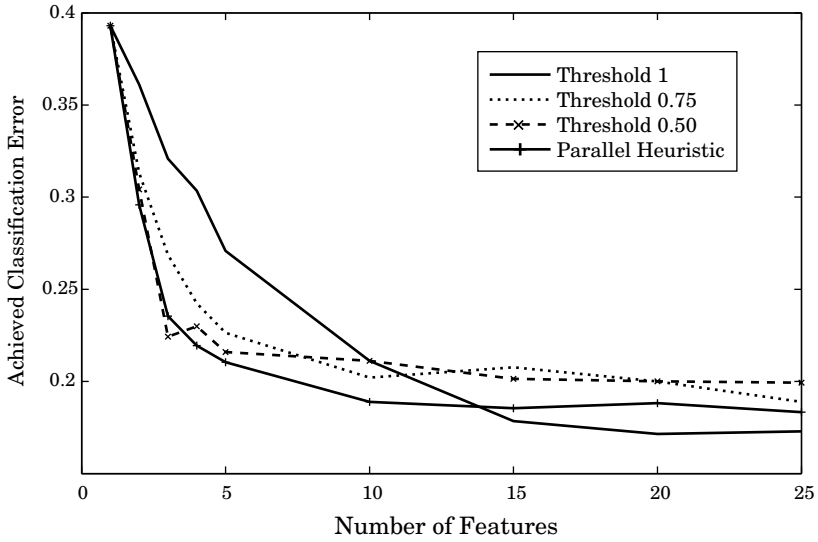

Fig. 7.1. Effect of correlation filtering on classification errors. Results of correlation filtering on the t-statistic score and SVM are shown.

7.4.3 Wrapper Methods

Wrapper methods search for the best subset of features by trying them in combination with a fixed classification method. However, there is a natural tradeoff between the quality of the feature set found, and the time taken to search for it. Table 7.7 displays performance statistics for two search methods: *Greedy forward selection* and *simulated annealing*.

The forward selection approach, also called the *greedy* approach, adds the feature which improves the set the most. The panel begins empty and is built incrementally, stopping when no improvement in the feature set is possible. Simulated annealing is a randomized algorithm and if it is left to search long enough all possible combinations may be reached and evaluated. Thus, simulated annealing may arrive at a better solution than the greedy method when given enough time. This quality/time tradeoff is captured in the table. The model based on the greedy forward selection method leads to average errors of 0.1750 while simulated annealing approaches 0.1660. To reach

Table 7.7. Wrapper methods with two search algorithms: Forward selection and simulated annealing. Standard deviations are given in parentheses.

	Greedy	*Simulated Annealing*
ACE	0.1750 (0.0668)	0.1660 (0.0603)
SN	0.8239 (0.1123)	0.8149 (0.1097)
SP	0.8261 (0.1100)	0.8614 (0.0784)
# steps	7037.4	10000

the result, 7 037.4 feature sets were evaluated on average by forward selection, while simulated annealing was run for 10 000 steps on every train/test split.

Evaluating a new feature set in any wrapper method is done by internal validation methods, such as k-fold cross-validation or leave-one-out validation. The overhead incurred by the evaluation step contributes to the running time of the algorithm. In general, using more internal splits improves the estimate of the error for each feature set. The price paid for it is an additional increase in the running time. Despite the downfalls, the results obtained from wrapper methods powered by various search heuristics are often quite good, especially when computational time is not an issue.

7.4.4 Embedded Methods

Table 7.8 shows the results of three classification methods with embedded feature selection: CART (Breiman et al., 1984), *regularized logistic regression* (RLR) (Hastie et al., 2001) and *support vector machines* (SVMs) (Burges, 1998). Each of these methods handles features differently, and consequently leads to different classification accuracies. We see that two of the methods, RLR and SVM, achieved results comparable or better than filter and wrapper methods. While this is not the rule, the linear SVM appears to be a very stable method across a large range of features so we always recommend to try it on the full feature set.

Table 7.8. Performance statistics for embedded methods. Standard deviations are given in parentheses.

	CART	*Regularized LR*	*SVM*
ACE	0.3681 (0.0897)	0.1382 (0.0584)	0.1382 (0.0623)
SN	0.6321 (0.1888)	0.8619 (0.1026)	0.8536 (0.0913)
SP	0.6361 (0.2088)	0.8624 (0.0942)	0.8769 (0.0881)

Embedded methods may not be optimal, if we want to use them for biomarker discovery, that is, if our objective is to find a small set of original features with a good discriminatory performance. The embedded methods may rely on too many features so a follow-up selection of a smaller subset

is necessary. Wrapper methods based on the backward feature elimination (Guyon and Elisseeff, 2003) achieve this by gradually eliminating the features that affect the performance the least.

7.4.5 Feature Construction Methods

To illustrate feature construction methods we use three unsupervised methods: sample clustering, feature clustering and PCA projections, all aimed to reduce the dimensionality of data. The results of these methods in combination with the linear SVM are in table 7.9.

Table 7.9. Construction methods: Sample clustering using squared Euclidean distance, feature clustering using correlation coefficient, and PCA. Standard deviations are given in parentheses.

	Sample Clustering	*Feature Clustering*	*PCA Projections*
ACE	0.4525 (0.0810)	0.2104 (0.0652)	0.1681 (0.0594)
SN	0.4721 (0.1604)	0.7932 (0.1426)	0.8223 (0.0984)
SP	0.6444 (0.1633)	0.7968 (0.0920)	0.8492 (0.0842)

The first entry in the table (sample clustering with Euclidean distance) illustrates the major weakness of clustering methods: The clustering does not give reasoning as to why the data components group together, other that their distance is close, which obviously depends on the choice of the metric. Thus, one has to assure that the distance selected is not arbitrary and makes sense for the data and the prediction task. The result for clustering of features based on the correlation metric also supports this point. There are many feature correlates in the proteomic data set, so grouping the features based on their mutual correlation and replacing the features in each cluster with a feature corresponding to the cluster center tends to eliminate high correlates in the new (reduced) data. This is very similar in spirit to the correlation filtering method. The difference is that the correlation filtering is closely combined with and benefits from the univariate score filtering, while correlation clustering works fully unsupervised.

PCA constructs features using linear projections of complete data. Since PCA arranges projections along uncorrelated axes, it helps to relieve us from identifying feature correlates. As a result, we see an improvement in classification error over some other construction and filtering methods. Note that PCA can be a good "one shot" technique, avoiding necessities like the choice of the number of clusters, k, in k-means clustering, or scoring metric in filtering methods. The effort saved by not choosing parameters is in exchange for knowledge about a targetable panel of biomarkers, but PCA can still be convenient if the only interest is constructing a predictive model.

7.4.6 Summary of Analysis Results and Recommendations

There are multiple feature selection/dimensionality reduction methods one may apply to reduce the feature size of the data and make it "comparable" to its sample size. Unfortunately, there is no perfect recipe for what method to choose but here are some guidelines.

- Having prior information about how features can be related to the prediction task will always help feature selection and its subsequent application. So whenever possible try to use this information. For example, when the biological relevance of features can be ascertained, the potentially irrelevant or obvious features can also be eliminated.
- In the presence of no prior information, more generic information can be used for steering feature selection in the right direction. The effect of a feature on the target class and the presence of multivariate dependencies (e.g., correlations) among feature candidates appear to be the most important ones. The importance of a feature is captured by a univariate scoring metric. Dealing with highly correlated features, either by grouping them or eliminating redundancies, can help the selection process by narrowing the choice of features.
- Feature selection coupled with more robust classification methods, like SVM, can perform extremely well on all features. Backward feature elimination methods can be applied if we would like to identify a smaller panel of informative features.
- The feature selection method applied to data does not have to match a single method. A combination of feature selection methods may be beneficial and may work much better (Xing et al., 2001). For example, it may help to exclude some features outright with a basic filtering method by removing the lowest-scoring features and apply other methods (e.g., wrapper or PCA methods) only on the remaining features.

Since there are many feature selection methods, one may be tempted to try many of them in combination with a specific classifier and pick the one that gives the best test set result *post hoc*. Note that in such a case the error is biased and does not objectively report on the generalizability of the approach. Model selection methods based, for example, on an internal cross-validation loop should be applied whenever a choice out of many candidates is allowed.

In closing, it is important to note that the selection of the feature selection technique should first be driven by prior knowledge about the data, and then by the primary goal you wish to accomplish by analyzing the data: Obtain a small, easy to interpret, feature panel or build a good classification model. Feature selection techniques vary in their complexity and interpretability, and the issues discussed above must be taken into careful consideration.

7.5 Conclusions

In this chapter, we have presented four basic approaches to feature selection and dimensionality reduction. Filter, wrapper, and embedded methods work with the available features and choose those which appear important. In slight contrast, feature construction methods build new features which can be more powerful than previous ones. To discuss the entire gamut of feature selection methods would be exhaustive, as researchers must constantly meet their needs of analyzing high-dimensional data. The techniques covered here are among the most effective for analyzing genomic and proteomic data, in terms of building predictive models and developing biologically relevant information.

7.6 Mathematical Details

Table 7.10. Formulae for popular filter scores.

Filter Name	*Formula*
Fisher Score	$score(i) = \frac{(\mu_+(i)-\mu_-(i))^2}{s^2_+(i)+s^2_-(i)}$
SAM Score	$score(i) = \frac{\lvert\mu_+(i)-\mu_-(i)\rvert}{s_{SAM}(i)+s_{SAM,0}}$
t–test	$score(i) = \frac{\lvert(\mu_+(i)-\mu_-(i))\rvert}{\sqrt{\frac{s^2_+(i)}{n_+}+\frac{s^2_-(i)}{n_-}}}$
Mutual Inform.	$score(i) = \sum_{\{x_i\}} \sum_{y\in\{+,-\}} p(X_i = x_i, Y = y) \cdot \log \frac{p(X_i=x_i,Y=y)}{p(X_i=x)\cdot(Y=y)}$
χ^2 (Chi–Square)	$score(i) = \sum_{\{x_i\}} \sum_{y\in\{+,-\}} \frac{(p(X_i=x_i,Y=y)-p(X_i=x_i)\cdot p(Y=y))^2}{p(X_i=x_i)\cdot p(Y=y)}$
AUC	$score(i)$ = Area under the ROC curve for feature i
$J5$ Score	$score(i) = \frac{\lvert\mu_+(i)-\mu_-(i)\rvert}{\frac{1}{m}\sum_{j=1}^{m}\lvert\mu_+(j)-\mu_-(j)\rvert}$

where $s^2(i) = \frac{1}{n-1}\sum_i (x_i - \mu(i))^2$. The SAM technique is meant to be used in a permutation setting, however, the employed statistics can still be used for feature filtering. The terms $s_{SAM}(i)$ and $s_{SAM,0}$ are computed as follows:

$$s_{SAM}(i) = \sqrt{\frac{(1/n_+)+(1/n_-)}{(n_++n_--2)}\left[\sum_{j=1}^{n_+}(x_j(i)-\mu_+(i))^2 + \sum_{j=1}^{n_-}(x_j(i)-\mu_-(i))^2\right]}$$

$$s_{SAM,0} = 1.$$

References

Bach, F. and Jordan, M. (2001). Kernel independent component analysis. *Tech. Rep. UCB//CSD-01-1166, UC Berkeley.*

Baldi, P. and Long, A.D. (2001). A bayesian framework for the analysis of microarray expression data: regularized t-test and statistical inferences of gene changes. *Bioinformatics*, 17:509–519.

Ball, G. and Hall, D. (1967). A clustering technique for summarizing multivariate data. *Behav. Science*, 12:153–155.

Ben-Dor, A., Bruhn, L., and Friedman, N., et al. (2000). Tissue classification with gene expression profiles. *J. Comp. Biol.*, 7:559–584.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *J. Roy. Stat. Soc. B*, 57:289–300.

Blum, A. and Langley, P. (1997). Selection of relevant features and examples in machine learning. *Art. Intell.*, 97(1-2):245–271.

Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J. (1984). *Classification and regression trees.* Wadsworth International Group, Belmont, CA.

Burges, C.J.C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167.

Chernoff, H. and Lehmann, E.L. (1954). The use of maximum likelihood estimates in chi2 tests for goodness-of-fit. *The Annals of Mathematical Statistics*, 25:576–586.

Cormack, R.M. (1971). A review of classification. *J. Roy. Stat. Soc. A*, 134:321–367.

Cox, T. and Cox, M. (1994). *Multidimensional Scaling.* Chapman & Hall, London.

Dempster, A.P., Laird, N.M., and Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Stat. Soc. B*, 34:1–38.

Denham, M.C. (1994). Implementing partial least squares. *Statistics and Computing.*

Dijkstra, T. (1983). Some comments on maximum likelihood and partial least squares methods. *J. Econometrics*, 22:67–90.

Duda, R.O., Hart, P.E., and Stork, D.G. (2000). *Pattern Classification.* Wiley-Interscience Publication.

Eisen, M.B., Spellman, P.T., Brown, P.O., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA*, 95(25):14863–14868.

Furey, T.S., Christianini, N., and Duffy, N., et al. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906–914.

Golub, T.R., Slonim, D.K., and Tamayo, P., et al. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537.

Good, P. (1994). *Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypotheses*. Springer.

Gosser, W. S. (1908). The probable error of a mean. *BIOMETRIKA*, 6:1–25.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *J. Machine Learning Res.*, 3:1157–1182.

Hanley, J. A. and McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer, New York/Berlin/Heidelberg.

Hauskrecht, M., Pelikan, R., and Malehorn, D.E., et al. (2005). Feature selection for classification of SELDI-TOF MS proteomic profiles. *Appl. Bioinf.*, 4(4):227–246.

Jolliffe, I. T. (1986). *Principal Component Analysis*. Springer, New York.

Jutten, C. and Herault, J. (1991). Blind separation of sources, part 1: An adaptive algorithm based on neuromimetic architecture. *Signal Process.*, 24(1):1–10.

Kendall, M.G. (1945). The treatment of ties in ranking problems. *Biometrika*, 33(3):239–251.

Kirkpatrick, S., Gelatt, C., and Vecchi, M. (1983). Optimization by simulated annealing. *Science*, 220:671–680.

Kohavi, R. and John, G. (1998). The wrapper approach. In Liu, H. and Motoda, H., editors, *Feature Selection for Knowledge Discovery and Data Mining*, pages 33–50. Kluwer Academic Publishers, Norwell, MA, USA.

Koza, J. (1995). Survey of genetic algorithms and genetic programming. *Proc. Wescon95:E2. Neural-Fuzzy Technologies and Its Applications, IEEE*, pages 589–594.

Krus, D.J. and Fuller, E.A. (1982). Computer-assisted multicrossvalidation in regression analysis. *Educational and Psychological Measurement*, 42:187–193.

Liu, H. and Setiono, R. (1995). Chi2: Feature selection and discretization of numeric attributes. *Proc. 7th IEEE Intl. Conf. Tools with Artificial Intelligence*, page 88.

MacKay, D. (1992). *Bayesian Methods for Adaptive Models*. PhD thesis, California Institute of Technology.

McLachlan, G., Peel, D., and Prado, P. (1997). Clustering via normal mixture models. *Proc. Am. Stat. Assoc.*, pages 98–103.

McQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proc. 5th Berkeley Symp. Math. Stat. Prob.*, pages 281–297.

Neal, R. (1998). Assessing relevance determination methods using DELVE. In Bishop, C.M., editor, *Neural Networks and Machine Learning*, pages 28–32. Springer.

Patel, S. and Lyons-Weiler, J. (2004). caGEDA: A web application for the integrated analysis of global gene expression patterns in cancer. *Appl. Bioinf.*, 3(1):49–62.

Pavlidis, P., Weston, J., Cai, J., and Grundy, W.N. (2001). Gene functional classification from heterogeneous data. In *Proc. 5th Ann. Intl. Conf. Comp. Mol. Biol.*, pages 242–248.

Ross, D.T., Scherf, U., and Eisen, M.B., et al. (2000). Systematic variation in gene expression patterns in human cancer cell lines. *Nat. Gen.*, 24:227–235.

Russel, S. and Norvig, P. (1995). *Artificial Intelligence*. Prentice Hall.

Schölkopf, B. and Smola, A. J. (2002). *Learning with Kernels*. MIT Press.

Slonim, D.K., Tamayo, P., Mesirov, J.P., Golub, T.R., and Lander, E.S. (2000). Class prediction and discovery using gene expression data. *Proc. 4th Ann. Intl. Conf. Comp. Mol. Biol.*, pages 263–272.

Speer, N., Spieth, C., and Zell, A. (2005). Spectral clustering gene ontology terms to group genes by function. *Lecture Notes in Bioinformatics*, 3692:001–012.

Storey, J.D. and Tibshirani, R. (2003). SAM thresholding and false discovery rates for detecting differential gene expression in DNA microarrays. In Parmigiani, G., Garrett, E.S., Irizarry, R.A., and Zeger, S.L., editors, *The Analysis of Gene Expression Data: Methods and Software*, pages 272–290. Springer, New York.

Tusher, V.G., Tibshirani, R., and Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *Proc. Natl. Acad. Sci. USA*, 98(9):5116–5121.

Tzannes, N.S. and Noonan, J.P. (1973). The mutual information principle and applications. *Information and Control*, 22(1):1–12.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer, New York.

Westfall, P.H. and Young, S.S. (1993). *ResamplingBased Multiple Testing: Examples and Methods for P-value Adjustment*. Wiley.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, 1:80–83.

Xing, E.P., Jordan, M.I., and Karp, R.M. (2001). Feature Selection for High-Dimensional Genomic Microarray Data. *Proc. 18th Intl. Conf. Machine Learning*, pages 601–608.