

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.

For more information visit www.intechopen.com



A Comparison of Biomarker and Fingerprint-Based Classifiers of Disease

Brian T. Luke and Jack R. Collins
*SAIC-Frederick, NCI at Frederick, Frederick,
USA*

1. Introduction

1.1 Early detection of disease

Early detection of a disease is very important since it greatly improves the individual's chance of responding well to treatment. For example, the 5-year survival rate from prostate cancer is nearly 100% if it is detected early [http://www.toacorn.com/news/2005/1027/Health_and_Wellness/077.html]. Similarly, the 5-year survival rate for ovarian cancer is 95% if caught early, but since 75% of the cases are first observed in the later stages of the disease, the overall 5-year survival rate is less than 50% [<http://www.information-about-ovarian-cancer.com/>]. It would be nice if there was a single test to determine if an individual had cancer somewhere in their body, but unfortunately such a test does not exist. While all cancers have many factors in common, tissue differences and the body's response to different cancers make the test for ovarian cancer (CA125) very different from the test for prostate cancer (PSA). The lack of sufficient sensitivity and specificity has recently resulted in the recommendation that PSA no longer be used as a potential marker of prostate cancer [<http://www.uspreventiveservicestaskforce.org/uspstf/uspSprca.htm>].

Even within the same tissue, all cancers are not necessarily the same. It is well known that there are two major types of lung cancer, small cell lung cancer (SCLG) and non-small cell lung cancer (NSCLC). It is also known that NSCLC has three major sub-types; adenocarcinoma (AC), squamous cell carcinoma (SCC), and large cell undifferentiated carcinoma (LCUC). Each of these has differences in the biochemical processes going on within the cancer cell and one should not expect that the detection, or necessarily the treatment, of these cancers will be the same. Of the four recognized forms of lung cancer (SCLG, AC, SCC and LCUC), the latter three are strictly differentiated by appearances of the cell under the microscope. It is possible that the underlying biochemical processes of an AC cell in one individual are significantly different than the biochemical processes in another individual with a cancer that appears similar. Therefore, each of these categories of lung cancer may be composed of one or more states. While the disease category represents the name of the disease based on some experimental observation, the disease state represents a grouping based on the underlying biochemical processes within the diseased cell. The detection of a disease and its treatment should be relative to specific disease states, not a disease category or individuals within that category.

1.2 Types of classifiers

A high-quality classifier would be a great aide in the early detection of disease. The general procedure is to obtain a biological specimen and search for one or more features that correctly classify the individual. The specimen can be blood, urine, mucous, or tissue sample, for example, and the feature can be the expression level of mRNA, a protein, or a metabolite. The construction of the classifier starts with obtaining a large number of features from individuals with known phenotypes, known as the training set, and constructing a classifier that sufficiently predicts each sample's phenotype. This classifier is then used on a second set of samples of known phenotype, called the testing set, to determine its overall accuracy. Since the number of features will be much larger than the number of samples in the training set, the construction of a classifier suffers from the "curse of dimensionality" [Bellman, 1957, 1961, 2003]. If the training set contains N_h healthy samples and N_d diseased samples, then virtually any classifier that uses the smaller of N_h and N_d features, such as their social security numbers, can correctly classify all samples in the training set. This extreme example would represent a case where the classifier is fitting the individuals in the training set and not their phenotype. The goal is to choose a relatively small number of features that correctly distinguishes the samples.

Two extremes in the total range of possible classifiers are fingerprint-based and biomarker-based classifiers. A fingerprint-based classifier uses a collection of features, which is also known as a panel of markers. If two individuals have the same pattern in this set of features (i.e. similar fingerprints), and one is known to have a particular disease, it is assumed that the other has this same disease. A biomarker-based classifier tries to find a very small number of features that distinguishes all healthy samples from all diseased samples. In other words, a biomarker-based classifier tries to cluster all samples of the same phenotype into the smallest possible number of clusters. The optimum biomarker would distinguish all healthy from all diseased samples, resulting in a single healthy cluster and a single disease cluster.

1.3 Selecting features

A major difference between fingerprint-based and biomarker-based classifiers is how the features are selected. In a fingerprint-based classifier, the actual classifier is used to determine how well a given set of features distinguishes the samples. An overriding heuristic is used to determine what set of features is tested in the classifier, and the quality of the classification can be used to determine which feature set is tried next. This is known as a *wrapper method*. In contrast, a biomarker-based classifier uses one or more procedures to determine which features successfully distinguish some or all of the samples in the training set. This set of putative biomarkers is then used in a different classifier, either individually or a small number together, to determine how well the healthy samples can be distinguished from the diseased samples. In other words, the selection of features for a biomarker-based classifier uses a *filter method*.

Many different procedures can be used in the wrapper method to find the optimal set of markers. Three major classes of procedures are forward-selection, reverse-selection, and multidimensional searches. The simplest forward-selection method is a Greedy Search. In this procedure all features are individually tested in the classifier and the one that performs

the best is retained. All remaining features are then tested in combination with this best feature to find the feature-pair that performs the best. This procedure continues until either the addition of an additional feature does not improve the classification or a pre-set number of features are selected. An extension of this Greedy Search is known as Branch-and-Bound. In this latter procedure multiple classifiers are retained at each cycle and the search results in a population of classifiers that have the highest accuracy.

Reverse-selection works in the opposite direction. Initially, all features are used in the classifier and features that are not important to the classification are removed. In cases where the number of features is larger than the number of samples, special procedures need to be used to ensure that an important feature is not removed in the early steps of the reduction.

Multidimensional searches use a pre-defined number of features and try different combinations of features in the classifier. Examples of multidimensional search techniques are Simulated Annealing, Tabu Search, Gibbs Sampling, Genetic Algorithm, Evolutionary Programming, Ant Colony Optimization, and Particle Swarm Optimization. The first three techniques modify a single set of features while the latter four use a population of sets, where each feature set is changed throughout the search to find one or more optimal sets. It should be noted that in a Genetic Algorithm the number of features in the set can be reduced, so the pre-defined number should be considered a maximum number of allowed features in the final set.

The goal of each wrapper search technique is to find the optimal set of features, and therefore are approximations to an exhaustive search. If the objective is to find the best set of k features from a total set of K features, the number of unique combinations is generally given by $\binom{K}{k}$. If there are a total of 300 features and the goal is to find the best set of seven features, an exhaustive search would require examining 4.04×10^{13} unique sets of features. The situation is slightly more complicated for a decision tree. In this case the order of the features is important since this order determines if the feature acts on the entire set of samples, or a particular subset of samples. Here, the number of possible combinations is $\frac{K!}{(K-k)!}$ and an exhaustive search of 300 features to find the best seven-node decision tree would require examining 2.04×10^{17} trees. Since this is not computationally feasible, any result from a fingerprint-based classifier should be considered as a lower-bound to the accuracy of the classification algorithm.

In contrast, the search for the search for the best biomarker-based classifier is exhaustive. All features are examined by each filtering method and all combinations of putative biomarkers can be used in the final classifier.

1.4 Fingerprint-based classifiers

Informatic analysis has led to a new paradigm for classification known as fingerprinting or pattern matching. In this paradigm, individuals are classified based upon a particular pattern of intensities [Petricoin et al., 2003]. If an untested individual has the same pattern as a known individual, then these two have the same classification. The simplest fingerprint-based classifier is a decision tree (Figure 1). In all known applications of a decision tree to produce a classifier using spectral data [Ho et al., 2006; Liu et al., 2005; Yang et al., 2005; Yu

et al., 2005], a single scoring metric (e.g. Gini Index, entropy gain, etc.) was used to determine the cut point at a given node so that the two daughter nodes were as homogeneous as possible for one or more categories (e.g. diseased versus healthy). Given the general structure of a decision tree in Figure 1, the root node (Node 1) would contain all training samples and an m/z value, or feature, and its cut point would be selected that best separates diseased and healthy individuals between Nodes 2 and 3. If there was still enough of a mixture in Node 2, for example, a second feature would be chosen based on the same metric that would separate the individuals into Nodes 4 and 5. The same process would be used for all heterogeneous nodes until there was a sufficient domination of one category over another.

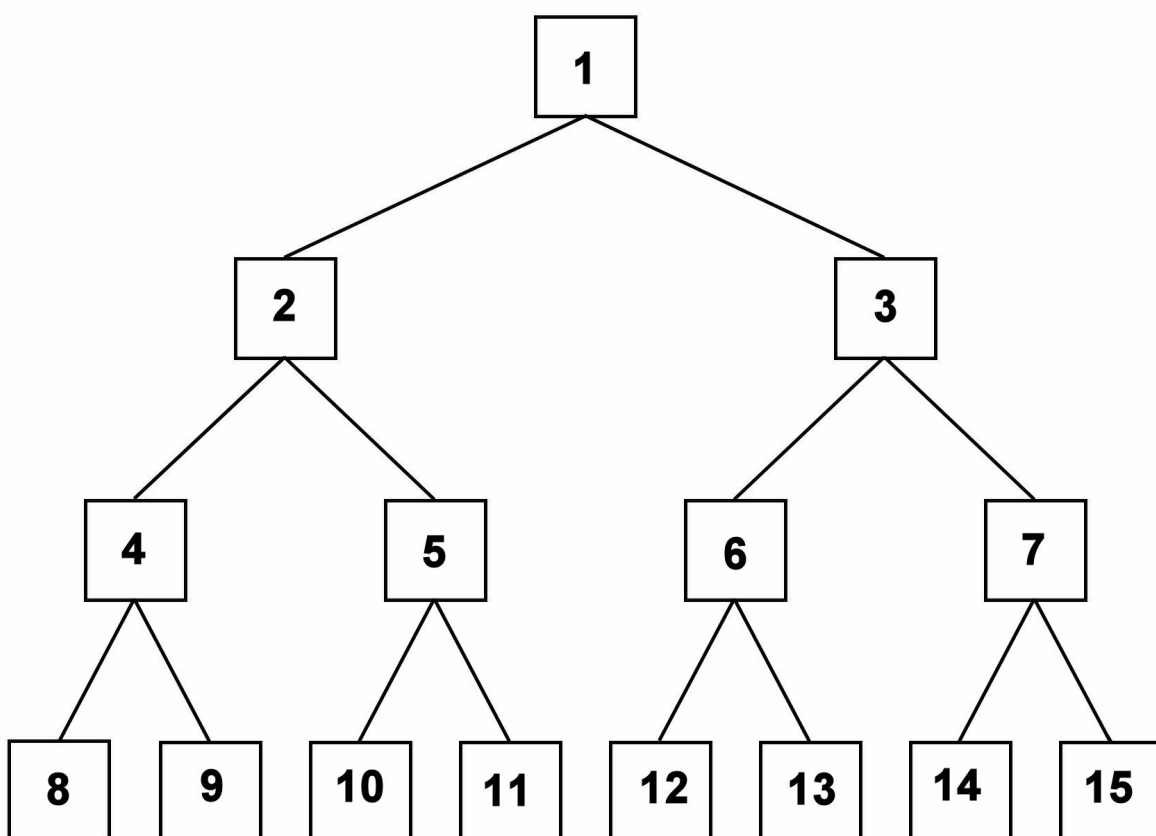


Fig. 1. Example of a 7-node decision tree.

Decision Support also uses decision trees, but an independent question is asked at each level in the tree. For example, Node 1 may be used to separate the individuals by gender, race, or other genetic difference, and then different features may be used to separate samples obtained from affected and healthy patients at a given level of stratification. Since the stratifying variables are not known ahead of time, there is no way to know the proper metric that should initially separate the training set. Therefore, the procedure used here is to construct unconstrained decision trees that best classify the training individuals.

The medoid classification algorithm is a best attempt at reproducing the algorithm used in many of the studies conducted in the laboratories of Emmanuel Petricoin and Lance Liotta [Browers et al., 2005; Conrads et al., 2004; Ornstein et al., 2004; Petricoin et al., 2004;

Srinivasan et al., 2006; Stone et al., 2005]. While these authors stated that their algorithm was quite similar to a Self-Organizing Map (SOM), their algorithm, as they described it, has virtually nothing in common with a SOM. In a SOM [Kohonen, 1988], the layout of the cells is determined *a priori*, as are the number of features, n , used in the separation. In general, the cells are placed in a rectangular or hexagonal pattern, with a maximum of four or six adjacent cells, respectively. The cells are seeded with random centroids that represent the n -dimensional coordinates of each cell. The first training sample is assigned to the cell with the closest centroid and the centroids of this and all the other cells are significantly shifted towards this sample. This procedure is repeated for all samples. Once all samples have been processed, the algorithm repetitively cycles through the list of training samples. In each subsequent cycle, the extent to which the centroid of the cell it is assigned to shifts towards that sample decreases, as does the extent to which the other centroids are affected. This shift becomes significantly smaller for cells that are further from the selected cell, as defined by the initial mapping. When finished, all samples are assigned to cells and each centroid represents an approximate average of the n features for all samples in that cell, and the distance between centroids increases as the cells become further apart in the pre-defined map.

In contrast, the algorithm used in the references cited above places the first training sample as the center of the first cell, and this cell is classified as the category of this sample. Since it is sample-centered, each cell has a medoid not a centroid. Each cell is given a constant trust radius, r . If the second sample has a distance that is larger than r from the first, it is assigned to a second cell and that cell is classified by its category; otherwise it is assigned to the first cell. This process continues until all training samples have been analyzed.

Therefore, a SOM has a fixed number of cells, each cell is described by a centroid, and the algorithm cycles through the training data many times to adjust the centroid's coordinates. The algorithm used by the groups of Petricoin and Liotta has an undefined number of cells, each described by a single sample, and the training data is only processed once.

Also grouped within the class of fingerprint-based classifiers are Support Vector Machines and Linear Discriminant Analysis. A Support Vector Machine (SVM) [Boser et al., 1992; Vapnik, 1998] is a kernel-based learning system. SVM searches for the optimal hyperplane that maximizes the margin of separation between the hyperplane and the closest data points on both sides of the hyperplane. Linear Discriminant Analysis (LDA) [Fukunaga, 1990] is a supervised learning algorithm. LDA finds the linear combination of features that maximize the between-class scatter and simultaneously minimize the within-class scatter to achieve maximum discrimination in a dataset. The within-class scatter matrix may become singular if the sample size is smaller than the dimensionality of the search space (number of features), but several techniques are available to handle this situation.

1.5 Biomarker-based classifiers

An example if a state-specific marker is shown in Figure 2. Each “+” represents, for example, the blood concentration of a particular biochemical. The individuals in the left column are in a specific disease state, while those in the right column are not and are therefore considered to be in a healthy state, at least with respect to this disease. Individuals in each state have different blood concentrations of this biochemical due to genetic and environmental

differences between individuals and any experimental uncertainty in the measurement. What is clear is that the range of concentrations for individuals in the disease state is significantly higher than for those not in this state. Such a marker can be used to classify the individuals into three groups; they are in the disease state if the blood concentration is above an upper threshold, they are not in this disease state if the concentration is below a lower threshold, and they are undetermined if the blood concentration is between these thresholds.

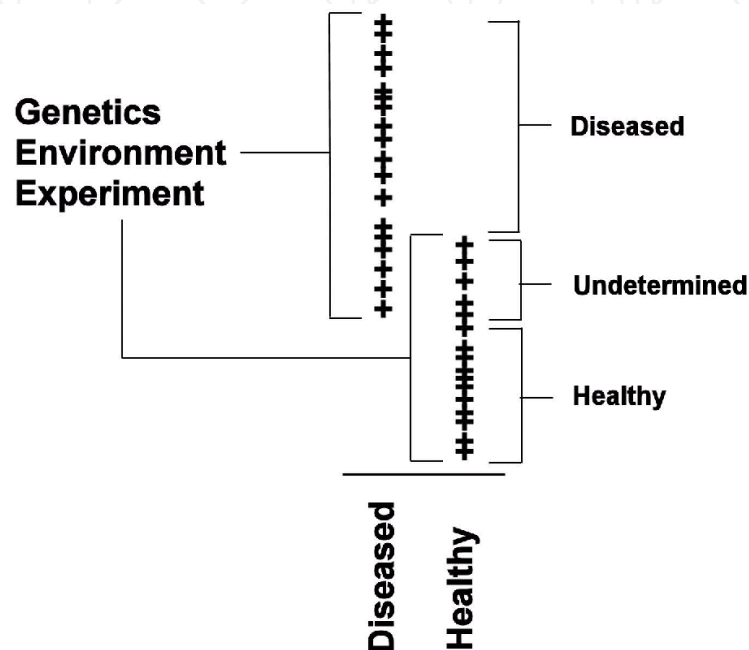


Fig. 2. Values of a state-specific marker for individuals in a disease state (left) and in a healthy state (right).

While blood concentrations of biochemicals are one possible means of examination, it is not the only one. Concentrations of biochemicals in the urine are another, but this can be extended to tears, mucous, or virtually any biofluid. Instead of directly measuring the concentration of specific compounds, mass spectra (with or without pre-fractionation) and 2D NMR of these biofluids can also be used to measure abundance. The difference with these spectral methods is that the abundance of a compound can be examined from one individual to the next without knowing the identity of this compound. Therefore, examining the intensity or area of spectral peaks is called an undirected search since a list of compounds to examine was not created before hand, while direct measurements of concentrations or intensity measurements from microarray experiments are directed searches since the search is over a set of pre-defined compounds.

In general, the set of biochemicals whose concentration is directly measured or examined by microarray analysis, as well as the set of peaks present in various spectra, are known as features. For each individual, each of these features has a corresponding value. This value can be the concentration, the logarithm of the relative fluorescence intensity, or the intensity or area of the spectral peak. The search for a putative biomarker is over the set of N available features, and each individual is represented by an array of N numbers representing the values of these features.

Since the values of many features are known for each individual, it is possible to construct classifiers using two or more features. An algorithm would search through sets of two or more features to find a set that optimally classified a given set of individuals, which is known as a training set. The goal is to maximize the number of correctly classified individuals, so if two features are used and one is that shown in Figure 2, the second feature would try to correctly resolve those in the undetermined region without upsetting the correct classification of the other individuals. Therefore, the action of this second feature in the classifier is to specifically act on those individuals in the undetermined region. The first feature (Figure 2) therefore is a state-specific marker since its intensity is largely controlled by the state of the individual, while the second feature is individual-specific since it only acts on those individuals who have an intensity of the first feature in the undetermined region.

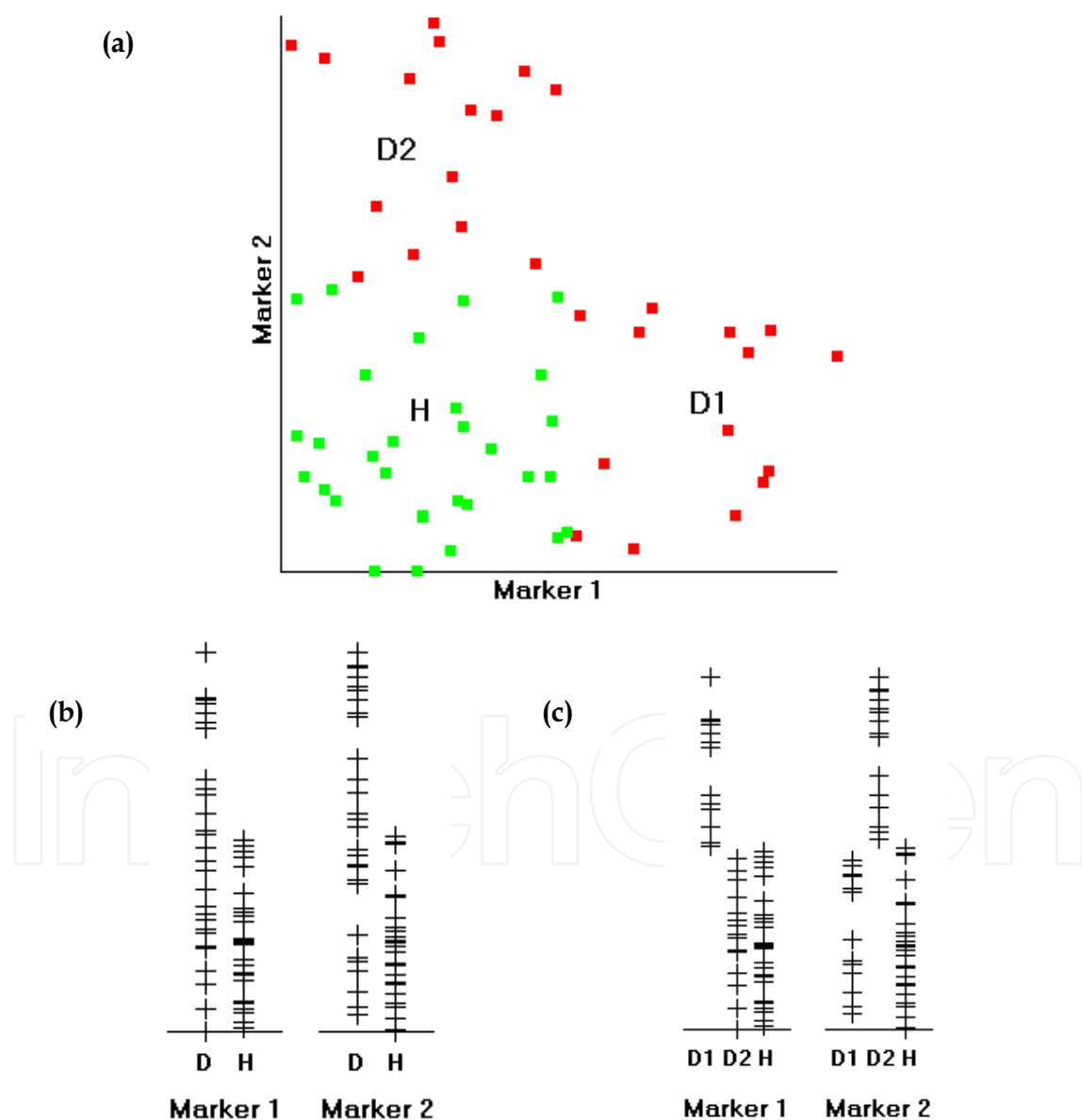


Fig. 3. (a) Scatter plot for diseased (red) and healthy (green) individuals using the values of a pair of correlated markers; (b) Values for these markers assuming one state for each category; and (c) Values for these markers assuming two disease states and one healthy state.

This argument suggests that statistical methods which find features that are significantly different in magnitude depending upon an individual's state is all that is needed to find any state-specific markers. These independent markers can be found if both the healthy and diseased categories are represented by a single state. Figure 3 displays a situation where the diseased category (shown in red) is actually composed of two states (D1 and D2). This can only be seen through the action of a concerted pair of markers, Marker 1 and Marker 2. State D1 has a high intensity in Marker 1 while State D2 has a high intensity in Marker 2, while the healthy individuals have a low intensity in both features. Figure 3 shows intensity plots for these two markers under the assumption that there is a single diseased state and a single healthy state. It is questionable whether a given statistical method would find the difference in the intensities of these features significant. Only by correctly distinguishing the state if each individual can one see that Marker 1 is a good classifier for State D1 and Marker 2 for State D2 (Figure 3).

1.6 Bias, chance and generalizability

Ransohoff [2005a, 2005b] has presented three factors that must be explored in any classification study; bias, chance and generalizability. Until now, any marker that clearly distinguishes individuals in different states, either alone or in a concerted action with another feature, is denoted as a putative biomarker. Before it can become a true biomarker one has to ensure that the marker is not due to an underlying bias. For example, if all individuals in the disease state are being given a particular drug, there is no way to determine if the change in the feature value is due to the disease or the drug. There is no way to remove this bias, and such situations should be excluded in the initial study design. As a second example, the individuals in the disease state may be significantly older than those in the healthy state. Many diseases are more prevalent in older individuals and it may be very difficult to find age-matched patients who are disease free or are not on a regular drug treatment. If a random collection of age-matched individuals without signs of the particular disease state are taken to be the healthy category, it is likely that this category will be composed of a number of states due to other diseases or drug responses. Markers separating each of these "healthy" states from the disease state would have to be found. Finding all required biomarkers would be very difficult within a single set of features. In addition, if the number of individuals in a particular healthy state was small, the significance of any biomarker may be suspect (see below). For this case, the affect of age can be examined. If there is no correlation between the feature value and the age of the individual in either the disease or healthy state, one can conclude that age is not the source of the difference in feature values.

If the available individuals in the disease and healthy states are divided into a training set and a testing set, it is theoretically possible to construct one or more classifiers using the training set that can accurately classify the individuals in the testing set without using a state-based marker. Such a classifier is a chance fit to the available data, and we have shown that accurate results can be obtained for certain classifiers without any state-specific marker being present in the set of available features. Therefore, simply constructing a good classifier is not sufficient to demonstrate the presence of a state-specific marker.

The basic assumption is that if a classifier is able to accurately classify both a training set and a testing set of data, then this classifier will be useful for all individuals in the population

from which these individuals were taken. In other words, any classifier that accurately classifies a sufficient sample from a population should be generalizable to the entire population. We assert that this assumption may be true only if the classifier is strictly composed of state-specific markers. Any classifier that is a chance fit to the available data will not be generalizable to the entire population.

1.7 Coverage, uniqueness, and significance

The simplest example of fingerprinting is a straightforward decision tree, like the one shown in Figure 4. Assuming that the entire dataset is composed of 60 diseased and 60 healthy individuals, the intensity of Feature 1 splits the dataset into two groups; 40 diseased and 20 healthy individuals if the intensity of this feature is below Cut-1 and 20 diseased and 40 healthy individuals if its intensity is above Cut-1. The left branch is further divided using Feature 2 into a diseased node (D1) that contains 38 diseased and 3 healthy individuals and a healthy node (H1) that contains 2 diseased and 17 healthy individuals. The right branch is divided using Feature 3 into a healthy (H2) and a diseased (D2) terminal node.

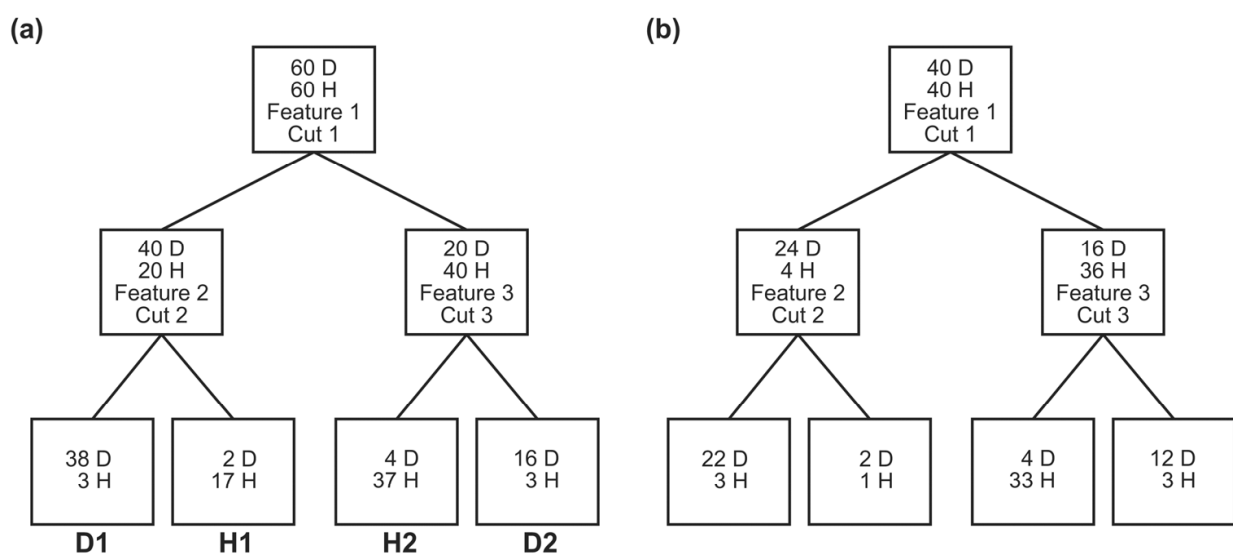


Fig. 4. (a) Hypothetical decision tree using all available data and (b) the corresponding tree when one-third of the samples are removed as testing data.

Overall this decision tree would yield a sensitivity and a specificity of 90%, but the general procedure is to divide the data into a training set and a testing set and construct the classifier using only the training data. If one-third of the data was removed to form the testing data, the situation in Figure 4b could be produced. In this example, 16 of the 20 healthy samples happened to come from H1 and 16 of 20 diseased samples from D1. This training distribution would make the use of Feature 2 unnecessary and may result in different features being used at each node. If only Features 1 and 3 were used, the training set would have a sensitivity of 90% and a specificity of 82.5%, while the testing data would have a sensitivity of 100% but a specificity of only 20%. The basic reason for this large change in sensitivity is that the fingerprint needed to describe the healthy subjects in Group H1 is no longer present in the training data.

Therefore, the first requirement of a fingerprinting method is that there must be a complete *coverage* of all required fingerprints in the training data. If a required fingerprint or proteomic pattern is missing from the training data (Figure 4), the quality of predictions for the testing data will either be greatly reduced or there will be a significant number of testing individuals that will receive an “undetermined” classification.

If a fingerprinting classifier is found that performs extremely well on classifying the training data, but classifies the testing data poorly, one can either state that the classifier is insufficient and therefore not biologically relevant, or that there was an incorrect separation of training and validation data so that effective coverage of all important fingerprints was not present in the training data. Since the discriminating fingerprints are not known, proper coverage cannot be known, and therefore proper selection of the training data cannot be known. In addition, since the quality of classifying the testing set is the metric used to determine biological relevance, the testing set is used in the process of constructing the classifier and is therefore part of the training process.

With these points in mind, an effective way to construct classifiers based on fingerprints is to include all data in the search for fingerprinting classifiers and then to selectively remove samples for the testing set in a way that preserves the coverage of the fingerprint in the training data. This statement does not suggest, in any way, that this procedure is used by other research groups who present fingerprinting classifiers, it simply states that this method is an effective way to ensure complete coverage in the training data and to effectively test for uniqueness. If Figure 4 was used as the basic classifier, all other possible three-node decision trees would have to be constructed and compared to a sensitivity and specificity of 90%. If no other three-node decision tree is found to have this overall accuracy, then the uniqueness of this classifier is established. Otherwise, each decision tree would have to be presented as a possible solution; since the important fingerprints are not known, the selection of the training set cannot be determined, and two different decision trees that imply different separations of training and validation data are therefore equally valid.

Finally, the *significance* of a fingerprinting classifier needs to be established. Permutation testing is often used to test significance, but can be used in three different ways. In the Random Forest algorithm [Breiman, 2001] the intensities of a given feature are scrambled among all data in each testing set (i.e. the out-of-bag samples) to determine the importance of that feature. The phenotypes of the samples can also be scrambled a large number of times to determine the probability that the accuracy of a given classifier occurred by chance. In this application, the phenotypes will be scrambled amongst all data to determine if a new classifier of the same form (e.g. a three-node decision tree) can be constructed with comparable accuracy. The probability that random phenotypes can be classified to a given accuracy determines the significance of a given model.

1.8 Proposed study

To test the classification ability of different algorithms, this study will attempt to build classifiers from sets of 300 possible features. In each case, the intensities of the features will be determined using a random number generator. In other words, each classifier will attempt to distinguish healthy samples from diseased ones using data that contains no information. Results using DT and MCA classifiers have been previously presented [Luke &

Collins, 2008], but these will be included here to compare to SVM, LDA, and biomarker-based classifiers. Exhaustive searches using DT, MCA, SVM and LDA are not computationally feasible, so the results presented here represent a lower bound to the accuracy that can be obtained from these methods with data that contains no information. It should also be stressed that 300 features is a very small number by current methods of analysis of biological samples, and the accuracy of all methods will not decrease as the number of features increases.

2. Methods

2.1 Decision tree

For the symmetric, 7-node decision tree shown in Figure 1, a modified Evolutionary Programming (mEP) procedure is used. Each putative decision tree classifier is represented by two 7-element arrays; the first contains the feature used at each node and the second contains the cut values. Both arrays assumed the node ordering listed in Figure 1. The only caveats are that all seven features must be different and that this ordered septet of features cannot be the same as any other putative solution in either the parent or offspring populations. When a new putative decision tree is formed, a local search is used to find optimum cut points for this septet of features.

The mEP procedure starts by randomly generating 2000 unique decision trees. Each decision tree has one or two of the features removed and unique features are selected, again requiring that the final septet is unique. The local search first tries to find optimum cut points for the new features that were added and then the search is over all seven cut points. The best set of cut points is combined with the septet of features to represent an offspring classifier. The score is the sum of the sensitivity and specificity for the training individuals over the eight terminal nodes. When the entire set of initial, or parent, decision trees have generated unique offspring, all 4000 scores are compared and the 2000 decision trees with the best score become parents for the next generation. This process is repeated for a total of 4000 generations and the best classifiers in the final population are examined.

2.2 Mediod classification algorithm

While the algorithm described by Petricoin and Liotta [Browsers et al., 2005; Conrads et al., 2004; Ornstein et al., 2004; Petricoin et al., 2004; Srinivasan et al., 2006; Stone et al., 2005] used a genetic algorithm driver to search for an optimum set of features, allowing for different putative solutions to use different numbers of features (5-20 features), our algorithm uses a mEP feature selection algorithm and all putative solutions have the same number of features n . For a given value of n , n features were selected and the intensities of these features were rescaled for each individual using the following formula [Browsers et al., 2005; Conrads et al., 2004; Ornstein et al., 2004; Petricoin et al., 2004; Srinivasan et al., 2006; Stone et al., 2005]:

$$I' = (I - I_{\min}) / (I_{\max} - I_{\min}) \quad (1)$$

In this equation, I is a feature's original intensity, I' is its scaled intensity, and I_{\min} and I_{\max} are the minimum and maximum intensities found for the individual among the n

selected features, respectively. If I_{\min} and I_{\max} were from the same features in all samples, a baseline intensity would be subtracted and the remaining values scaled so that the largest intensity was 1.0. Each individual would then be represented as a point in an $(n-2)$ -dimensional unit cube. As designed, and as found in practice, I_{\min} and I_{\max} do not represent the same features from one individual to the next, so this interpretation does not hold. Therefore, each individual represents a point in an n -dimensional unit cube.

As stated in the Background, the first training sample becomes the medoid of the first cell, with this cell being classified as the category of this sample. Each cell has a constant trust radius r , which is set to $0.1(n)^{1/2}$, or ten percent of the maximum theoretical separation in this unit hypercube. If the second sample is within r of the first, it is placed in the first cell; otherwise it becomes the medoid of the second cell and that cell is characterized by the second sample's category. This iteration continues until all training samples are processed. Each cell is then examined and the categories of all samples in the cell are compared to the cell's classification. This calculation allows a sensitivity and specificity to be determined for the training data, and their sum represents the score for this set of n features.

The mEP algorithm initially selects 2000 sets of n randomly selected features. The only caveat is that each set of n features must be different from all previously selected sets. The medoid classification algorithm then determines the score for each set of features. Again, each parent set of features generates an offspring set of features by randomly removing one or two of the features and replacing them with randomly selected features, requiring that this set be different from all feature sets in the parent population and in all offspring generated so far. The score of this feature set is determined and the score and feature set is stored in the offspring population. After all 2000 offspring have been generated the parent and offspring populations are combined. The 2000 feature sets with the best score are retained and become the parents for the next generation.

It should be noted that for a set of n features, the number of unique cells that can be generated is on the order of 10^n . Since no training set is ever this large (n is 5 or more), only a small fraction of the possible cells will be populated and classified. As will be shown in the next section, this limitation causes a significant number of the testing samples to be placed in an unclassified cell, though none of the publications that used this method [Browsers et al., 2005; Conrads et al., 2004; Ornstein et al., 2004; Petricoin et al., 2004; Srinivasan et al., 2006; Stone et al., 2005] reported an undetermined classification for any of the testing samples. Instead of searching through a large number of solutions that classified the training samples to a significant extent and find those that minimized the number of unclassified testing samples, we decided to use all samples and limit the number of cells. All samples were placed in the training set and the algorithm was run with the added requirement that any set of n features that produced more than a selected number of cells was given a score of zero. If the number of healthy and disease medoids are sufficiently small, all other samples could then be divided to place the required number in the testing set and the remainder would be part of the training.

2.3 Support Vector Machine

A support Vector Machine (SVM) [Boser et al., 1992; Vapnik, 1998] is a kernel-based learning system. SVM searches for the optimal hyperplane that maximizes the margin of separation between the hyperplane and the closest data points on both sides of the hyperplane.

Many features in a genomic or proteomic data are irrelevant or redundant that may likely hinder the performance of a classifier. It is essential to select informative features to build a classifier. A new selection criterion is presented with a performance found to be better than or comparable to the other criteria and is applied to LDA and linear SVM as a classification method.

Support Vector Machines (SVM) [Boser, 1992; Vapnik, 1998], are becoming increasingly popular in biological problems [Noble, 2004]. SVM finds the optimal hyperplane that maximizes the margin of separation between the hyperplane and the closest data points on both sides of the hyperplane. Instead of error-risk minimization, the parameters of SVMs are determined on the basis of structural risk minimization. Thus, they have the tendency to overcome the overfitting problem. SVMs have been successful with a recursive procedure in selecting important features for cancer prediction [Guyon et al., 2002; Tang et al., 2007].

The decision functions (used to determine the class of a sample) of SVM and LDA can be expressed as a linear combination of features. They differ with regard to how the weights are determined. The weights (coefficients in the decision function) of features, which reflect the significance of the features for classification, can be served as a feature ranking criterion. This criterion corresponds to removing a feature whose elimination changes the objective function least [LeCun et al., 1990]. The criterion has been used with a recursive feature elimination scheme [Guyon et al., 2002], as described before.

Instead of judging a feature by its contribution to the classification on the full dataset, this study uses a leave-one-out cross-validation to evaluate a feature's contribution to the ensemble of classifiers. In other words, a classifier is re-trained on a new dataset formed by removing a sample from original dataset to obtain a weight for every feature. If a feature is important in differentiating samples, it should remain so when any sample is removed from a dataset. This can be indicated by the coefficient of variation of the weight value for each feature. The coefficient of variation is defined as the ratio of the standard deviation to the mean. A small coefficient of variation indicates smaller variation and a more consistent contribution of a feature to the sample classification. There are two ways to incorporate this criterion into the recursive selection process. One is to pre-select the number of iterations and the number of features at each iteration. This can be implemented by determining the coefficient of variation for each feature in current feature set and selecting k features with smallest coefficient of variation, where k is the predefined number of features for this iteration. In the second implementation, the number of iterations and the number of features at each iteration are determined during the selection process. It starts with all the features and can be described as follow:

- Step 1. Compute the coefficient of variation for each feature in current feature set. In every selection cycle, the procedure initially eliminates at least certain number of features. In this study, 10% of the current features, or 1 whichever is larger, with largest coefficient of variation are eliminated.
- Step 2. Let c_{min} denote the minimum coefficient of variation and c_{max} denote the maximum coefficient of variation among the remaining features.
- Step 3. Select k coefficient of variation, c_1, c_2, \dots, c_k , such that $c_{min} < c_1 < c_2 < \dots < c_k = c_{max}$ and c_1, c_2, \dots, c_k divided the interval $[c_{min}, c_{max}]$ into k subintervals of equal lengths except possibly the interval $[c_{k-1}, c_{max}]$. In this study, we choose $k=8$.

- Step 4. Estimate, for each c_i , $i=1,2,\dots,k$, the performance of a classifier, which uses all the features whose coefficient of variation is less than or equal to c_i , from a cross-validation, such as, leave-one-out cross-validation.
- Step 5. Find the smallest c_i from c_1, c_2, \dots, c_k which gives the best performance in Step 3.
- Step 6. Choose all the features from current feature set whose coefficient of variation are less than or equal to c_i as the feature subset for this selection cycle.

The selection process is repeated until only one feature remains.

2.4 Linear Discriminate Analysis

Linear Discriminant Analysis (LDA) [Fukunaga, 1990] is a supervised learning algorithm that finds the linear combination of features that maximize the between-class scatter and simultaneously minimize the within-class scatter to achieve maximum discrimination in a dataset. The within-class scatter matrix may become singular if the sample size is smaller than the dimensionality of the search space (number of features). To overcome the singularity problem, the pseudo-inverse [Golub & Van Loan, 1983] of the within-class scatter matrix is computed in this study.

The computation of a pseudo-inverse in LDA may be demanding if the dimension of within-class scatter matrix is too large. In this study diagonal LDA is used, which is the same as LDA except that the covariance matrices are assumed to be diagonal. The diagonal LDA has been reported to be performed remarkably well compared to more sophisticated methods [Dudoit et al., 2002]. A leave-one-out procedure is again used to determine the coefficient of variation for all remaining features, and procedure outlined above is used to reduce the feature set.

2.5 Biomarker Discovery Kit

The BioMarker Discovery Kit (BMDK) represents a suite of programs with the eventual goal of constructing one or more biomarker-based classifiers. Each biomarker represents a particular feature that is associated with a particular disease state represented by a subset of the available individuals. BMDK uses 10 different methods of analysis to identify putative biomarkers. These methods determine how well each feature distinguishes some or all of the individuals in a given histology. Descriptions of each filtering method are given elsewhere [<http://isp.ncifcrf.gov/abcc/abcc-groups/simulation-and-modeling/biomarker-discovery-kit/>]. The union of all features that have one of the top five scores for each of the 10 methods produces the set of putative biomarkers.

A single biocompound may produce more than one putative biomarker if the features are obtained from a mass spectroscopic investigation. For example, separate peaks for the +1 and +2 ion or the biocompound alone and complexed with another compound are possible. Therefore, the Pearson's correlation coefficient between all pairs of putative biomarkers across all samples is used to combine the putative biomarkers into groups. All other features in the dataset are then compared to the putative biomarkers within each group and are selected for examination if the correlation coefficient is 0.70 or higher. Each group is then represented by the single feature with the largest maximum value; all other features are discarded.

The final classifier is based on a distance-dependent K-nearest neighbor (DD-KNN) algorithm. In this classifier the un-normalized probability that an unknown sample belongs to the same group as a neighbor is given by the inverse of their distance. To account for the situation where an unknown sample has no nearest neighbors, the classifier also contains a probability that its group is unknown. This probability linearly increases from 0.0 to 0.01 as the probability of being in a neighbors group decreases from 1.0 to 0.8. For smaller probabilities of belonging to the neighbor's group, the probability of being unknown stays constant at 0.1. These probabilities are summed over all neighbors and scaled to a total probability of 1.0. Therefore, each unknown sample is described by a probability of belonging to Group-1 (e.g. healthy), Group-2 (e.g. diseased), and Undetermined. The final classification is given by a probability of membership of at least 0.5, or Undetermined if the probabilities of belonging to either group is less than 0.5.

All of the putative biomarkers are individually used to find the best 1-feature DD-KNN algorithm, and this is followed by an exhaustive search over all sets of two and three putative biomarkers. In practice, six nearest neighbors are generally used but this number can be increased if there are a large number of samples; the number of neighbors should not decrease below six. The quality of the classifier is determined using a leave-one-out procedure since this method preserves the coverage (range of intensities) for the samples to the greatest extent. Each time an optimum classifier is found, the distribution of samples in feature-space is plotted to determine the number of disease state present for each category.

3. Results

Since the datasets examined in this investigation are produced using a random number generator, the goal is to simply determine a lower-bound to the accuracy that can be obtained for 300 features for different numbers of Cases and Controls. Since these labels really have no meaning, the accuracy of a classifier will be given by the sum of the sensitivity and specificity. These are lower bounds since only five different datasets are examined for each Case/Control combination, and for all methods but BMDK only a small fraction of all possible feature-combinations are explored.

3.1 DT and MCA classifiers

For the DT and MCA classifiers, it is assumed that the dataset is divided such that two-thirds of the samples in each group are used in the training set and one-third is used in the testing set. The accuracy of the classifier is the sum of the sensitivity and specificity of the testing set.

For the DT algorithm, all samples are used in the construction of the decision tree. After the best decision tree is constructed from the evolutionary programming search over ordered sets of seven features, one-third of the samples are removed to build the testing set. This is done in a way that does not change the description of each terminal node (i.e. it stays as either a healthy or diseased node) and the sensitivity and specificity of the training and testing sets are approximately equal. This may appear to be cheating, but the goal of this investigation is to determine the minimum accuracy that could be obtained from data that contains no information.

The best quality from the five datasets for each number of Cases and Controls is given in Table 1. For 30 Cases and 30 Controls, a 7-node decision tree was able to correctly

classify all 60 samples. In other words, all eight terminal nodes only contained samples from a single group. For 60 Cases and 60 Controls, a decision tree correctly classifies over 89% of the testing samples, and for 90 Cases and Controls, over 83% of the testing samples were correctly classified. It is only when the number of samples is as large or larger than the number of features that the DT classifier drops below 80% accuracy for the testing data.

Cases and Controls	DT	MCA		
		5-Feat	6-Feat	7-Feat
30	200.0	200.0	200.0	200.0
42	190.5	197.6	197.6	197.6
60	178.3	193.3	193.3	195.0
90	166.7	187.8	188.9	191.1
150	155.3	183.3	185.3	187.3
300	138.3	170.3	179.0	180.3

Table 1. Highest quality obtained using absolute differences in un-scaled peak intensities, from a decision tree (EPDT) and the medoid classifier algorithm (MCA). *Note:* Taken from [Luke & Collins, 2008] where the quality is the sum of the sensitivity and specificity.

The classifier constructed by the MCA is order dependent in that the first training sample automatically becomes the medoid of the first region. Therefore, this analysis uses all samples to construct a classifier with the requirement that the samples used as medoids cannot exceed two-thirds of either the Cases or Controls. One-third of the samples from each group are then selected as testing samples and are chosen such that the accuracy of the training set is at least as high as the testing set.

The results in Table 1 show that an MCA classifier performs excellently using datasets that contain no information. If only seven of the 300 features are used, one can find a classifier with an average sensitivity and specificity of over 90% even when there are 300 Cases and 300 Controls (200 of each in the training set and 100 of each in the testing set). If the number of features is reduced to five or six, the accuracy stays above 90% for all but the largest datasets.

3.2 SVM and LDA classifiers

The results for the SVM and LDA classifiers are shown in Table 2. As described above, all samples are used to determine which features will be used in the classifier, but the accuracy of the final classification uses the sum of the sensitivity and specificity of the testing set for 10-fold cross-validation, averaged over 100 runs where the order of the samples are scrambled before each run. SVM has an average classification accuracy over 97% when there are 90 or fewer Cases and Controls in the dataset. For LDA, the average accuracy stays above 99.7%.

For the larger datasets, the accuracy decreases significantly. When there are 150 Cases and 150 Controls, the SVM and LDA classifiers have an average accuracy of about 87 and 84%,

respectively. This is better than a 7-node decision tree, but not as good as any of the MCA classifiers examined. Datasets that contain 300 Cases, 300 Controls, and only 300 features have SVM and LDA accuracies of 61.3 and 63.7% respectively, which is below the accuracy of the DT and MCA classifiers.

Cases and Controls	SVM	LDA
30	198.50	199.06
42	199.88	199.84
60	199.02	199.68
90	194.82	199.52
150	174.38	167.90
300	132.66	135.44

Table 2. Highest quality obtained using normalized feature values from support vector machine (SVM) and linear discriminate analysis (LDA) classifiers. *Note:* The quality is the sum of the sensitivity and specificity for the testing set averaged over 100 runs of 10-fold cross-validation.

3.3 BMDK Classifier

The accuracy of the BMDK classifier is shown in Table 3. This accuracy is determined from a leave-one-out cross-validation, a procedure that is known to exaggerate the accuracy of a classifier. After each sample is classified, the overall accuracy is the sum of the sensitivity and specificity minus the percentage of samples that were classified as unknown. For the smallest dataset (30 Cases and 30 Controls) a 3-feature DD-KNN classifier correctly classified 78.9% of the samples. In general, the accuracy decreased as the number of samples increased.

Cases and Controls	BMDK		
	1-Feat	2-Feat	3-Feat
30	147.4	153.3	157.8
42	142.9	136.4	137.3
60	151.7	140.0	140.1
90	123.3	137.9	137.3
150	118.0	127.3	125.3
300	115.7	121.7	122.1

Table 3. Highest quality obtained from the BioMarker Development Kit (BMDK) using absolute differences in un-scaled peak intensities. *Note:* The quality is the sum of the sensitivity and specificity minus the percent of samples undetermined using a leave-one-out cross-validation of the entire set of data.

The only exception was for one dataset with 60 Cases and 60 Controls. This increased accuracy was due to an unusual pattern in one of the randomly generated features. The intensities for this feature are shown in Figure 5, where the “+” marks in the left column are the intensities of the 60 samples in Group-1 and the marks in the right column are for the 60 samples in Group-2. While there is no overall difference between these columns, a closer examination shown a clumping of intensities in one group at values that have gaps in the other group.

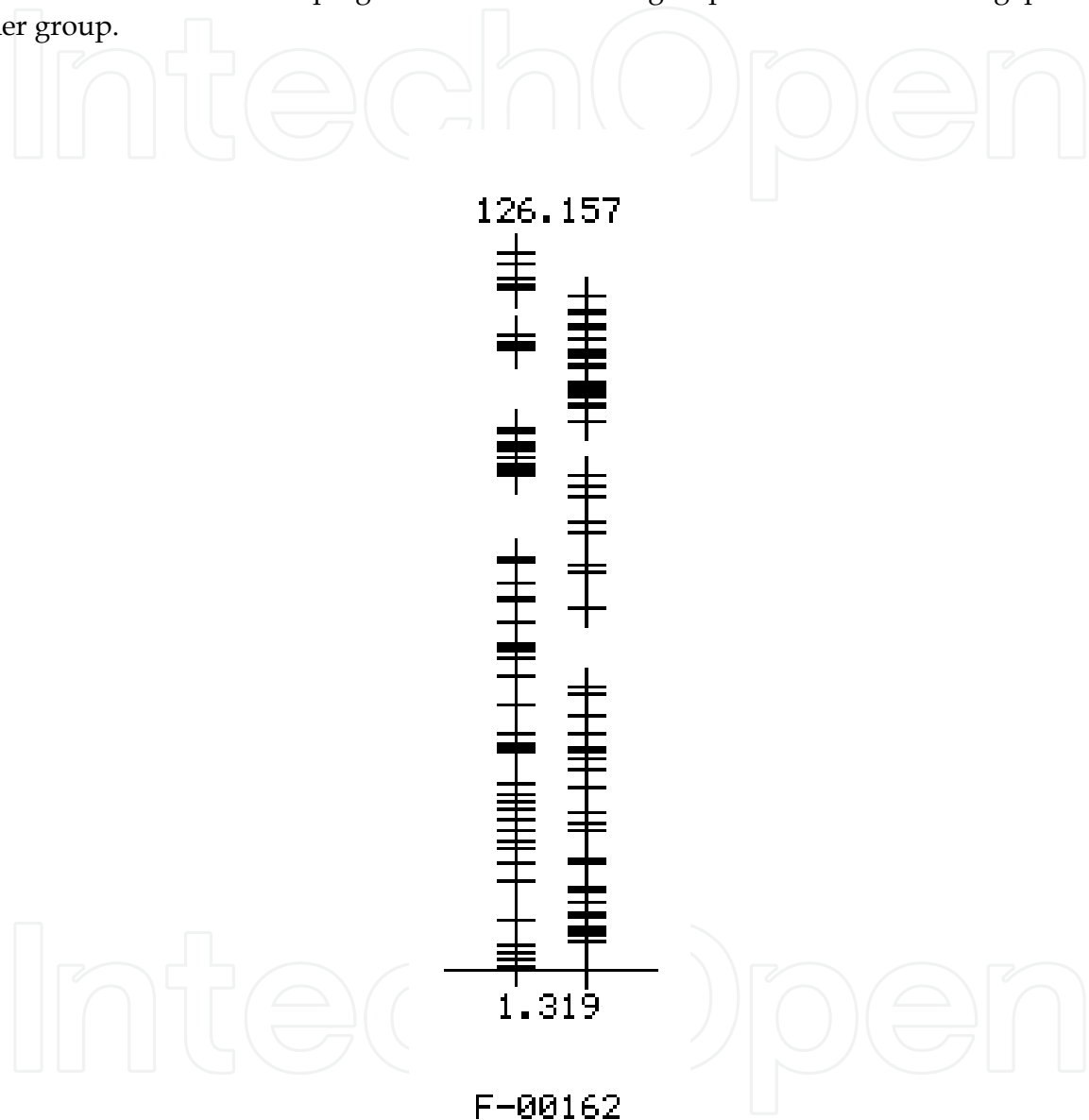


Fig. 5. Intensities for the 60 cases (left column) and 60 controls (right column) for the peak that yielded a quality score of 151.7 (sensitivity=78.3%, specificity=73.3%, undetermined=0.0%) in the dataset of random peak intensities.

In many cases the accuracy of a 3-feature classifier is not significantly better than a 2-feature classifier. This is due to the fact that as the dimensionality of the classification space

increases, the separation between the samples becomes larger. This causes more samples to be classified as Undetermined. For this reason, no 4-feature classifier did better than the 3-feature classifier in any of the 30 datasets.

4. Discussion

The results presented in the tables above show that very good results can be obtained from DT, MCA, SVM and LDA classifiers for datasets that contain no information. It can be argued that the procedures used here are selected to obtain the maximum possible accuracy, and that is exactly the point. If a 7-node decision tree used 40 Cases and 40 Controls in the training set and 20 Cases and 20 Controls in the testing set and obtained an accuracy of 87.5% for the testing samples, one could propose that the set of seven features denotes a fingerprint that accurately classifies the samples. The results in Table 1 show that this accuracy can be obtained from a dataset with only 300 randomly generated feature values for each sample. A 7-feature MCA classifier is able to achieve an average accuracy of over 90% when the dataset contains 300 Cases, 300 Controls, and only 300 non-informative features. This should draw into question the results of any study that uses this classification method.

SVM and LDA classifiers have testing set accuracies above 97.4 and 99.7%, respectively, for all but the largest datasets. It is only when the number of samples is at least as large as the number of features that these methods break down. Current methods for obtaining information from biological samples generate many more features than the 300 used here.

The BMDK classifier did not achieve an average accuracy above 80% for even the smallest dataset. This result is not unexpected. Since the datasets do not contain any information, there are no biomarkers and a biomarker-based classifier should not perform well. Fortuitous results can be obtained and a closer examination of the putative biomarkers should be performed (Figure 5).

For the DT and MCA methods there is some selection of which samples should be placed in the training and testing sets, but this is basically what is required because of the coverage problem. If a given terminal node in a DT classifier contains 7 Cases and 4 Controls, and 4 of the Cases were moved to the testing set, this terminal node would change from a Case-node to a Control-node and the classification accuracy of the testing data would be decreased. The MCA classifier is based on the premise of a fingerprint that associates a sample in the testing set with a sample in the training set. If that sample were removed from the training set, the association could not be made and the accuracy of the classifier would be decreased.

5. Conclusions

The results presented here show that very good classification results can be obtained from DT, MCA, SVM, and LDA classifiers, even if the dataset contains no information. Studies using any of these methods should carefully examine whether the results are due to some underlying biology or are just fortuitous. Performing comparable examinations on randomly generated feature values, or performing analysis of the same data after the

group labels of the samples has been scrambled, should be a necessary part of these investigations.

In contrast, a biomarker-based classifier such as BMDK should perform poorly if the dataset contains no biological information. Even when reasonably good results are obtained, the putative biomarkers used in the classifier should be carefully examined to ensure that they really distinguish samples in one group from the other.

6. Acknowledgement

The authors would like to thank Dr. Jih-H Chen who constructed the recursive SVM and LDA programs and generated their results. This work was funded in whole with federal funds from the National Cancer Institute, National Institutes of Health, under Contract No. HHSN261200800001E and the Center for Biomedical Informatics and Information Technology (CBIT)/Cancer Biomedical Informatics Grid (caBIG) ISRCE yellow task #09-260 to NCI-Frederick. The content of this publication does not necessarily reflect the views or policies of the Department of Health and Human Services, nor does mention of trade names, commercial products, or organizations imply endorsement by the United States Government.

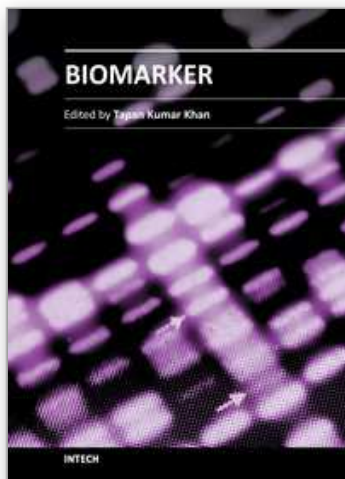
7. References

- Bellman, R.E. & Rand Corporation (1957). *Dynamic Programming*, Princeton University Press
- Bellman, R.E. (1961). *Adaptive Control Processes: A Guided Tour*, Princeton University Press
- Bellman, R.E. (2003). *Dynamic Programming*, Courier Dover Publications
- Boser, B., Guyon, I. & Vapnik, V. (1992). A Training Algorithm For Optimal Margin Classifiers, In: *Proceedings of the 5th Annual Workshop on Computational Learning Theory*, pp. 114-152, New York, NY, USA
- Breiman, L. (2001). Random Forests. *Machine Learning*, Vol. 45, pp. 5-32
- Browsers, F.M., Petricoin, E.F. 3rd, Ksinantova, L., Breza, J., Rajapakse, V., Ross, S., et al. (2005). Low Molecular Weight Proteomic Information Distinguishes Metastatic From Benign Pheochromocytoma. *Endocrine-Related Cancer* Vol. 12, pp. 263-72
- Conrads, T.P., Fusaro, V.A., Ross, S., Johann, D., Rajapakse, V., Hitt, B.A., et al. (2004). High-Resolution Serum Proteomic Features For Ovarian Cancer Detection. *Endocrine-Related Cancer*, Vol. 11, pp. 163-78
- Dudoit, S., Fridlyand, J., & Speed, T.P. (2002). Comparison of Discrimination Methods For Classification of Tumors Using Gene Expression Data. *Journal of the American Statistical Association*, Vol. 97, No. 457, pp. 77-87
- Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*, Academic Press, 2nd Edition, Boston, Mass, USA
- Golub, G. & Van Loan, C. (1983). *Matrix Computations*. The John Hopkins University Press, Baltimore, USA

- Guyon, I., Weston, J., & Barnill, S. (2002) Gene Selection for Cancer Classification Using Support Vector Machines. *Machine Learning*, Vol. 46, No. 1, pp. 389-422
- Ho, D.W., Yang, Z.F., Wong, B.Y., Kwong, D.L., Sham, J.S., Wei, W.I., et al. (2006). Surface-Enhanced Laser Desorption/Ionization Time-of-Flight Mass Spectrometry Serum Protein Profiling to Identify Nasopharyngeal Carcinoma. *Cancer*, Vol. 107, pp. 99-107
- Kohonen, T. (1988). *Self-Organizing and Associative Memory*, 2nd ed., New York, Springer-Verlag
- LeCun, Y., Denker, J., & Solla, S. (1990). Optimal Brain Damage, In *Advances In Neural Information Processing System II*, Morgan Kaufmann, San Mateo, CA, USA
- Liu, W., Guan, M., Wu, D., Zhang, Y., Wu, Z., Xu, M., & Lu, Y. (2005). Using Tree Analysis Pattern and SELDI-TOF-MS to Discriminate Transitional Cell Carcinoma of the Bladder Cancer From Noncancer Patients. *European Urology*, Vol. 47, pp. 456-62
- Luke, B.T., & Collins, J.R. (2008). *Examining the Significance of Fingerprint-based Classifiers*. *BMC Bioinformatics*, Vol. 9, No. 1, (December 17), p. 545
- Noble, W. (2004). Support Vector Machine Applications in Computational Biology, In: *Kernel Methods in Computational Biology*, pp. 71-92, MIT Press, Cambridge, Mass, USA
- Ornstein, D.K., Rayford, W., Fusaro, V.A., Conrads, T.P., Ross, S.J., Hitt, B.A., et al. (2004). Serum Proteomic Profiling Can Discriminate Prostate Cancer From Benign Prostates In Men With Total Prostate Specific Antigen Levels Between 2.5 and 15.0 ng/ml. *Journal of Urology*, Vol. 172, pp. 1302-1305
- Petricoin, E.F., Fishman, D.A., Conrads, T.P., Veenstra, T.D., & Liotta, L.A. (2003). Proteomic Pattern Diagnostics: Producers and Consumers in the era of correlative science. *BMC Bioinformatics*, Vol. 4, p. 24/comments
- Petricoin, E.F., Rajapakse, V., Herman, E.H., Arekani, A.M., Ross, S., Johann, D., et al. (2004). Toxicoproteomics: Serum Proteomic Pattern Diagnostics For Early Detection of Drug Induced Cardiac Toxicities and Cardioprotection. *Toxicologic Pathology*, Vol. 32, Suppl. 1, pp. 122-30
- Ransohoff, D.F. (2005a). Lessons From Controversy: Ovarian Cancer Screening and Serum Proteomics. *Journal of the National Cancer Institute*, Vol. 97, pp. 315-319
- Ransohoff, D.F. (2005b). Bias as a Threat to the Validity of Cancer Molecular-marker Research. *Nature Reviews Cancer*, Vol. 5, pp. 142-149
- Srinivasan, R., Daniels, J., Fusaro, V., Lundqvist, A., Killian, J.K., Geho, D., et al. (2006). Accurate Diagnosis of Acute Graft-versus-host Disease Using Serum Proteomic Pattern Analysis. *Experimental Hematology*, Vol. 34, pp. 796-801
- Stone, J.H., Rajapakse, V.N., Hoffman, G.S., Specks, U., Merkel, P.A., Spiera, R.F., et al. (2005). A serum proteomic approach to gauging the state of remission in Wegener's granulomatosis. *Arthritis & Rheumatism*, Vol. 52, pp. 902-10
- Tang, Z., Han, L., Lin, H., et al. (2007). Derivation of Stable Microarray Cancer Differentiating Signatures Using Consensus Scoring of Multiple Random Sampling and Gene-ranking Consistency Evaluation. *Cancer Research*, Vol. 67, No., 20, pp. 9996-10003

- Vapnik, V.N. (1998). *Statistical Learning Theory*, John Wiley & Sons, New York, NY, USA
- Yang, S.Y., Xiao, X.Y., Zhang, W.G., Zhang, L.J., Zhang, W., Zhou, B., et al. (2005). Application of Serum SELDI Proteomic Patterns in Diagnosis of Lung Cancer. *BMC Cancer*, Vol. 5, p. 83
- Yu, Y., Chen, S., Wang, L.S., Chen, W.L., Guo, W.J., Yan, H., et al. (2005). Prediction of Pancreatic Cancer by Serum Biomarkers Using Surface-enhanced Laser Desorption/Ionization-based Decision Tree Classification. *Oncology*, Vol. 68, pp. 79-86

IntechOpen



Biomarker

Edited by Prof. Tapan Khan

ISBN 978-953-51-0577-0

Hard cover, 392 pages

Publisher InTech

Published online 27, April, 2012

Published in print edition April, 2012

Clinicians, scientists, and health care professionals use biomarkers or biological markers as a measure of a person's present health condition or response to interventions. An ideal -biomarker should have the following criteria: (I) ability to detect fundamental features of the disease, (II) ability to differentiate from other closely related diseases, (III) ability to detect early stages and stages of progression, (IV) the method should be highly reliable, easy to perform and inexpensive, and (V) sample sources should be easily accessible from body. Most of the chapters in this book follow the basic principle of biomarkers.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Brian T. Luke and Jack R. Collins (2012). A Comparison of Biomarker and Fingerprint-Based Classifiers of Disease, Biomarker, Prof. Tapan Khan (Ed.), ISBN: 978-953-51-0577-0, InTech, Available from: <http://www.intechopen.com/books/biomarker/a-comparison-of-biomarker-and-fingerprint-based-classifiers-of-disease>

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

© 2012 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the [Creative Commons Attribution 3.0 License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IntechOpen

IntechOpen