# PATTERN RECOGNITION ASSISTED INFRARED LIBRARY SEARCHING

By

KADAMBARI NUGURU

Bachelor of Engineering in Electronics and

Communication Engineering

Osmania University

Hyderabad, Andhra Pradesh, India

2006

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
May, 2009

**PATTERN RECOGNITION ASSISTED INFRARED**

**LIBRARY SEARCHING**

Thesis Approved:

Dr. Alan Cheville
Associate Professor in Electrical and Computer Engineering
Thesis Adviser

Dr. R.G. Ramakumar
Regents Professor, PSO/Albrecht Naeter Professor in
Electrical and Computer Engineering

Dr. Weihua Sheng
Assistant Professor in Electrical and Computer Engineering

Dr. Barry K. Lavine
Associate Professor in Department of Chemistry

Dr. A. Gordon Emslie
Dean of the Graduate College

## TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

# Chapter 1

# INTRODUCTION

Pattern recognition has its origins in the field of image and signal processing where techniques were developed to categorize samples on the basis of regularities in the observed data. Personal computers now make it possible for these methods to be applied on a regular basis to text classification [1], oil spill identification [2], and speech recognition [3]. The first applications of pattern recognition to chemistry were studies involving low resolution mass spectrometry [4-8] which appeared in the chemical literature in 1969. Since then, pattern recognition has been applied to problems involving Raman spectroscopy [9, 10], liquid chromatography [11, 12], and nuclear magnetic resonance spectroscopy [13, 14].

Pattern recognition techniques are well suited for analyzing chemical data because of the characteristics of the procedures. No exact functional form is fitted to the data; rather, relationships are sought which provide definitions of similarity among diverse groups of data. In essence, pattern recognition techniques can be thought of as providing relations that uncover common properties. Once such relationships are developed, they may be used to infer the properties of objects that were not part of the original data set. These techniques are also capable of dealing with high-dimensional data where more than three measurements are used to represent each object. Furthermore, pattern recognition methods can handle data from multiple sources where each measurement can be the result of a separate independent experiment. Finally, techniques are available for selecting important features from a large set of parameters. Thus, studies can be performed on systems where the exact relationships are not fully understood.

Pattern recognition methods were originally designed to solve the class membership problem (e.g., differentiating between two classes, diabetes/normal). In a typical pattern recognition study, objects or samples are classified according to a specific property using measurements that are indirectly related to that property. An empirical relationship or classification rule is developed from a set of samples for which the property of interest and the measurements are known. The classification rule is then used to predict this property in samples that are not part of the original training set. The property in question may be the type of jet fuel responsible for an underground fuel spill, and the measurements are the areas of selected peaks from a gas chromatogram. The set of samples for which the property of interest and measurements are known is called the training set, whereas the set of measurements that describe each sample in the data set is called a pattern. The determination of the property of interest by assigning a sample to its respective class is called recognition, hence the term “pattern recognition.”

In pattern recognition analysis, each object is represented as a point in a high dimensional space. The number of dimensions of the space corresponds to the number of descriptors (measurements) that are available for each object or sample. A basic assumption is that distances between pairs of points are inversely related to their degree of similarity. Thus, points representing objects from one class will cluster in a limited region of this space distant from the points corresponding to the other class. Pattern recognition is a set of methods for investigating data represented in this manner to assess the degree of clustering and general structure of the data. The four main subdivisions of

pattern recognition methodology are mapping and display, clustering, discriminant development, and modeling.

The development of a genetic algorithm (GA) for pattern recognition analysis of infrared data is the focus of this thesis. The pattern recognition GA selects features that optimize the separation of the classes in a plot of the two or three largest principal components of the data. Because the largest principal components capture the bulk of the variance in the data, the features chosen by the GA primarily convey information about differences between the classes in the data set. Hence, the principal component analysis routine embedded in the fitness function of the GA serves as an information filter, significantly reducing the size of the search space since it restricts the search to feature subsets whose principal component plots show clustering of the spectra on the basis of the class label of the samples. In addition, the algorithm focuses on those classes and or samples that are difficult to classify as it trains using a form of boosting to modify the class and sample weights. Boosting addresses the problem of convergence to a local optimum since the fitness function of the pattern recognition GA will change as the population evolves towards a solution. Samples that consistently classify correctly are not as heavily weighted as samples that are more difficult to classify. Over time the algorithm learns its optimal parameters in a manner similar to a neural network. The proposed algorithm integrates aspects of artificial intelligence and evolutionary computations to yield a smart one-pass procedure for feature selection and pattern recognition.

The efficacy, flexibility, and efficiency of the pattern recognition GA, as an engine for knowledge discovery, has been investigated through analysis of infrared

spectral library data to discover the spectral features that provide the desired discrimination between different functional groups and properties of materials. The wavelet packet transform is used to denoise and deconvolute the spectra by decomposing each spectrum into wavelet coefficients that represent both the high and low frequency components of the signal. This decomposition process is iterated through successive wavelet packets until the required level of signal decomposition is achieved. The genetic algorithm for pattern recognition analysis is used to identify the wavelet coefficients that can classify the IR spectra by the functional group of the compound.

# Chapter 2

# Pattern Recognition

## 2.1 Introduction

Many relationships in multivariate chemical data cannot be expressed in quantitative terms. These relationships are better expressed in terms of similarity and dissimilarity among diverse groups of data. The task which confronts a scientist or engineer when investigating these types of relationships is twofold: (1) Can the data be divided into categories for the prediction of some property, and (2) Can the features that differentiate the categories be identified? For the first task, a set of known samples is used to separate information and noise sources, with the information sources combined to develop a discriminant, which is used to predict the class membership of samples that are not part of the original training set. The second task is called feature selection [15]. The development of suitable models to isolate groups or classes of data according to their properties is known as classification or pattern recognition [16]. The basic premise underlying the use of these methods is that clustering of the data into less similar subgroups is associated with some underlying structure or property of the data.

In this chapter, pattern recognition methods are discussed. (Feature selection is discussed in Chapter 3.) What are the operations that must be performed in order to apply pattern recognition methods to chemical problems of interest? There are several

texts on this subject which describe in detail the theory and application of pattern recognition techniques [17, 18]. A summary of the techniques used in the studies described in this thesis will be included in several of the following sections. Special emphasis will be placed on the application of these techniques to problems in spectral pattern recognition.

## 2.2 Data Representation

The first step in any pattern recognition study is to convert the raw data into computer compatible form. Normally, the computer compatible form used is a string of scalar measurements comprising an n-tuple called the pattern vector.

$$x = (x_1, x_2, x_3, \ldots\ldots\ldots\ldots x_p)$$

Each component of the pattern vector represents a physically measurable quantity. For spectral data, each component of the pattern vector is the absorbance or spectral intensity at a specified wavelength from a baseline corrected digitized spectrum. The pattern vectors, in turn, constitute a data matrix. The rows of the matrix represent observations, and the columns represent the values for each descriptor.

$$\begin{bmatrix} x_{11} & x_{12} & x_{13}\ldots\ldots\ldots\ldots\ldots x_{1p} \\ x_{21} & x_{21} & x_{23}\ldots\ldots\ldots\ldots\ldots x_{2p} \\ x_{31} & x_{32} & x_{33}\ldots\ldots\ldots\ldots\ldots x_{3p} \\ . & . & \ldots\ldots\ldots\ldots\ldots\ldots . \\ . & . & \ldots\ldots\ldots\ldots\ldots\ldots . \\ . & . & \ldots\ldots\ldots\ldots\ldots\ldots . \\ x_{n1} & . & \ldots\ldots\ldots\ldots\ldots\ldots x_{np} \end{bmatrix}$$

It is essential that descriptors encode the same information for all objects in the data matrix. For example, if descriptor 2 is the absorbance at 2500nm in object 1, it must also be the absorbance at 2500nm in objects 2, 3, … N. Hence, peak matching is crucial when spectra are translated into data vectors. For spectral data, peak matching is not a problem since absorbance or spectral intensity is measured as a function of the excitation wavelength. For chromatographic data, however, the problem of peak matching is often quite formidable [19].

## 2.3 Data Preprocessing

The next step involves scaling the data. The scaling procedure which should be used for a given data set will depend upon the nature of the problem. This aspect of pattern recognition has not been adequately investigated for spectral data. In the applications discussed herein, two techniques have been used – normalization and autoscaling. Although preprocessing of the data is (strictly speaking) not part of the pattern recognition process, it definitely affects the results obtained and for this reason is discussed here.

Normalization involves setting the sum of the squares of the components of each pattern vector equal to the same arbitrary constant. For infrared and Raman data, each data vector is normalized to unit length. This is accomplished by dividing each data vector by the square root of the sum of the squares of the components composing the vector. Normalization will compensate for variation in the data due to differences in sample size or optical path length. However, normalization can introduce dependence between variables which could have an effect on the results of the investigation. One must take this into account when deciding whether or not to normalize the data.

Normalization is an effective procedure for removing variation in spectral data due to sample size if the noise is homoscedastic [20]. A recent paper provides a discussion about the effects of normalization on the classification of spectral data [21].

Autoscaling (see Equation 2.1) involves standardizing the measurement variables such that each descriptor or measurement has a mean of zero and a standard deviation of unity, that is,

$$x_{i,new} = \frac{(x_{i,orig} - \bar{x}_{i,orig})}{s_{i,orig}} \tag{2-1}$$

where $\bar{x}_{i,orig}$ is the mean and $s_{i,orig}$ is the standard deviation of the original measurement variable. Autoscaling removes any inadvertent weighing of the variables that otherwise would occur due to differences in magnitude among the various measurements. Consider a data set where each sample is described by two variables: the concentration of Na and the concentration of Mg as measured by flame emission spectroscopy. The concentration of Na in the samples varies from 10ppm to 100ppm, whereas the concentration of Mg in the same samples only varies from 1ppm to 10ppm. A 10% change in Na concentration will have a greater effect on the Euclidean distance than a 10% change in Mg concentration. If the data are autoscaled, a 10% change in Mg concentration will have the same effect as a 10% change in Na concentration. After autoscaling, all of the measurement variables have equal weight in the analysis. Autoscaling influences the spread of the data, placing the data points inside a hypercube. However, autoscaling does not change the relative distribution of the data points in the high-dimensional measurement space.

## 2.4 Principal Component Analysis

Scientists and engineers often use graphical methods to study data. If there are only two or three measurements per object or sample, the data can be directly displayed as points in a two-or three- dimensional measurement space with the coordinate axes of the space defined by the measurement variables. By examining the plot, a scientist can search for similarities and dissimilarities among samples, find natural clusters in the data, and even gain information about the overall structure of the data set. If there are $p$ measurements per sample ($p$>3), a two- or three-dimensional representation of the measurement space is needed to visualize the relative position of the data points in $p$-space. This representation should reflect in some manner the distribution of the data points in the higher-dimensional measurement space. A popular approach to this problem is to use a mapping and display technique called principal component analysis [22-24]. A detailed treatment of principal component analysis (PCA) is not provided here. However, those aspects of PCA related to the genetic algorithm for pattern recognition analysis and the studies discussed in this chapter are summarized here.

PCA is the most widely used multivariate analysis technique in science and engineering. It is a method for transforming the original measurement variables into new, uncorrelated variables called principal components. Each principal component is a linear combination of the original measurement variables. Using this procedure, a set of orthogonal axes that represents the direction of greatest variance in the data is found. (Variance is defined as the degree to which the data are spread in the $p$-dimensional measurement space.) Typically, only two or three principal components are necessary to explain a significant fraction of the information (variance) present in the data. Hence,

PCA can be applied to multivariate data for dimensionality reduction, identification of outliers, display of data structure, and classification of samples.

Dimensionality reduction is possible with PCA because of correlations that exist among measurement variables. Consider Figure 2.1, which shows a plot of 15 samples in a two-dimensional space. The coordinate axes of this pattern space are defined by the variables $x_1$ and $x_2$. Both $x_1$ and $x_2$ are correlated, since fixing the value of $x_1$ limits the range of values possible for $x_2$. If $x_1$ and $x_2$ were uncorrelated, the enclosed rectangle (shown in Figure 2.1) would be completely filled by the data points. Because of the correlative relationship between $x_1$ and $x_2$, the data points occupy only a fraction of the pattern space.

Information can be defined as the scatter of points in a vector space. Correlations between measurement variables decrease the scatter and subsequently the information content of the vector space because the data points are restricted to a small region of the pattern space due to correlations among the measurement variables. If the measurement variables are highly correlated, the data points could even reside in a subspace. This is shown in Figure 2.2. Each row of the data matrix is a sample, and each column is a measurement variable. Here $x_3$ is perfectly correlated with $x_1$ and $x_2$, since $x_3$ (third column) equals $x_1$ (first column) plus $x_2$ (second column). The seven data points lie in a plane (or two-dimensional subspace), even though each point is characterized by three measurements. Because $x_3$ is a totally redundant, it does not contribute any additional information, which is why the data points lie in a plane.

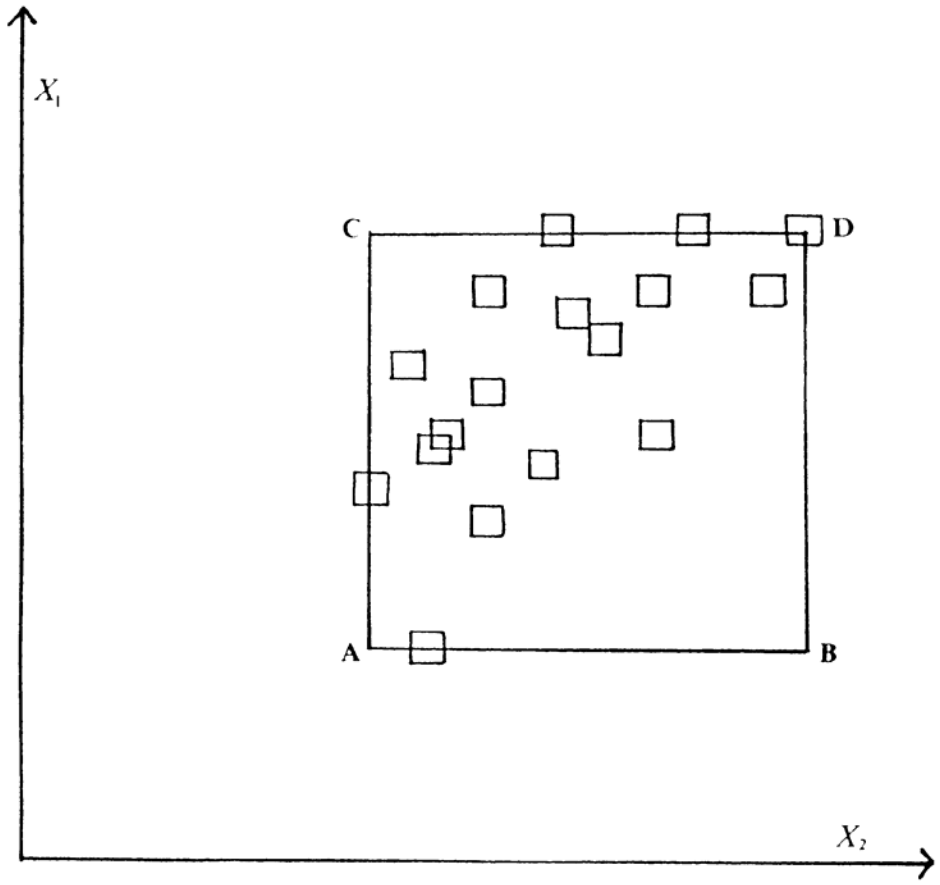



Figure 2.1. Fifteen samples projected onto a two-dimensional data space. Because $x_1$ and $x_2$ are correlated, the data points are restricted to a small region of the measurement space defined by the vertices A-D of the rectangle. (Adapted from *NBS J. Res.*, 1985, 190(6), 465-476.)

$$\mathbf{X} = \begin{bmatrix} 0.5 & 0.5 & 1.0 \\ 1.9 & 0.7 & 2.6 \\ 2.0 & 2.0 & 4.0 \\ 0.3 & 1.8 & 2.1 \\ 1.9 & 1.7 & 3.6 \\ 1.2 & 0.2 & 1.4 \\ 1.9 & 0.9 & 2.8 \end{bmatrix}$$

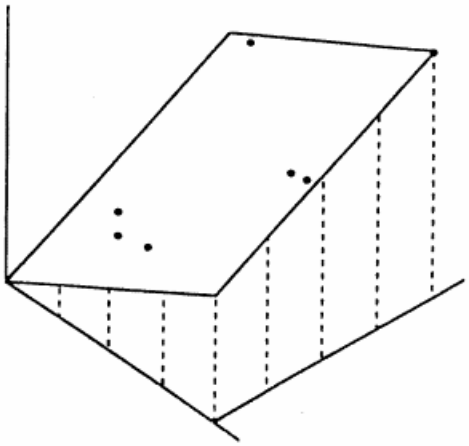


Figure 2.2. In the case of strongly correlated measurement variables, the data points may even reside in a subspace of the original measurement space. (Adapted from *Multivariate Pattern Recognition in Chemometrics*, Elsevier Science Publishers, Amsterdam, 1992.)

Measurement variables that have redundant information are said to be collinear. High collinearity between measurement variables is a strong indication that a new coordinate system can be found that is better at conveying the information present in the data than one defined by the original measurement variables. The new coordinate system for displaying the data is based on variance. (The scatter of the data points in the measurement space is a direct measure of the data's variance.) Each principal component of the data defines the variance-based axes of this new coordinate system. The first principal component is formed by determining the direction of largest variation in the original measurement space of the data and modeling it with a line fitted by linear least squares (see Figure 2.3) that passes through the center of the data. The second largest principal component lies in the direction of next largest variation. It passes through the center of the data and is orthogonal to the first principal component. The third largest principal component lies in the direction of next largest variation, and it also passes through the center of the data. It is orthogonal to the first and second principal component, and so forth.

Each principal component describes a different source of information because each defines a different direction of scatter or variance in the data. (Information can be defined as the scatter of points in a vector space; the scatter of the data points in the vector space is also a direct measure of the data's variance; hence, variance and information are synonymous.) The orthogonality constraint imposed by the mathematics of PCA also ensures that each variance-based axis is independent.

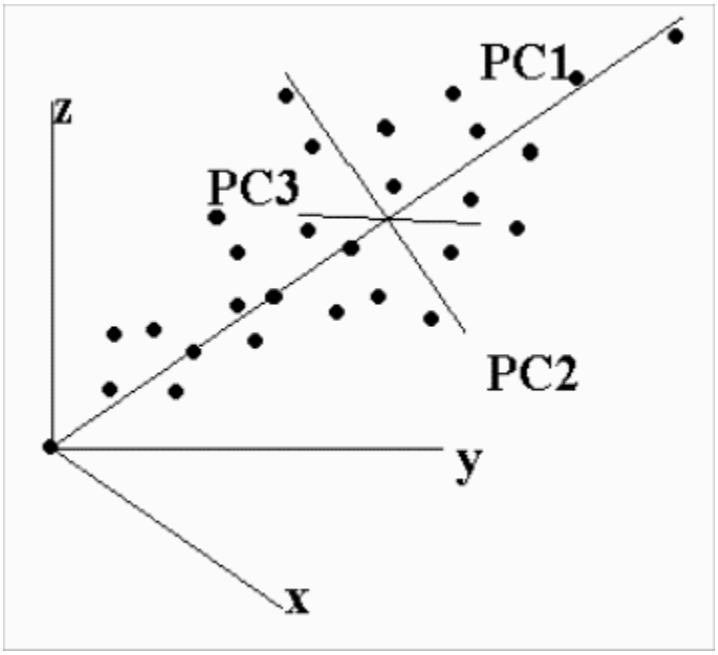



Figure 2.3. Principal component axes defining a new set of basis vectors for the measurement space defined by the variables X, Y, and Z. The third principal component describes only noise in the data.

A measure of the amount of information conveyed by each principal component is the variance of the data that it explains which can be expressed in terms of the eigenvalue that is associated with the principal component. For this reason, principal components are arranged in order of decreasing eigenvalues. The largest principal component, which is the first, is the most informative, whereas the smallest principal component, which is the least informative, is the last. The amount of information contained in a principal component relative to all of the original measurement variables, i.e., the fraction of the total cumulative variance explained by the principal component, is equal to the eigenvalue of the principal component in question divided by the sum of all the eigenvalues for the data. The maximum number of principal components that can be extracted from the data is the smaller of either the number of samples or the number of variables in the data set, as this number defines the largest possible number of independent axes in the data.

Principal components are computed directly from the data using an algorithm called singular value decomposition [25]. This algorithm generates the score and loading matrices, and the eigenvalues of the covariance matrix of the data. The score matrix defines the coordinates of the data points in the principal component space, whereas the loading matrix defines the relationship between the original measurement variables and the new basis vectors describing variation. In other words, the principal components of the data can be reconstructed from the original measurement variables using information contained in the loading matrix. The eigenvalues of the covariance matrix tell how much variation in the data is captured by each principal component.

If the data are collected with due care, one would expect that only the larger principal components would contain information about the property of interest (i.e., the class membership), since most of the information in the data should be about the effect of interest. However, the situation is not always as straightforward as has been implied. Each principal component describes some amount of signal and some amount of noise because of accidental correlations between signal and noise. The larger principal components contain information primarily about signal, whereas the smaller principal components primarily describe noise. By discarding the smaller principal components, noise is discarded, but so is a small amount of signal. However, the gain in signal to noise that is accrued will usually more than compensate for the biased representation of the data that occurs when plotting only the largest principal components of the data.

PCA takes advantage of the fact that a large amount of data generated by spectrochemical methods has a great deal of redundancy and therefore a great deal of collinearity. Because the measurement variables are correlated, 600-point spectra do not require 600 independent orthogonal axes to define the position of a sample point in the measurement space. Using principal component analysis, the original measurement variables that constitute a correlated-axes system can be converted into an orthogonal-axes system, which dramatically reduces the dimensionality of the data, since only a few independent axes are needed to describe the data. Spectra for a set of samples often lie in a subspace of the original measurement space, and a plot of the two or three largest principal components of the data can help one to visualize the relative position of the spectra in this subspace. Using PCA, the data can be plotted in a new coordinate system based on variance. The origin of the new coordinate system is the center of the data, and

the coordinate axes of this new system are the principal components of the data. By utilizing this new coordinate system, we can uncover relationships present in the data; for example, we can find distinct samples subgroups or classes within multivariate data.

Clearly, there are a number of advantages in using principal components to describe or model chemical data, e.g. dimensionality reduction, classification, and signal enhancement. However, there will be a loss of information when changing coordinate systems. The loss is characterized by the sample residual, which is the distance between a sample and the principal components used to model the signal in the data, i.e., the subspace. The projection distance is large for samples that are not well fitted by the principal component model.

The approach of describing a data set in terms of important and unimportant variation is known as soft modeling in latent variables. This approach to modeling is possible because chemical data sets often contain a large number of interrelated measurement variables. All of the important variation in this data can be explained by a small number of surrogate variables (usually principal components) because of the redundancies in the data. By examining these surrogate variables, it is possible to identify important relationships in the data, that is, find similarities and differences among the samples in a data set, since each surrogate variable captures a different source of information. The surrogate variables that describe important variation in the data (i.e., signal) can be identified and used to develop a classification model. Surrogate variables that describe a property of interest are called latent variables.

With PCA, we are able to plot the data in a new coordinate system based on variance. The origin of the new coordinate system is the center of the data, and the

coordinate axes of the new system are the principal components of the data which primarily contain the signal. This variance based coordinate system will be different for each data set. With this new coordinate system, we can uncover relationships present in the data. PCA is actually using the data to suggest the model, which is a new coordinate system for the data. The model is local since the model center and the principal components will be different for each data set. The focus of PCA is signal, not noise. PCA based soft models are both linear and additive.

## 2.5 Classification

So far in this chapter, only PCA has been discussed. The structure of the data is analyzed without any *a priori* information about samples belonging to distinct groups. Unsupervised pattern recognition techniques such as PCA are not always sufficient for developing classification rules. However, the overall goal of a pattern recognition study is the development of a classification rule that can accurately predict the class membership of an unknown sample. In the final section of this chapter, supervised methods of pattern recognition that are referred to in subsequent chapters of this thesis are discussed.

This section on classification is not intended to be a comprehensive review, which should come as no surprise since this field is substantially too complex for such a brief treatment. Indeed, one of the most complete summaries to date was written over fifteen years ago by McLachlan [26] and, even with his abbreviated, outline-type presentation, it took well over 500 pages to complete. For this reason, the material in this section is focused on a set of introductory topics and a logical progression that is relevant to the needs of a physical scientist or engineer who wants to understand something about

classification and who, as often as not, must analyze data that has an abundance of features. The approach will be to reason from examples in an effort to bring forward subtle ideas. In all cases, the goal will be to help the reader develop an intuitive feeling for the tools, techniques, and some of the limitations involved when using classification methods.

### Example 1 – Distinguishing Midges

What can you do with more than one variable that you cannot do as well with a single variable? Well, many things, but group discrimination is one. Consider the data shown in Table 2.1. These data were reported by Grogan and Wirth [27] and are the wing and antenna lengths of two recently discovered species of gnat-like insects known as a "midge". These two species - *Amerohelea fasciata (*Af) and *A. pseudofasciata* (Apf) - are difficult to distinguish without the use of an elaborate laboratory procedure. It would be desirable to be able to distinguish the two by some simple procedure such as measuring the length of their antenna and wings.

A two-dimensional plot of antenna length versus wing length is shown for the two midge species in Figure 2.4. The two species can be completely separated using these two measurements by drawing a straight line through this two dimensional measurement space (which is also known as a pattern space). Let $d(x) = w_1x_1 + w_2x_2 + w_3 = 0$ be the equation of the line (or boundary surface) separating the two species, where $w_1$, $w_2$, and $w_3$ are the parameters (or weights of the linear combination of the measurement variables) and $x_1$ and $x_2$ are the actual values of the measurement variables (antenna and wing length) for each midge. As shown in the figure, it is evident that neither antenna length nor wing length alone can provide such a clear separation of the data. A linear combination of the original variables has served a purpose that could not be served as

well by any single one of the original two variables. The weights in this linear combination are, of course, not the same as those found in principal component analysis since they are chosen for purposes of discrimination and not for variance summary. In the next subsection, Fisher's original approach to classification will be discussed since this approach to linear discrimination is more consistent with the ideas from principal component analysis than are the other approaches. We will eventually see the sense in which this original perspective on discrimination aligns with the more popular likelihood (i.e., Mahalanobis distance) based approach.

### 2.5.1 Canonical Discriminant Analysis

Although Sir Ronald Fisher (1890-1962) is generally referred to as the "father of modern inference", his initial approach to discrimination [28] was not couched in probabilistic terms. To understand this "canonical" approach, it is necessary to introduce some matrix notation, but the intuition behind this idea can be understood with just a basic grasp of analysis of variance (ANOVA). Recall, in a simple design, an ANOVA is concerned with the relative sizes of three basic constructs: the total sums of squares (SST), the among-treatments sums of squares (SSTr), and the error sums of squares (SSE), typically pooled across treatment groups. In balanced designs we rely heavily on the fact that SST = SSTr + SSE, and we reason with strong justification that if SSTr is sufficiently larger than SSE, then treatment groups are different.

In canonical discriminant analysis (CDA), there are three completely analogous constructs: Total Sum of Squares (Equation 2.2), Among-groups Sums of Squares (Equation 2.3), and Pooled Within-Groups Sums of Squares (Equation 2.4)

$$\mathbf{S}_X = \left(\frac{1}{n-1}\right) \sum_{i=1}^{g} \sum_{j=1}^{n_i} (x_{ij} - \bar{x})(x_{ij} - \bar{x})^t \tag{2.2}$$

$$\mathbf{H} = \sum_{i=1}^{g} n_i (\bar{x}_i - x)(\bar{x}_i - x)^t \tag{2.3}$$

$$\mathbf{E} = \sum_{i=1}^{g} \sum_{j=1}^{n_i} (x_{ij} - \bar{x}_i)(x_{ij} - \bar{x}_i)^t \tag{2.4}$$

where $x_{ij}$ is the value of the $j^{th}$ descriptor for sample i, $\bar{x}$ is the mean of the $j^{th}$ descriptor for all samples in the data set. As with ANOVA, it can be shown that $(n-1)\mathbf{S}_X = \mathbf{H} + \mathbf{E}$.

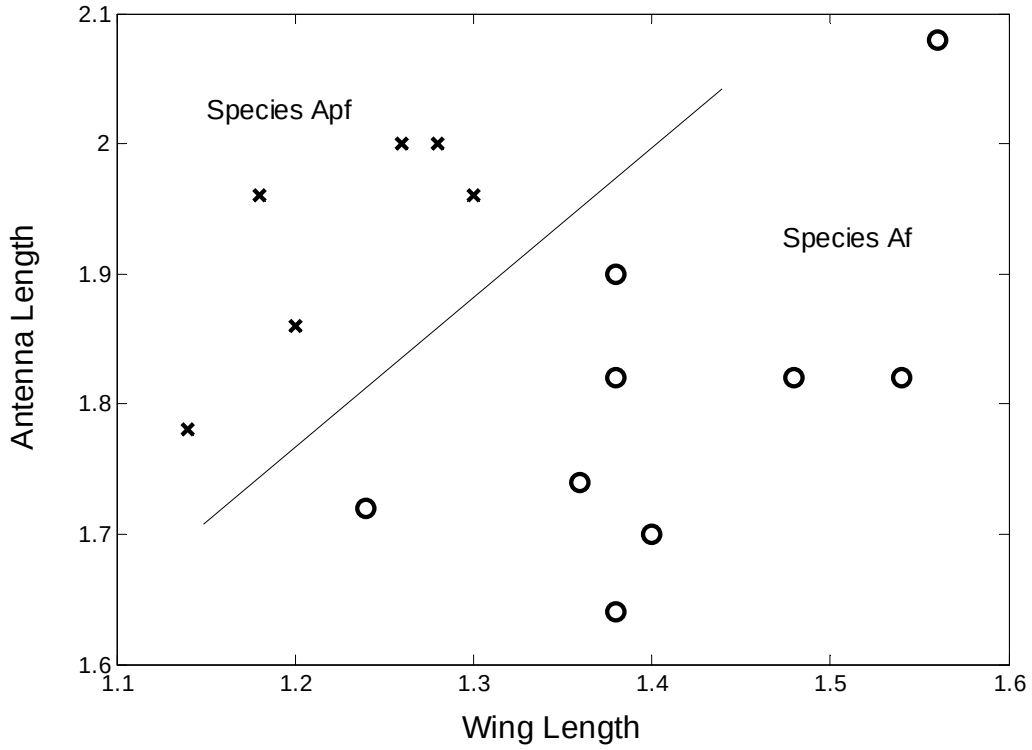



Figure 2.4. Scatter plot of antenna length versus wing length for gnat like insects known as midges. (x = Species *A. pseudofasciata* and o = Species *Amerohelea fasciata*.)

With CDA the goal is to successively find linear combinations of the original measurement variables that exhibit maximum between-groups variability, relative to the pooled within-groups variability. These linear combinations are called "canonical discriminant scores" or just "scores" if the context is clear, and the weights that define them are found in such a way that classes are optimally separated at the score level. This is similar to the reasoning used in PCA, except that a map of the data is being developed with the focus of the optimization on "between-groups variability" not "total variability".

A comment on terminology is in order. It is fairly common in the literature for CDA to be called "canonical variates analysis," which can be confusing since canonical variates analysis is a generic phrase used to refer to regression, multivariate analysis of variance, canonical correlations analysis, and other various forms of discriminant analysis. In the rest of this chapter and thesis, the phrase "canonical variates" will be used to explicitly refer to the scores that result from a canonical discriminant analysis.

The optimization problem that defines CDA, as well as the corresponding solution, are well-known in the literature, but will not be restated here in keeping with a commitment to suppress any unnecessary mathematical notation. However, the weights used to produce these canonical variates are contained in the eigenvectors of a matrix based on **E** and **H**.

**CDA Result**

Fisher's optimal weighting scheme is provided by the entries in the normalized eigenvectors from the matrix $\mathbf{E}^{-1}\mathbf{H}$

If there are a total of N observations on each of p variables, across g groups, with N-g > p (in our "training set"), then the matrix **E** will be full rank, **H** will have rank g-1,

and, in a formal sense ("with probability one") the matrix $\mathbf{E}^{-1}\mathbf{H}$ will have rank g-1. Hence there will be g-1 nonzero eigenvalues associated with this matrix and, in turn, g-1 non-arbitrary weighting schemes provided by the corresponding normalized eigenvectors. Hence, there are g-1 possible scores that can be computed using this paradigm.

**Example 1 – Differentiating Midges (continued)**

A data set of wing and antenna lengths of two recently discovered species of gnat-like insects known as "midges" will be analyzed by CVA. In this example there are two groups and two variables, so $\mathbf{E}^{-1}\mathbf{H}$ will be (2 x 2), and have rank 1. The optimal weighting scheme, therefore, is contained in the normalized eigenvector corresponding to the lone nonzero eigenvalue of this matrix construct. These weights or loadings are shown below, as are the means for each species.

| Weights | Variable |
|---|---|
| 0.8371 | Antenna Length |
| -0.5471 | Wing Length |

| Species | Mean Antenna Length | Mean Wing Length |
|---|---|---|
| Af | 1.413333333 | 1.804444444 |
| Apf | 1.226666667 | 1.926666667 |

The line through the group means has a slope of -1.53 which is also the slope of the eigenvector: (.8371/-0.5471) = -1.53. Therefore it is clear that the weights on the original variables form a vector that is parallel to the line connecting the two group means. In fact, if the original data were corrected for their (grand) mean (which is usually the case since the data is autoscaled), then this vector would define the line between the two group means. Note, however, that this line is not the line shown in

Figure 2.5 for discriminating between the two species of Midges. Rather, it is the line "perpendicular" (with respect to an inner product induced by the pooled covariance array) to the dividing boundary shown in Figure 2.5 The line with slope -1.53 (passing through the grand mean) is the subspace that the original data would be projected onto in order to produce the so-called discriminant scores. These scores are the distances between the projected grand mean of the raw data and the projected ordered pairs. The distinctions between dividing boundaries in the original measurement space and the lower-dimensional scores will become relevant later as we work to understand the differences between CDA and the likelihood approach.

There is no inherent classification rule that emerges as part of the language of canonical variates. Indeed, the primary focus of canonical variates analysis is visual: take data in higher dimensions, compute the first two or three "canonical scores" and plot them where you can see them and look for separation. The two-dimensional score plots are often called "territory plots" and reflect the imposition of an ad hoc classification rule. The way the paradigm has developed; there is no discussion of an actual classification rule, only a prescription for how one can create optimally separated scores. For this reason, scientists and engineers have often employed the following reasoning when using canonical variates.

- Start with p-dimensional data on g groups
- Compute the first k (often k = 2) discriminant scores
- Compute the k-dimensional mean for each of the score groups
- Classify an unknown into the group that exhibits the closest group mean (score), where "closest" is in terms of the Euclidean distance.

- Evaluate the goodness of this rule by cross-validating on the original training set data and record the percentage of misclassifications.

This is a perfectly rational way for a scientist or engineer to proceed. However, there can be no immediate claim of optimality with the imposition of this ad hoc rule, other than the mathematical assurance that the scores originate from a process that is optimal in the sense of the original problem posed by Fisher, which, granted, is an important sense of optimality. However, this rule can also be claimed to minimize theoretical misclassification rates if enough scores are used, which will be explained below.

### Example 2: Iris Data Set

The data consists of 50 samples from each of three species of Iris (*Iris setosa*, *Iris versicolor*, and *Iris virginica*). Each sample is represented by four features: septal length, septal width, petal length, and petal width. Robert Fisher [29] developed a classification model to identify the species of Iris using a combination of these four features. Figure 2.5 shows a score plot of the two canonical variates of the autoscaled Iris data, and Figure 2.6 shows a score plot of the two largest principal components of the autoscaled Iris data. If information about species lies in the directions of maximum variance, then it should not come as a surprise that principal components analysis and canonical variates yield similar scatter plots. The conclusion that one can draw from this data is that the bulk of the information encoded by these four features is about species.

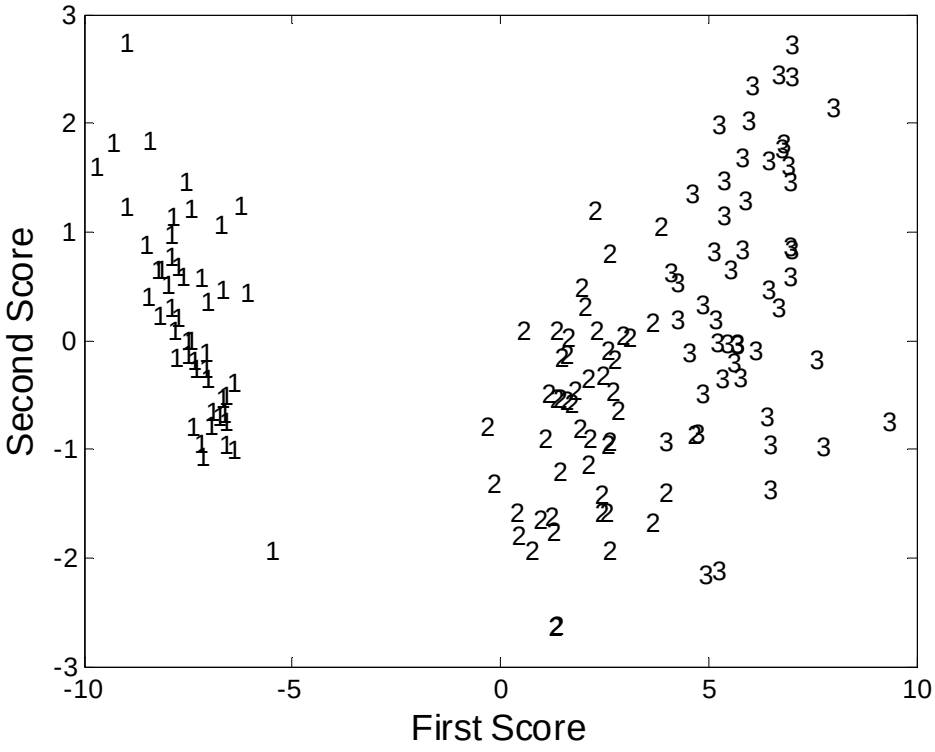



Figure 2.5. Score plot of the two largest canonical variates of the autoscaled Iris data. (1 = *Iris setosa*, 2 = *Iris versicolor*, and 3 = *Iris virginica*)

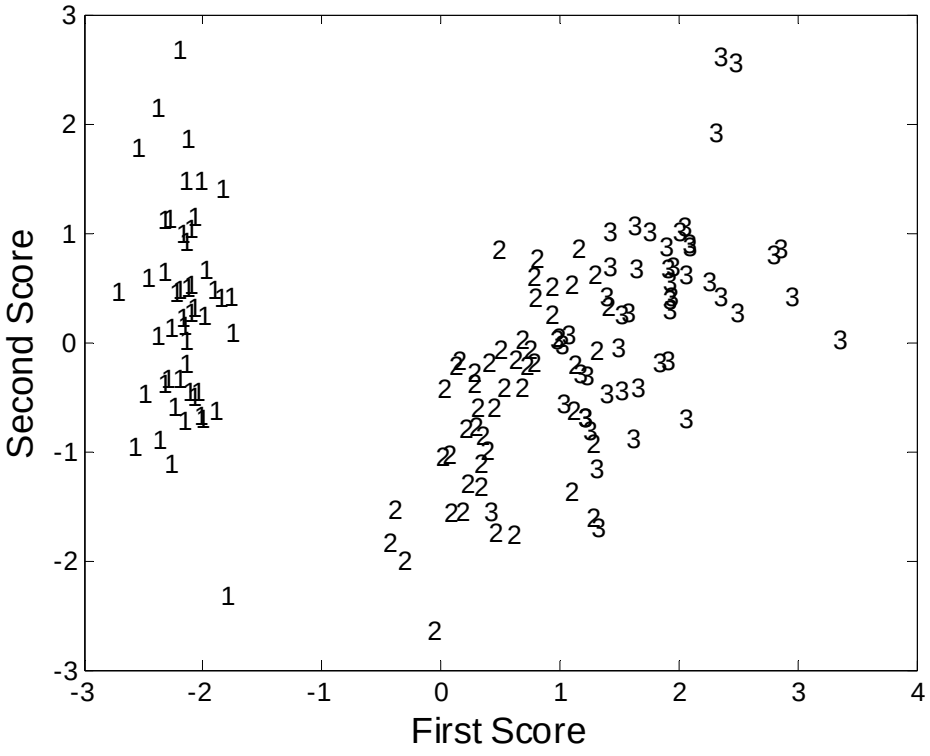



Figure 2.6. Score plot of the two largest principal components of the autoscaled Iris data set. (1 = *Iris setosa*, 2 = *Iris versicolor*, and 3 = *Iris virginica*)

Although PCA is a nonsupervised pattern recognition technique, chemists often use it to classify data. There are a number of advantages in using PCA to classify chemical data. If the data have been collected with due care, one would expect that only the larger principal components would contain information about the class membership of the samples in the data set since most of the information in the data should be about the effect of interest. That is to say, the variability in the data would, ideally, be very small, except for that induced by class differences. When this is the case, PCA will usually be all that one needs. Further, if our problem is unstructured in the sense that no *a priori* class structure is known, then classical discriminant techniques will not apply and PCA may be an adequate alternative for data where the among groups variability dominates.

### Example 3: Ant Data Set

This data set consists of gas chromatograms of cuticular hydrocarbon extracts obtained from the cuticles of 134 red fire ant samples. Each sample contains the hydrocarbons extracted with hexane from the cuticles of 100 individual ants. The hydrocarbon fraction analyzed by gas chromatography was isolated from the concentrated hexane washings by means of a silicic acid column. Five major hydrocarbon peaks were identified and quantified by gas chromatography/mass spectrometry: heptacosane, 13-methylheptacosane, 13,15-dimethylheptacosane, 3-methylheptacosane, and 3,9-dimethylheptacosane in order of elution from the packed OV-17 GC column used. An internal standard was used for quantitation. Each gas chromatogram was normalized using the weights of the collected ants. In this study, ant samples were obtained from four different colonies (E, J, P, and Q). Figure 2.7 shows a score plot of the first two canonical variates of the autoscaled data, and Figure 2.8 shows

a score plot of the two largest principal components of the autoscaled data. It is evident from an examination of these two score plots that information about colony is not oriented along the direction of maximum variance, which would suggest that most of the information in the five features is not about the colony of origin of the ants. Further details about this data can be found elsewhere [30, 31].

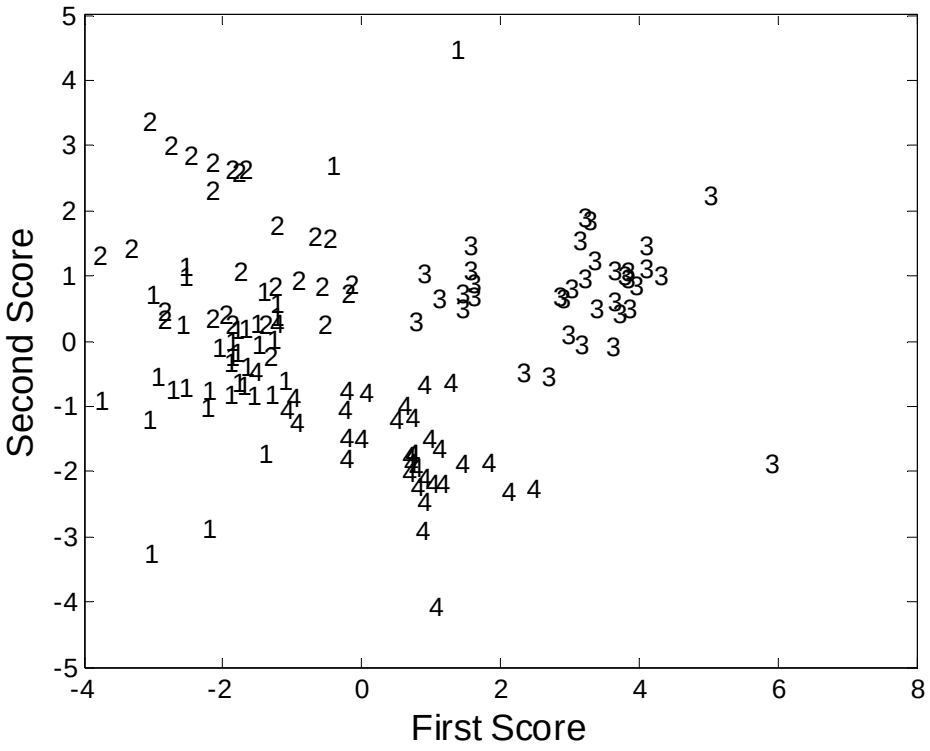



Figure 2.7. Score plot of the two largest canonical variates of the autoscaled and normalized ant data. (1 = Colony E, 2 = Colony J, 3 = Colony P, and 4 = Colony Q)

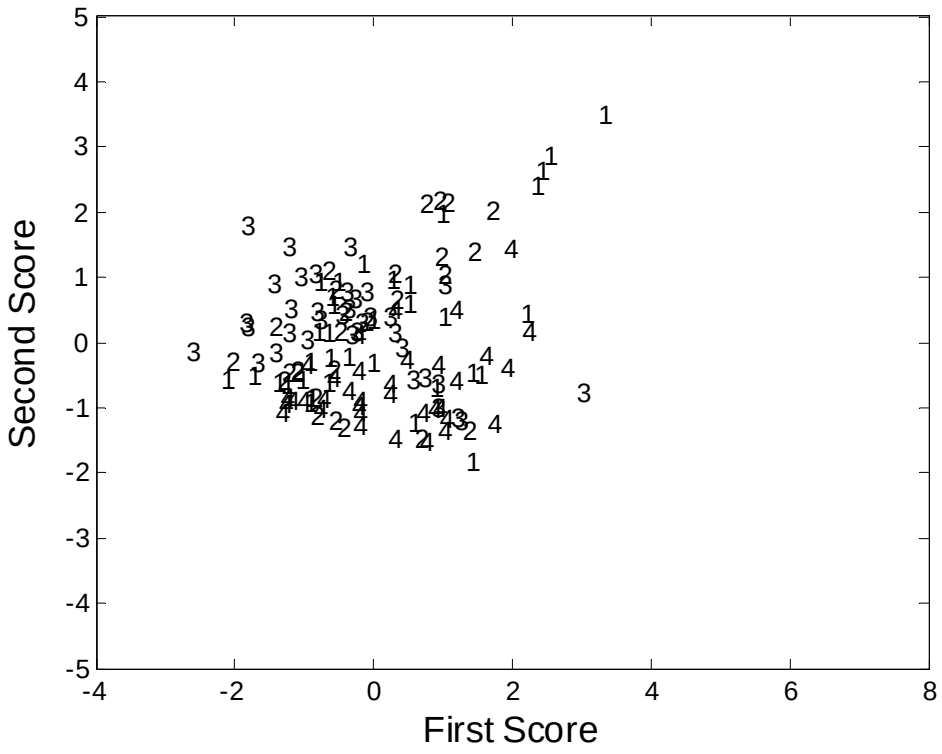



Figure 2.8. Score plot of the two largest principal components of the autoscaled and normalized ant data. (1 = Colony E, 2 = Colony J, 3 = Colony P, and 4 = Colony Q)

Canonical discriminant analysis, in the spirit of PCA and many other exploratory multivariate techniques, focuses on the optimal construction of linear combinations of the original measurement variables. Unlike PCA, the goal of CDA is to find linear combinations that best separate a pre-defined group structure. In this sense "canonical variates" are better suited for discrimination in structured problems than are principal component scores. However, the weights (eigenvectors) that ultimately define the canonical scores do not partition the original measurement space into orthogonal subspaces (which is the case with PCA). This means, for example, that distances between points in the original measurement space will not necessarily be the same as the

distances between their corresponding canonical scores. Furthermore, canonical variates cannot be extracted from data when the number of features exceeds the number of samples, which is a situation discussed at length in this chapter and subsequent chapters in this thesis.

### 2.5.2 Linear Discriminant Analysis (LDA)

As noted above, canonical discriminant analysis is couched in a language of dimension reduction, scores, and loadings, in much the same way as PCA. However, the formal statistical notion of linear discrimination is not that of canonical variates, but rather is derived from a more formal encounter with likelihoods and, in the case of multivariate normal data, intimately related to the work of Mahalanobis. On the surface, the goals of canonical variates and linear discriminant analysis are very different. Canonical variates focuses on optimal group separation with the classification rule often imposed in a rational, though ad hoc fashion. The so-called "likelihood" based approach is focused on the derivation of an optimal classification rule at the outset of the process. In that sense, it is more steeped in statistical theory. The basic set up, which is presented for two groups, to keep the notation simple, is as follows

- Assume two groups: $G_1$ and $G_2$, governed by densities $f_1$ and $f_2$.
- Have some notion of prior probability of group membership: $\pi_1$ and $\pi_2$

The rule that minimizes the probability of misclassification is given by:

**Likelihood Discriminant Rule**

Assign an unknown $\mathbf{x} \Rightarrow G_1$ iff $\mathbf{x} \in \left\{ \mathbf{x}: \frac{\pi_2}{\pi_1} < \frac{f_1}{f_2} \right\}$

In the case of (multivariate) normal densities, with group means $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$, equal covariances $\Sigma$, and equal prior probabilities, this rule takes the following form:

**Normal Likelihood Discriminant Rule**

Classify an unknown $\mathbf{x} \rightarrow G_1$ iff the $(\mathbf{x}-\boldsymbol{\mu}_1)^t\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_1) \leq (\mathbf{x}-\boldsymbol{\mu}_2)^t\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_2)$

In practice, the population means, $\boldsymbol{\mu}_i$, and the theoretical covariance matrix, $\boldsymbol{\Sigma}$, are replaced by estimates of these parameters from the data. This likelihood perspective, which for multivariate normal data amounts to classifying the data according to a minimum Mahalanobis ("standardized") distance, will minimize the total (theoretical) probability of misclassification. That is why this perspective on classification is often referred to interchangeably as both a "Mahalanobis distance" approach and a "likelihood" approach.

Canonical variates and likelihood-based linear discriminant analysis provide the most elementary perspectives on linear classification of data. However, there are some other issues that need to be addressed which offer unification of these two perspectives on classification. An important question to ask is whether the perspective provided by canonical variates, with the ad hoc rule imposed at the score level is the same as that provided by the Mahalanobis distance approach. Assuming g groups, if g-1 canonical scores are computed, and the Euclidean distance is calculated between each sample point in the (g-1)-dimensional score plot and each class mean and the sample is assigned to the class with the smallest mean Euclidean distance, then the aforementioned classification rule will have the same theoretical error rate as the Mahalanobis distance rule applied in the original p-dimensional space.

While this is an intriguing mathematical observation, it has profound ramifications. When canonical variates are employed, typically only one or two scores are retained since the emphasis is on visual separation. If a Euclidean distance nearest-mean classification scheme is employed based on only two scores in a problem where $g > 3$, then no theoretical claim to "best rule" can be given, at least not in the sense of minimizing misclassification rates. This is not to say that our ad hoc rule is unreasonable or even significantly less than optimal in some sense; it may in fact perform quite well. We would only caution that language matters. In the absence of theoretical claims one can always assess the practical performance of any classification rule by cross-validation [32] on the training set, or by how it performs on a prediction set.

In what sense is this rule "linear"? This approach is linear in the data $\mathbf{x}$ in the sense that an optimal discriminant is of the general (matrix) form $\mathbf{Ax} + \mathbf{b}$. Geometrically, this corresponds to imposing "flat" (linear) boundaries in p-dimensional space to partition up the pattern space into g mutually exclusive regions for classification. Linearity occurs (for multivariate normal problems) because of the assumption that individual class covariances are the same in all classes. If this assumption is not true, then these linear rules would not be optimal and a quadratic rule would be preferred. This is discussed in the next section.

Claims to optimality, while important, are theoretical. However, it is useful to have theoretical ideas of how to proceed at the practical level. For example, it would not be possible to understand the original purpose of "regularized" discrimination (to be discussed later) without understanding that practical misclassification rates do not always mirror theory. It is possible for class covariance arrays to be markedly different, but for

linear classification rules to outperform quadratic rules in practice, when individual group sample sizes are very small, which gives rise to errors in estimating the class covariance matrix from the original data..

### 2.5.3 Quadratic Discrimination (QDA)

Let us assume that it is not plausible to treat covariance matrices in each of the g groups as equal. Therefore, the linear rule would no longer be optimal. If one were to allow for each class covariance matrix, $\mathbf{\Sigma_i}$, to be different then the following classification rule would be optimal for (multivariate) normal data:

**Quadratic Discriminant Rule (Alternate Statement)**

Assign an unknown $\mathbf{x} \Rightarrow G_i$ iff i = index for the $\min_j \left\{ \log\left( \left| \mathbf{\Sigma}_j \right| \right) + (\mathbf{x} - \boldsymbol{\mu}_i)^t \mathbf{\Sigma}_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i) \right\}$

There are several things worth noting about this rule:

- This rule is not limited to the Mahalanobis distance computed for each class but is specific for each class covariance matrix. The determinant of the individual covariance matrix (which corresponds to the volume of space occupied by the sample points representing the class is also a part of the rule.

- This rule is not linear. It is quadratic in the data **x**, by virtue of these Mahalanobis-like terms depending on the group. Geometrically this simply means that "flat" boundaries are no longer used to separate groups in the p-dimensional space. Instead, curved ("quadratic") boundaries are now employed to do the classification.

- There is no analogy to canonical variates, so any discussion involving canonical discrimination is no longer relevant. Canonical variates analysis is a linear paradigm. If the covariances are different, then the rule stated above will, in theory, outperform the linear rule.

### 2.5.4 Shrinkage and Covariance Stabilization

When the number of features is large compared to the number of samples in the training set, a linear classification rule will outperform a quadratic rule. There are a few

ways to think about this. On the one hand, this is an issue about numerical stabilization, with the condition number of the (pooled) common covariance estimate used in linear discriminant analysis being notably better (less noisy) than that of any of the individual covariance matrices. From a statistical perspective one can think of this in terms of having fewer parameters to estimate from the available data and, therefore, the variances associated with those estimates will be smaller.

In any case, the concept of "partially pooled" estimates originated out of a desire to have some of the benefits of increased stability of the linear approach, without being so restrictive as to ignore the essential difference in the covariance that may be present. This has led to a number of different pooling strategies, but the one that is most widely known was popularized by Frank and Friedman [33], who called their method "regularized discriminant analysis" (RDA), which is the generic name that chemometricians have attached to the entire collection of ideas and algorithms that focus on shrinkage and stabilization. One should be cognizant of the fact that RDA is a very specific type of shrinkage model and uses a very specific call on how parameters in that model are estimated. For the sake of convenience and expedience, there will be no attempt to differentiate among the various shrinkage methods that have been published in the literature.

To make some sense out of this, it will be necessary to introduce some notation that has been ignored until now. In LDA, there is an assumption of a common covariance matrix for all groups. The usual estimate of this matrix is found by pooling the individual estimates, in exactly the same way pooled variances are estimated in t-tests on means, assuming a common variance across the populations:

$$\mathbf{S}_p = \frac{(N_1 - 1)\mathbf{S}_1 + (N_2 - 1)\mathbf{S}_2}{N_1 + N_2 - 2} \tag{2.5}$$

When an estimate is referred to as having been "shrunk", what is meant is that a new matrix has been formed which is a convex combination of the original matrix and one that is more stable (i.e., lower condition number), such as $\mathbf{S}_p$ or the identity matrix. Sometimes the covariance matrix is shrunk towards both $\mathbf{S}_p$ and **I**. In Friedman's RDA, for example, each $\mathbf{S}_i$ is first shrunk toward $\mathbf{S}_p$ in a particular way, and then the resulting compound matrix is, in turn, shrunk towards the **I** matrix in another carefully prescribed way. The resulting estimate is a function of two "shrinkage parameters," $\lambda$ for the pooled covariance matrix and $\gamma$ for the identity matrix. Both $\lambda$ and $\gamma$ will have values that vary between 0 and 1. If $\lambda = 0$ and $\gamma = 0$, then we have QDA, whereas if $\lambda = 1$ and $\gamma = 0$, we have LDA. Thus, QDA and LDA are special cases of RDA. In RDA, the optimum value for $\lambda$ and $\gamma$ is estimated using a simple, but elegant cross-validated error rate performed on a unit square defined by $\lambda$ and $\gamma$. These two parameters are varied by increments of 0.1 on the grid. In other words, a vector of misclassifications as a function of the shrinkage parameters, $\lambda$ and $\gamma$, is generated, with the values of the evaluated parameters corresponding to the lowest error rate selected.

The problem of covariance stabilization in classification was first tackled by Svante Wold in 1976 [34]. Wold recognized the fact that QDA and LDA are guaranteed to produce an optimal classification surface, even though these two methods could seldom be applied to classification problems in chemistry because there are usually too few samples to reliably estimate the inverse of the covariance matrix. Wold addressed the

problem of covariance stabilization in discriminant analysis by developing a biased estimate of the covariance matrix. He called his method SIMCA [35], which is similar in form to QDA, where the inverse of the covariance matrix for each class in the data set is approximated by a principal component representation involving the so-called secondary eigenvectors. In other words, the inverse of the class k covariance matrix $C_k^{-1}$ can be represented by the spectral decomposition

$$C_k^{-1} = \sum_{j=1}^{p} [\nu_{jk} \nu_{jk}^{T}] / \lambda_{jk}] \tag{2-6}$$

where $\upsilon_{jk}$ is the $j^{th}$ principal component of $C_k$, $\lambda_{jk}$ is the corresponding eigenvalue, and p is the number of features in the data. When reconstructing $C_k^{-1}$, it is the smaller eigenvalues, not the larger ones that are the most important. However, smaller eigenvalues are difficult to reliably estimate in small sample high dimensional settings. By taking the average of these smaller eigenvalues, Wold hoped to filter out the noise in them and hence obtain more reliable estimates of them:

$$C_k^{-1} = \left[ \sum_{j=A+1}^{p} [\nu_{jk} \nu_{jk}^{T}] / \sum_{J=A+1}^{p} \lambda_{jk}] \right] \tag{2-7}$$

where A is the number of principal components necessary to describe class k. (Thus, the maximum likelihood estimate of the inverse of the covariance matrix, which conveys information about the size, shape, and orientation of the data cloud for each class, is being replaced by a principal components estimate.) For problems with a low object to descriptor ratio, Wold has shown that his bias estimate is usually a better approximation of the inverse of the variance covariance matrix than maximum likelihood estimates.

### 2.5.5 Summary

From the point of view of "optimality," LDA or QDA should be used for classification when it can be used. That is, with the context properly defined, there is no better classification paradigm from the point of view of minimizing theoretical misclassification probabilities than LDA or QDA, at least within the rational framework provided by statistical theory. We have already established the sense in which LDA and CDA, paired with a sensible classification rule, are equivalent, and when they are not equivalent. Both are useful and rational, and they are sometimes identical.

A fundamental problem associated with the use of LDA or QDA is the requirement that the associated pooled or individual within-groups covariance matrices be invertible. For data sets with collinear features or with significantly more features than observations, this may not be the case. In these situations, a form of ridging or shrinkage can be employed to stabilize these matrices. Alternatively, one can use PCA to reduce the dimensionality of the data, and then use the scores from principal component analysis as descriptors for LDA or QDA. Either paradigm will work provided that most of the features in the data set are not noisy. Otherwise, signal is averaged with noise over a large number of variables with a discernible loss in signal amplitude if the noisy features are not removed from the data. In these situations, feature selection, which is the subject of the next chapter, must be applied to increase the signal to noise ratio of the data by discarding variables that are not related to the classification problem of interest.

While it may not be a particularly startling revelation that PCA is not the best paradigm to use for discrimination (since it is only capable of identifying total variability and is not focused on distinguishing "between-groups" and "within-groups" variability),

principal component scores, nevertheless, are often used in classification problems, sometimes as an end in and of itself, and sometimes as a first step to facilitate the reduction in dimension. The literature persists in publishing papers wherein LDA (or QDA) could have been used to classify the data, since there were no apparent problems with dimensionality or numerical stability, but instead PCA was used. This may be because the user is more familiar with PCA than LDA and QDA or because of readily available software for PCA. Granted that some scientists prefer to use principal component analysis when it "works", because it allows for an unstructured type of classification as there is no *a priori* inherent group structure that has to be specified, it does not constitute a general solution to problems in discriminant analysis involving high dimensional data. If between-groups variability dominates within-groups variability, as is often the case in chromatographic studies, for example, principal component analysis score plots will show clustering on the basis of the class membership distribution of the samples and provide results that are comparable to LDA or QDA. [36].

# CHAPTER 3

# GENETIC ALGORITHMS FOR FEATURE SELECTION AND PATTERN RECOGNITION

## 3.1 INTRODUCTION

Pattern recognition or classification is one of the more common techniques used in chemometrics. Pattern classification involves the assignment of unknown samples into distinct categories using a vector of variables that are viewed as a pattern. Thus, samples are categorized using characteristic features or "patterns" that are contained in a data vector. Pattern recognition got its start in chemistry when Edmund Malinowski began to apply principal component analysis to problems involving the modeling of chemical data [37]. He discovered that it was possible to elucidate the nature of the factors influencing the data, often the fundamental question in a chemical problem. Later, a few scientists began to apply these techniques to problems in analytical chemistry. They recognized the fact that many problems in analytical chemistry could be structured in a form amendable to solution by the same techniques used in optical and audio recognition. Algorithms could be trained to recognize chemical structures based on mass spectral or infrared absorbance data or classify samples from a set of chemical measurements [38, 39]. The dramatic increase in the number and sophistication of chemical instruments triggered interest in the development of pattern recognition

techniques that could extract information from the large arrays of data that were being routinely generated. Much of the growth in the field of pattern recognition that has occurred was and continues to be driven by the press of too much data.

Problems arise when applying pattern recognition methods to multivariate chemical data. First, classification success rates often vary with the pattern recognition method chosen. Second, low classification success rates for the prediction set can be obtained despite a linearly separable training set. Third, discriminant development is difficult to automate. Potentially, these problems can be remedied since all discriminant analysis techniques will perform equally well when the problem is simple. By identifying the appropriate features, a "hard" problem can be reduced to a "simple" one. Therefore, feature selection is the crucial step in any pattern recognition study. The feature selection method used should be multivariate in nature (see Figure 3.1) to ensure that crucial features are not discarded from the analysis. Filters, which select variables by ranking them using either the Fisher ratio or the variance weight [40], are often preferred because of their computational and statistical scalability. However, the variables selected by filters are usually not optimal for a given predictor because they score features individually and independent of each other and as such are at a loss to determine which feature combinations give the best classification results.

Our own experience in pattern recognition is that irrelevant features often introduce so much noise that a good classification of the data cannot be obtained. When these irrelevant features are removed, a clear and well-separated class structure can be found. The deletion of irrelevant variables is, therefore, a major goal of any pattern recognition study since noisy variables increase the chances of false classification and

decrease the classification success-rates obtained with new data. For averaging techniques such as LDA and QDA (see Chapter 2), feature selection is vital since signal is averaged with noise over a large number of variables with a loss of discernible signal amplitude when noisy features are not removed from the data. With neural networks (see Chapter 4), the presence of irrelevant measurement variables may cause the network to focus its attention on the idiosyncrasies of individual samples due to the net's ability to approximate a variety of complex functions in higher dimensional space, thereby causing it to loose sight of the broader picture, which is essential for generalizing any relationship beyond the training set. There are many other potential benefits associated with feature selection, for example, better data visualization and better understanding of the essential features that play an important role in governing the behavior of the system under investigation.

Feature selection is also necessary because of the sheer enormity of many classification problems, e.g., DNA array data, which consists of thousands of descriptors per observation but only 50 or 100 observations distributed equally between two classes. Feature selection can improve the reliability of a classifier because noisy variables will increase the chances of false classification and decrease classification success-rates obtained on new data. It is important to identify and delete features from the data set that contain information about experimental artifacts or other systematic variations in the data not related to legitimate chemical differences between the classes represented in the study. Feature selection can reduce measurement and storage requirements for the classifier as well as training and utilization times.

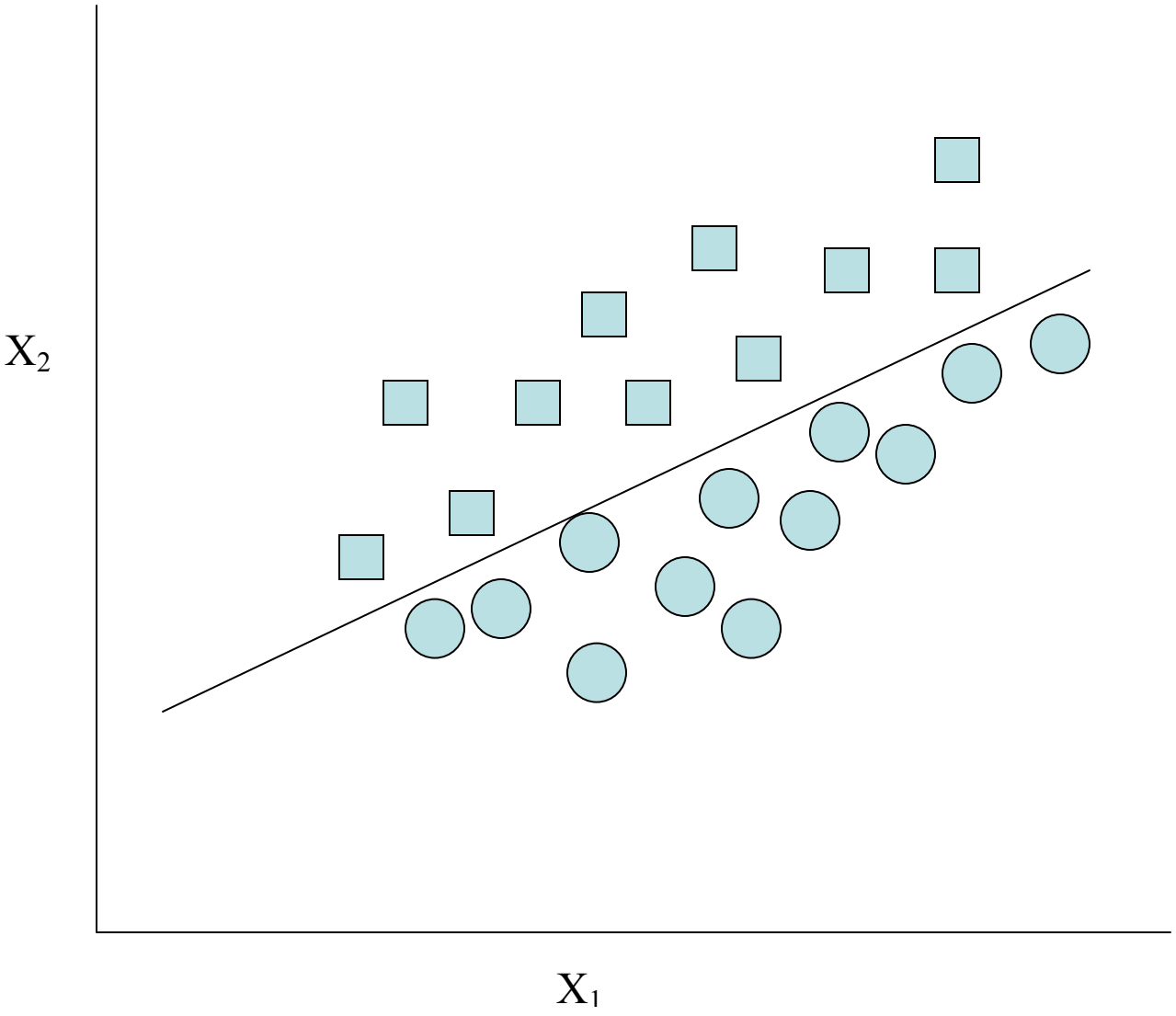
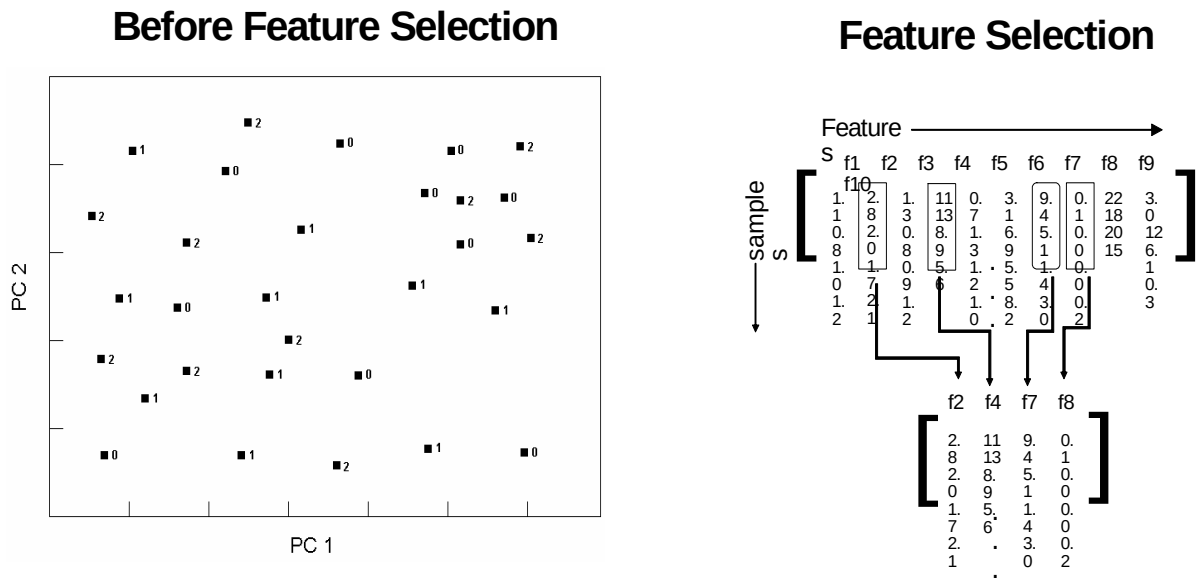



Figure 3.1. Data set consisting of two classes: ovals are acceptable and rectangles are unacceptable samples. Each sample is characterized by two measurements: $x_1$ and $x_2$. Univariate criteria for feature selection such as the Fisher ratio or the variance weights would rank $x_1$ and $x_2$ as uninformative variables.




# After Feature Selection

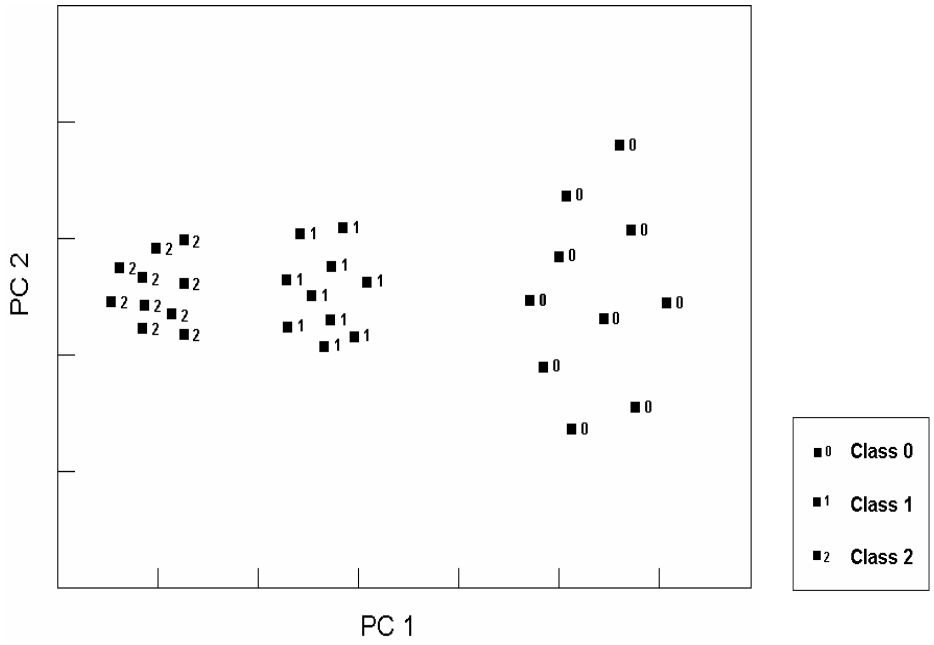



Figure 3.2. A plot of the two largest principal components developed from all of the features in the data set does not show class separation. When principal components are developed from the four features that contain information about class membership, sample clustering on the basis of class is evident in a principal component plot of the feature selected data.

The approach to feature selection described in this chapter is based on a very simple idea - identify a set of measurement variables that optimize the separation of the classes in a plot of the two or three largest principal components of the data. Because principal components maximize variance, the bulk of the information encoded by these features is about differences between classes in the data set. This idea is demonstrated in Figure 3.2, which shows a plot of the two largest principal components of a data set prior to feature selection. The data set consists of 30 samples distributed between 3 classes (good, better, and best). Each sample is characterized by 10 measurements. However, only 4 of these measurements contain information about the classification problem. When a principal component map of the data is developed using only these 4 measurements, sample clustering on the basis of class is evident.

Using this approach to feature selection, an eigenvector projection of the data is developed that discriminates classes in the data set by maximizing the ratio of between- to within-group variance (which is the same criterion that is used in CVA to develop projections of the data for classification). This approach to feature selection has a number of advantages. It avoids overly complicated solutions that do not perform as well on the prediction set because of over-fitting, which is a serious problem with most wrapper methods. Although a principal component plot is not a sharp knife for discrimination, if we have a principal component plot that shows clustering, then our experience is that we will be able to predict robustly using this set of descriptors. For redundant features, noise reduction and better class separation can be achieved if PCA is used to characterize the information content of the redundant measurement variables.

Furthermore, a principal component plot displays variability between large numbers of samples and shows the major clustering trends present in the data; the user can visually identify the presence of confounding relationships in the data, thereby gaining insight into how the decision is made for a classification.

To identify these features, it is necessary to use a genetic algorithm [41, 42], which employs a survival of the fittest approach. Genetic algorithms exploit knowledge contained in a population of solutions (i.e., feature subsets) to generate new and better solutions while simultaneously using random choice as a tool to guide a highly exploitive search of the data. Genetic algorithms do not make any assumptions about the geometry of the response surface beyond the fitness of a potential solution to the optimization problem. Discontinuities or singularities, which can rule out the use of some derivative based methods, do not pose a problem for genetic algorithms because many points in different regions of the search space are simultaneously investigated while searching for the best solution. Therefore, results can be quite robust in terms of the starting location. The genetic algorithm's search of the solution space is efficient, and the computational environment offered by a genetic algorithm can be readily adjusted to match a particular application. Genetic algorithms are best suited to problems whose underlying optimization function is unknown, poorly understood, exceedingly complex, or error prone, or some combination thereof, which is the case with feature selection.

### 3.2 Genetic Algorithms

Genetic algorithms were developed by Holland [43] as part of a study on adaptive processes. They are based on the principles of natural evolution and selection. The procedure builds a population of binary strings, each of which represents a possible

solution. Fit solutions are allowed to live and breed. A block diagram of the genetic algorithm for pattern recognition analysis is shown in Figure 3.3.

Implementation of a genetic algorithm requires a population of strings (i.e., candidate solutions) and heuristics to manipulate them. The actual procedure involves several interrelated steps. First, an initial population of strings is generated. Each binary string represents a potential solution to the problem, i.e., a unique subset of features. (For our applications, the length of the chromosome is equal to the number of features in the data set.) For a feature to be included in the subset, it is necessary for the corresponding bit in the string to be set at 1. If the bit is set to 0, the feature is not included in the subset. During each generation, the strings are decoded yielding the actual parameter set, which is sent to the fitness function for evaluation. Each string is assigned a value by the fitness function, which is a measure of the quality of the proposed feature subset for the classification problem.

Reproduction is implemented using three operators: selection, recombination, and mutation. In selection, the fitness is used to select strings for recombination. Solutions with a high fitness have a higher probability of being selected. These solutions then undergo a structured yet randomized exchange of information with the expectation that good solutions will generate even better ones (i.e., recombination). Usually, a crossover operator is employed to generate new strings or solutions (see Figure 3.4). Additional randomness or variability is achieved by the mutation operator, which flips the state of single bits based on certain probabilities. This allows the genetic algorithm to explore other regions of the solution space. If the genetic algorithm finds a better point, the genes from this point can invade the population, with the optimization continuing in a new

direction. The boosting algorithm adjusts the genetic algorithm's internal parameters for the next iteration. The aforementioned procedure (fitness evaluation, reproduction, and adjustment of internal parameters) is repeated until a specified number of generations have been executed or a feasible solution is found.

The selection criterion used for reproduction exhibits bias for the higher-ranking strings. Hence, the new population is expected to perform better on average than its predecessor. However, the reproductive operators also assure a significant degree of diversity in the population because of crossover and mutation. This is shown by the schema theorem. A schema also referred to as a similarity template [44] represents a set of chromosomes. For example, the schema {1 * * * * * * 0} will match all chromosomes with eight bits that start with 1 and end with 0 with either 0 or 1 in positions 2 thru 7. Genetic algorithms use schema implicitly. For a genetic algorithm operating on a population of chromosomes of fixed length l, there are $3^l$ unique schema or patterns. Each chromosome is a member of $2^l$ of them. For example, {0.1} is a member of the following schema: {01}, {*1}, {0*}, and {**}. When the fitness function of a genetic algorithm is evaluating a chromosome, it is also evaluating many schemas. In a population of identical chromosomes, there are $2^l$ schema present, and in a population of n unique chromosomes, there can be as many as $n2^l$ schema represented. Evaluating different chromosomes which are members of the same schema can be thought of as estimating the average value of that pattern. Even though these averages are not explicitly calculated, the survival of the pattern and the number of representative chromosomes can be expressed in terms of these averages.

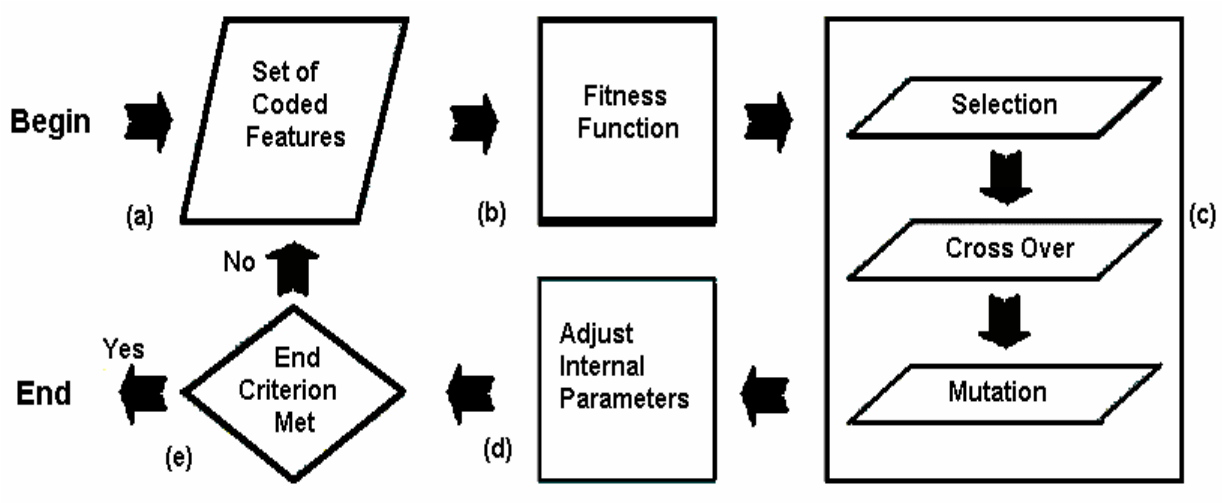



Figure 3.3. A block diagram of the genetic algorithm for pattern recognition analysis

Let S be a schema present in the population at generation g. Its multiplicity, m(S, g) is defined as the number of instances of S in the population at generation g. The expected number of chromosomes which represents S in the next generation is given by Equation 3-1 (the schema theorem):

$$m(S, g+1) \geq m(S,g) \cdot \frac{f(c)}{\bar{f}} \left( (1 - p_c \cdot \frac{d(S)}{l-1}) \right) \left( (1-p_m)^{o(S)} \right) \qquad (3\text{-}1)$$

where m(S, g + 1) is the expected number of chromosomes representing schema S in the next generation (g + 1) based on the number of chromosomes representing schema S in the current generation, m(S, g). The ratio of the fitness of the chromosome representing schema S to the average fitness of the population is given by $\frac{f(S)}{\bar{f}}$, and collectively m(S, g) $\frac{f(S)}{\bar{f}}$ is the likelihood that schema S is represented in the population. Due to selection pressure alone, schema will grow or decay depending on their fitness [45]. However, chromosomes selected for reproduction will undergo crossover and mutation. These operators can disrupt the schema S such that S is not present in the next generation. The second factor in Equation 3-1, $\left( (1 - p_c \cdot \frac{d(S)}{l-1}) \right)$, accounts for the probability that S survives crossover; $p_c$ is the probability that a chromosome undergoes crossover. The last factor in Equation 3-1, $\left( (1-p_m)^{o(S)} \right)$, accounts for the probability that S survives the mutation operator. Here, $p_m$ is the probability that a given bit is flipped, and o(S) is the order of S or the number of non-* bits in S.

The consequence of a genetic algorithms use of schema is an implicit parallelism [45]. At each evaluation, the genetic algorithm is aware of a particular point in the fitness landscape because of the chromosomes it is evaluating. According to the schema theorem, the genetic algorithm makes observations about areas of the search space based on the schema, which allows the genetic algorithm to focus its attention on "hot spots" or areas likely to have a high fitness in the solution space similar to that of a gradient descent search. (In a gradient descent search, the value at a random position is calculated. The points around it are also inspected to calculate the direction and magnitude of greatest local descent. A new point in that direction is sampled and the process is repeated until the minimum is reached.) An obvious difference between these two methods (genetic algorithm versus gradient descent) is the number of points sampled per iteration. Even when the gradient descent method is modified to sample multiple points per iteration, the next point or set of points is near the last in the solution space, whereas genetic operators such as crossover and mutation produce points which are near or are distant from the parents in the solution space depending on the bits that are exchanged or flipped in the chromosome. Consequently, a genetic algorithm is less likely to get stuck in a local minimum in the solution space.

Genetic algorithms are probabilistic, neither random nor deterministic. This is demonstrated in the selection process where a chromosome's chances of being selected are weighted against its fitness. It is preferable that offspring are not produced in the same way each time. This is addressed by assigning a probability to each reproductive function. When two chromosomes are selected for reproduction, a mechanism is chosen according to its probability (P). The user can assign $p_m$ equal to 0.01 and $p_c$ equal to 0.5.

This mixing of reproductive operations preserves a certain amount of variation in the population.

### 3.3 PCKaNN

The development of a genetic algorithm (GA) for pattern recognition analysis of chemical data, PCKaNN, has recently been reported in the literature [46-52]. The pattern recognition GA selects features that optimize the separation of the classes in a plot of the two or three largest principal components of the data. Because principal components maximize variance, the bulk of the information encoded by these features is about differences between classes in the data set. The pattern recognition GA is able to focus on those classes and/or samples that are difficult to classify as it trains by adjusting the values of the class and sample weights. Over time, the algorithm learns its optimal parameters in a manner similar to a neural network. The pattern recognition GA integrates aspects of artificial intelligence and evolutionary computations to yield a "smart" one -pass procedure for feature selection and pattern classification. The various components of the pattern recognition GA are described below.

#### *Fitness Function*

The fitness function of the pattern recognition GA scores the principal component plots and thereby identifies a set of features that optimize the separation of the classes in a plot of the two or three largest principal components of the data. Because principal component analysis is used to determine the information present in a given subset of features, it is precisely this variation in principal components (different coordinate system for each feature subset) that allows for meaningful comparisons to be made between sets of features.

The principal component plot used in the fitness function of the pattern recognition GA acts as an information filter. Features sets are selected based on their principal component plots. A good principal component plot can only be generated by features that maximize between to within group differences. Hence, principal component analysis limits the search to these types of feature subsets, significantly reducing the size of the search space and also the probability of spurious or chance classification. (An exhaustive search of the feature space is plausible using a genetic algorithm equipped with this fitness function, which is not the case with other wrapper methods that have been previously reported in the literature.) Features that contain discriminatory information about a particular classification problem are often correlated, which is why feature selection methods based on principal component analysis or other variance based methods are preferred to display the information content of the data.

To facilitate the tracking and scoring of the principal component plots, class and sample weights, which are an integral part of the fitness function, are computed (see equations 3-2 and 3-3) where CW(c) is the weight of class c (with c varying from 1 to the total number of classes in the data set). $SW_c(s)$ is the weight of sample s in class c. The class weights sum to 100, and the sample weights for the objects comprising a particular class sum to a value equal to the class weight of the class in question.

$$CW(c) = 100\frac{CW(c)}{\sum_{c} CW(c)} \qquad (3-2)$$

$$SW(s) = CW(c)\frac{SW(s)}{\sum_{s \in c} SW(s)} \qquad (3-3)$$

Each principal component plot generated for each feature subset after it has been extracted from its chromosome is scores using the K-nearest neighbor classification algorithm [53]. For a given data point, Euclidean distances are computed between it and every other point in the principal component plot. These distances are arranged from smallest to largest. A poll is taken of the point's $K_c$ nearest neighbors. For the most rigorous classification, $K_c$ equals the number of samples in the class to which the point belongs. (Thus, $K_c$ usually has a different value for each class.) The number of $K_c$ nearest neighbors with the same class label as the sample point in question, the so-called sample hit count, SHC(s), is computed ($0 \leq \text{SHC(s)} \leq K_c$) for each sample. It is then a simple matter to score a principal component plot (see Equation (3-4). First, the contribution to the overall fitness by each sample in class 1 is computed, with the scores of the samples comprising the class summed to yield the contribution by this class to the overall fitness. This same calculation is repeated for classes 2, 3, etc., with the scores from each class summed to yield the overall fitness, *F (d).*

$$F(d) = \sum_{c} \sum_{s \in c} \frac{1}{K_c} \times \text{SHC(s)} \times \text{SW(s)} \qquad (3-4)$$

To understand scoring, consider a data set with two classes, which have been assigned equal weights. Class 1 has ten samples, and class 2 has 20 samples. For uniformly distributed sample weights, class 1 samples will have a weight of 5 and class 2 samples will have a weight of 2.5, since each class has a weight of 50 and the sample weights in each class are uniformly distributed. Suppose a sample in class 1 has, as its nearest neighbors, seven class 1 samples in a principal component plot developed from a particular feature subset. Hence, $\text{SHC(s)}/K_c = 7/10$, and the contribution of the sample to the fitness function for the particular feature subset equals 0.7*5 or 3.5. Multiplying

$SHC/K_c$ by SW(s) for each sample and summing up the corresponding product for the 30 samples in the data set yields the value of the fitness function for this particular feature subset.

***Reproduction***

Selection, crossover, and mutation operators are applied to the chromosomes. Fit strings are retained and selected for breeding, a process called selection, which is the first step toward population reorganization. The fit feature subsets are then broken-up, swapped, and recombined, creating new feature subsets, which are introduced into the population of potential solutions. This process is called crossover (see Figure 3.4). In this study, the selection and crossover operators are implemented by ordering the population of strings, i.e. potential solutions, from best to worst, while simultaneously generating a copy of the same population and randomizing the order of the strings in this copy with respect to their fitness (see Figure 3.5). A fraction of the population is then selected as per the selection pressure, which is set at 0.5. The top half of the ordered population is mated with strings from the top half of the random population, guaranteeing the best 50% are selected for reproduction, while every string in the randomized copy has a uniform chance of being selected. This is due to the randomized selection criterion imposed on strings from this population. If a purely biased selection criterion were used to select strings, only a small region of the search space would be explored. Within a few generations, the population would consist of only copies of the best strings in the initial population.

For each pair of strings selected for mating, two new strings are generated using three-point crossover. A mutation operator is then applied to the new strings. The mutation probability of the operator is usually set at 0.01, so 1% of the feature subsets are

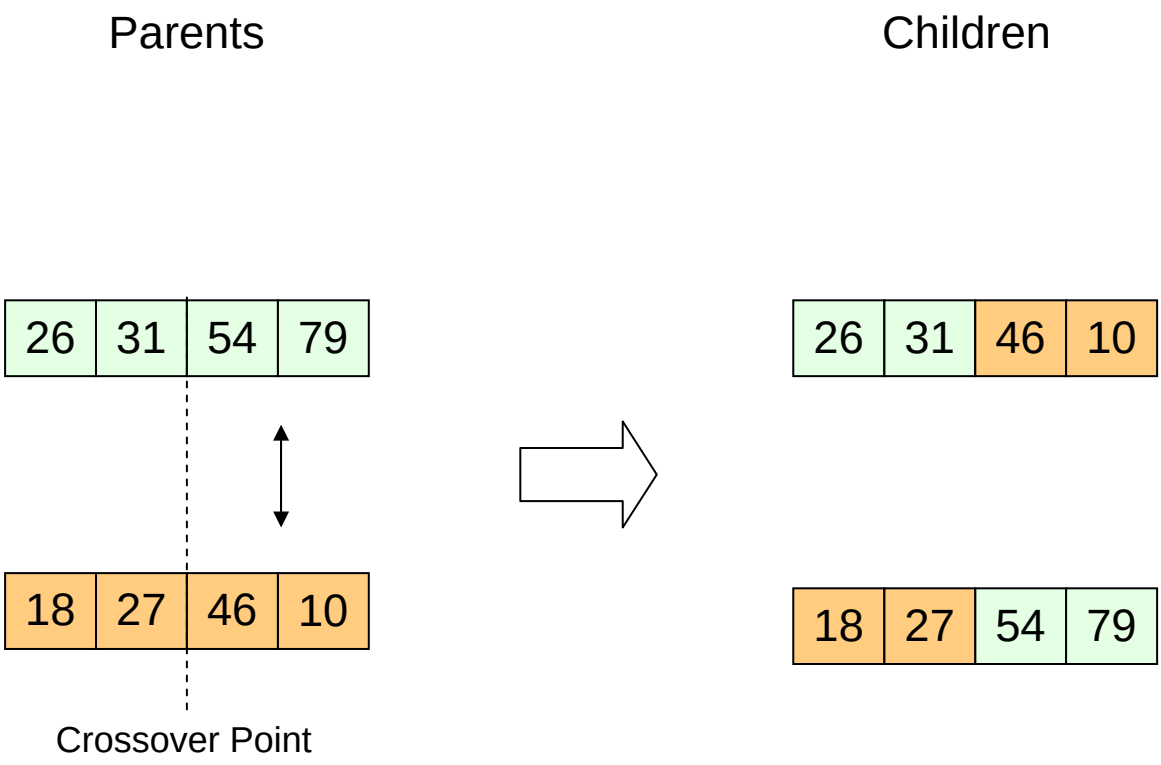



Figure 3.4. Single point crossover: alleles are swapped while simultaneously preserving their position.

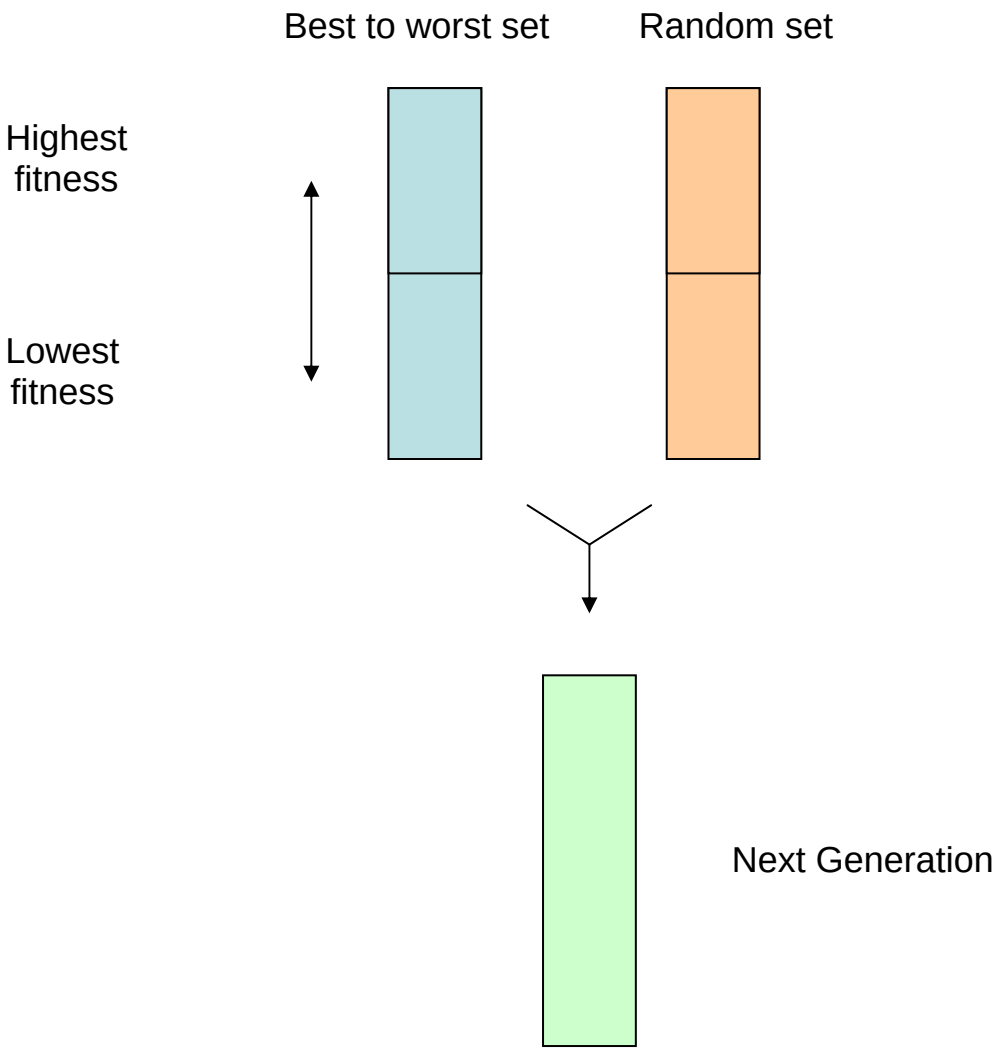



Figure 3.5. The top half of the ordered population is mated with strings from the top half of the random population, guaranteeing the best 50% are selected for reproduction, while every string in the randomized copy has an equal chance of being selected to ensure sufficient diversity in the population.

selected at random for mutation. A chromosome marked for mutation has a single random bit flipped, which allows the GA to explore other regions of the parameter space. The resulting population of strings, both the parents and children, are sorted by fitness, with the top $\phi$ strings retained for the next generation. Because the selection criterion used for reproduction exhibits bias for the higher-ranking strings, the new population is expected to perform better on average than its predecessor. The reproductive operators used, however, also assure a significant degree of diversity in the population, since the crossover points of each chromosome pair is selected at random.

***Adjusting Internal Parameters***

The fitness function of the GA is able to focus on those samples and classes that are difficult to classify by boosting their weights over successive generations. (Boosting the weights is referred to as adjusting the internal parameters in the block diagram of the genetic algorithm shown in the previous section.) In order to boost, it is necessary to compute both the sample-hit rate (SHR), which is the mean value of $SHC/K_c$ over all feature subsets produced in a particular generation (see Equation 3-5), and the class-hit rate (CHR), which is the mean sample hit rate of all samples in a class (see Equation 3-6). $\phi$ in equation 3-5 is the number of chromosomes in the population, and AVG in Equation 3-6 refers to the average or mean value. During each generation, class and sample weights are adjusted by a perceptron (see Equations 3-7 and 3-8) with the momentum, P, set by the user. (g + 1 refers to the current generation, whereas g is the previous generation.) Classes with a lower class hit rate and samples with a lower sample hit rate are boosted more heavily than those classes or samples that score well.

$$\mathrm{SHR}(s) = \frac{1}{\phi} \sum_{i=1}^{\phi} \frac{\mathrm{SHC}_i(s)}{\mathrm{K}_c} \qquad (3-5)$$

$$\mathrm{CHR}_g(c) = AVG(\mathrm{SHR}_g(s) : \forall_{s \in c}) \qquad (3-6)$$

$$\mathrm{CW}_{g+1}(s) = \mathrm{CW}_g(s) + \mathrm{P}(1 - \mathrm{CHR}_g(s)) \qquad (3-7)$$

$$\mathrm{SW}_{g+1}(s) = \mathrm{SW}_g(s) + \mathrm{P}(1 - \mathrm{SHR}_g(s)) \qquad (3-8)$$

The change in the class weights is monitored throughout the run. If the average change in the class weights is greater than some tolerance, the genetic algorithm is said to be learning its optimal class weights. Once this tolerance has been reached, the class weights become fixed. This initiates the second stage. The momentum, which controls the rate at which the sample weights are changed, is initially assigned a value of 0.8 while the genetic algorithm is learning, but the momentum is, adjusted to 0.4 once the class weights become fixed. These values have been chosen in part because they facilitate learning by the genetic algorithm but do not cause a particular sample or class to dominate the calculation, which would result in the other samples or classes not contributing to the scoring by the fitness function.

Boosting is crucial for the successful operation of the pattern recognition GA because it modifies the fitness landscape by adjusting the values of the class and sample weights. This helps to minimize the problem of convergence to a local optimum. Hence, the fitness function of the GA changes as the population is evolving towards a solution

using information from the population to guide these changes. A block diagram of the boosting algorithm for the pattern recognition GA is shown in Figure 3.6.

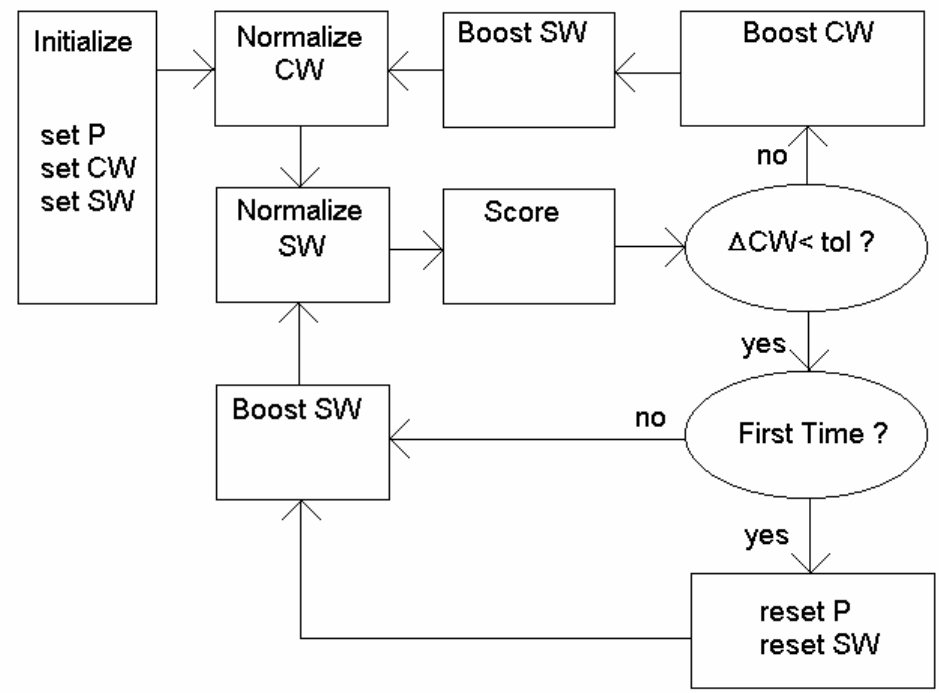



Figure 3.6. Block diagram of the pattern recognition GA with boosting which is used to adjust the weights of difficult classes and/or samples.

***End Criterion***

During each generation, class and sample weights are updated using the class and sample hit-rates from the previous generation. Evaluation, reproduction, and boosting of potential solutions are repeated until a specified number of generations are executed or a feasible solution is found.

***GA Parameters***

There are a number of parameters that affect the performance of the pattern recognition GA including the choice of crossover and mutation rate and the configuration of the initial population. Our experience with the pattern recognition GA has shown that

3-point crossover works. However, the number of features in each feature subset of the initial population should also be treated as an important parameter. If the feature sets are initially sparse, the probability of including features, which are neither good nor bad, is low since the principal component based fitness function does not provide additional points for adding them. Conversely, the probability of removing these features from less sparse feature subsets is also low since there is no advantage in deleting them. For data sets with a large number of good features, it is probably best not to employ sparse feature subsets in the initial population. Otherwise, it may take thousands of generations to ensure the inclusion of all good features in the solution.

To ensure removal of features, which are neither good nor bad, the corresponding loading plot that is generated with each principal component plot can be examined by the pattern recognition GA. If the loadings for a particular feature are near zero for both principal components, the feature is a likely candidate for removal since its contribution to the principal component plot is negligible. Culling can be implemented every 10 generations, for example, to check for features, which are neither good nor bad. During the generation when culling is implemented, crossover is not performed on the strings.

Varying the composition of the initial population or the mutation rate can prove beneficial in optimizing a solution but this fact should not be viewed negatively as suggested by some workers since it allows the user to vary the search of the solution space ensuring a more careful analysis of the data. Given the small number of iterations required for a solution (usually less than 100), the advantages of using these two GA parameters as search variables outweighs any disadvantage that might be incurred due to increased complexity.

A drawback of a genetic algorithm is that one cannot control the rate of convergence, but convergence is not what we are seeking. A genetic algorithm can evade local optima, but this does not mean that convergence necessitates an optimal solution. Convergence as a benchmark for the success of a GA would suggest that any genetic algorithm provides a deficient solution. However, the quality of the best solution found – and how quickly and reproducibly it is found – is the guide being used to determine the success of this method. The ease, speed, and reproducibility of our pattern recognition GA have been demonstrated on a variety of data sets. We attribute the success of the pattern recognition GA to the large number of optimum solutions that exist in the data as a result of the high degree of collinearity between measurement variables in the data set.

### 3.4 *Incorporation of Transverse Learning in PCKaNN*

Projections of data that reveal clustering are valuable for identifying relationships between properties and the measurement variables for sets of observations. Feature subsets that reveal interesting projections of the data can be found using the Hopkins statistic [54] to score the principal component plots. The Hopkins statistic (H), which is a fast and simple method, does not make any assumptions about the data in order to assess clustering. It is defined as

$$H = \frac{\Sigma U_j}{\Sigma U_j + \Sigma W_j} \tag{3-9}$$

where $U_j$ is the distance between a randomly selected location on the PC plot and the data point nearest to it. The x, y coordinates for each location are obtained from a random number generator. $W_j$ is the distance between a randomly selected data point

and its nearest neighbors in the same principal component plot. The number of random locations and the number of data points selected will be the same and are usually set as 10% of the number of data points in the data set. If the data are well clustered, $\Sigma U_j$ will be significantly larger than $\Sigma W_j$ since most of the PC plot will be barren of points and it is likely that a randomly selected location would not be occupied by a data point. If the data are randomly distributed in the PC plot, then one would expect that $\Sigma U_j$ and $\Sigma W_j$ would be similar in magnitude since the random location would either be occupied by a data point or a data point would be in its vicinity. For this reason, the value of the Hopkins statistic varies from 0.5 (no clustering, $\Sigma U_j = \Sigma W_j$) to 1.0 (perfect clustering, $\Sigma U_j >> \Sigma W_j$). Because H varies from 0.5 to 1.0, we have found that it is necessary to scale its values using a sigmoid transfer function.

For underdetermined data sets (i.e., more features than objects), we recognize the fact that even quite well behaved multivariate normal distributions with no outliers will have variables that produce eigenvector projections containing points that appear as outliers in a principal component plot. An index such as the Hopkins statistic, which generates high values for such projections, will tend to be distracted from other types of structures. Therefore, the Hopkins statistic must be enhanced. Recently an influence function [55] for principal components has been developed, which can identify observations with high leverage (i.e., outliers) and deweight their contribution to the Hopkins statistic score thereby ensuring that the Hopkins statistic is a meaningful metric to assess clustering of the data points in a principal component plot. The robustification procedure used for the Hopkins statistic is defined in Equation 3-10 where

max(influence$_i$) is the influence value (which is a fraction) for the data point having the greatest influence on the eigenvalue of the i$^{th}$ principle component.

$$H = H - H\sum_{i=1}^{PC} \max(\inf luence_i) \tag{3-10}$$

The alternative to the Hopkins statistic is a moment-based index, which is used in projection pursuit [56] or independent component analysis [57]. However, a moment-based index such as the kurtosis also achieves high values for projections that contain these so-called outliers. Unlike the Hopkins statistic, it is not apparent how one would go about robustifying a moment-based index while simultaneously preserving its calculability. According to Friedman [58], trimming or other methods based on projected rank do not work.

Previously, projection pursuit and independent component analysis had been considered in lieu of principal component analysis for identification of features that will cause the data to cluster. These two techniques could easily have been substituted for principal component analysis in the pattern recognition GA but this idea was rejected for several reasons. First, the added computational burden due to substitution of projection pursuit or independent component analysis dramatically increased the run time. Second, independent component analysis has the problem of ordering. One does not know in advance which independent components would contain information about class membership. Third, many of the variables in the data sets studied had small Fisher or variance weights even though they were found to be informative in a multivariate setting. In some cases, we have examined histograms of individual descriptors identified by the pattern recognition GA, and the histograms revealed a mound shape pattern suggesting that these variables are most likely Gaussian. Therefore, principal component analysis is

the appropriate technique. Fourth, we have compared the performance of projection pursuit and PCA using data sets and have found no significant difference between these methods for classifying multivariate data. For all of these reasons, this line of investigation has not been pursued further although we are currently rethinking this issue.

Another advantage of the Hopkins statistic is that it can be directly coupled to PCKaNN, creating a fitness function that utilizes transverse learning [59]. For data sets with only a small amount of labeled data and a large amount of unlabeled data (e.g., 100 to 500 observations in a 10,000 object database), this approach is preferred since it will perform better than a learning model developed from a set of features whose selection is based solely on the dichotomization power of the features for observations with known responses. Feature subsets are selected to optimize clustering using all the data points (see Hopkins statistic) and to create class separation using the labeled data points (see PCKaNN). Employing this approach, we have been able to demonstrate marked improvements in our ability to predict future data [60]. The advantage of this approach over support vector machines [61] is that transverse learning is being used not only to predict future data, but also to identify truly informative features in the data set, thereby ensuring a better classification of the data. As in the case of an embedded method, e.g., CART [62], this approach to feature selection also makes efficient use of the data since all of it is used in training.

Although the Hopkins statistic when combined with PCKaNN produces a fitness function that has many advantages, the new fitness function does not take full advantage of the power that is associated with boosting. To overcome this problem, each

chromosome (or feature subset) can be scored using a modified version of the Hopkins statistics (see Equation 3-11). $U$ is the number of unlabelled data points, $USW_j$ is the weight of the j$^{th}$ unlabelled data point, and $d_{ij}$ is the distance between the unlabelled data point $j$ and its nearest neighbor (labeled data point) in the PC plot. Each unlabelled data point is assigned an initial weight of 100/U. To boost the weight of each sample, an average distance vector is computed (see Equation 3-12) and the weights are boosted using the relationship defined in Equation 3-13. The modified Hopkins statistic focuses on the unlabelled data points. By coupling PCKaNN (labeled data points) with the modified Hopkins statistics (unlabelled data points), a second approach to transverse learning, which utilizes boosting, can be used to perform feature selection.

$$MH_i = \sum_{j=1}^{U} \frac{1}{1+d_{ij}} USW_j \qquad (3-11)$$

$$AvgD_j = \frac{1}{\phi} \sum_{i=1}^{\phi} \frac{1}{1+d_{ij}} \qquad (3\text{-}12)$$

$$USW_{g+1}(j) = USW_g(j) + P(1 - AvgD_g(j)) \qquad (3\text{-}13)$$

Successful mining of data requires the user to combine empirical data with careful analysis and prior knowledge and reasoning. Supervised learning represents a systematic approach to this problem, which can be defined as the search for significant structure in data. The pattern recognition GA described in this chapter is designed to search for significant structure in multivariate data. The fitness function of the pattern recognition

GA can be tuned, enabling it to explore the structure of a large data set and to uncover hidden relationships in the data, for example, the discovery of new classes by varying the contribution of the Hopkins statistic and PcKaNN to the fitness function used to score the feature subsets identified by the GA during each generation.

## 3.5 Applications of the Pattern Recognition GA

Pattern recognition methods are usually implemented in four distinct stages: data preprocessing, feature selection, mapping and display, and classification. However, the process is iterative with the results of classification often determining further preprocessing steps and reanalysis of the data. Although the procedures selected for a given problem are highly dependent upon the nature of the problem, it is still possible to develop a general set of guidelines for applying the pattern recognition GA to actual data sets. In this section, a framework for solving the class membership problem rotted on feature selection is presented by way of two studies that have been performed as part of the research described in this thesis using the pattern recognition GA.

### *Classification of Hardwoods, Softwoods, and Tropical Woods by Raman Spectroscopy*

Wood identification is usually accomplished by forestry experts who employ visual microscopy, hardness testing, and/or leaf analysis [62]. Vibrational spectroscopy offers another means of elucidating the structure of wood and characterizing wood types. In this study, Raman spectroscopy and pattern recognition techniques have been used to develop a potential method to characterize wood by type. The test data consisted of 98 Raman spectra of temperate and tropical woods. The temperate woods consisted of 31 hardwoods and 28 softwoods from North America, and the tropical woods consisted of 15 Brazilian and 24 Honduran woods. The Raman spectra were measured on a Perkin-

Elmer System 2000 Fourier-transform spectrometer fitted with the standard Perkin-Elmer Raman attachment and a modified Spectron 301 $Nd^{3+}$ laser ($\lambda$ = 1064 nm). The spectra were measured at a resolution of $4cm^{-1}$. Each Raman spectrum, which was an average of 500 scans, was stored from 3600 to 250 $cm^{-1}$. Further details about the experimental conditions used for the measurement of the FT-Raman spectra can be found elsewhere [63].

All Raman spectra were normalized to unit length to adjust for variations in the scattering cross-section of each sample. For pattern recognition analysis, each wood sample was represented by a data vector $x = (x_1, x_2, x_3 \ldots x_j \ldots x_{3352})$ where $x_j$ is the Raman intensity of the $j^{th}$ point in the normalized Raman spectrum. The data were standardized and autoscaled so that each variable had a mean of zero and a standard deviation of unity within the entire set of 98 Raman spectra.

The first step in this study was to apply PCA to the entire data set. PCA is a powerful method for uncovering hidden relationships in multivariate data. Using this procedure is analogous to finding a new coordinate system that is better at conveying the information present in the data than axes defined by the original measurement variables. The basis vectors of this new coordinate system are the principal components of the data. Each principal component is a linear combination of the original measurement variables. Often, only two or three principal components are necessary to explain all of the information present in a data set when there are a large number of interrelated measurement variables.

Figure 3.7a shows a plot of the scores of the two largest principal components of the 3352-point spectra that comprise this data set. The two largest principal components

of the data explain 41% of the total cumulative variance. Each spectrum is represented as a point in the score plot (1 = soft, 2 = hard, and 3 = tropical). There is overlap between the tropical woods, hard woods, and soft woods in the score plot of the data.

Feature selection was the next step, since deletion of uninformative features would ensure that discriminatory information about wood type would be the major source of variation in the data. A genetic algorithm (PcKaNN) for pattern recognition analysis was used to uncover features characteristic of the Raman profile of each wood-type. In this study, the population consisted of 5000 chromosomes, and the mutation rate was 0.2. Three point cross-over was used, and K for each class was set equal to the number of samples in the class. The genetic algorithm identified informative features in the data by sampling key feature subsets, scoring their principal component plots, and tracking those samples or classes that were most difficult to classify. The boosting routine used this information to steer the population to an optimal solution. After 300 generations, the genetic algorithm identified 11 wavelengths whose principal component plot showed clustering of the Raman spectra according to wood type (see Figure 3.7b). The hardwoods, softwoods, and tropical woods are well separated from each other in the score plot. For these 11 features, between group differences are large compared to within group differences, which would suggest that all pattern recognition methods will work well with this data. An advantage of using a score plot to display the classification results instead of submitting the 11 features to linear or quadratic discriminant analysis for development of a classifier is that it allows the user to better understand how a classification decision is made for a particular sample.

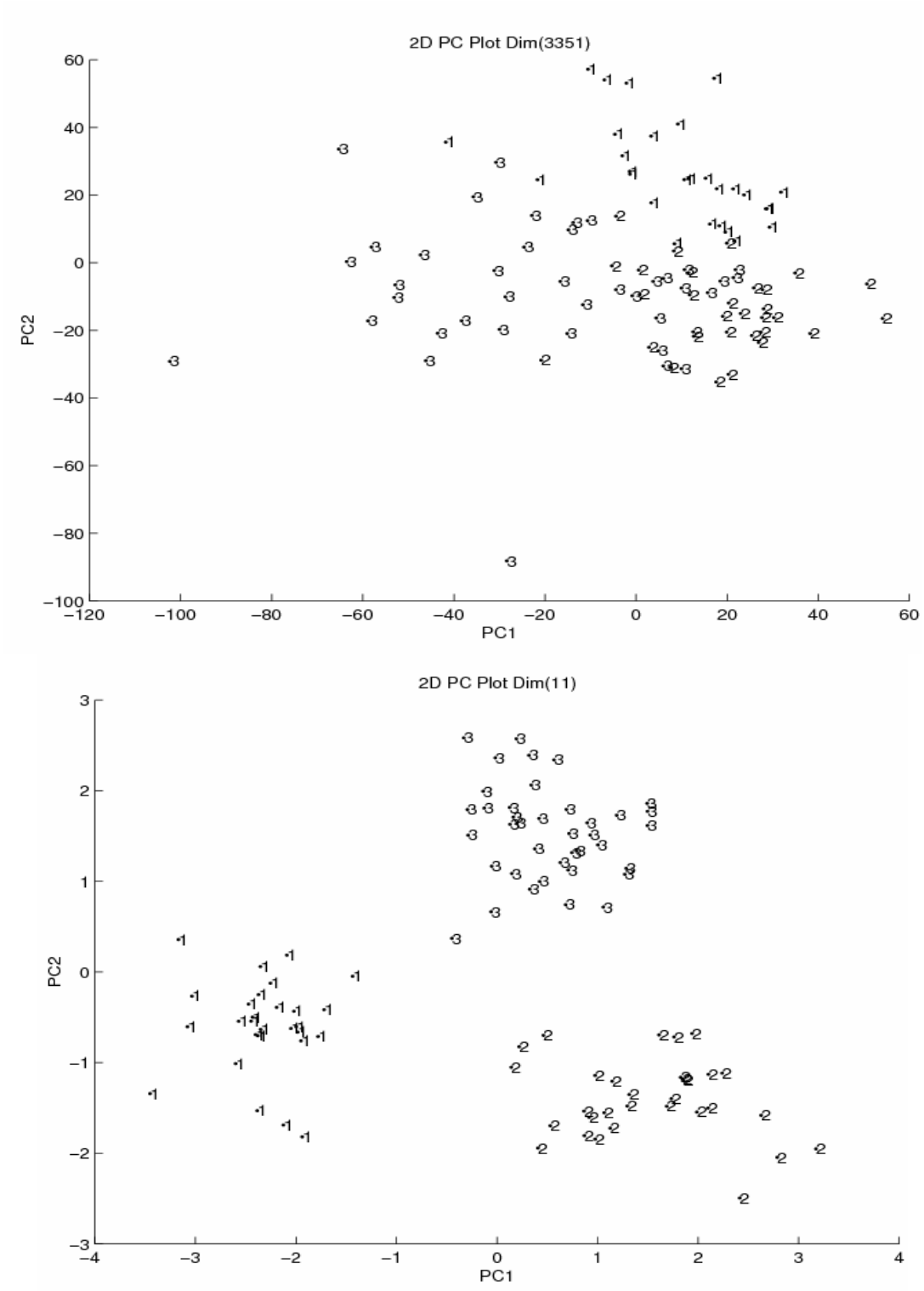



Figure 3.7. A score plot of the two largest principal components of the Raman spectra that comprise the wood data set. Each spectrum is represented as a point in the score plot (1 = soft, 2 = hard, and 3 = tropical). a) 3352 wavelengths, and b) 11 wavelengths identified by the pattern recognition GA.

The ability of a classifier to predict the class membership of a simulated unknown wood sample was tested using a procedure known as segmented cross validation. The data set was divided into N training set prediction set pairs. A classifier is developed for each training set and then tested on the corresponding prediction set. Each sample was present in only one of the N prediction sets generated.

For this study, two training set prediction set pair combinations were investigated: 80%/20% (5 training set prediction pairs with 80% of the samples in each training set and the remaining 20% in each prediction set), and 20%/80% (5 training set prediction set pairs with 20% of the samples in each training set and the remaining 80% in each prediction set). Figures 3.8 thru 3.12 summarize the results of the segmented cross validation for the 80%/20% case. CVA (see Chapter 2) and PCA plots of the features identified as informative using PCKaNN are shown for each training set. Each object in the training set is represented as "1" (softwoods), "2" (hardwoods), or "3" (tropical woods). The prediction set samples are represented as "S" (softwoods), "H" (hardwoods), and "3" (tropical woods). From an examination of the CVA and PCA plots, it is evident that all prediction set samples in this validation were correctly classified. Furthermore, the CVA and PCA plots are comparable. If information about wood-type lies in the directions of maximum variance, then it should not come as a surprise that PCA and CVA yield similar scatter plots (see Iris data set on pages 25-27 of thesis). The conclusion that can be drawn from this validation study is that the bulk of the information encoded by the feature subsets identified by the pattern recognition GA is about wood-type. Using the pattern recognition GA, the CVA paradigm (optimizing

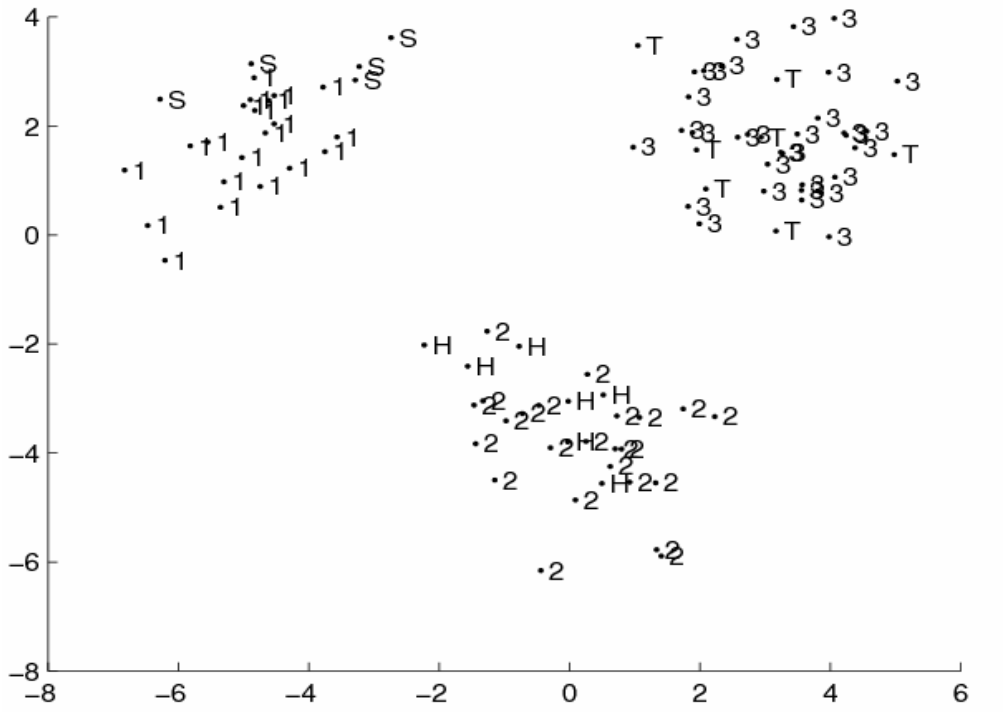
between to within group differences in the data) can be extended to include data sets that are underdetermined (i.e., data sets that contain more features than objects).

The genetic algorithm for pattern recognition analysis with transverse learning was also used to analyze each training set/prediction set pair in the 80%/20% segmented cross validation study. Figures 3.13 thru 3.17 summarize the results for the Hopkins statistic (see pages 62-65) and for the modified Hopkins statistic (page 66). Each object in the training set is represented as "1" (softwoods), "2" (hardwoods), or "3" (tropical woods) in these figures. The prediction set samples are represented as "S" (softwoods), "H" (hardwoods), and "3" (tropical woods) in these figures. From an examination of these plots and the plots in Figures 3.8 thru 3.12, it is evident that all of the prediction-set samples in the validation sets are correctly classified and that all of the plots (principal components and canonical variates) are comparable. Table 3.1 which shows the results of LDA, RDA, and the 1-NN for the features selected by the pattern recognition GA using the three fitness functions are comparable to the results obtained from the principal component and canonical variate analysis plots for the same data.

**Table 3.1 Discriminant Analysis Results for 80%/20% Cross Validation Study**

| **Method** | **Average Tset % classification** | | | **Average Pset % classification** | | |
|---|---|---|---|---|---|---|
| | **Normal** | **Modified Hopkins** | **Hopkins** | **Normal** | **Modified Hopkins** | **Hopkins** |
| **LDA** | 100 | 100 | 100 | 98 | 99 | 100 |
| **RDA(auto)** | 100 | 100 | 100 | 97 | 98 | 99 |
| **1-NN** | 99.25 | 100 | 98.75 | 97 | 98 | 98 |

**TP1- CVA**



**TP1- PCKaNN**

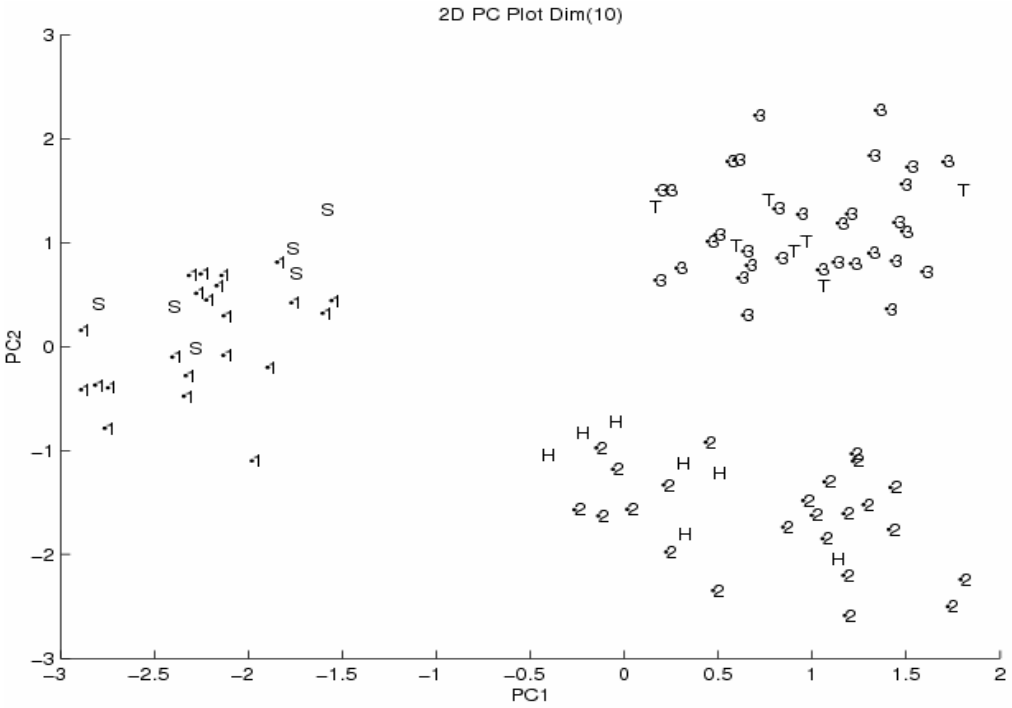
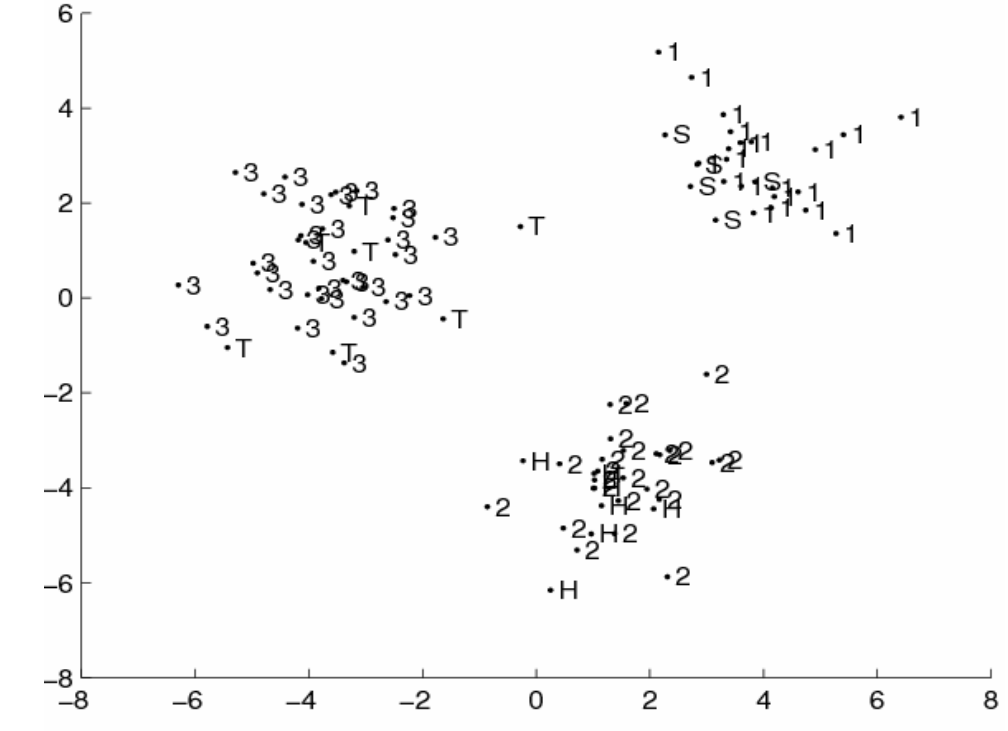



Figure 3.8. Segmented cross validation results for the first training set prediction set pair for CVA and PCA using 10 features identified by the pattern recognition GA. "1" = softwoods, "2" = hardwoods, and "3" = tropical woods for the training set samples. "S" = softwoods, "H" = hardwoods, and "T" = tropical woods for the prediction set samples.

**TP2-CVA**



**TP2- PCKaNN**

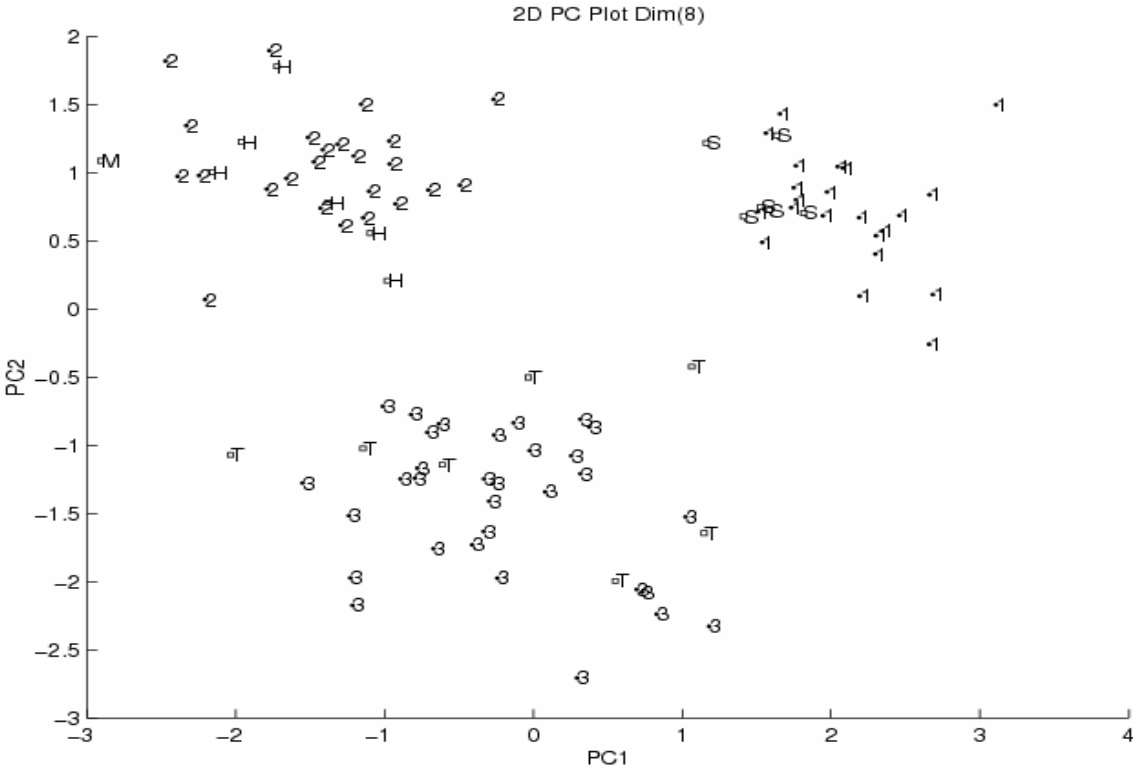
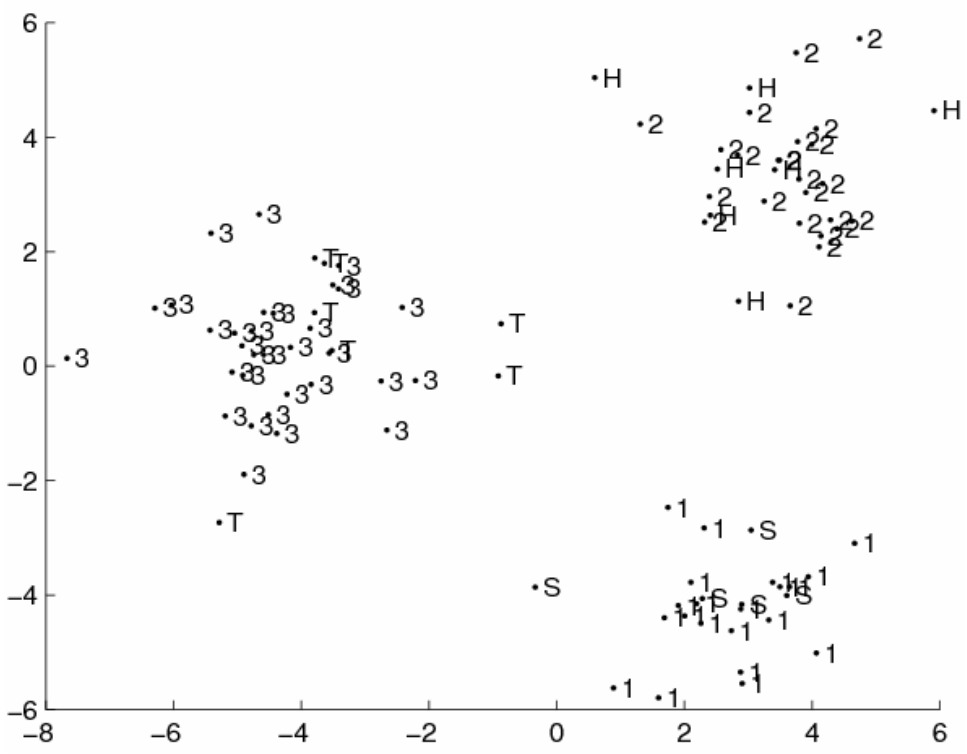



Figure 3.9. Segmented cross validation results for the second training set prediction set pair for CVA and PCA using 8 features identified by the pattern recognition GA. "1" = softwoods, "2" = hardwoods, and "3" = tropical woods for the training set samples. "S" = softwoods, "H" = hardwoods, and "T" = tropical woods for the prediction set samples.

**TP3-CVA**



**TP3- PCKaNN**

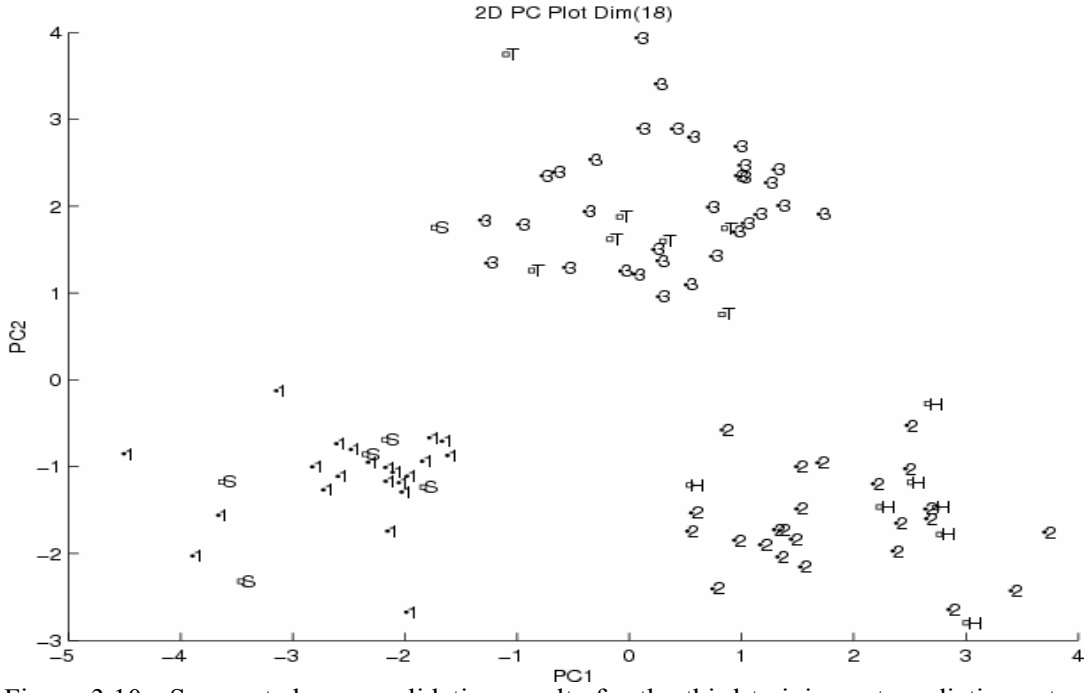
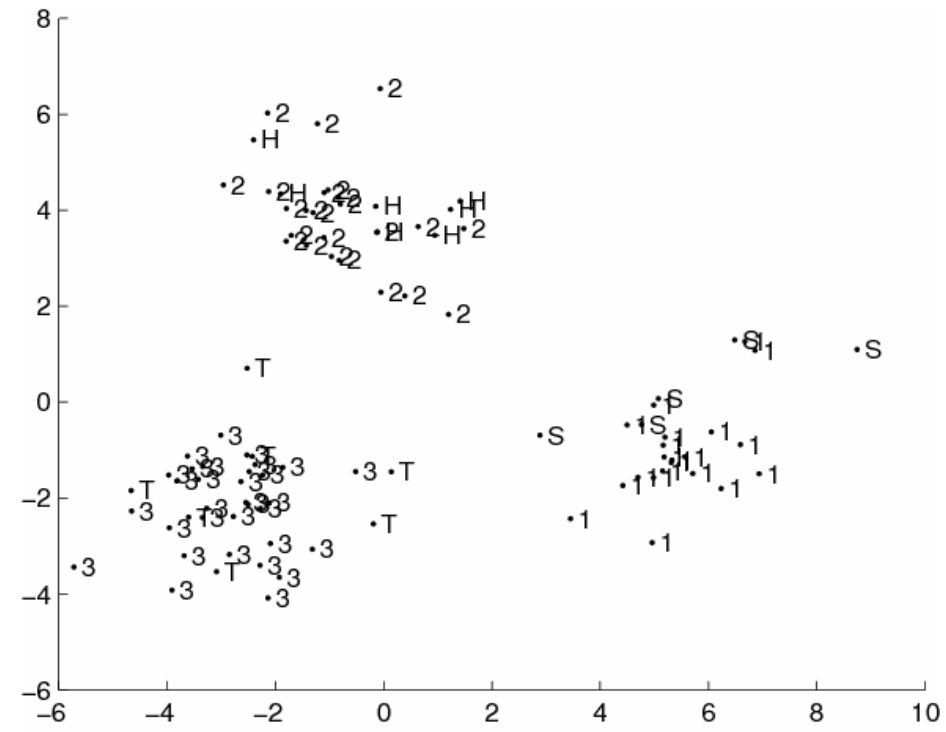



Figure 3.10. Segmented cross validation results for the third training set prediction set pair for CVA and PCA using 18 features identified by the pattern recognition GA. "1" = softwoods, "2" = hardwoods, and "3" = tropical woods for the training set samples. "S" = softwoods, "H" = hardwoods, and "T" = tropical woods for the prediction set samples.

**TP4-CVA**



**TP4- PCKaNN**

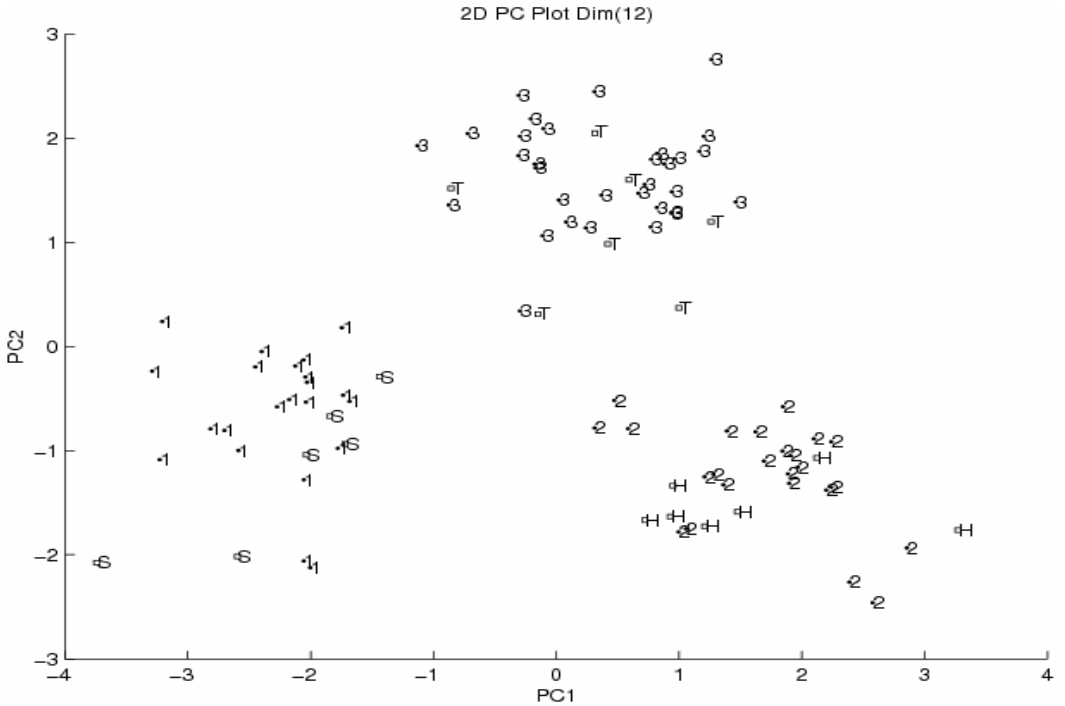
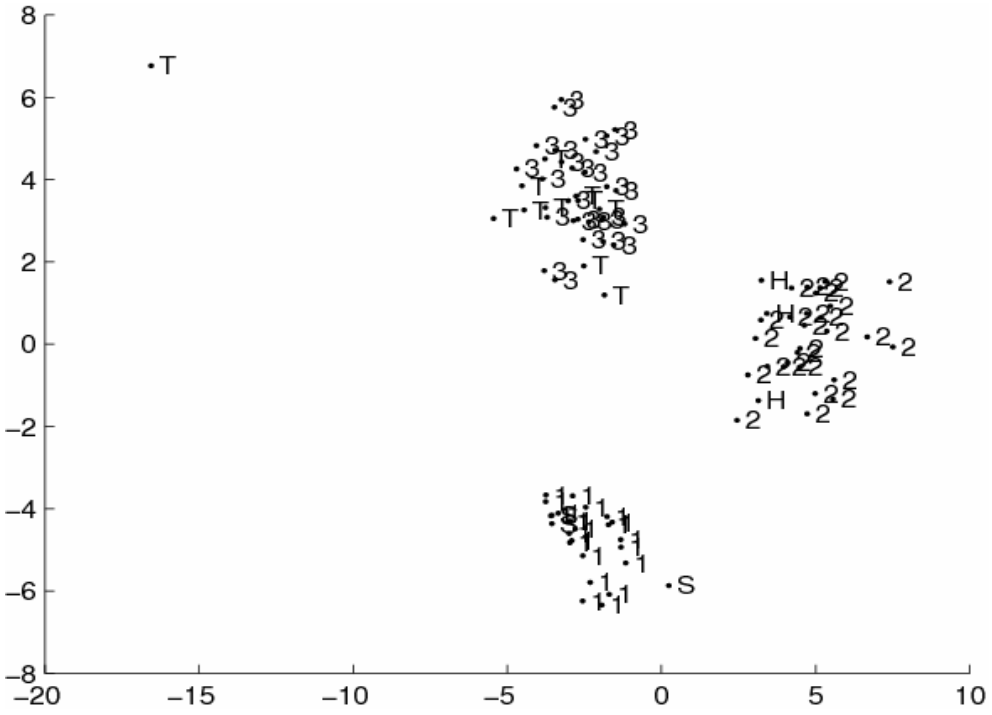



Figure 3.11. Segmented cross validation results for the fourth training set prediction set pair for CVA and PCA using 12 features identified by the pattern recognition GA. "1" = softwoods, "2" = hardwoods, and "3" = tropical woods for the training set samples. "S" = softwoods, "H" = hardwoods, and "T" = tropical woods for the prediction set samples.

**TP5-CVA**




**TP5- PCKaNN**

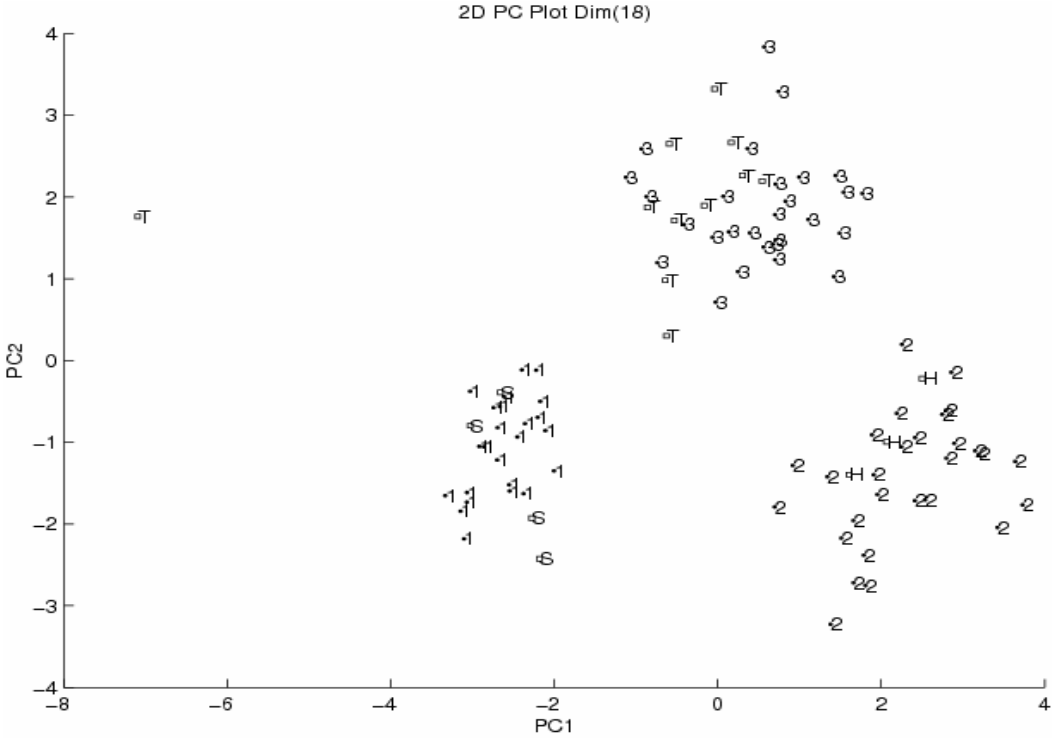
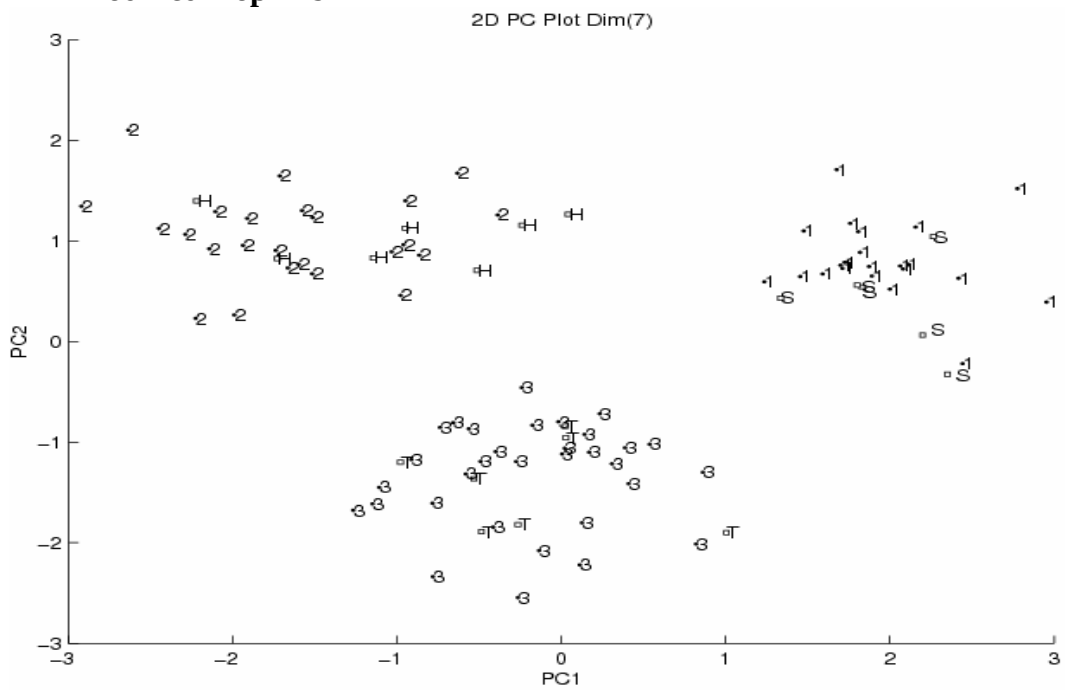



Figure 3.12. Segmented cross validation results for the fifth training set prediction set pair for CVA and PCA using 18 features identified by the pattern recognition GA. "1" = softwoods, "2" = hardwoods, and "3" = tropical woods for the training set samples. "S" = softwoods, "H" = hardwoods, and "T" = tropical woods for the prediction set samples.

**TP1-Modified Hopkins**




**TP1- Hopkins**

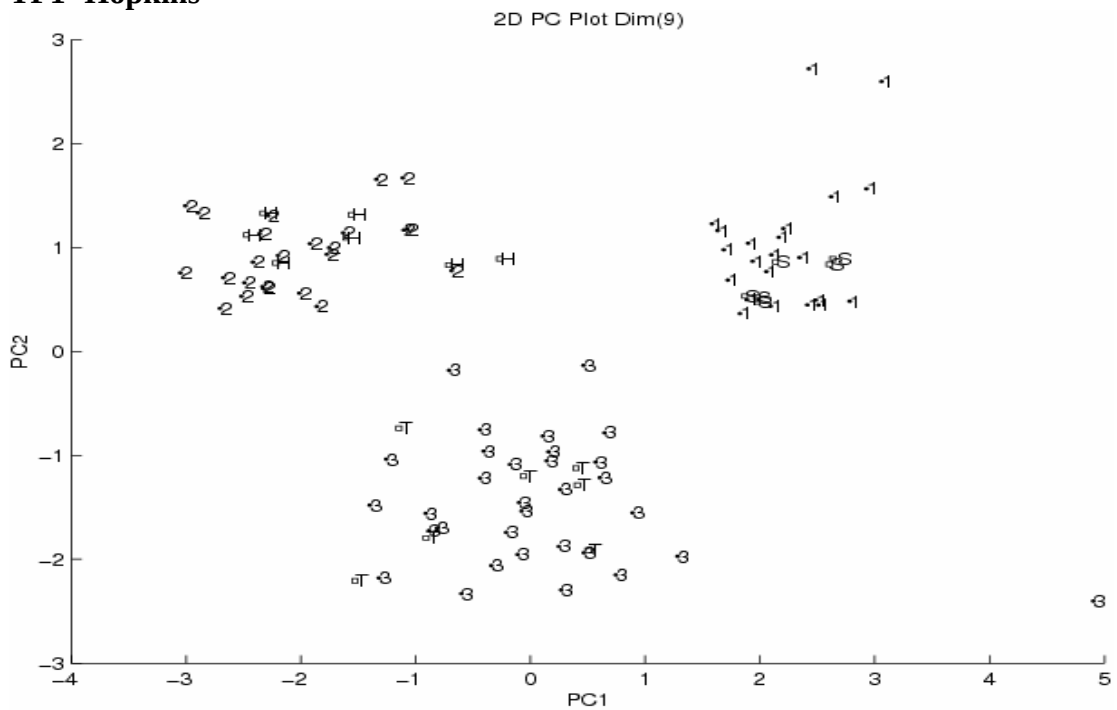
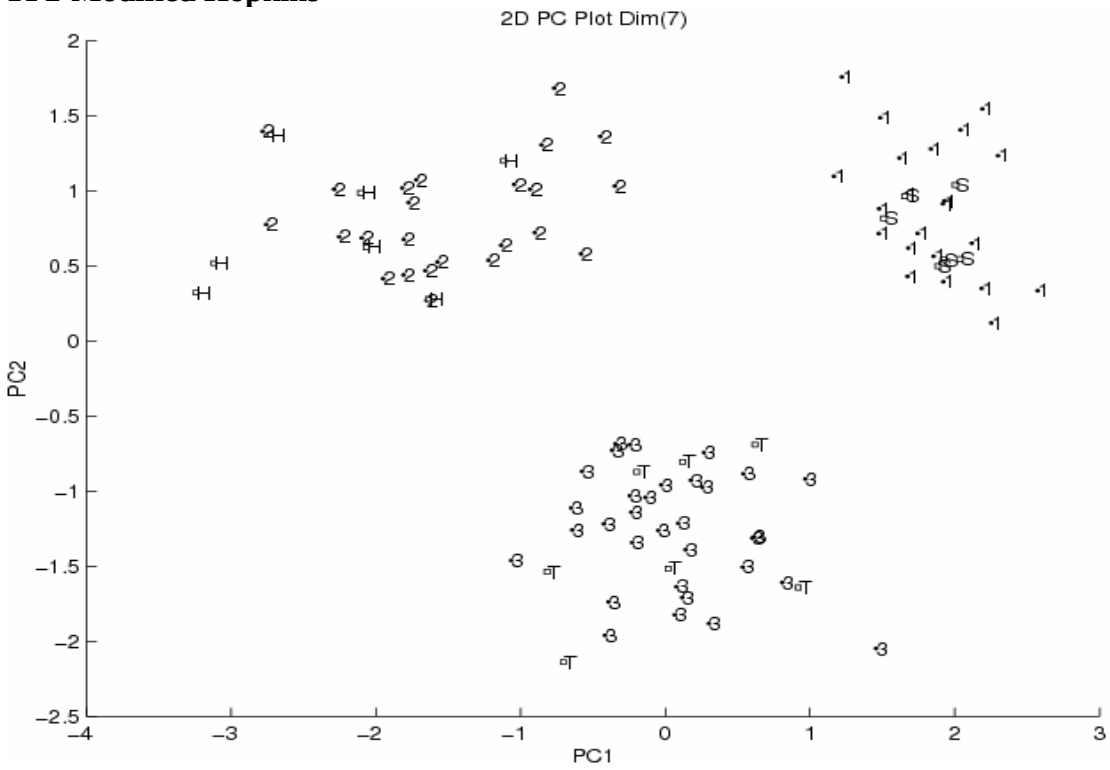



Figure 3.13. Segmented cross validation results for the first training set prediction set pair for features identified by PCKaNN with the modified Hopkins statistic as the fitness function and for features identified by PCKaNN with the Hopkins statistic as the fitness function. "1" = softwoods, "2" = hardwoods, and "3" = tropical woods for the training set samples. "S" = softwoods, "H" = hardwoods, and "T" = tropical woods for the prediction set samples.

**TP2-Modified Hopkins**




**TP2- HOPKINS**

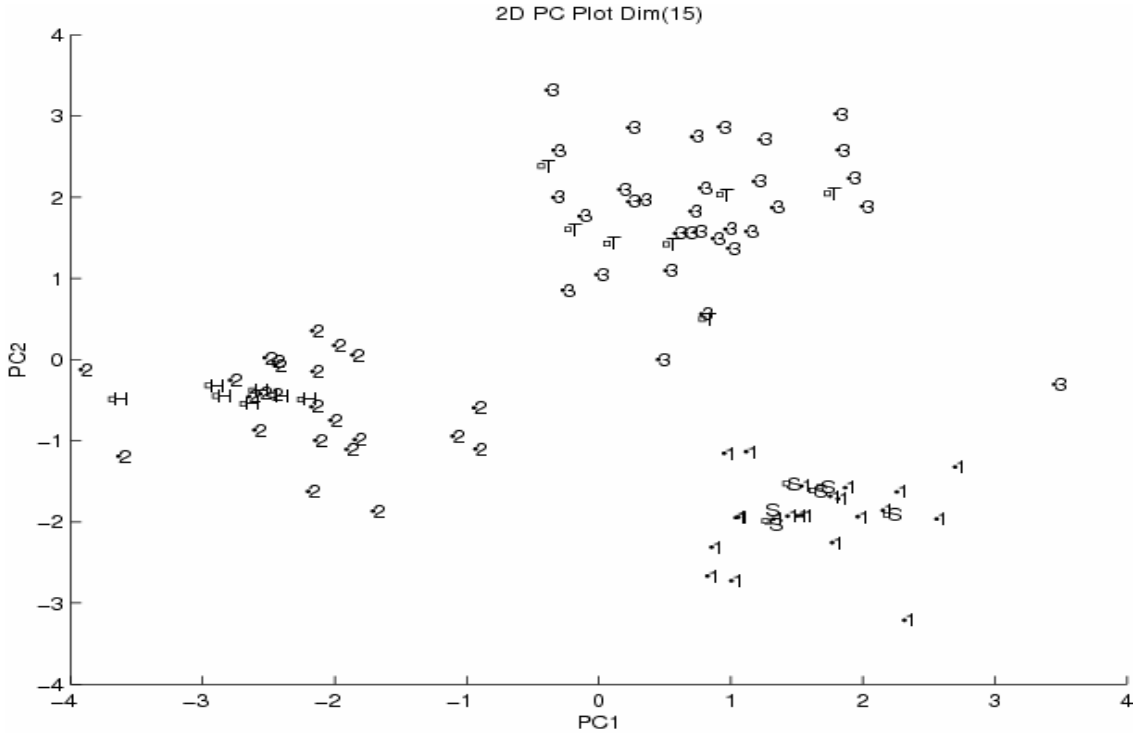
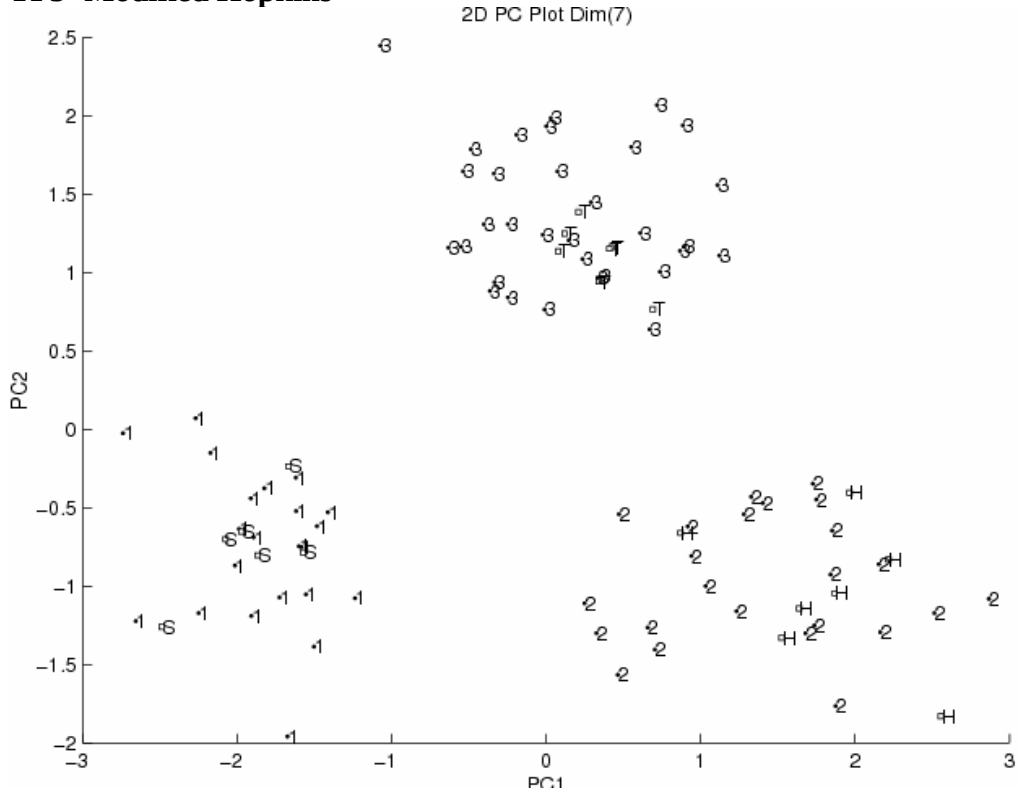



Figure 3.14. Segmented cross validation results for the second training set prediction set pair for features identified by PCKaNN with the modified Hopkins statistic as the fitness function and for features identified by PCKaNN with the Hopkins statistic as the fitness function. "1" = softwoods, "2" = hardwoods, and "3" = tropical woods for the training set samples. "S" = softwoods, "H" = hardwoods, and "T" = tropical woods for the prediction set samples.

**TP3- Modified Hopkins**




**TP3- HOPKINS**

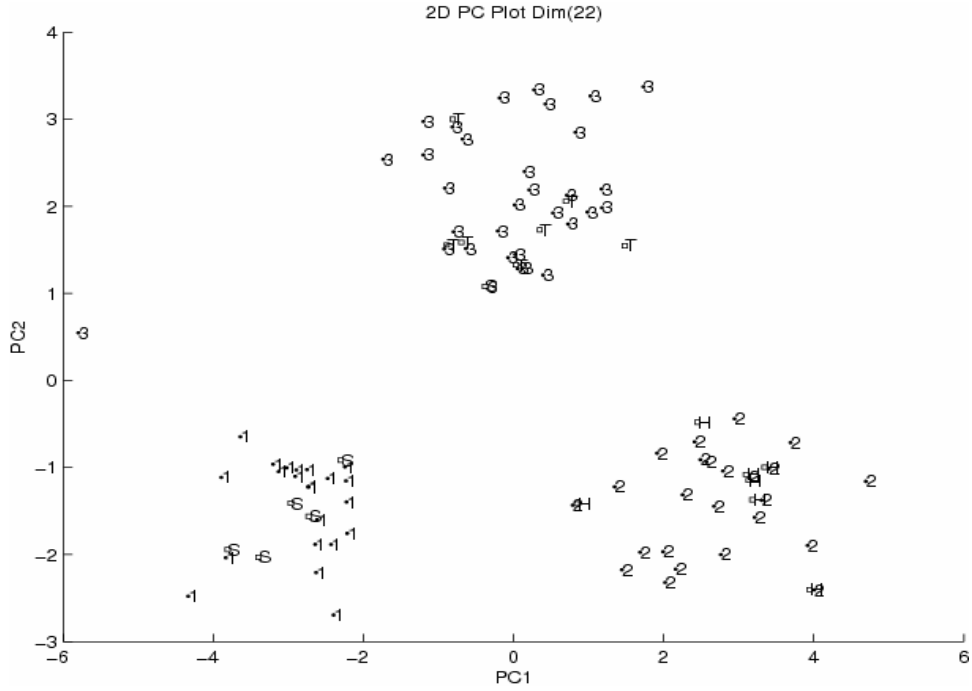
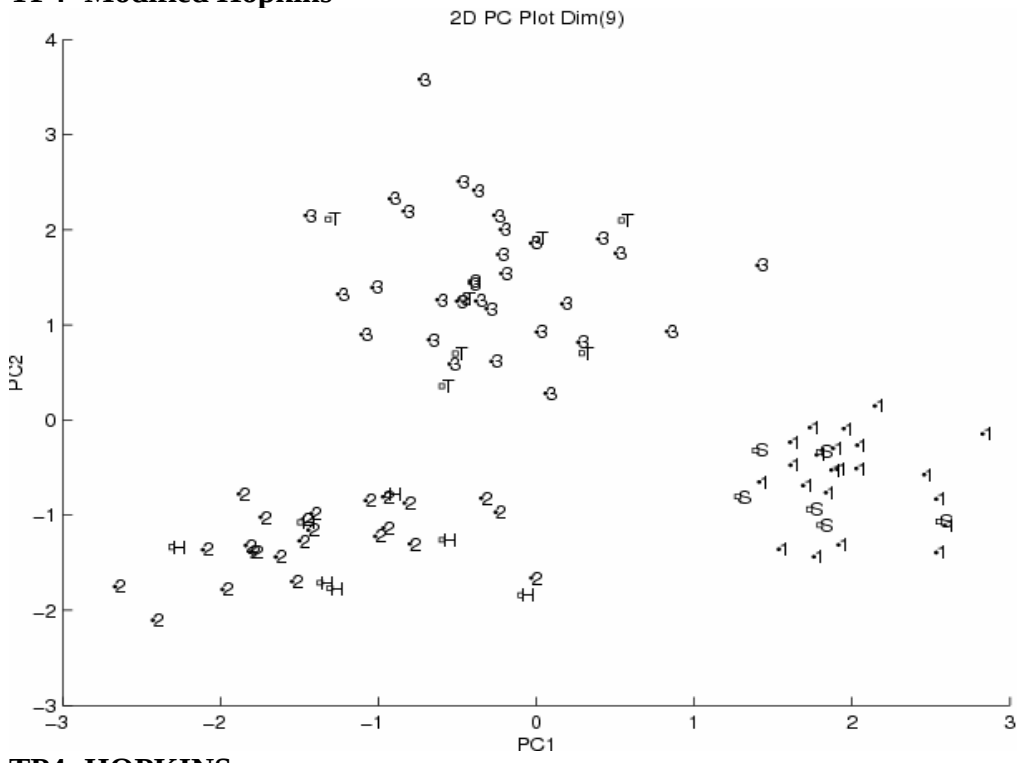



Figure 3.15. Segmented cross validation results for the third training set prediction set pair for features identified by PCKaNN with the modified Hopkins statistic as the fitness function and for features identified by PCKaNN with the Hopkins statistic as the fitness function. "1" = softwoods, "2" = hardwoods, and "3" = tropical woods for the training set samples. "S" = softwoods, "H" = hardwoods, and "T" = tropical woods for the prediction set samples.

**TP4- Modified Hopkins**




**TP4- HOPKINS**

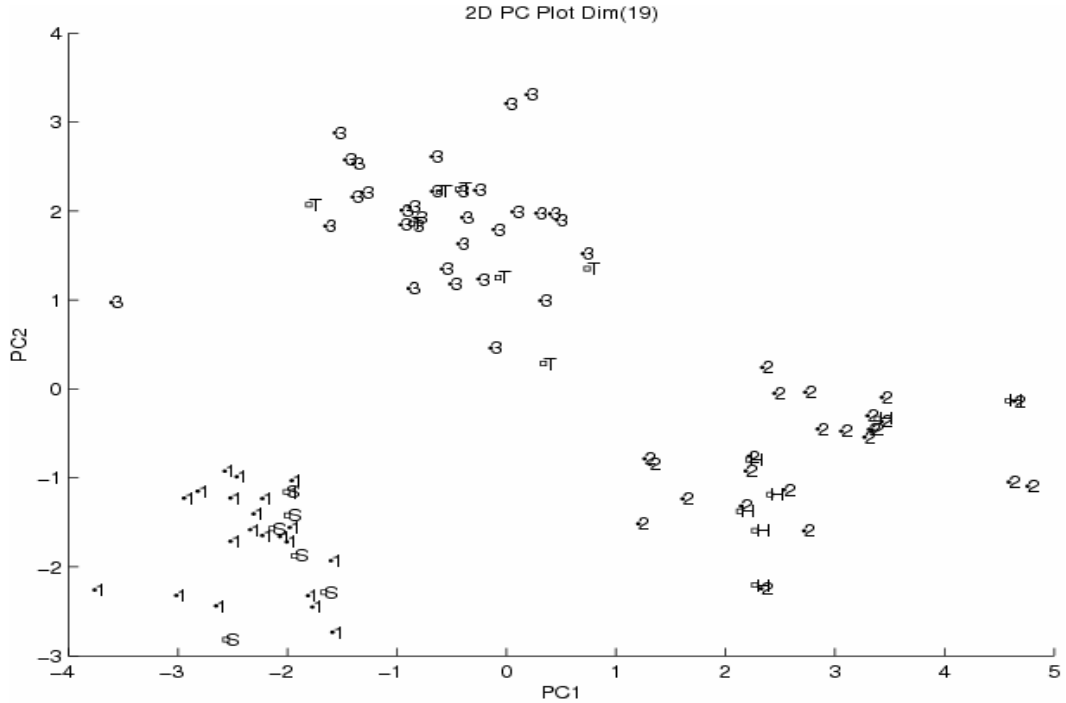
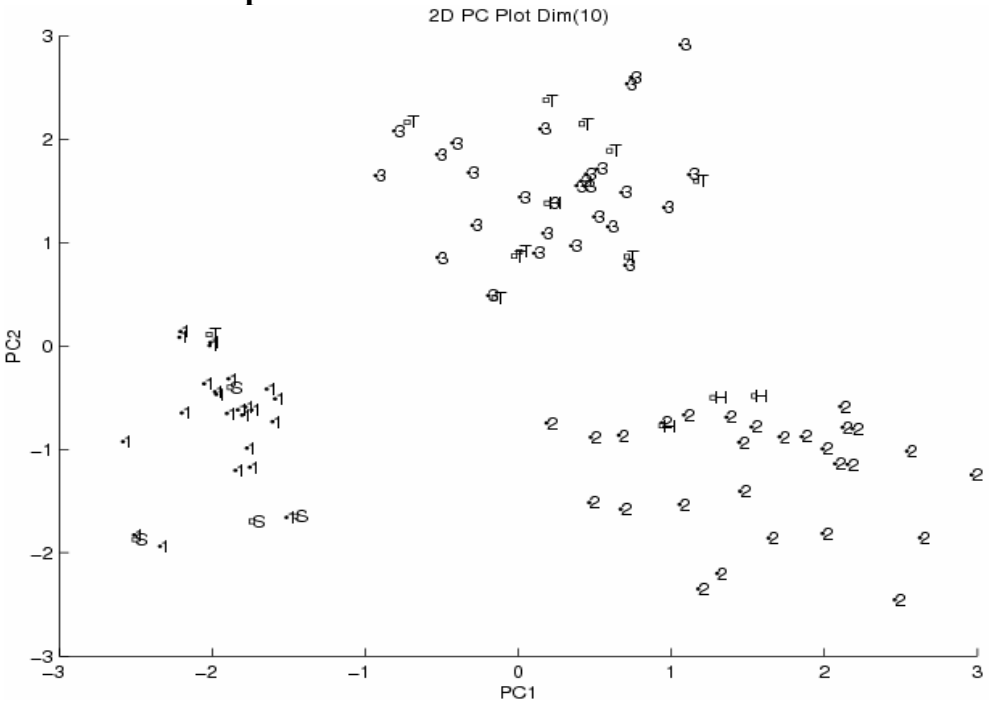



Figure 3.16. Segmented cross validation results for the fourth training set prediction set pair for features identified by PCKaNN with the modified Hopkins as the fitness function and for features identified by PCKaNN with the Hopkins statistic as the fitness function. "1" = softwoods, "2" = hardwoods, and "3" = tropical woods for the training set samples. "S" = softwoods, "H" = hardwoods, and "T" = tropical woods for the prediction set samples.

**TP5-Modified Hopkins**




**TP5- HOPKINS**

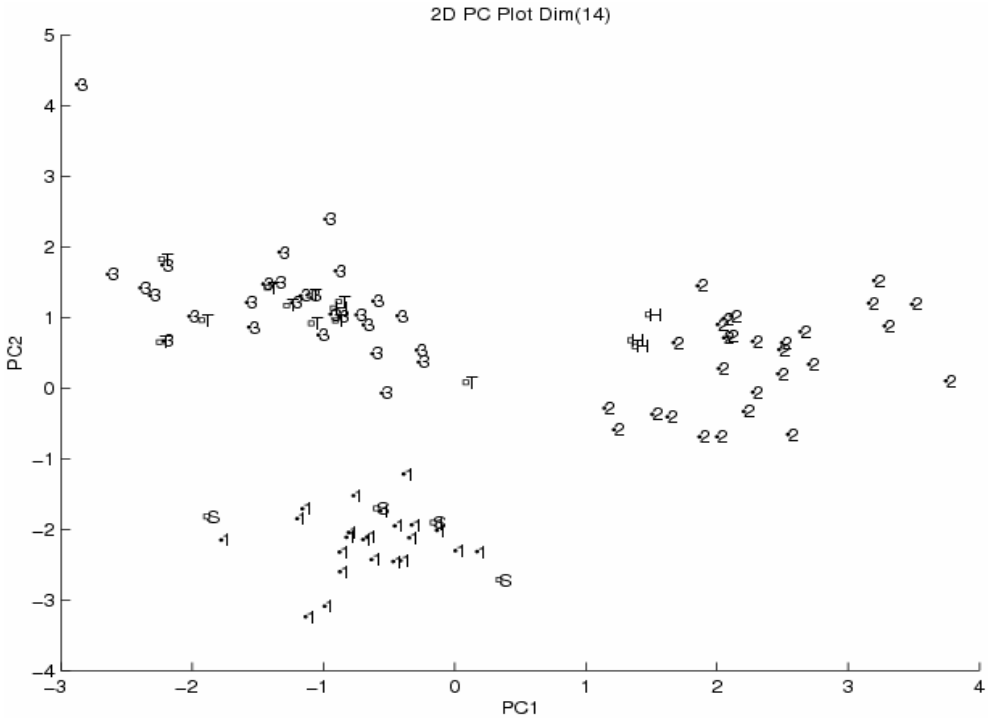



Figure 3.17. Segmented cross validation results for the fifth training set prediction set pair for features identified by PCKaNN with the modified Hopkins statistic as the fitness function and for features identified by PCKaNN with the Hopkins statistic as the fitness function "1" = softwoods, "2" = hardwoods, and "3" = tropical woods for the training set samples. "S" = softwoods, "H" = hardwoods, and "T" = tropical woods for the prediction set samples.

Table 3.2 summarizes the results from the 20%/80% segmented cross validation study (5 training set prediction set pairs with 20% of the samples in each training set and the remaining 80% in each prediction set) using LDA, RDA, and 1-NN for the features selected by the pattern recognition GA with transverse learning and without transverse learning. (The results from the 20%/80% validation study are not displayed using principal component plots because PCA does not scale up well when the number of prediction set samples is comparable or greater than the number of training set samples.) From this table, it is evident that classifiers developed from features selected by the pattern recognition GA using transverse learning performed better than classifiers developed from features selected by the pattern recognition GA using only PCKaNN. For training sets with small amounts of data with class labels and large amounts of unlabeled data, transverse learning usually performs better since information in the unlabeled data is used by the fitness function to guide feature selection. Features are selected to optimize clustering and to maximize the distance between the different classes in the data set. This will ensure that features selected by the pattern recognition GA using transverse learning will perform better than a learning model developed from a set of features whose selection is based solely on the dichotomization power of the features for the labeled data points. The superior performance of classifiers developed from features selected by the pattern recognition GA using the modified Hopkins statistic (as compared to classifiers developed from features selected by the pattern recognition GA using PCKaNN with only the Hopkins statistic) can be attributed to the modified Hopkins statistic taking advantage of the power associated with boosting. The results of these two

segmented cross validation studies demonstrate the efficacy and flexibility of the pattern recognition GA to identify linear structure in undetermined multivariate data sets.

**Table 3.2 Discriminant Analysis Results for 20%/80% Cross Validation Study**

| Method | Average Tset % classification | | | Average Pset % classification | | |
|---|---|---|---|---|---|---|
| | Normal | Modified Hopkins | Hopkins | Normal | Modified Hopkins | Hopkins |
| **LDA** | 100 | 100 | 100 | 72 | 85.5 | 75 |
| **RDA(auto)** | 100 | 100 | 100 | 68.22 | 79.25 | 70 |
| **1-NN** | 100 | 100 | 98 | 78.5 | 88 | 85.5 |

***Treatment of Prostrate Cancer***

Prostrate cancer is a leading cause of death in men. Although only 1 in 10,000 men under the age of 40 are diagnosed with prostrate cancer, the rate is 1 in 39 for ages 40 to 59 and 1 in 14 for ages 60 to 69. Treatment options currently available for men with prostrate cancer include surgery, radiation therapy, hormone therapy, and chemotherapy. In this study, a potential method has been developed to determine if surgery is a viable treatment option for prostrate cancer patients based on pattern recognition analysis of DNA microarray data obtained from tumor biopsy material. The tumor biopsy material was obtained from 100 patients: 49 patients who exhibited no reoccurrence of prostrate cancer within 60 months after surgery, and 51 patients who had died from cancer within a 5 year time period after surgery. Each biopsy specimen was represented by 44,928 gene expressions, the concentration levels of all mRNA sequences in a cell or tissue at any given time. However, the expression levels of some genes in the tumor biopsy material recovered from the patients used in this study were invariant.

After removal of these genes, the number of features per sample was reduced to 30,167. The gene expression data in this study was obtained from Dr. William J. Catalona's research group at Washington University in St. Louis.

The first step in the study was to apply PCA to the autoscaled data. Figure 3.18 shows a principal component plot developed from the 100 data set samples and 30,167 genes. The 1's are no reoccurrence and the 2's are reoccurrence of cancer. Clustering of the samples (patients) by class label is not evident in the principal component plot. The pattern recognition GA (PCKaNN fitness function) was used to find a set of descriptors from which a discriminating relationship could be found. Because of the large number of features present in this data, it was decided to develop a seed to facilitate the searching of the solution space by the pattern recognition GA. (Hardware limitations on the number of chromosomes that could be used to search the solution space was another reason to use a seed.) The following experimental protocol was employed to develop the seed. First, the initial population of chromosomes, which was randomly generated, was fixed at 5000. This ensured that each feature was present in at least one of the chromosomes in the initial population. Next, ten runs were performed, with each run proceeding for 100 generations. The seed was then formed from the top 10% of the final population in each run which was mixed with a randomly generated population whose number of chromosomes was comparable to the number of chromosomes in the top 10% of the final population selected from each run. Figure 3.19 shows a principal component plot developed from 41 features identified by the pattern recognition GA using PCKaNN as the fitness function and the seed as the initial population for this run. The mutation rate

of the GA was set at 0.3, and culling (every six generations with a threshold of 0.0875) was employed.

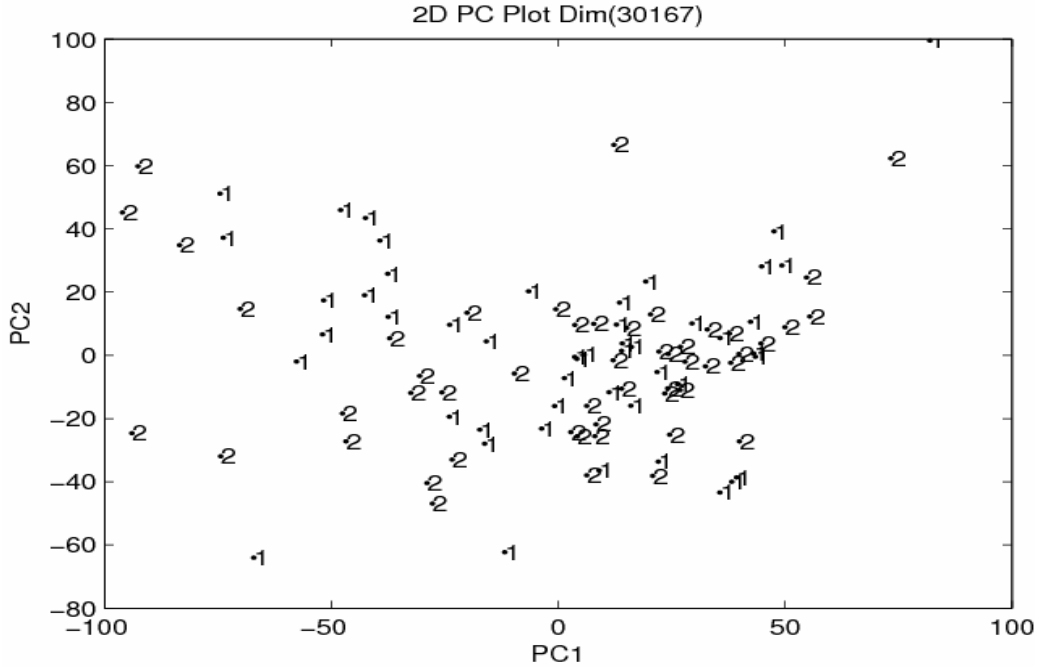



Figure 3.18. A plot of the two largest principal components developed from the 100 samples and 30,167 features. 1 = no reoccurrence and 2 = reoccurrence.

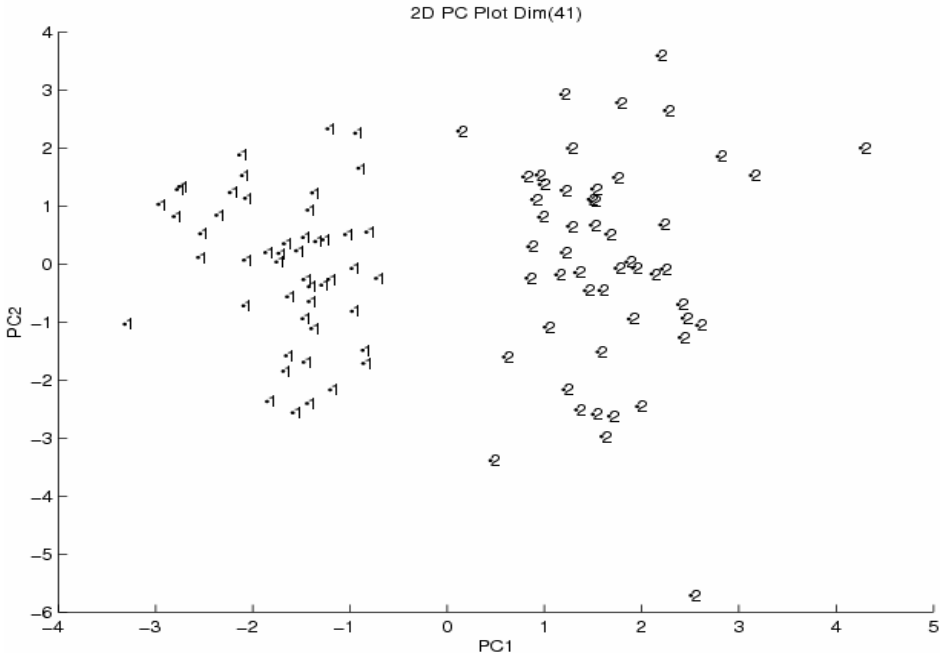
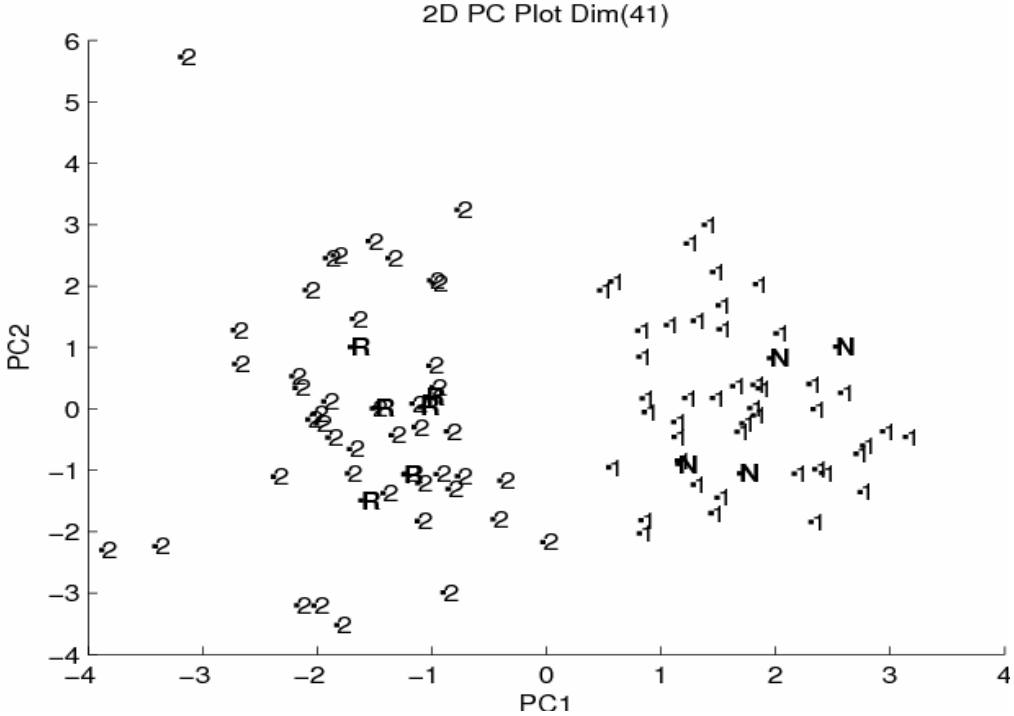



Figure 3.19. A plot of the two largest principal components developed from the 100 samples and 41 features identified by PCKaNN. 1 = no reoccurrence and 2 = reoccurrence.

The ability of a classifier to predict the class membership (reoccurrence or no reoccurrence of cancer after surgery) of a simulated unknown biopsy specimen was tested using segmented cross validation. The data set was divided into 10 training set prediction set pairs. A classifier is developed for each training set and then tested on the corresponding prediction set. Each sample was present in only one of the 10 prediction sets generated. Figures 3.20 thru 3.34 summarize the results of the segmented cross validation for the pattern recognition GA using PCKaNN with transverse learning and without transverse learning. Each object in the training set is represented as "1" (no reoccurrence), and "2" (reoccurrence). The prediction set samples are represented as "N" (no reoccurrence), and "R" (reoccurrence). From an examination of the PCA plots, it is evident that 99 of the 100 validation set samples are correctly classified regardless of the fitness function used. Previous work in our laboratory [60] has shown that selecting features for classification of microarray data using transductive inference (which is also known as transverse learning) yields smaller prediction errors then selecting features for classification based on inductive inference (i.e., PCKaNN) where the goal is to construct a good classifier, which is applied to any future data. We attribute this discrepancy to the use of the seed. Furthermore, we do not believe that our results are biased because the principal component plots generated for the best feature subsets identified by the pattern recognition GA for the runs used to produce the seed did not show separation correlated to the success of the treatment.

**TP1-Normal**




**TP1- Hopkins**

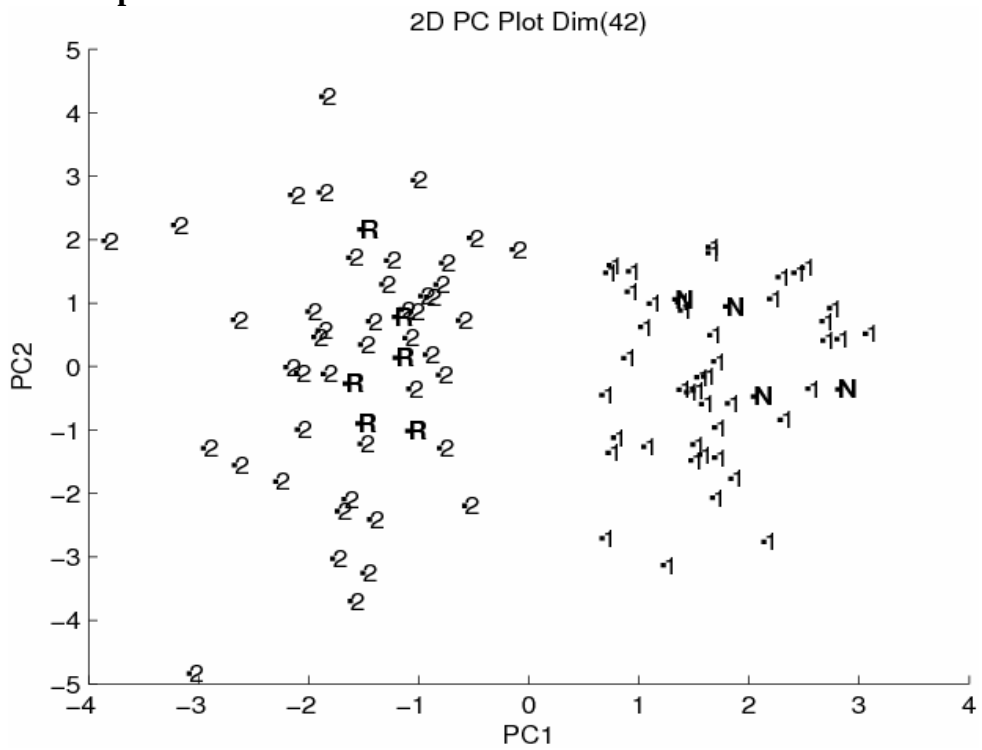
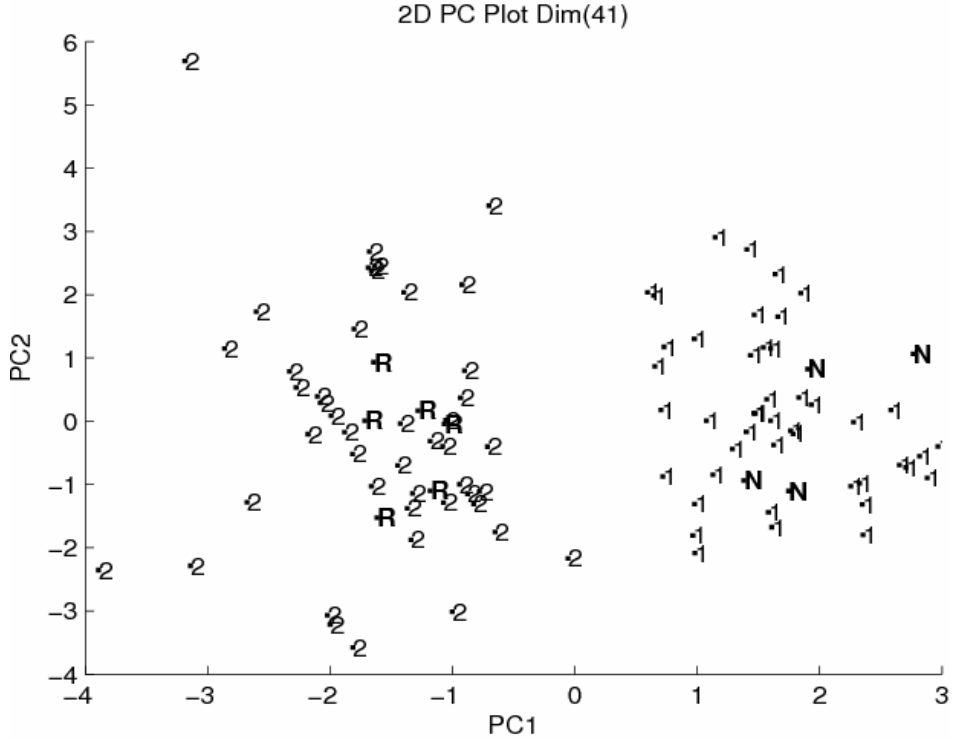



Figure 3.20. A plot of the two largest principal components developed from 90 samples and 41 features identified by the pattern recognition GA using PCKaNN as the fitness function for training set 1, and 90 samples and 42 features identified by the pattern recognition GA using PCKaNN and the Hopkins statistic as the fitness function for training set 1. 1 = no reoccurrence and 2 = reoccurrence (training set). R = reoccurrence and N = no reoccurrence (validation set)

**TP1- Modified Hopkins**




**TP2-Normal**

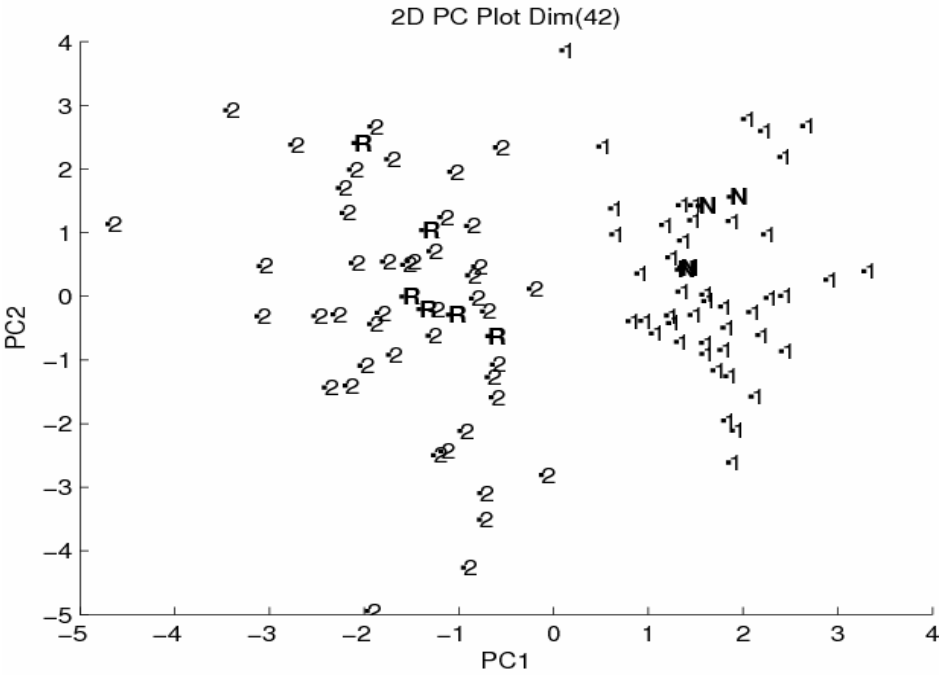
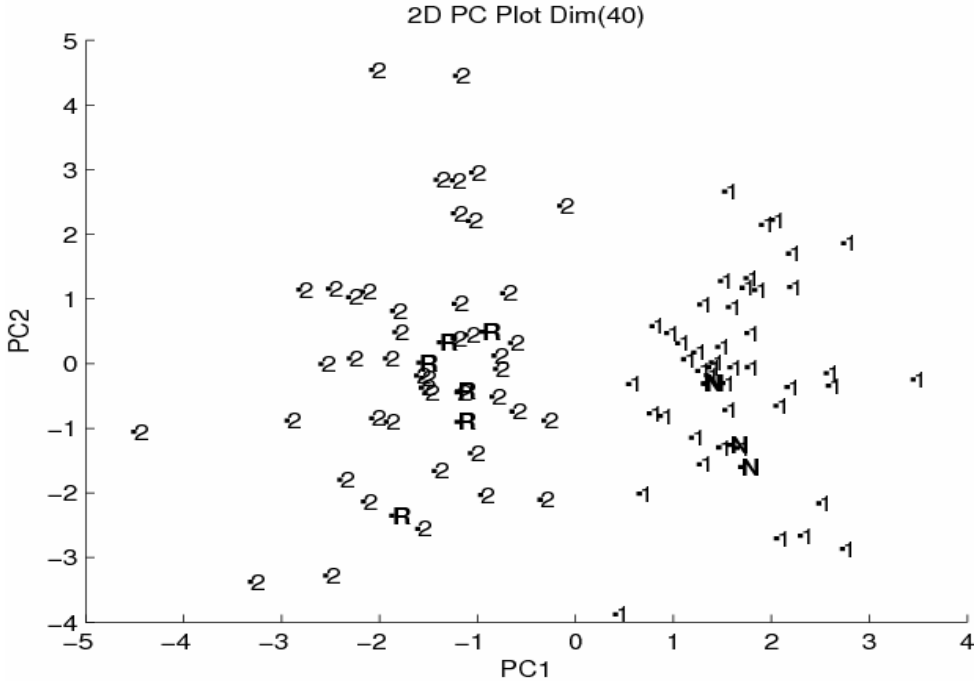



Figure 3.21. A plot of the two largest principal components developed from 90 samples and 41 features identified by the pattern recognition GA using PCKaNN with the modified Hopkins statistic as the fitness function for training set 1 and 90 samples and 42 features identified by the pattern recognition GA using PCKaNN as the fitness function for training set 2. 1 = no reoccurrence and 2 = reoccurrence (training set). R = reoccurrence and N = no reoccurrence (validation set)

**TP2- Hopkins**




**TP2- Modified Hopkins**

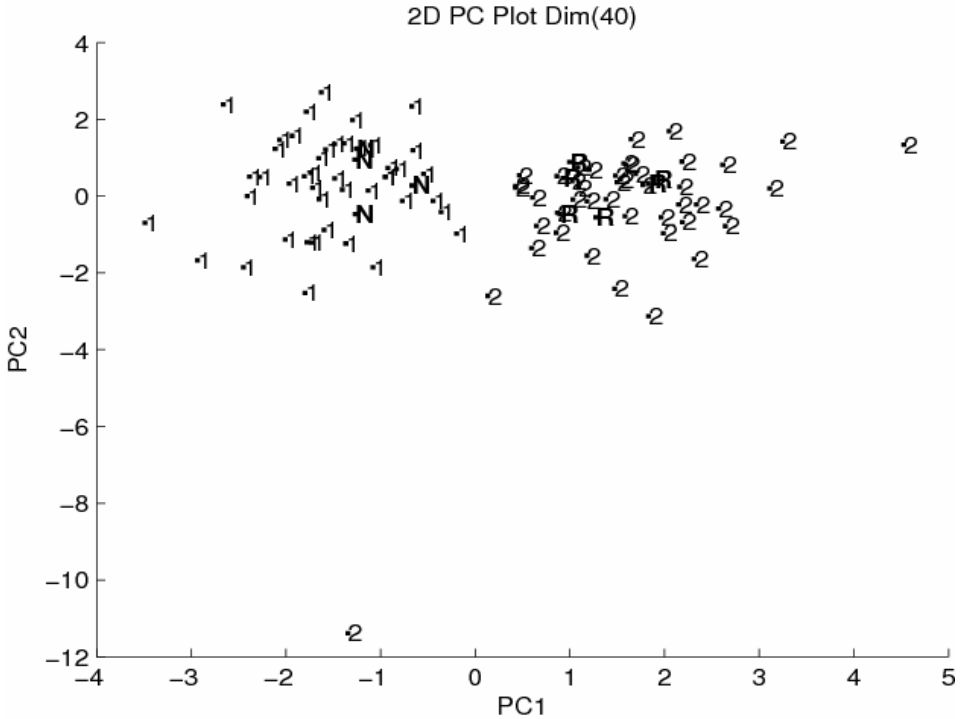
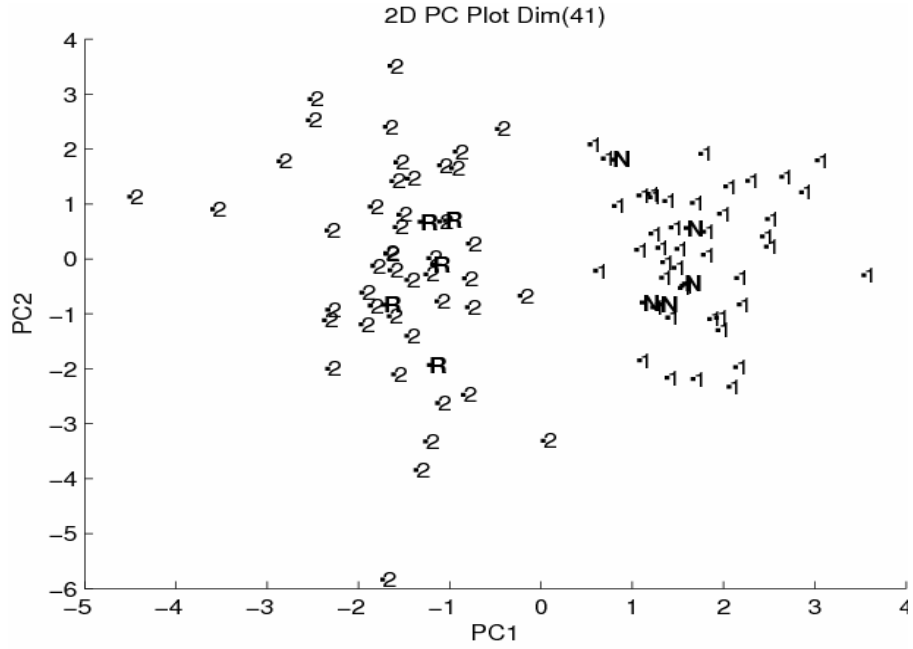



Figure 3.22. A plot of the two largest principal components developed from 90 samples and 40 features identified by the pattern recognition GA using PCKaNN with the Hopkins statistic as the fitness function for training set 2 and 40 features identified by the pattern recognition GA using PCKaNN with the modified Hopkins statistic as the fitness function for training set 2.. 1 = no reoccurrence and 2 = reoccurrence (training set). R = reoccurrence and N = no reoccurrence (validation set)

**TP3- Normal**




**TP3- Hopkins**

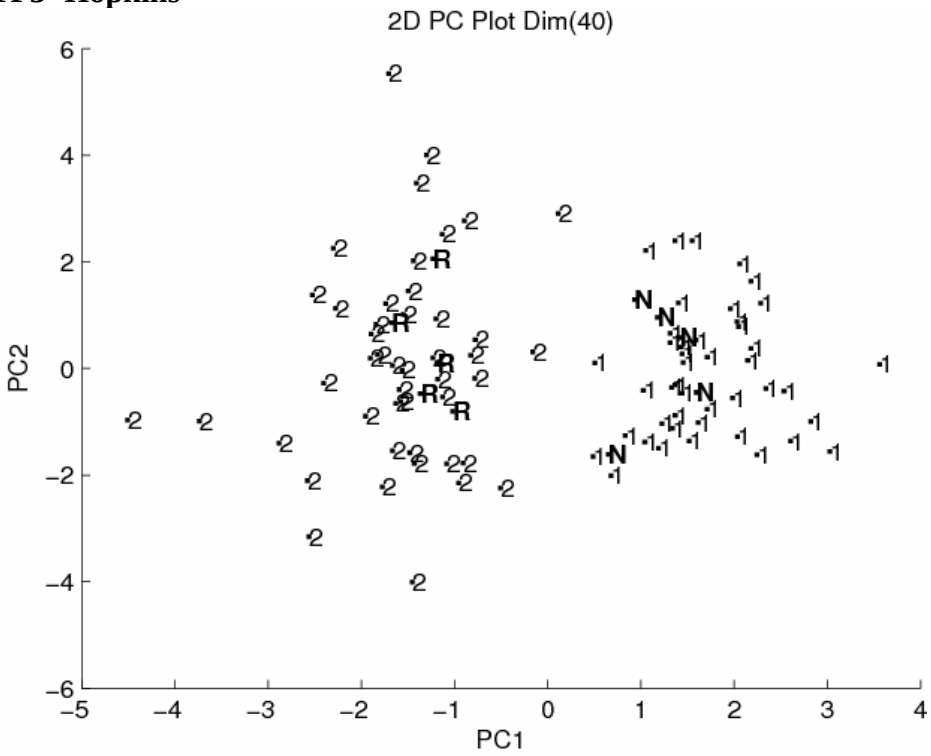
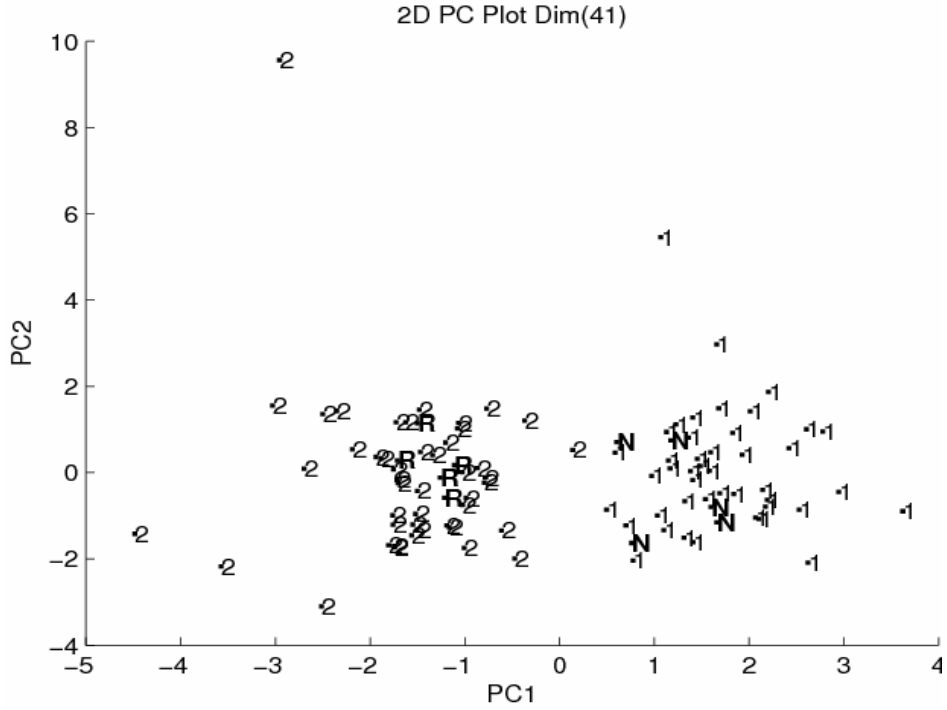



Figure 3.23. A plot of the two largest principal components developed from 90 samples and 41 features identified by the pattern recognition GA using PCKaNN as the fitness function for training set 3 and 90 samples and 40 features identified by the pattern recognition GA using PCKaNN and the Hopkins statistic for training set 3. 1 = no reoccurrence and 2 = reoccurrence (training set). R = reoccurrence and N = no reoccurrence (validation set)

**TP3- Modified Hopkins**




**TP4- Normal**

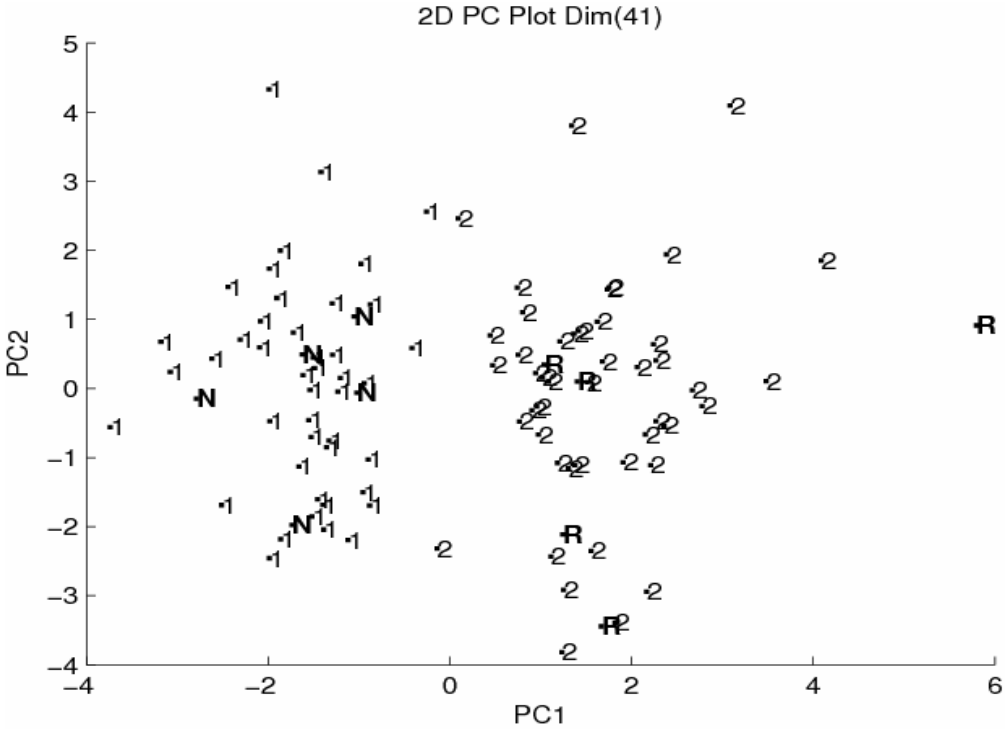
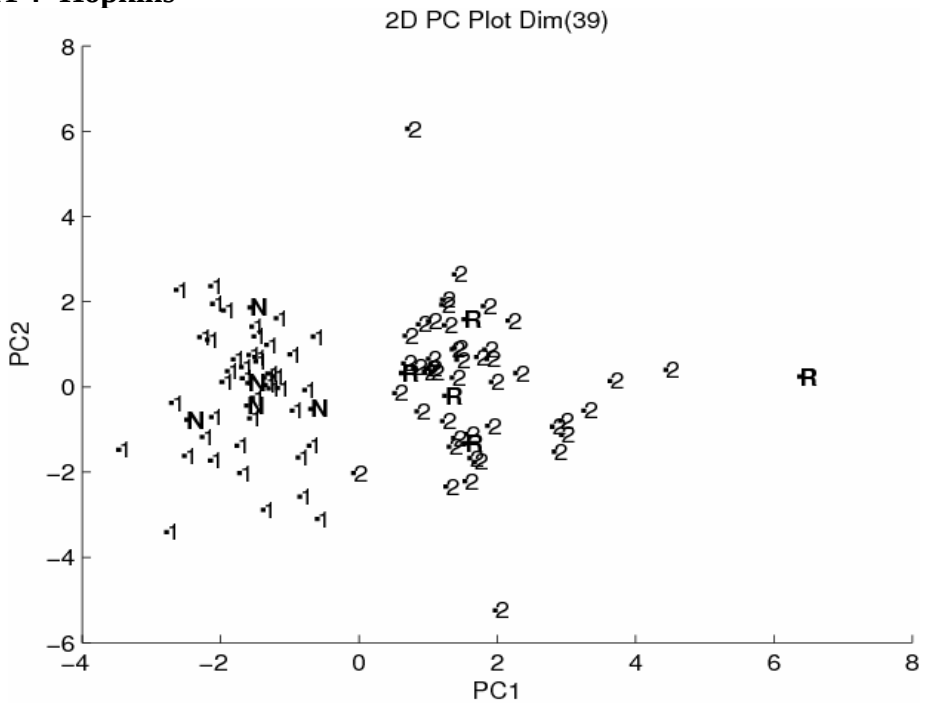



Figure 3.24. A plot of the two largest principal components developed from 90 samples and 41 features identified by the pattern recognition GA using PCKaNN with the modified Hopkins statistic as the fitness function for training set 3 and 90 samples and 41 features identified by the pattern recognition GA using PCKaNN as the fitness function for training set 4. 1 = no reoccurrence and 2 = reoccurrence (training set). R = reoccurrence and N = no reoccurrence (validation set)

**TP4- Hopkins**




**TP4- Modified Hopkins**

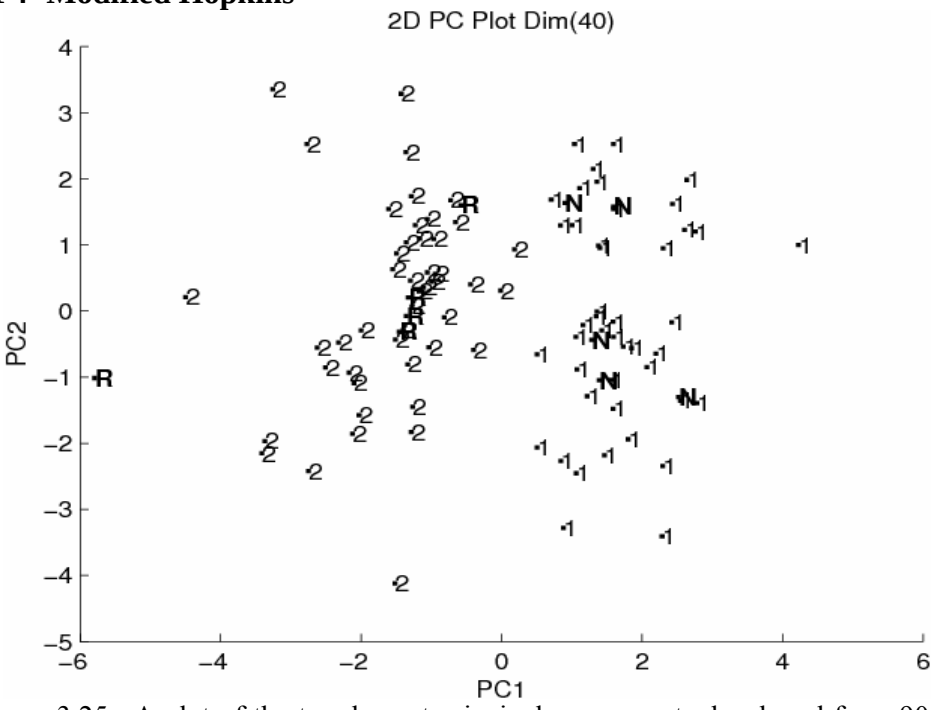
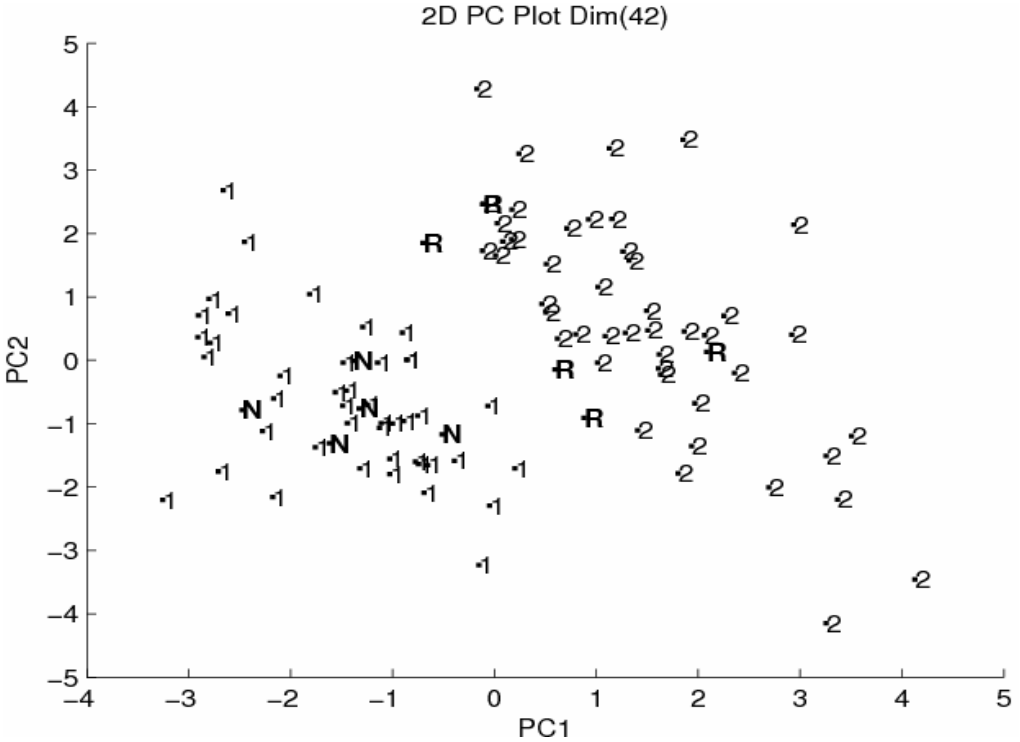



Figure 3.25. A plot of the two largest principal components developed from 90 samples and 39 features identified by the pattern recognition GA using PCKaNN with the Hopkins statistic as the fitness function for training set 4, and 90 samples and 40 features identified by the pattern recognition GA using PCKaNN with the modified Hopkins statistic for training set 4. 1 = no reoccurrence and 2 = reoccurrence (training set). R = reoccurrence and N = no reoccurrence (validation set)

**TP5 – Normal**




**TP5- Hopkins**

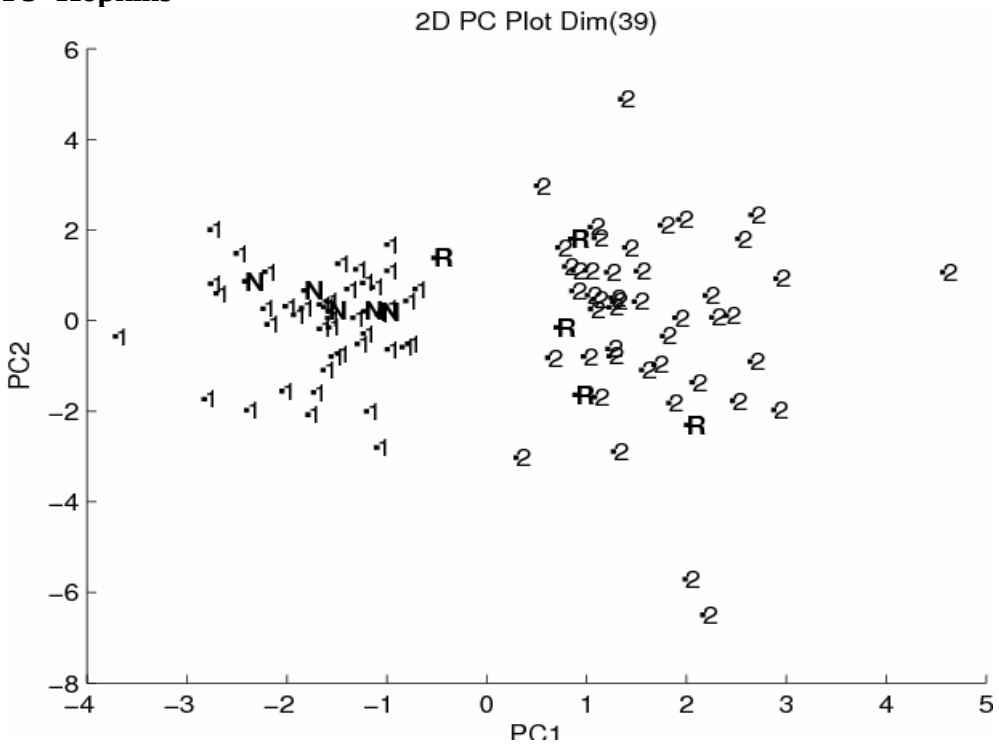
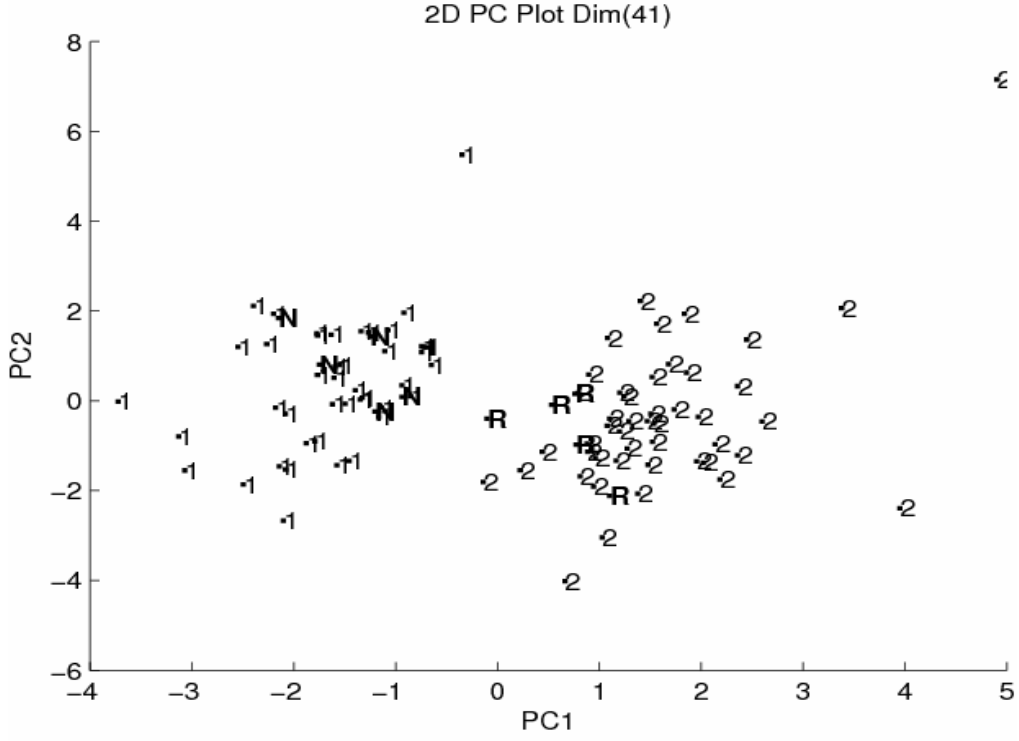



Figure 3.26. A plot of the two largest principal components developed from 90 samples and 42 features identified by the pattern recognition GA using PCKaNN as the fitness function for training set 5 and 90 samples and 39 features identified by the pattern recognition GA using PCKaNN and the Hopkins statistic as the fitness function for training set 5. 1 = no reoccurrence and 2 = reoccurrence (training set). R = reoccurrence and N = no reoccurrence (validation set)

**TP5- Modified Hopkins**




**TP6- Normal**

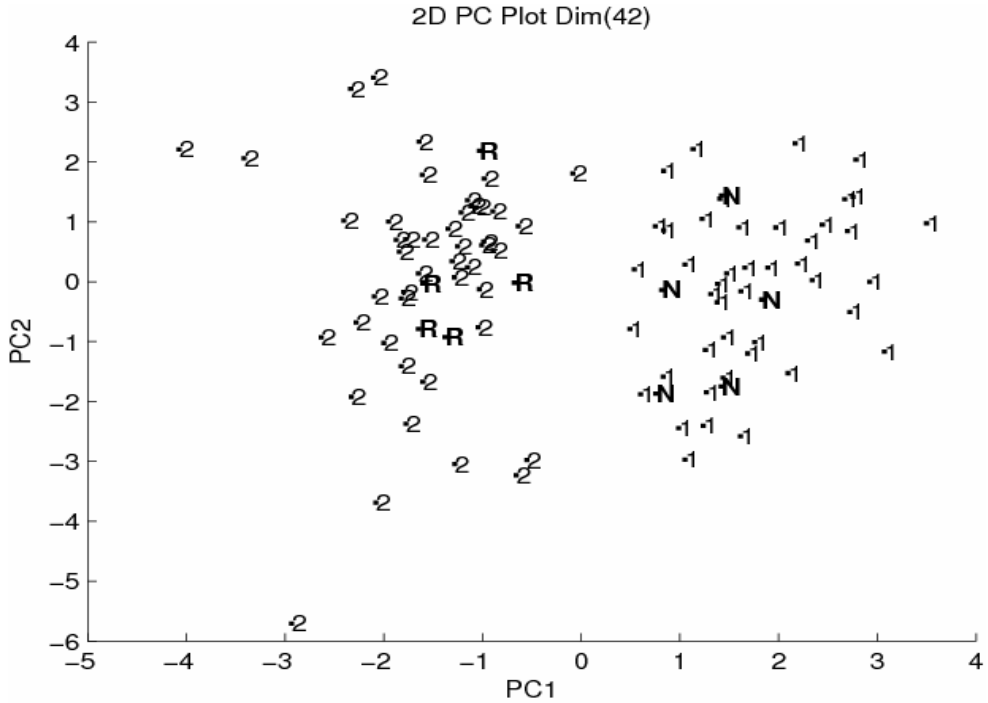
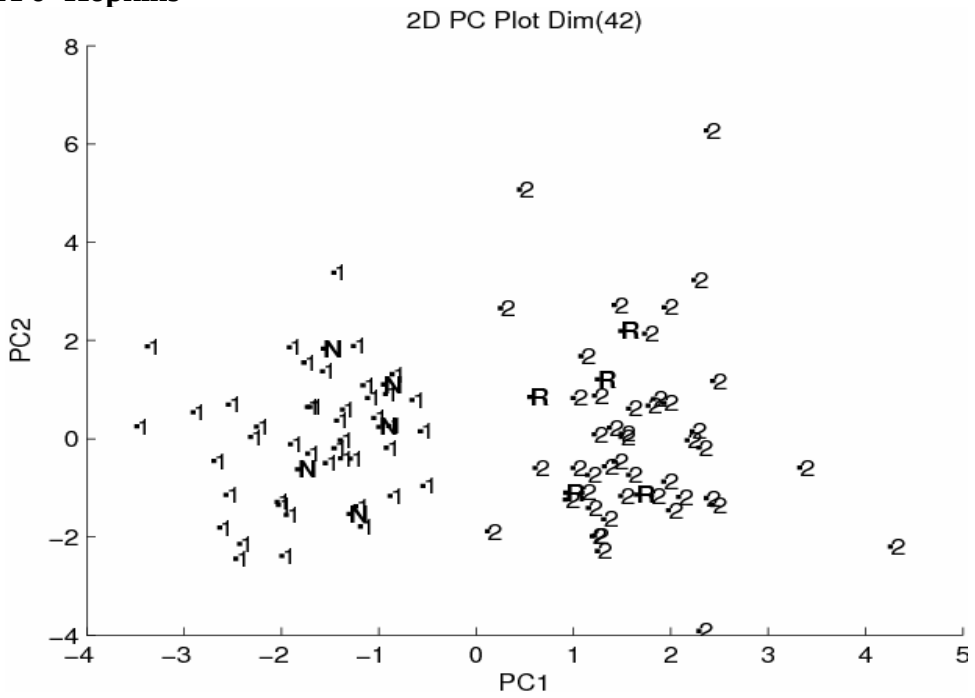



Figure 3.27. A plot of the two largest principal components developed from 90 samples and 41 features identified by the pattern recognition GA using PCKaNN with the modified Hopkins statistic as the fitness function for training set 5 and 90 samples and 42 features identified by the pattern recognition GA using PCKaNN as the fitness function for training set 6. 1 = no reoccurrence and 2 = reoccurrence (training set). R = reoccurrence and N = no reoccurrence (validation set)

**TP6- Hopkins**




**TP6- Modified Hopkins**

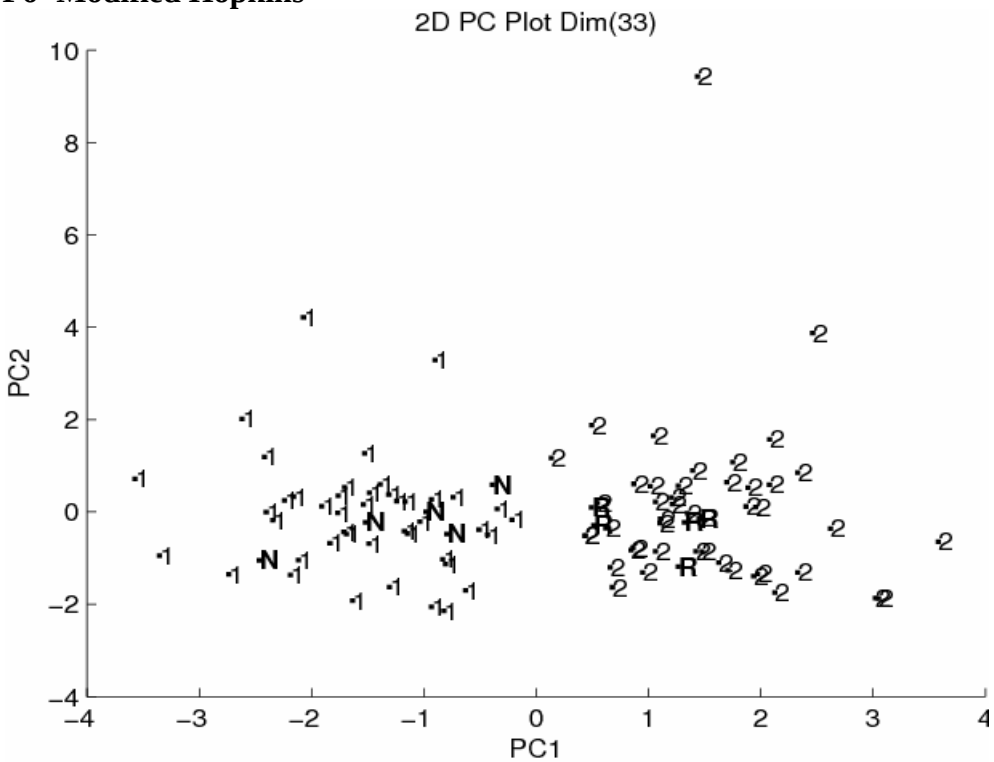
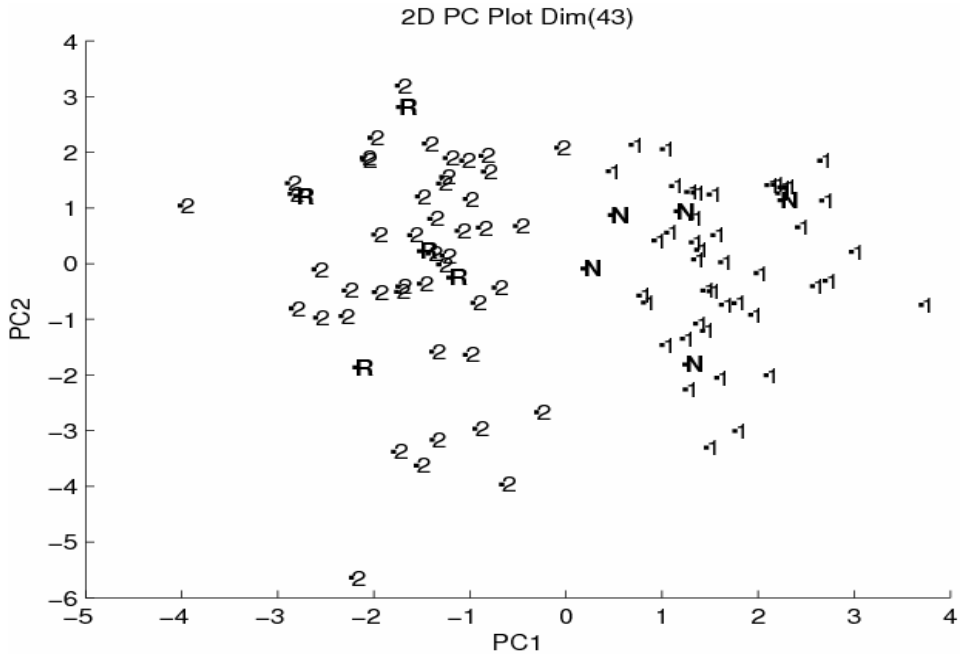



Figure 3.28. A plot of the two largest principal components developed from 90 samples and 42 features identified by the pattern recognition GA using PCKaNN with the Hopkins statistic as the fitness function for training set 6, and 90 samples and 33 features identified by the pattern recognition GA using PCKaNN with the modified Hopkins statistic as the fitness function for training set 6. 1 = no reoccurrence and 2 = reoccurrence (training set). R = reoccurrence and N = no reoccurrence (validation set)

**TP7- Normal**




**TP7- Hopkins**

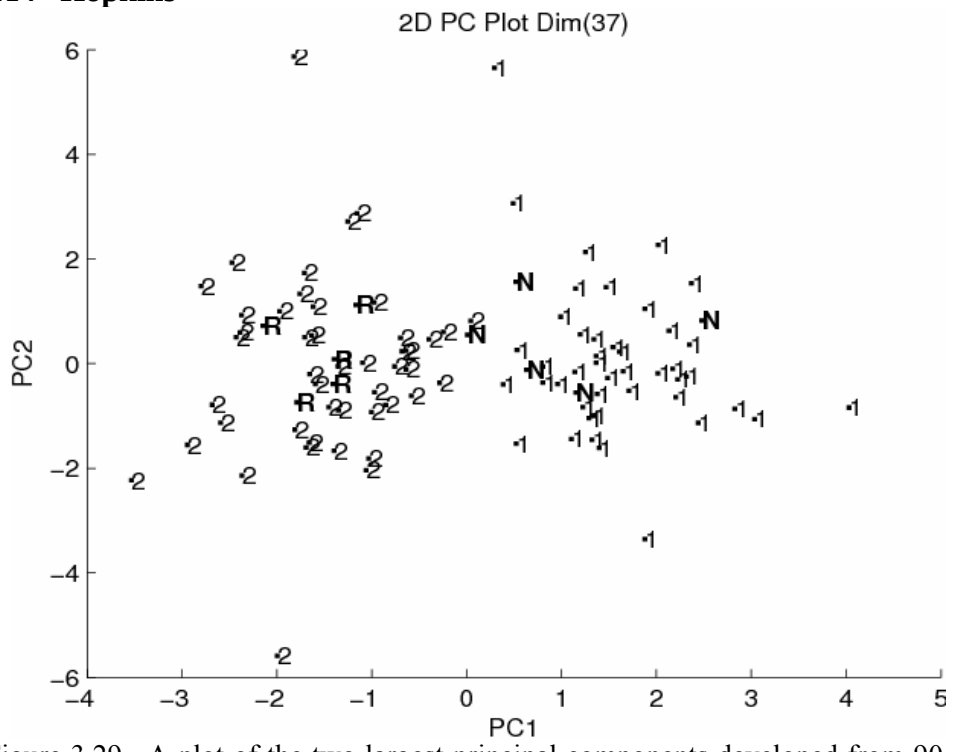
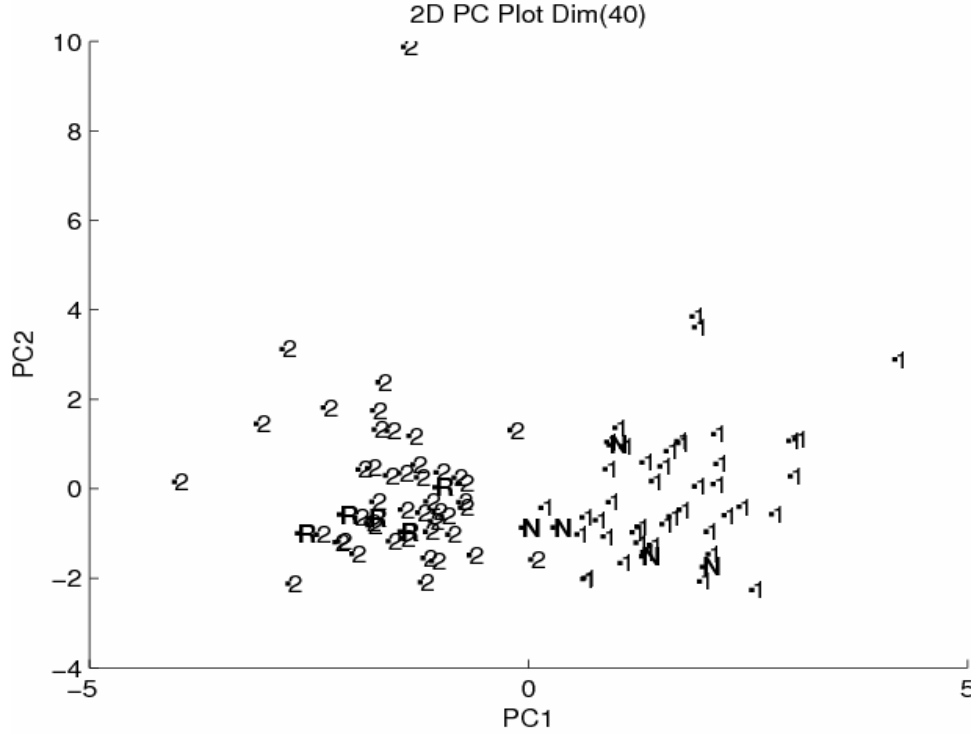



Figure 3.29. A plot of the two largest principal components developed from 90 samples and 43 features identified by the pattern recognition GA using PCKaNN as the fitness function for training set 7 and 90 samples and 37 features identified by the pattern recognition GA using PCKaNN with the Hopkins statistic as the fitness function for training set 7. 1 = no reoccurrence and 2 = reoccurrence (training set). R = reoccurrence and N = no reoccurrence (validation set)

**TP7- Modified Hopkins**




**TP8- Normal**

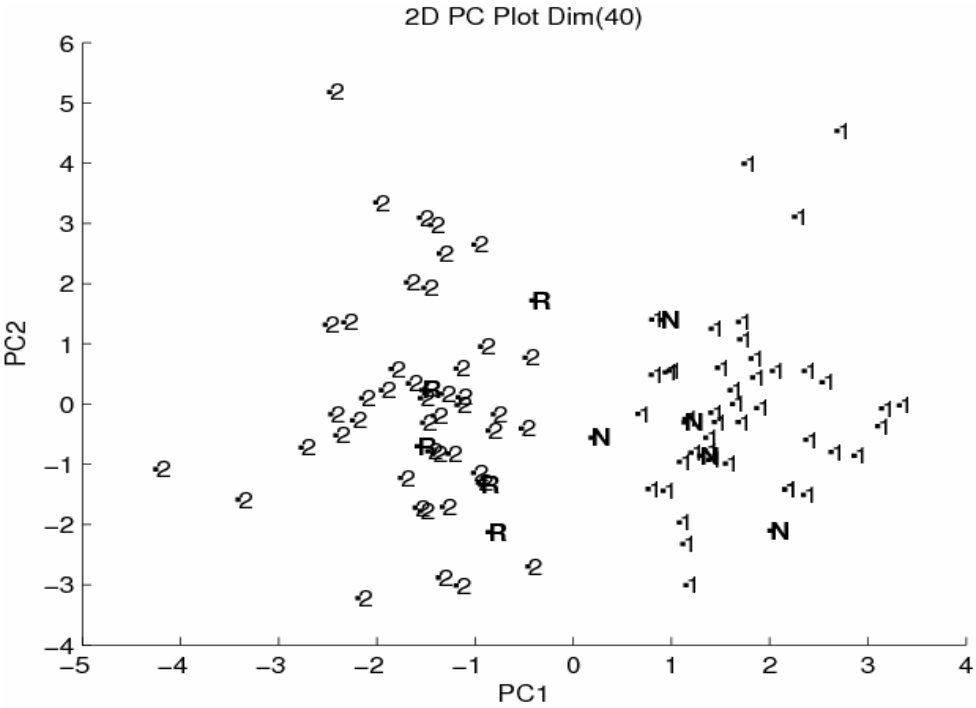
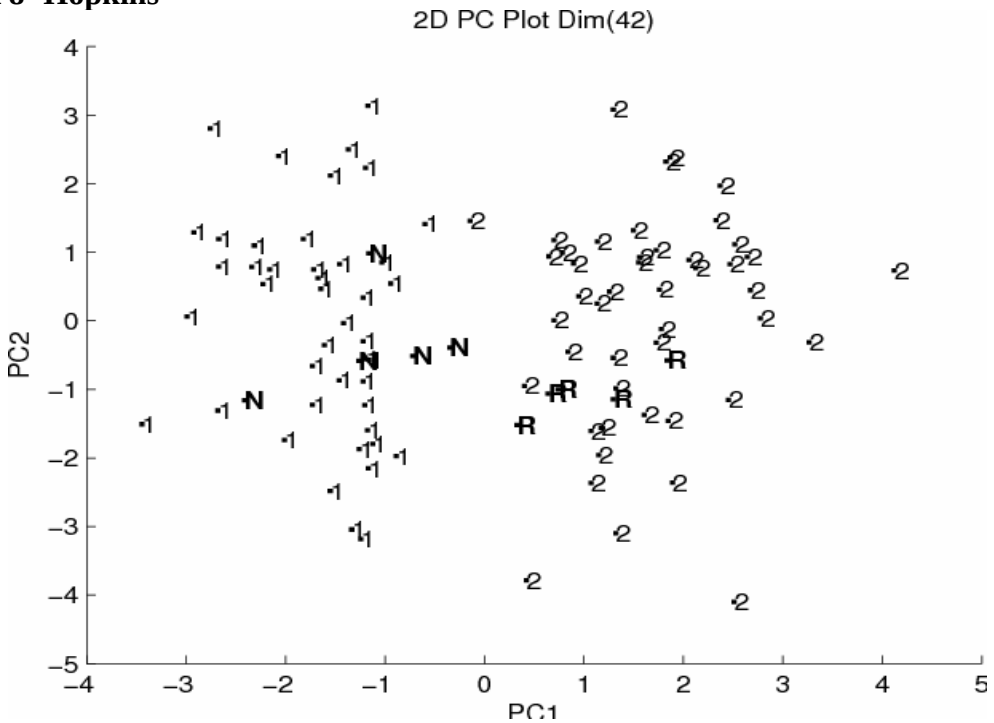



Figure 3.30. A plot of the two largest principal components developed from 90 samples and 40 features identified by the pattern recognition GA using PCKaNN with the modified Hopkins statistic as the fitness function for training set 7 and 90 samples and 40 features identified by the pattern recognition GA using PCKaNN as the fitness function for training set 8. 1 = no reoccurrence and 2 = reoccurrence (training set). R = reoccurrence and N = no reoccurrence (validation set)

**TP8- Hopkins**




**TP8- Modified Hopkins**

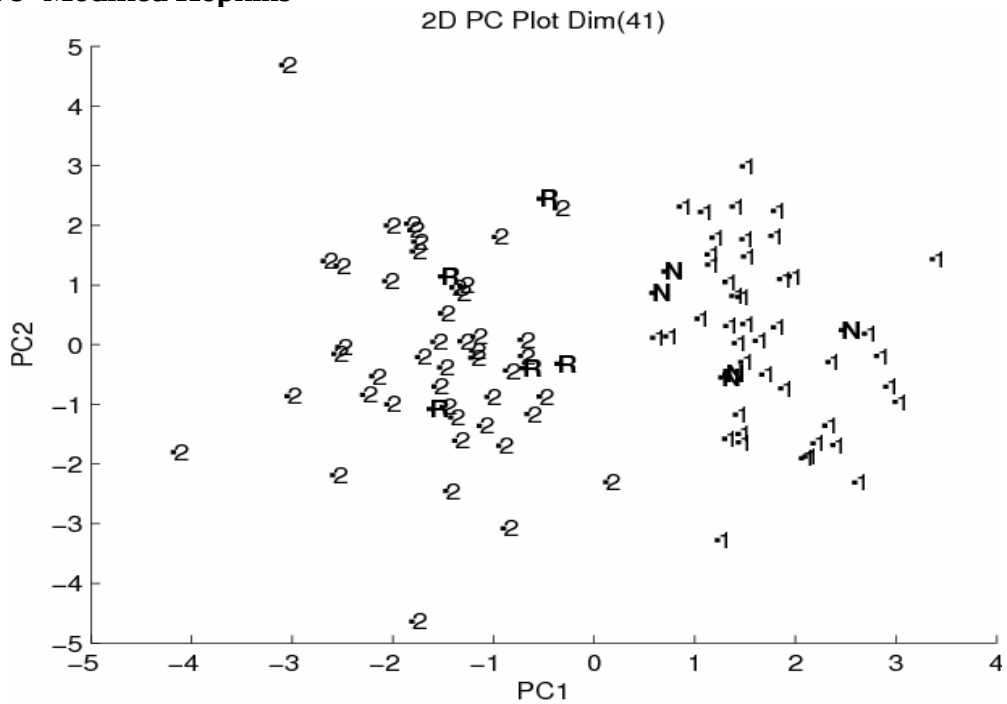
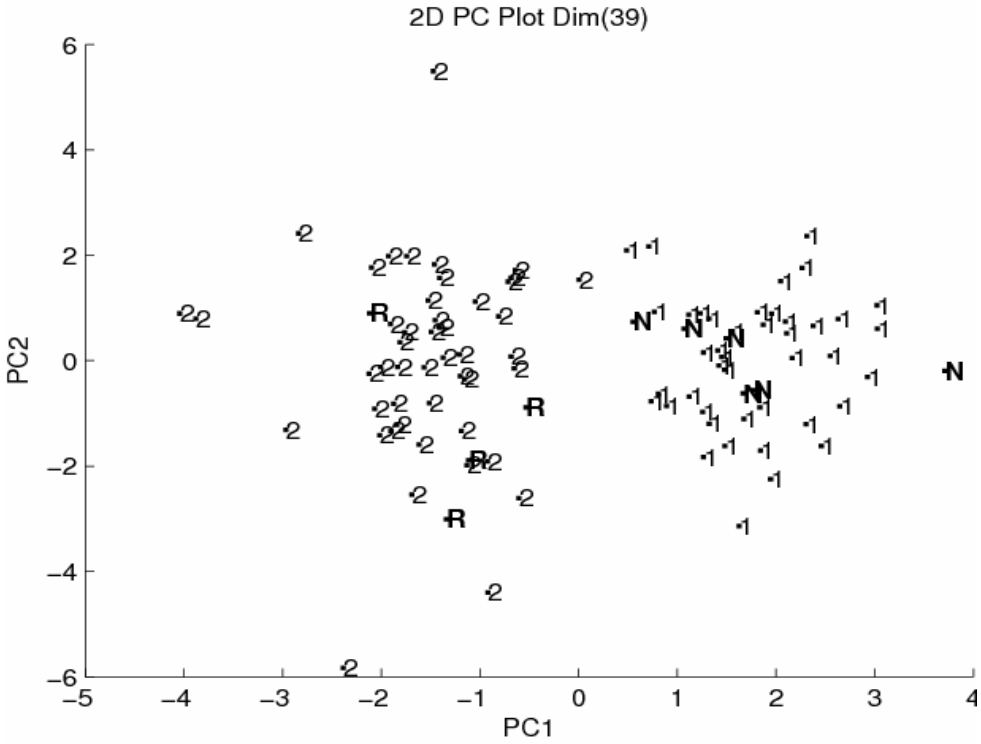



Figure 3.31. A plot of the two largest principal components developed from 90 samples and 42 features identified by the pattern recognition GA using PCKaNN with the Hopkins statistic as the fitness function for training set 8 and 90 samples and 41 features identified by the pattern recognition GA using PCKaNN with the modified Hopkins statistic as the fitness function for training set 8. 1 = no reoccurrence and 2 = reoccurrence (training set). R = reoccurrence and N = no reoccurrence (validation set)

**TP9 – Normal**




**TP9- Hopkins**

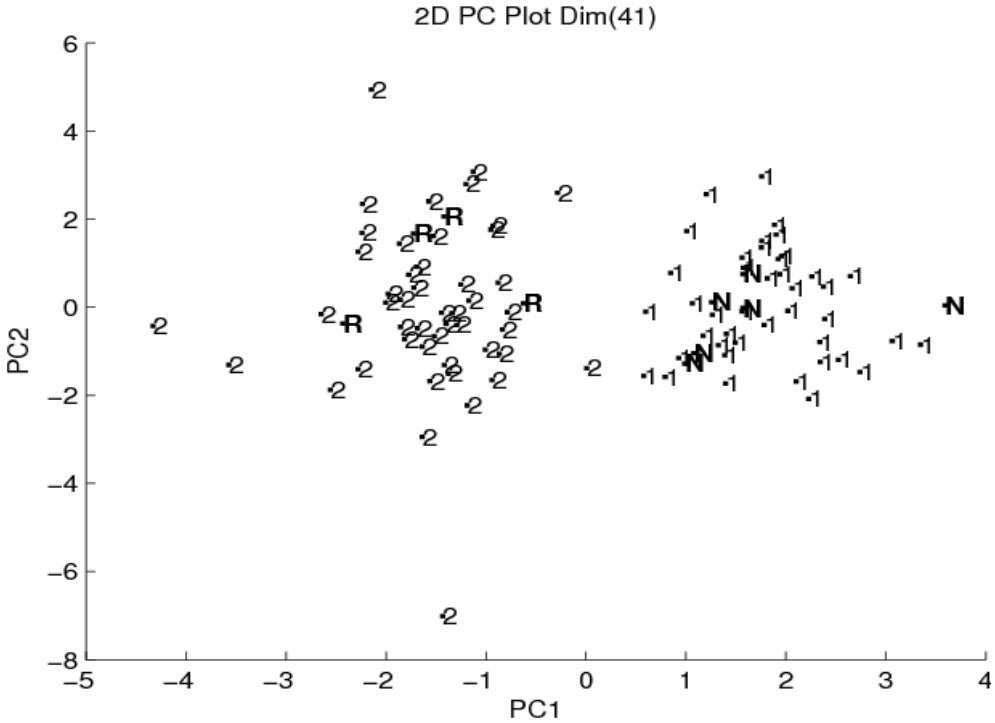
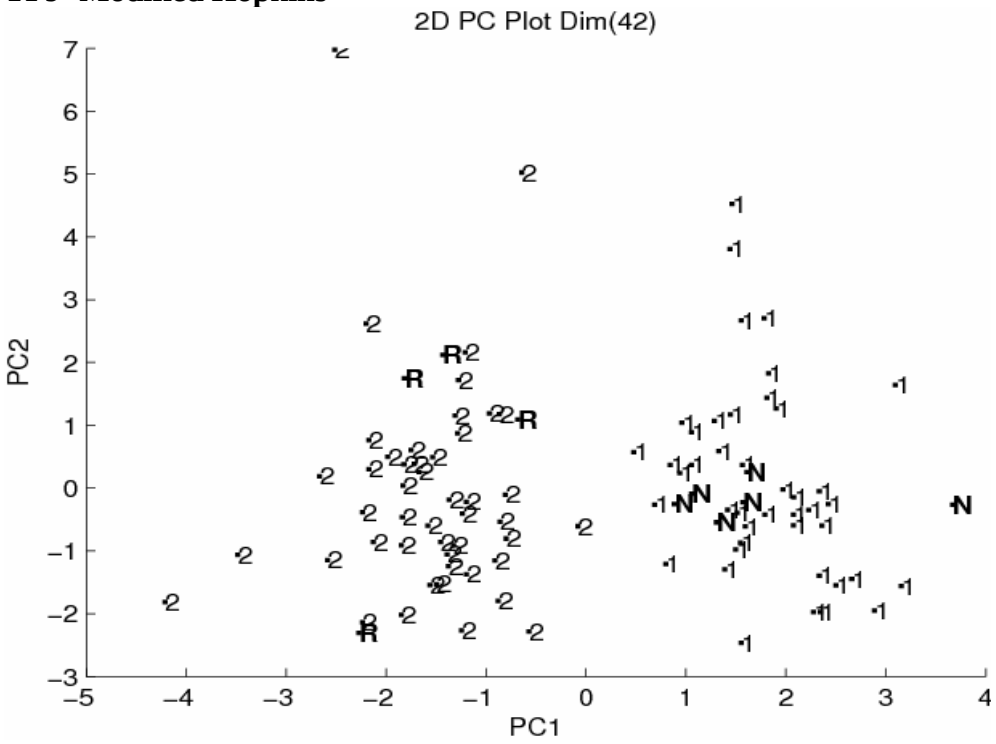



Figure 3.32. A plot of the two largest principal components developed from 90 samples and 39 features identified by the pattern recognition GA using PCKaNN as the fitness function for training set 9 and 90 samples and 41 features identified by the pattern recognition GA using PCKaNN with the Hopkins statistic for training set 9. 1 = no reoccurrence and 2 = reoccurrence (training set). R = reoccurrence and N = no reoccurrence (validation set)

**TP9- Modified Hopkins**




**TP10- Normal**

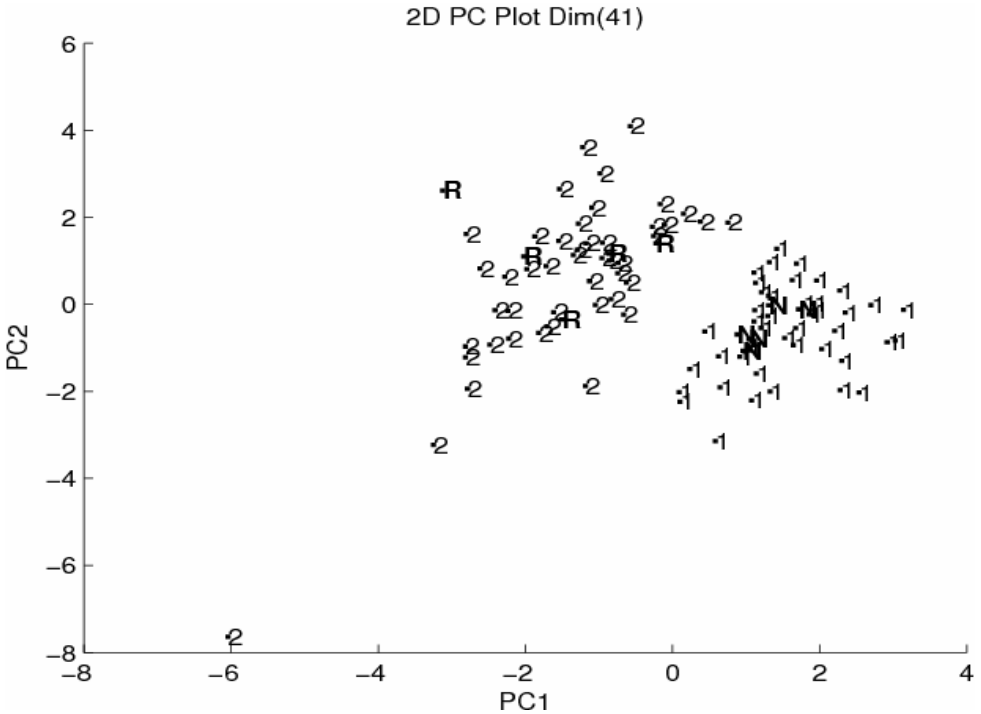
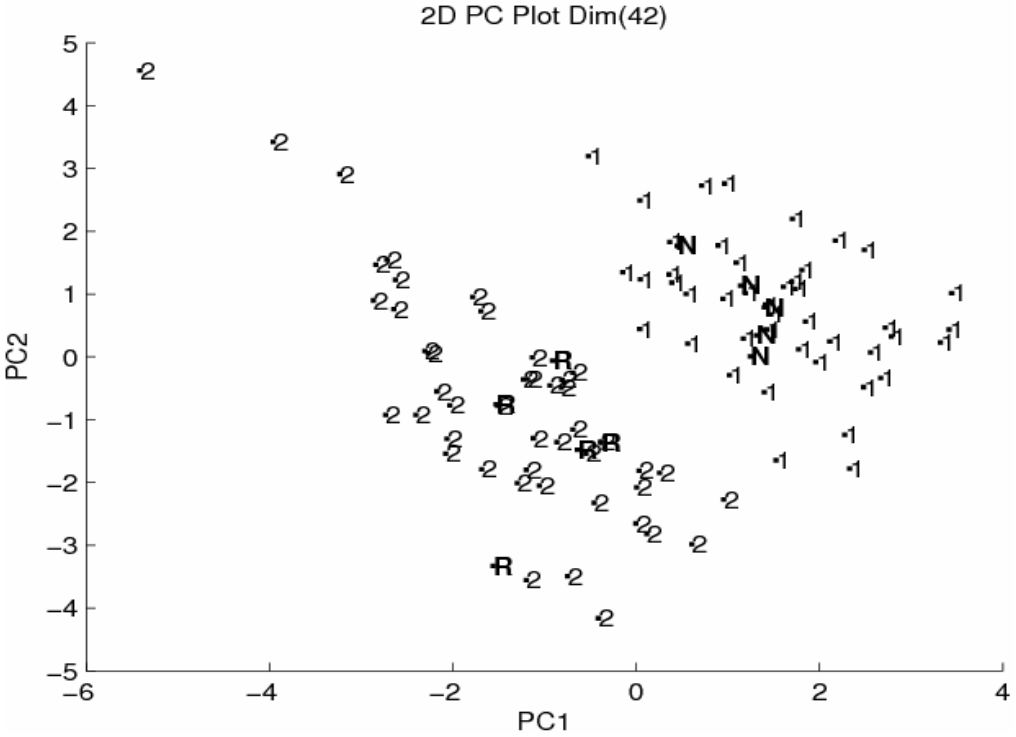



Figure 3.33. A plot of the two largest principal components developed from 90 samples and 42 features identified by the pattern recognition GA using PCKaNN with the modified Hopkins statistic as the fitness function for training set 9, and 90 samples and 41 features identified by the pattern recognition GA using PCKaNN as the fitness function for training set 10. 1 = no reoccurrence and 2 = reoccurrence (training set). R = reoccurrence and N = no reoccurrence (validation set)

**TP10- Hopkins**




**TP10- Modified Hopkins**

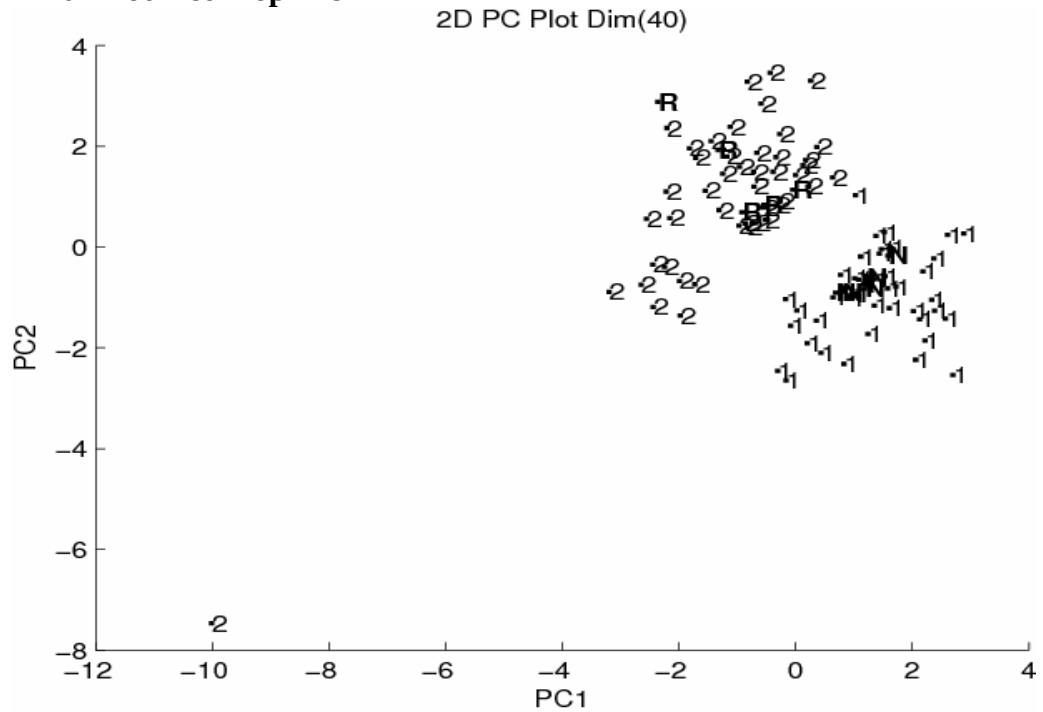



Figure 3.34. A plot of the two largest principal components developed from 90 samples and 42 features identified by the pattern recognition GA using PCKaNN with the Hopkins statistic as the fitness function for training set 10 and 90 samples and 40 features identified by the pattern recognition GA using PCKaNN with the modified Hopkins statistic for training set 10. 1 = no reoccurrence and 2 = reoccurrence (training set). R = reoccurrence and N = no reoccurrence (validation set)

## 3.6 Conclusion

The pattern recognition GA through the principal component plots that it generates allows any user to interpret the meaning of underlying relationships in multivariate data and to understand how a decision is made for a classification. The approach used by the pattern recognition GA for feature selection is the same approach that many statisticians would like to use for solving their classification problems, which is identifying a set of features whose principal component plot shows clustering on the basis of class. If separation by class is evident in the principal component plot, then all pattern recognition methods will perform well since between-group differences will be large compared to within group differences for this set of features. Using the pattern recognition GA, it is feasible to examine a large number of feature subsets, score their principal component plots, and thereby identify the truly informative features in a data set. Feature selection, supervised and unsupervised learning where the number of features is much greater than the number of samples, and pattern classification where only a few samples have class labels are examples of some of the shared challenges in data mining and computational biology that can be addressed using the pattern recognition GA.

Another interesting feature of this methodology is that an important problem in multivariate data analysis, feature selection for classification, has been reformulated as an optimization problem. Motivation for doing so is simple enough – the feature selection problem in pattern recognition is usually intractable for very large and noisy data sets, and clustering is often difficult to quantify. Clearly, the underlying optimization problem is both complex and error prone, which justifies the use of a genetic algorithm. The

computational environment offered by genetic algorithms is well suited for this problem, whereas it is not evident how one would formulate the feature selection problem in pattern classification using so-called traditional methods.

# CHAPTER 4

# Search Prefilters for Infrared Library Searching

## 4.1 Introduction

Infrared (IR) spectroscopy is an accepted method for chemical identification. The unique fingerprinting and identification ability provided by an IR spectrum results from the fact that peaks in the spectrum correspond to vibrational modes characteristic of the entire molecule (i.e., fingerprint region) or other modes directly related to the fundamental vibrations of specific functional groups. The combination of group frequencies and the fingerprint region has made the comparison of an unknown spectrum to a standard spectrum in a reference library a widely used method of identification.

Recently, there has been renewed interest in IR spectral matching because of the higher quality and larger amounts of IR data, improvements in computing power, and workers who are less well trained in the art of interpreting IR spectra. However, a concern in the use of reference library spectra for identification is the degree to which a search possesses true interpretive ability [65]. Most comparison schemes involve some type of point-by-point numerical comparison between the full spectrum of an unknown and each member of the library [66]. These algorithms lack interpretive ability because they treat the spectrum as a set of points rather than as a collection of specific bands. Band shifting is not handled well and bands of low intensity, which may be highly informative, are often ignored [67]. As a result, current IR library search algorithms are

restricted to identity searches. Because the number of compounds in an IR library is typically 20,000 whereas the total number of organic compounds in existence is several million, there is need for library search algorithms that can perform similarity searches.

Utilizing search prefilters, most of the problems encountered in IR library searching can be circumvented. The idea of search prefilters is based on the fact that most spectral comparisons performed during a search are of little use because the spectra in question are very dissimilar. A prefilter is a quick test to spot dissimilar spectra, thereby avoiding a complete spectral comparison. Prefilters would allow for more sophisticated and correspondingly for more time-consuming comparison search algorithms to be used for spectral matching since the size of the library can be culled down to allow for a specific match. From an interpretive standpoint, the information contained in the search prefiliter should be rotted on chemical structure. However, any substructural searching function used should have an appropriate degree of fuzziness. If the range of the function is too narrow, the substructure element of interest in an exotic environment may be missed. On the other hand, if the range is too wide, there may be too many false positives.

Pattern recognition methods have been used to develop search prefilters with mixed success. The reasons can be attributed to the nature of the modeling problem, which is sometimes quite complex. Structure-spectrum relationships cannot always be successfully modeled using a single spectral band. Some of the most significant wavelengths used to develop substructural classifiers from a spectral library often have no relationship with the characteristic frequencies of the functional group in question [68]. Closer inspection has shown that some wavelengths should be included in the

classifier for negative classification of potential interfering compounds rather than for an affirmative answer. (In other words, the inclusion of wavelengths that would exclude compounds containing functional groups that could be confused with carboxylic acids.) Previous published studies from our laboratory have shown that the fitness function of the pattern recognition GA is able to identify these types of features in large data sets, which makes the approach to feature selection described in this thesis well suited for this proposed application.

Using PCKaNN and the Hopkins statistic as the fitness function for the pattern recognition GA, a search prefilter based on the response function to the simple binary classification problem, carboxylic acids versus other compounds including carbonyl containing compounds, has been developed that allows for the specific detection of carboxylic acids. Carboxylic acids have highly characteristic features but there are also complications that confound the interpretation of their spectra. They can exist as either a dimer or a monomer which will affect the intensity of their most characteristic bands. Experts do not agree on the exact positions of peaks in their spectra. For example, a relatively broad absorption frequently occurs near 920 $cm^{-1}$, due to out of plane bending of the dimeric OH bond. The intensity of this peak is variable. Some authors consider this peak to be strong evidence for the presence of an acid whereas others consider this peak to be of little diagnostic value. There is general agreement that absorption due to the coupling of the OH-bending and C-O stretching around 1420 $cm^{-1}$ and 1300 $cm^{-1}$ (frequently lower especially in the presence of an electronegative group) can provide corroborative evidence for the presence of an acid dimmer, but these peaks have little diagnostic value by themselves. Because of these complications, the successful

development of a search prefilter to identify carboxylic acids in IR spectra has eluded workers.

In the two carboxylic acid classification studies described in this chapter, nonacids were selected to make the classification problem both challenging and informative. The nonacids selected included esters, aldehydes, ketones, amides, and other carbonyl containing compounds such as hydroxyl ketones. The wavelet packet transform, which was used to denoise and deconvolute IR library spectra, decomposed each spectrum into wavelet coefficients that represented both the high and low frequency components of the signal. This decomposition process was iterated through successive wavelet packets until the required level of signal decomposition was achieved. The pattern recognition GA was successfully used to identify wavelet coefficients characteristic of the carboxylic acid functional group.

## 4.2 Wavelets

The successful development of a carboxylic acid search prefilter described in this chapter can be attributed to the preprocessing of the IR spectral data by wavelets. The wavelet transform is based on small waves called **wavelets**. A wavelet is a localized waveform of effectively limited duration that has a varying frequency and an average value of zero. Wavelets are mathematical functions that have the ability to decompose the data into a set of different frequency components called wavelet coefficient packets, where each frequency component can be analyzed separately with a resolution matching its scale [69]. Wavelets can denoise and deconvolute overlapping bands in a spectrum; enhancing their features.

The wavelet transform is similar in some ways to the Fast Fourier Transform (FFT) which decomposes a signal into a combination of sine and cosine waves of different frequencies. However, wavelet analysis involves a non-redundant decomposition of the data into a set of approximation and difference functions by projecting the data onto shifted and dilated versions of finite-length and fast decaying oscillating waveform called the “mother wavelet”. The approximation function generates a sequence of the averages between two consecutive data in the input sequence. The difference function generates a sequence of the differences between two consecutive data in the current approximation sequence. These functions are applied recursively until the number of the elements in the difference sequence is equal to one [70].

Sine and cosine waves used in the Fourier Transform have infinite extent and can only extract global information from the signal. However, local features are better described by wavelets since they have local extent. The template of a typical wavelet basis function, the so-called ‘mother wavelet,’ is shown in Figure 4.1.

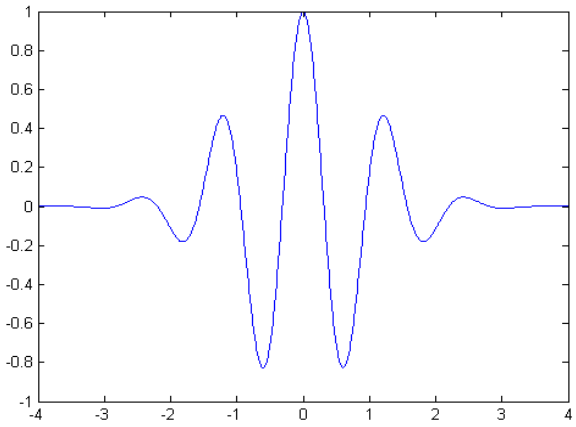



Figure 4.1. Template of a typical Wavelet basis function

The dilating or scaling functions provide the starting point for wavelet analysis. Dilating the wavelet simply means stretching or compressing the wavelet basis function along the horizontal direction by a scaling factor, in order to fit different scales of the signal. The dilated versions of the wavelets basis function can then be shifted to different locations of the signal to extract the local information. The general shape of the wavelet, like that of the sine and cosine functions in the Fourier transform, is the same for all the sizes of the wavelets that compose the waveform. Wavelet analysis is performed by dilating the wavelet basis function and shifting it to cover all parts of the signal (spectrum). Wavelet coefficients represent the correlation of different sections of the spectrum with the scaled versions of the wavelet basis function. Higher scales correspond to highly stretched wavelets. The more stretched the wavelet, the longer the portion is of the signal with which it is being compared, and the coarser will be the features being captured by the wavelet coefficients (see Figure 4.2). Slowly changing coarse features represent the low frequency components of the signal. Similarly, lower scales (see Figure 4.3) correspond to compressed wavelets that measure rapidly changing details and give the high frequency components of the signal.

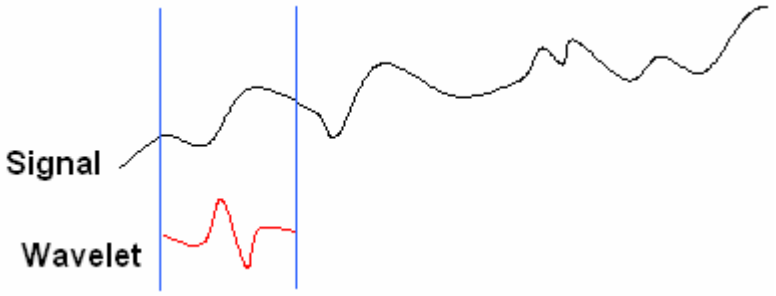



Figure 4.2. High scale representation of the signal by wavelets

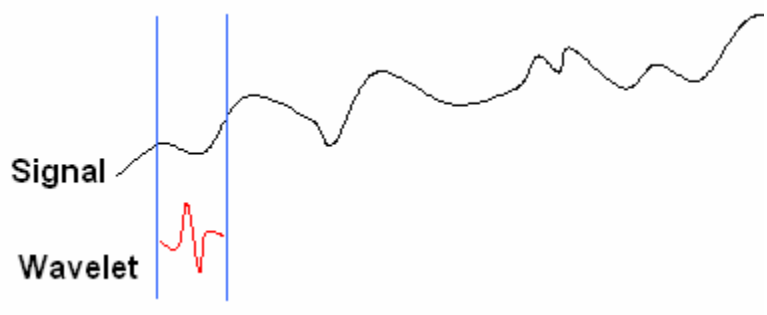



Figure 4.3. Low scale representation of the signal by wavelets.

The wavelet transform is implemented by passing each spectrum through two scaling filters: a high-pass filter and a low-pass filter (see Figure 4.4). The high-pass scaling filter generates the wavelet coefficients that represent the high frequency components of the signal, i.e. the details. The wavelet coefficients generated by the low-pass scaling filter represent the low frequency component of the signal, i.e. the approximations. Thus, the signal is decomposed into a low-scale high frequency packet and high-scale low frequency packet [71]. This process of decomposing a spectrum into both low frequency and high frequency packets allows the analyst to see both the details and the major trends in the signal. There are several kinds of wavelet transforms: continuous, discrete, fast, complex transforms, and wavelet packet transforms.

Consider a sine wave with noise as shown in Figure 4.5, in which the first and second levels of filtering are shown for an input sine wave. This decomposition process can be iterated using successive packets until the required level of signal decomposition is achieved to give what is called a wavelet packet tree (see Figure 4.6).

There are many different types of mother wavelets: Daubechies, symmlets, coiflet, Haar and biorthogonal. The Haar wavelet is the simplest wavelet. It is one period of a square wave. A major drawback of using the Haar wavelet is that it is not continuous

and therefore not differentiable. Daubechies are compactly supported orthonormal wavelets suitable for discrete wavelet analysis. Symmlets are nearly symmetrical wavelets. They are related to the Daubechies family as they share similar properties. Figure 4.7 shows the basic templates of the mother wavelets belonging to these families.

The criterion used to select the mother wavelet for the two studies discussed in this chapter was empirical and largely based on the ability of the wavelet to denoise and deconvolute the spectral data such that it could be separated into its respective classes using wavelet coefficients identified by the pattern recognition GA. There were also empirical rules used to guide this selection process. If the signal contained sharp peaks or discontinuities, Haar or other compact wavelets would be used. If the signal comprises broad peaks, a smoother wavelet such as Daubechies or Symmlet would be employed.

It is evident from these discussions that wavelets have attributes (unlike the Fourier transform) which makes them ideal to use for preprocessing IR spectra. Wavelet basis functions are localized in space, unlike Fourier sine and cosine functions, which are not local and often do a poor job in approximating sharp peaks. As the wavelets themselves are sharp, asymmetric and irregular and have a finite domain, they have the ability to represent functions that have both sharp peaks and discontinuities. Furthermore, the wavelet transform does not generate a single set of basis functions like the Fourier transform. It has an infinite set of potential basis functions that provide immediate access to information that is obscured by other time-frequency methods since wavelet basis functions can be tuned for specific applications.

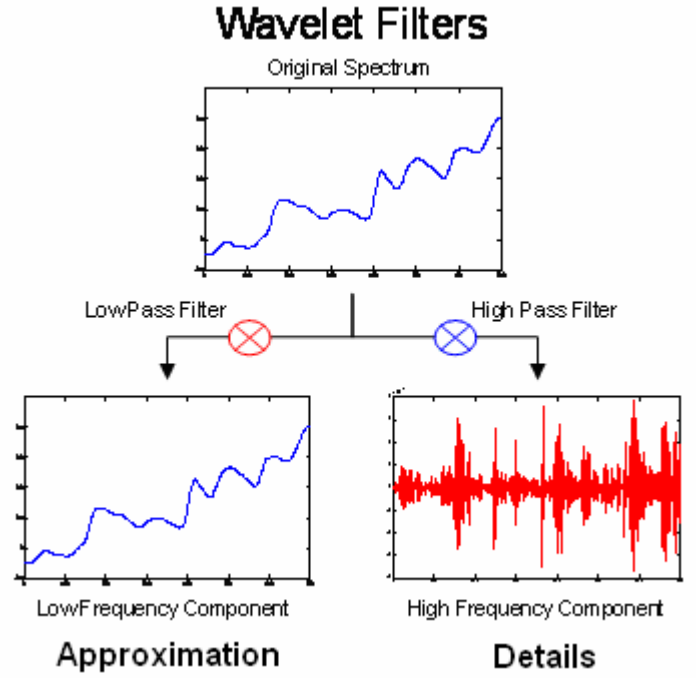



Figure 4.4. Decomposition of the spectrum using wavelet filters.

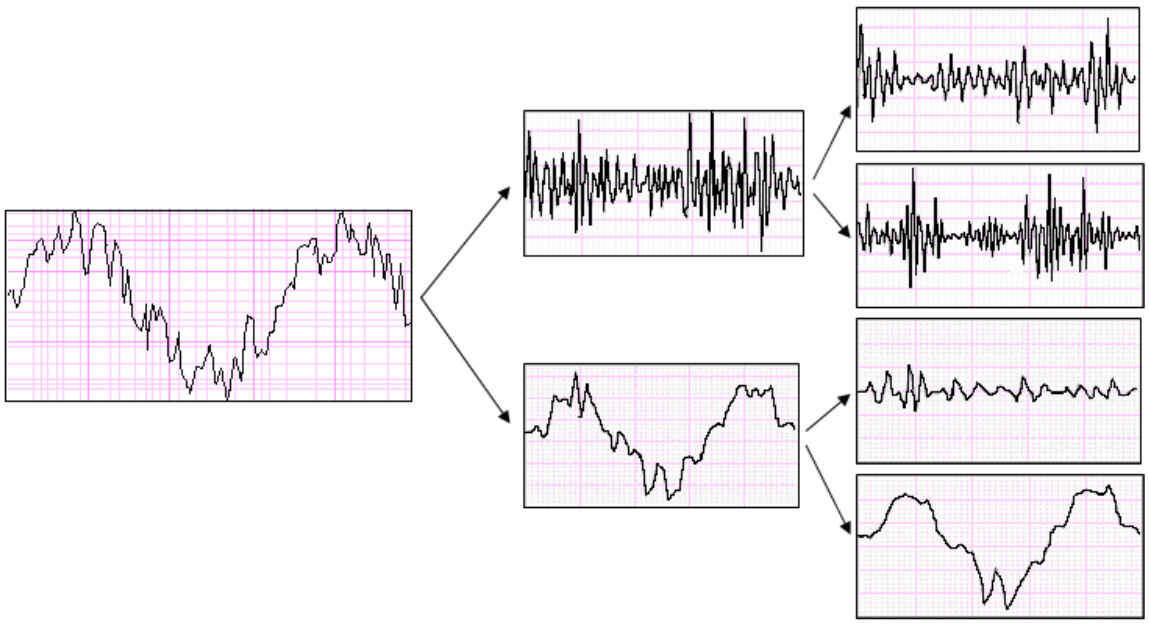


Figure 4.5. Second level decomposition of a sine wave using wavelet filters

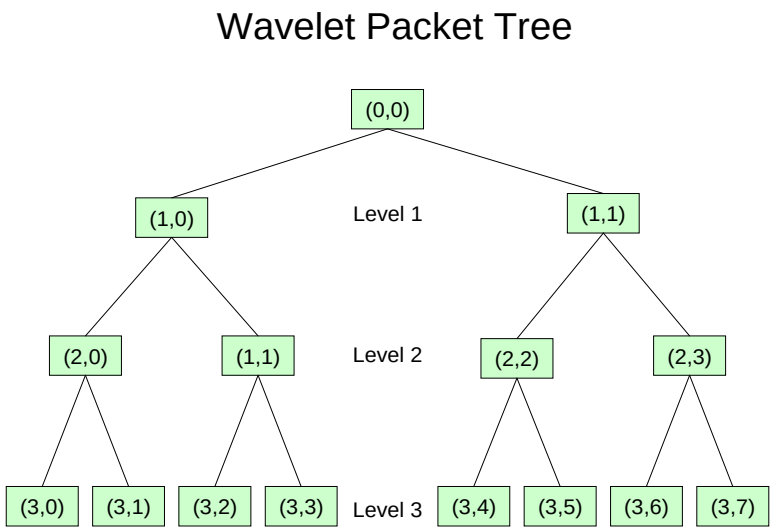



Figure 4.6. Wavelet packet tree decomposition of a signal at different levels.

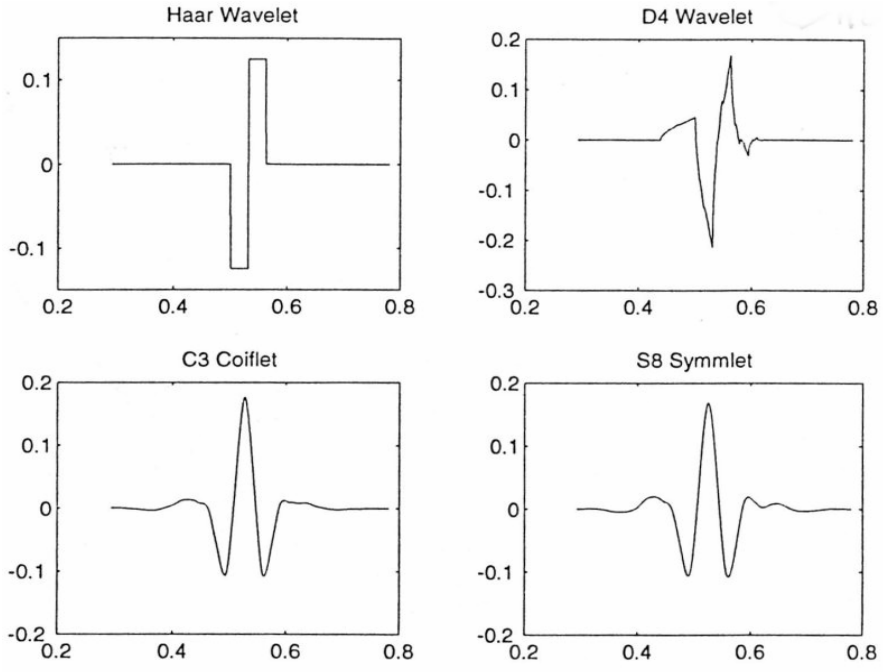



Figure 4.7. Templates of several “mother” wavelets

## 4.3 Data Collection and Preprocessing

The IR spectra used in this study were obtained from Nicolet (Madison, WI) and Biorad (King of Prussia, PA). Each IR spectrum, which was represented by 460 points, was normalized to unit length. For pattern recognition analysis, each IR spectrum was initially represented as a data vector, $x = (x_1, x_2, x_3,\ldots.x_j,\ldots..x_{420})$ where $x_j$ is the infrared absorbance of the $j^{th}$ point from the normalized first spectrum. The wavelet packet tree using a symmlet mother wavelet was used to denoise and deconvolute each pattern vector. For pattern recognition analysis, each IR spectrum was represented as a set of wavelet coefficients. The coefficients were autoscaled to ensure equal weighting in the analysis. Wavelet analysis was performed using the wavelet toolbox in MATLAB R2006a.

## 4.4 Results and Discussion

*Nicolet Vapor Phase Library Study*: The training set used in this study consisted of 476 IR spectra of carboxylic acids and noncarboxylic acids (see Table 4.1). Most of the spectra in the training set were acquired by Aldrich using as samples their products. The remaining spectra were obtained from the EPA gas phase IR collection and from the Bayerische Julious Maximilian Universitat Wurzburg. Each IR spectrum was measured in a heated cell or light pipe connected to the outlet of a gas chromatograph. The spectra were originally acquired at 0.5-2cm-1 spectral resolution. All spectra were mathematically deresolved during conversion to the Omnic Library format.

Aldehydes, ketones, esters and amides were included in the training set to ensure that the classification problem (identification of carboxylic acids) was challenging. Figures 4.8 and 4.9 are examples of the compounds (spectra) that comprised the training

set. Butyric acid has several characteristic carboxylic acid bands in its spectrum, whereas the spectrum of cyclopropanedicarboxylic acid more closely resembles octanoyl chloride or propionic anhydride. Spectra that comprise the prediction set (see Table 4.2) included alcohols and ketones, esters, and amides containing the OH functionality. The total number of compounds in the prediction set was 95.

**Table 4.1 – Nicolet Spectra Training Set**

| Functional Group | No. of compounds |
|---|---|
| Carboxylic acid (Two contain COOR) | 156 |
| Negative class (phosphates, alkenes, alkynes, alkanes) | 220 |
| Aldehydes | 25 |
| Ketones | 25 |
| Esters | 25 |
| Amides | 25 |
| **Total Number of Compounds** | **476** |

**Table 4.2 – Nicolet Spectra Prediction Set**

| Functional Group | No. of compounds |
|---|---|
| Carboxylic acid (Two contain COOR) | 27 |
| Negative class (phosphates, alkenes, alkynes, alkanes) | 25 |
| Alcohols | 10 |
| Ketones (5 ketones contained OH) | 15 |
| Esters (2 esters contained OH) | 13 |
| Amides (3 amides contained OH) | 15 |
| **Total Number of Compounds** | **95** |

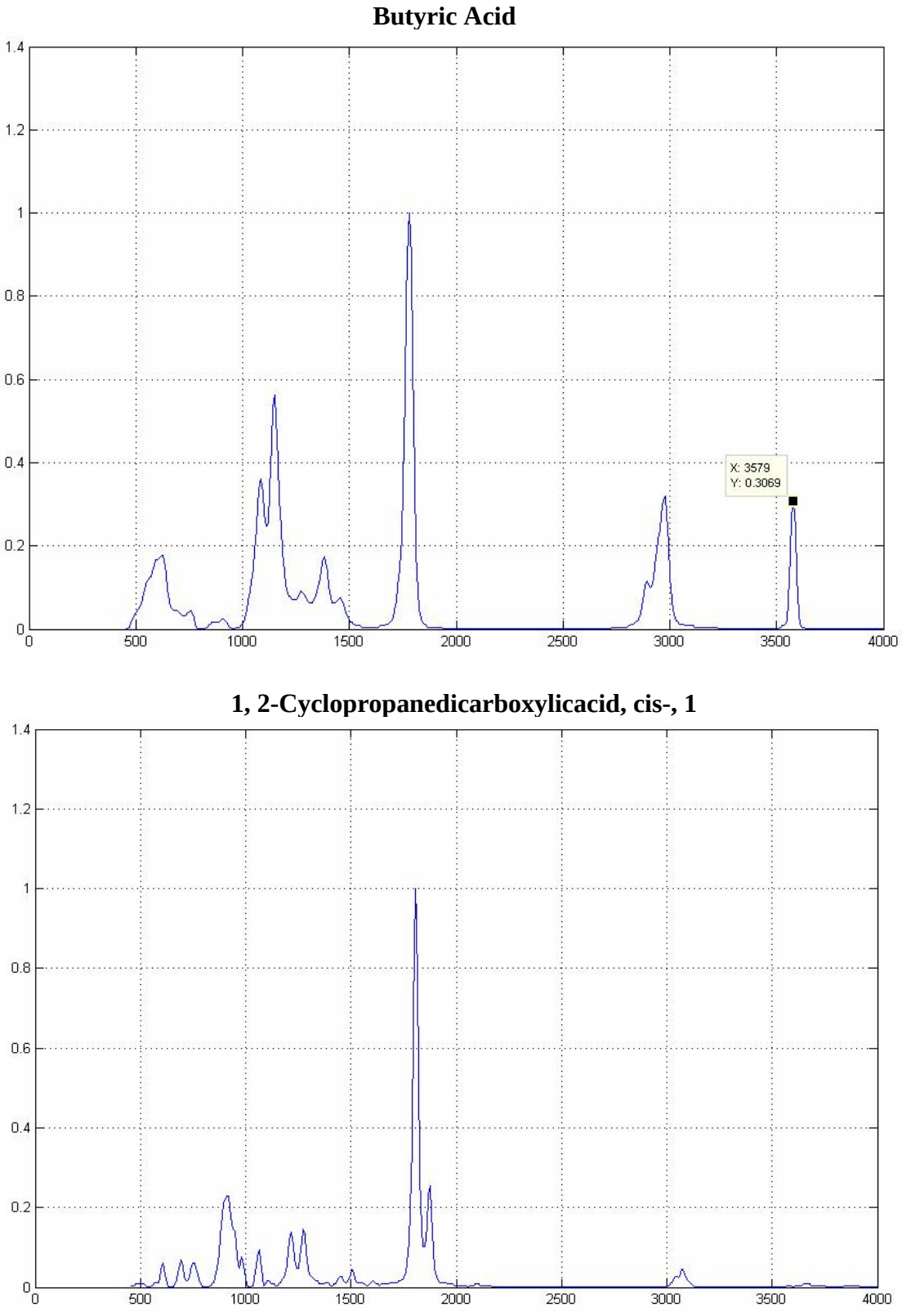



Figure 4.8. Infrared absorbance spectra of butyric acid and cis 1, 2-cyclopropanedicarboxylic acid.

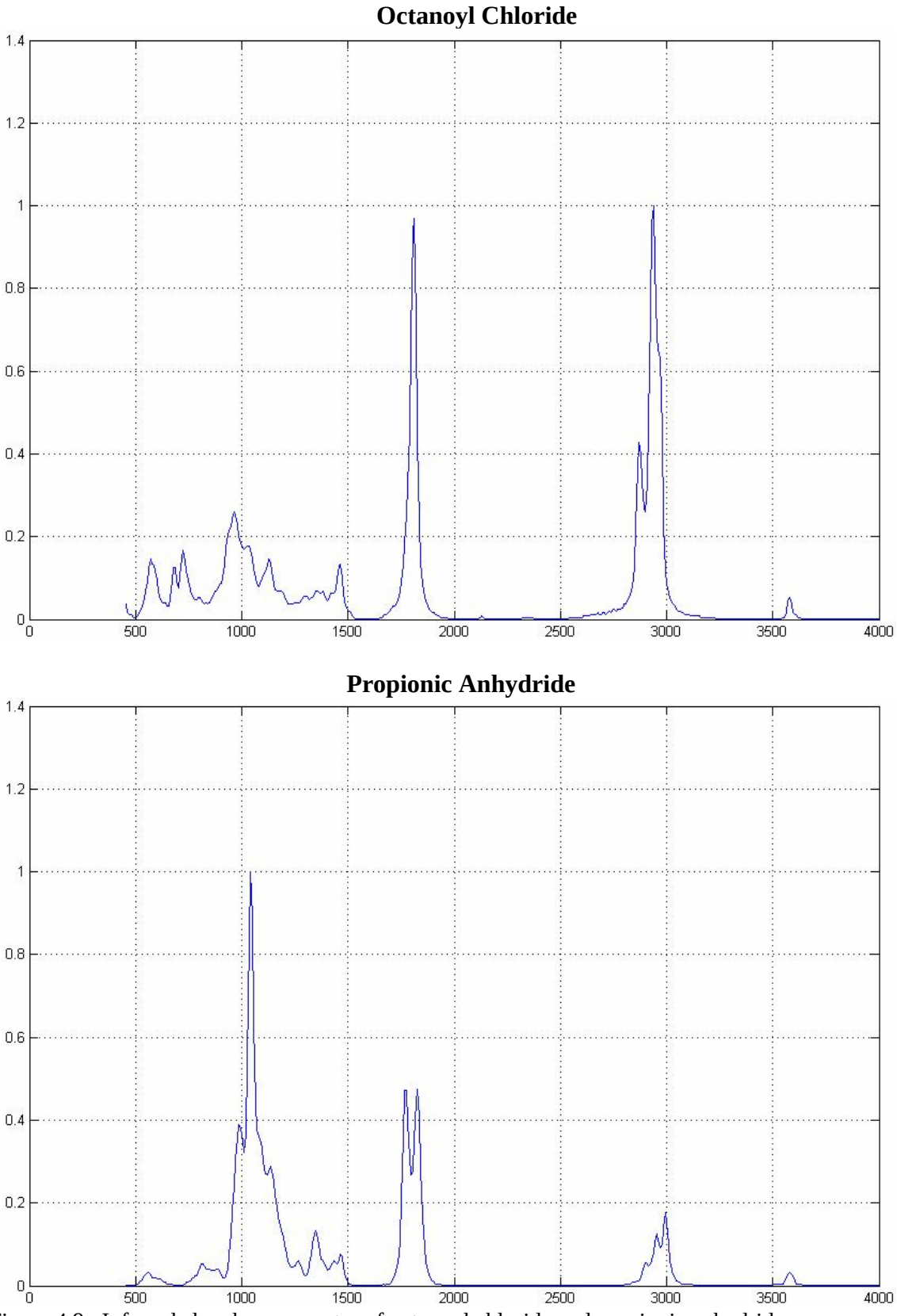


Figure 4.9. Infrared absorbance spectra of octanoyl chloride and propionic anhydride.

The first step in the study was to apply PCA to the raw training set data. The data were auto-scaled to ensure that each wavelength had equal weight in the analysis. Figure 4.10 shows a plot of the two largest principal components of the 460-point IR spectra that comprised the training set. Each spectrum is represented as a point in the principal component plot (1 = carboxylic acid and 2 = noncarboxylic acid). The overlap between the two groups (carboxylic acids and noncarboxylic acids) in the principal component plot is evident.

The pattern recognition GA (PCKaNN fitness function) was used to identify wavelengths characteristic of the IR absorption profile of each class. Features were identified by sampling key feature subsets, scoring their principal component plots, and tracking classes and/or samples, which were most difficult to classify. The boosting routine used this information to steer the population to an optimal solution. After 300 generations, the pattern recognition GA identified 22 spectral features whose principal component plot (Figure 4.11) showed some clustering of the IR spectra on the basis of class.

The symmlet 6 wavelet at the $8^{th}$ level of decomposition was applied to the IR spectra to deconvolve overlapping spectral bands and to denoise the data. Figure 4.12 shows a plot of the two largest principal components of the 476 spectra and 9200 wavelet coefficients used to represent each spectrum. There is a definite improvement in the separation between the two groups (carboxylic acids and noncarboxylic acids) after wavelet analysis is applied to the IR spectra. Figure 4.13 shows a principal component plot of the 41 wavelet coefficients identified by the pattern recognition GA using PCKaNN with the Hopkins statistic as the fitness function. For underdetermined data

sets, even quite well behaved multivariate normal distributions with no outliers will have sets of variables that produce eigenvector projections containing points that appear as outliers in the principal component plot. This problem was encountered with the symmlet 6 mother wavelet, which prevented us from performing the analysis using PCKaNN without the Hopkins statistic as the fitness function.

A prediction set of 95 compounds was used (see Table 4.2) to assess the predictive ability of the 41 wavelet coefficients identified by the pattern recognition GA. The prediction set spectra were projected onto the principal component map developed from the 476 spectra and 41 wavelet coefficients. Figure 4.14 shows the projection of the prediction set spectra onto a principal component map defined by the 41 coefficients selected by the pattern recognition GA. Each projected infrared spectrum lies in a region of the map occupied by spectra possessing the same class label. Evidently, the GA can identify wavelet coefficients characteristic of the compounds' functional group. This suggests that wavelet analysis and the pattern recognition GA can be used to extract structural information from spectral data.

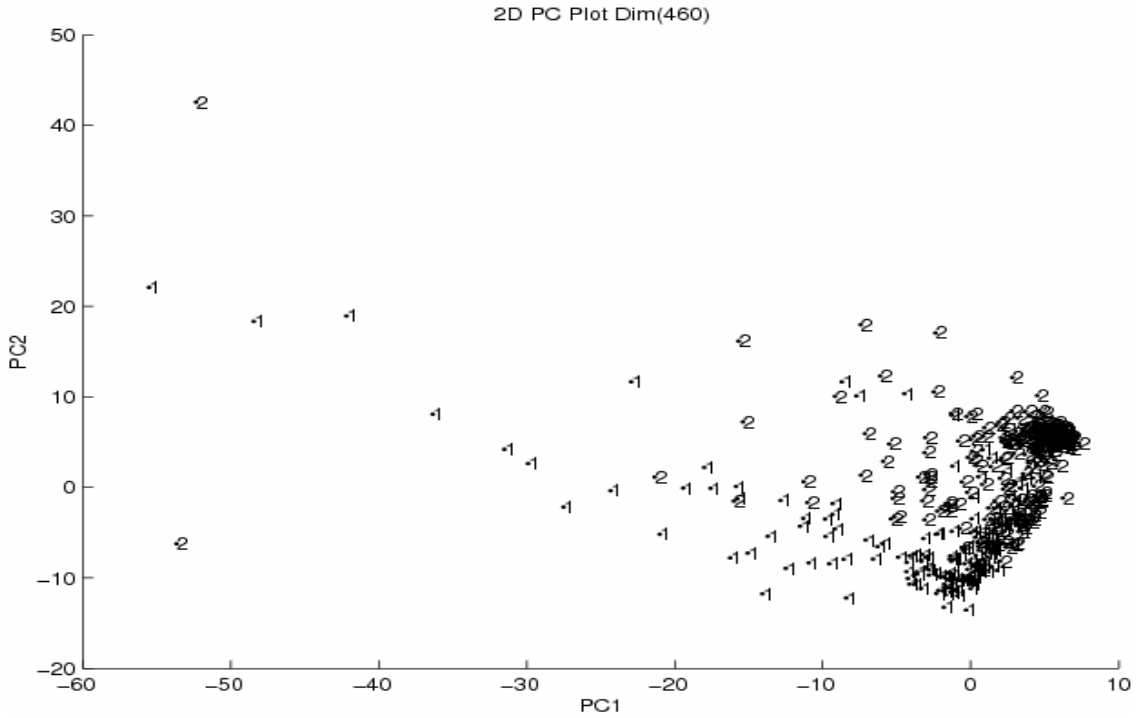



Figure 4.10. Plot of the two largest principal components of the 460-point IR spectra that comprised the Nicolet training set. Each spectrum is represented as a point in the principal component plot (1 = carboxylic acid and 2 = noncarboxylic acid).

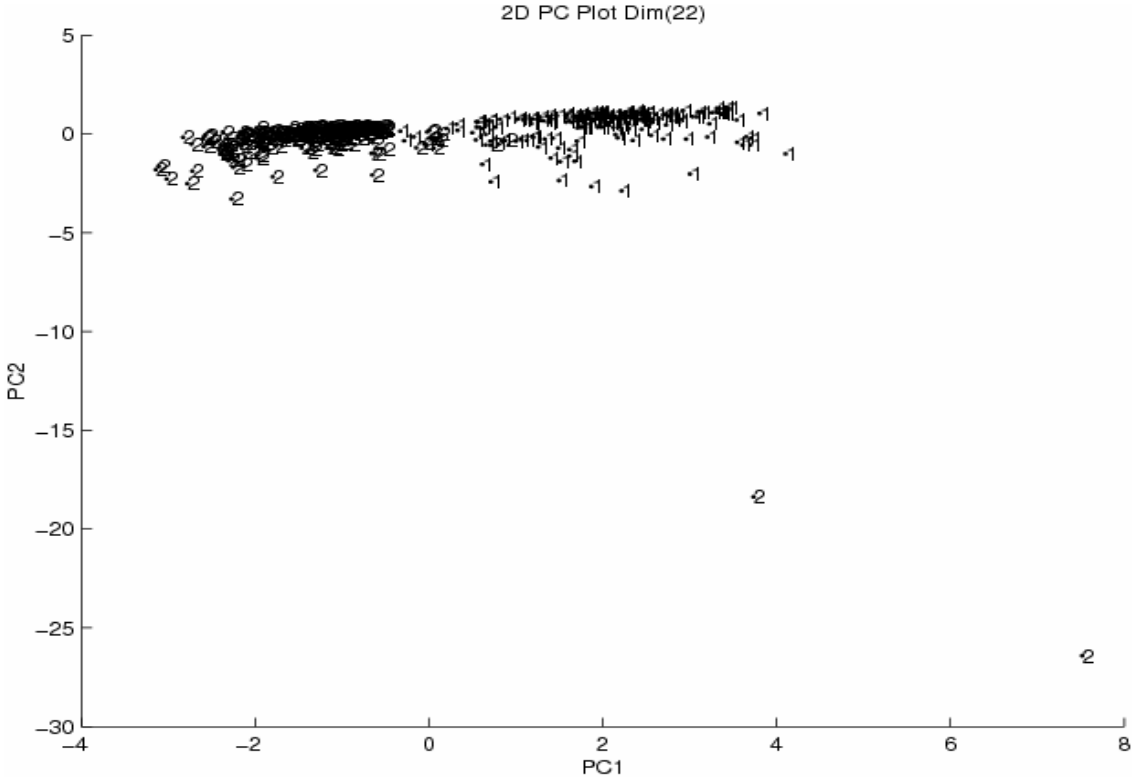



Figure 4.11. Plot of the two largest principal components of the 476 IR spectra and the 22 spectral features identified by the pattern recognition GA. Each spectrum is represented as a point in the principal component plot (1 = carboxylic acid and 2 = noncarboxylic acid).

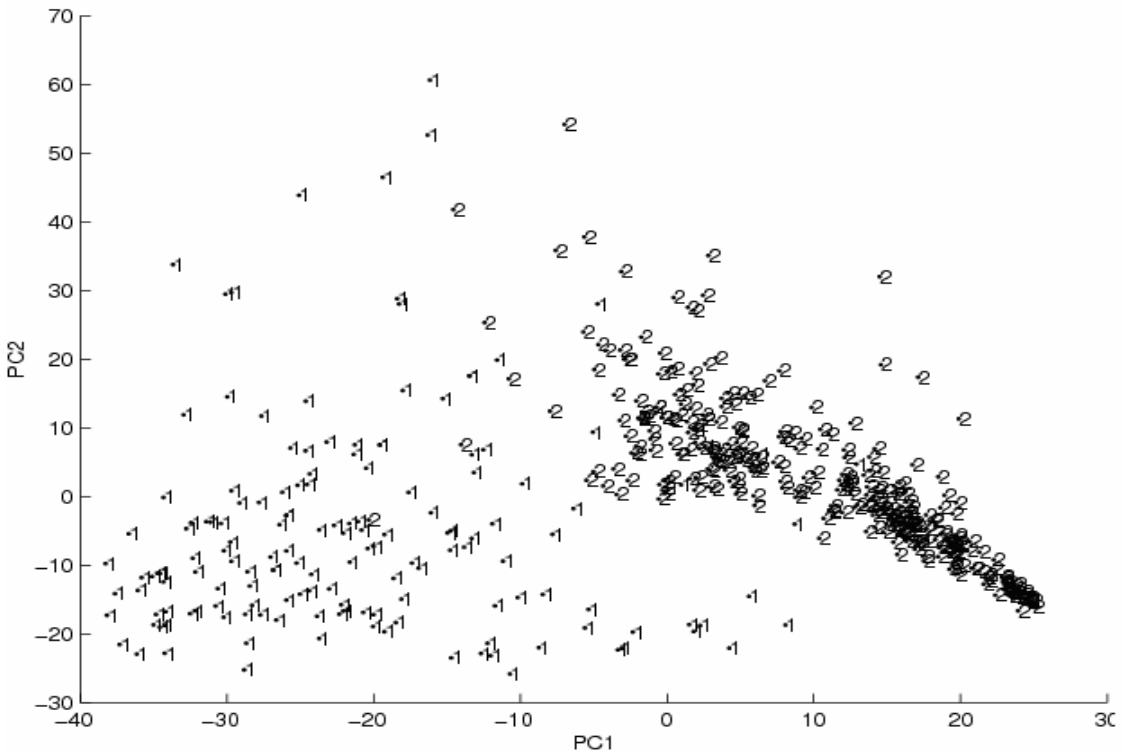



Figure 4.12. Plot of the two largest principal components of the 476 Nicolet training set spectra and 9200 wavelet coefficients that comprised the training set. Each spectrum is represented as a point in the principal component plot (1 = carboxylic acid and 2 = noncarboxylic acid).

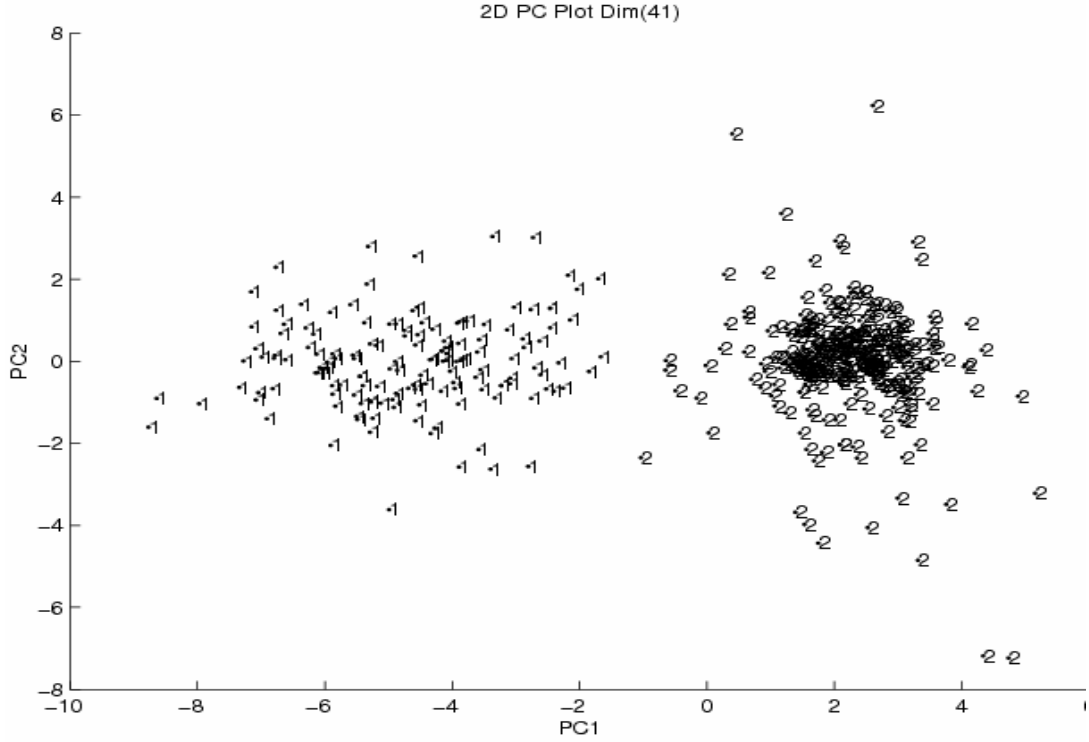



Figure 4.13. Plot of the two largest principal components of the 476 spectra and the 41 wavelet coefficients identified by the pattern recognition GA. Each spectrum is represented as a point in the principal component plot (1 = carboxylic acid and 2 = noncarboxylic acid).

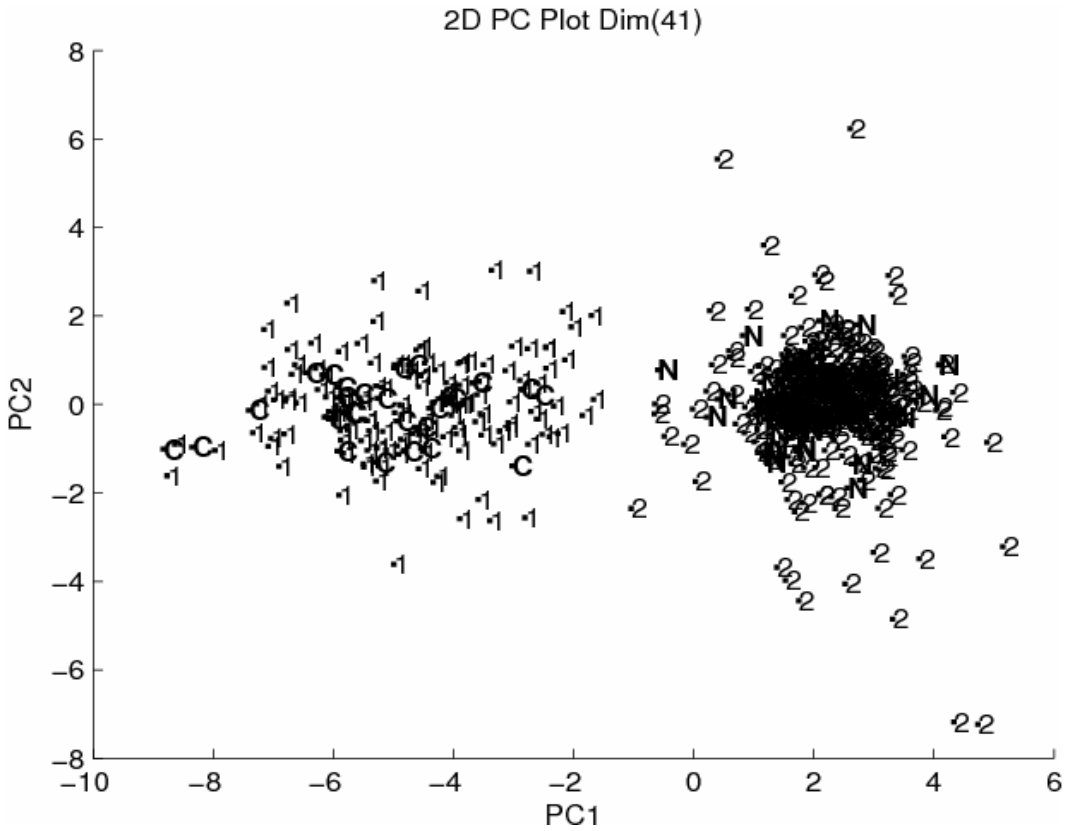


Figure 4.14 Plot of the two largest principal components of the 476 spectra and the 41 wavelet coefficients identified by the pattern recognition GA. Each spectrum is represented as a point in the principal component plot. 1 = carboxylic acid and 2 = noncarboxylic acid (training set). C = carboxylic acid and N = noncarboxylic acid (validation set).

*EPA Vapor Phase Spectral Library* (Biorad): The training set consisted of 435 IR spectra of carboxylic acids and noncarboxylic acids (see Table 4.3). The noncarboxylic acids contained aldehydes, ketones, esters and amides as well as compounds containing at least two of these functional groups. The presence of these compounds in the training set made the classification problem (identification of carboxylic acids) more challenging. Figures 4.15 and 4.16 are examples of the compounds (spectra) that comprise the training set. The spectrum of 2, 2-dimethylbutyric acid is representative of a carboxylic acid, whereas the spectrum of sarcosine, which is also a carboxylic acid, lacks several bands characteristic of the COOH group. The spectrum of the diethyl ester of tartaric acid and the IR spectrum of 6-bromo vanillin could easily be mistaken for that of a carboxylic

acid. Spectra that comprised the first prediction set are shown in Table 4.4. Spectra in the first prediction set included esters, ketones, amides, alcohols, and acid chlorides. Many of these compounds also contain an OH moiety. The total number of compounds in the first prediction set is 85.

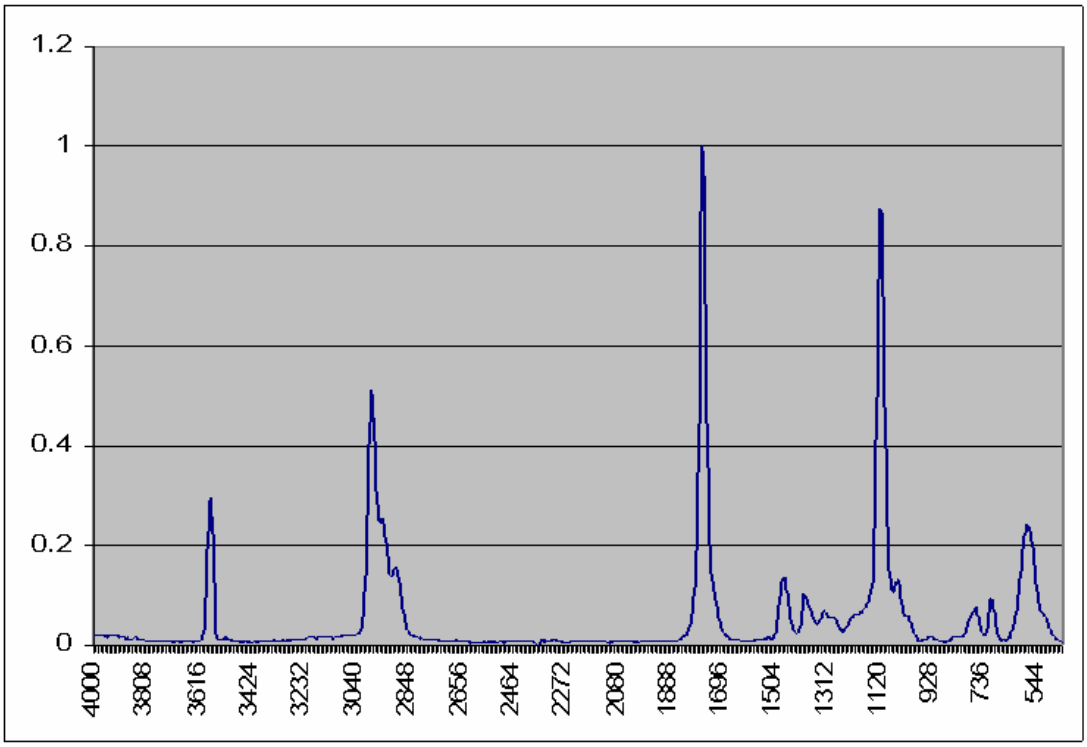
**Table 4.3- EPA Vapor Phase Library Spectra Training Set**

| Functional Group | No. of compounds |
|---|---|
| Carboxylic acids | 140 |
| Negative class (phosphates, alkenes, alkynes, alkanes) | 202 |
| Aldehydes (10 aldehydes also contained Ethers, 2 contained Amine, 2 had Esters and 1had Amide) | 25 |
| Ketones (6 Ketones also contained alcohol, 2 had Amine, 1 had Ester and 1 had Ether) | 20 |
| Esters (7 were also Amides, 4 had Ether, 3 had amine, 2 had alcohol, 2 had Aldehyde, 1 had Ketone and 1 had acid chloride ) | 26 |
| Amides ( 7-Esters, 4-Amines, 2-Alcohols, 1-aldehyde) | 30 |
| Acid Chlorides | 3 |
| **Total Number of Compounds** | **435** |

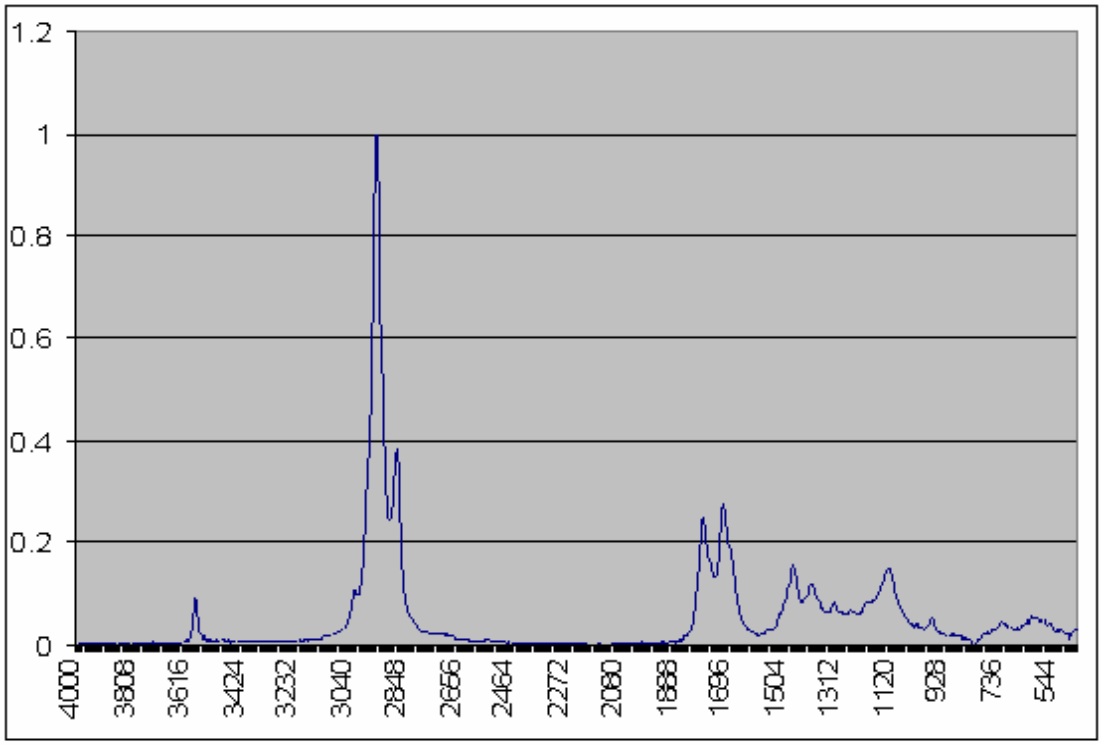
**Table 4.4- EPA Vapor Phase Library Spectra Prediction Set I**

| Functional Group | No. of compounds |
|---|---|
| Carboxylic acids | 24 |
| Negative class (phosphates, alkenes, alkynes, alkanes) | 20 |
| Esters (1 Ester contained OH, 3 esters contained amine and 2 contained amide) | 16 |
| Ketones (4 Ketones contained OH and 1 contained amine) | 12 |
| Amides (1 amide contained OH and 2 contained esters) | 13 |
| Alcohols | 6 |
| Acid chloride | 1 |
| Aldehyde | 1 |
| **Total Number of Compounds** | **85** |

**Butyric acid, 2, 2- dimethyl-,**

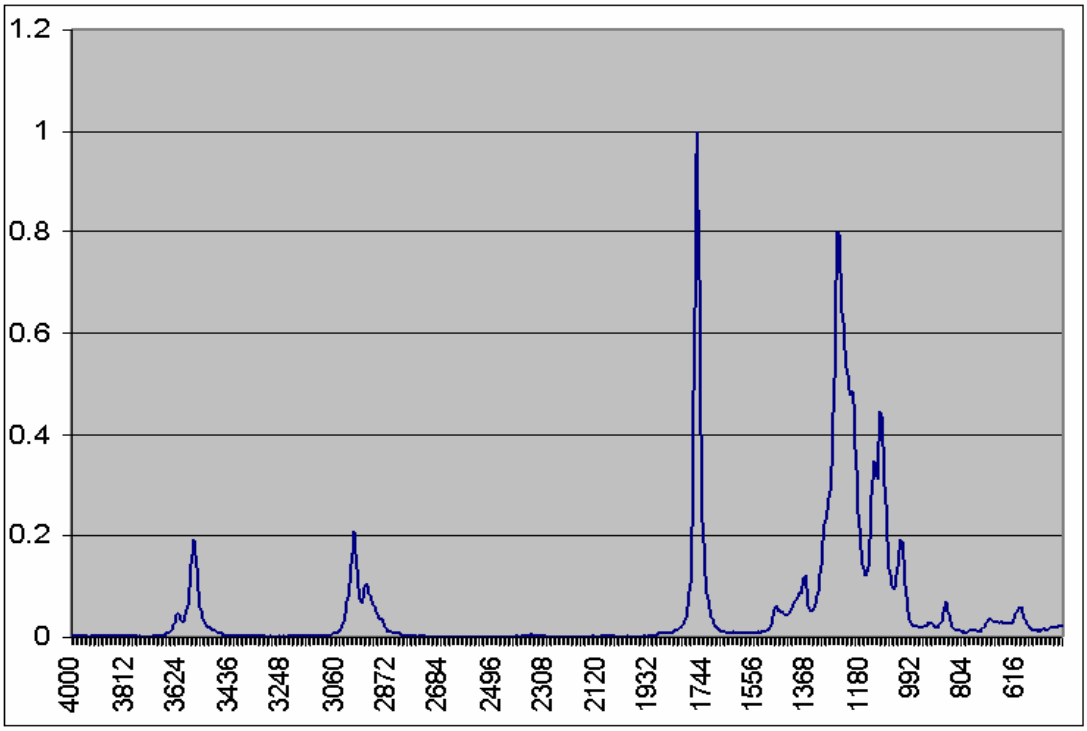



**Sarcosine, n-cis-9-octadecenoyl-,**

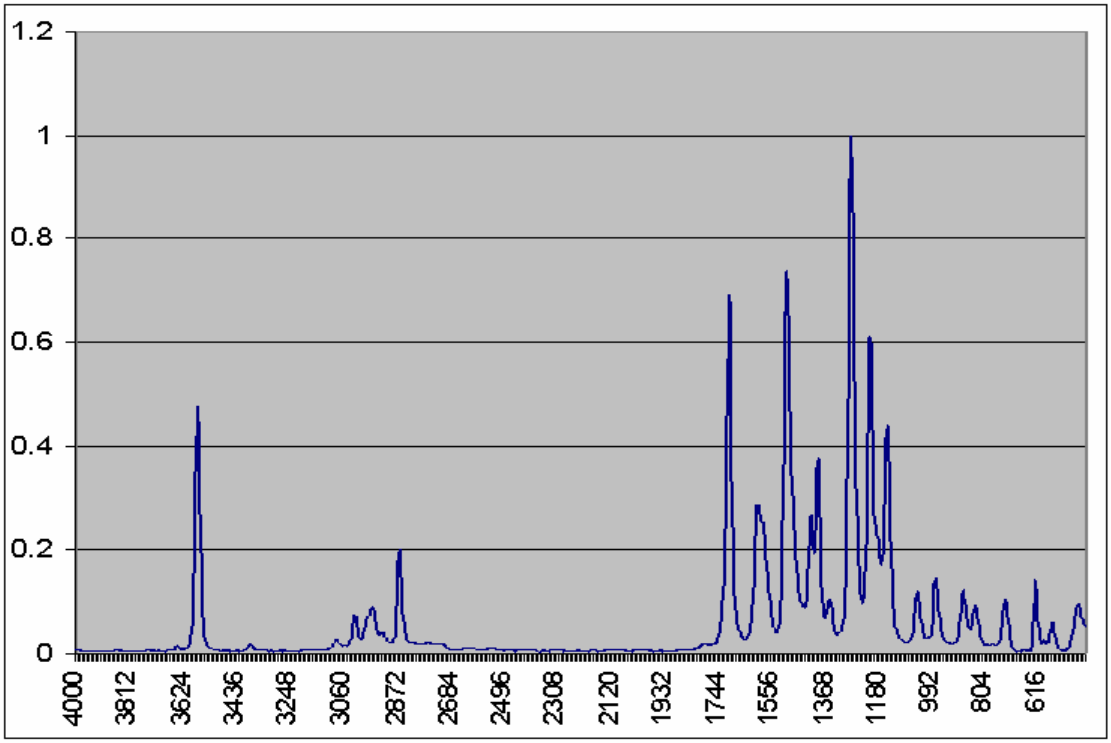



Figures 4.15 Infrared absorbance spectra of Butyric acid, 2, 2- dimethyl-, and Sarcosine, n-cis-9-octadecenoyl-,

**Tartaric acid, diethyl ester**




**Vanillin, 6-bromo-,**




Figures 4.16 Infrared absorbance spectra of Tartaric acid, diethyl ester and 6-bromo vanillin

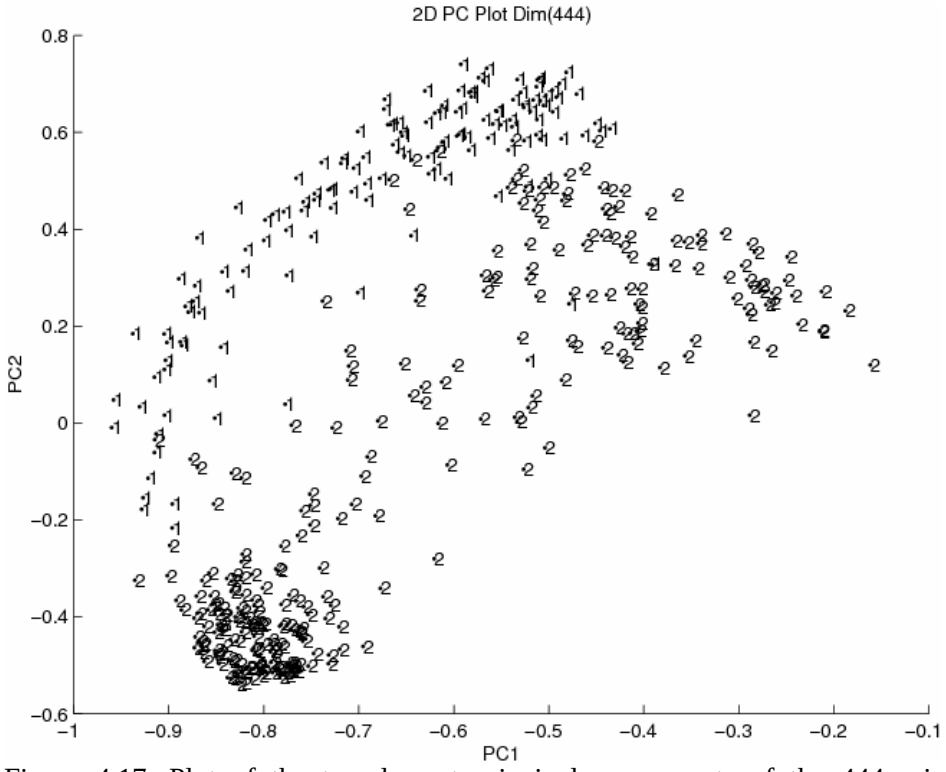



Figure 4.17. Plot of the two largest principal components of the 444-point IR spectra that comprised the training set. Each spectrum is represented as a point in the principal component plot. (1 = carboxylic acid and 2 = noncarboxylic acid).

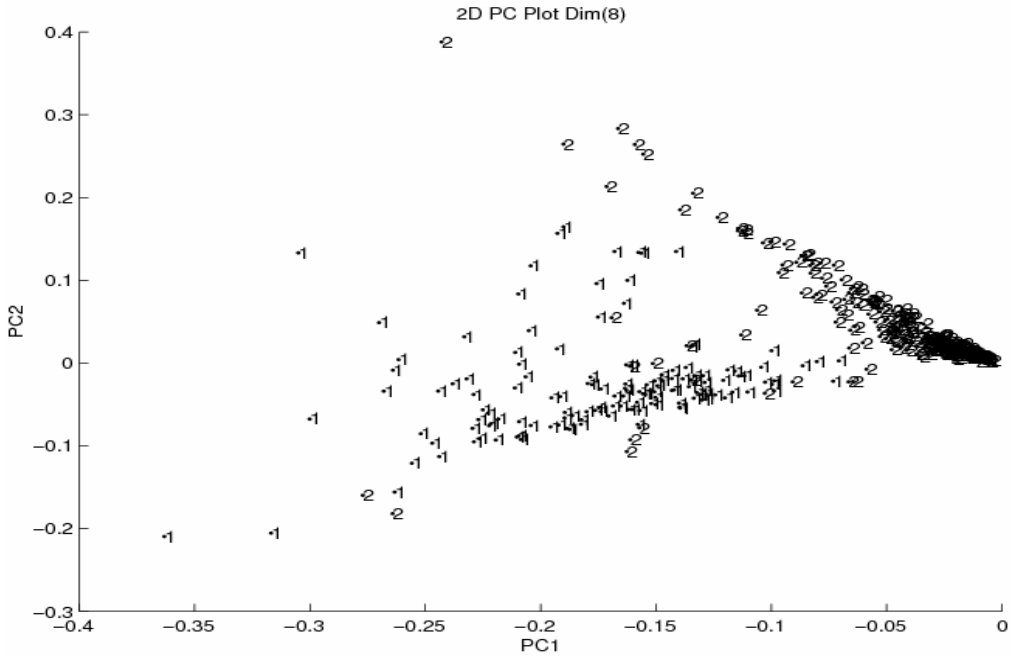



Figure 4.18. Plot of the two largest principal components of the 435 spectra and the 8 spectral features identified by the pattern recognition GA. Each spectrum is represented as a point in the principal component plot. (1 = carboxylic acid and 2 = noncarboxylic acid).

Figure 4.17 shows a plot of the two largest principal components of the 444-point IR spectra that comprised the training set. Figure 4.18 shows a plot of the 8 wavelengths identified by the pattern recognition GA. Each compound in the training set is represented as a point in the principal component plot (1 = carboxylic acid and 2 = noncarboxylic acid). The lack of clustering exhibited by the spectra in the principal component plot of the data on the basis of the presence or absence of the carboxylic acid functional group is evident.

The symmlet 4, symmlet 6 and symmlet 8 mother wavelets were applied to the IR spectral data to deconvolve overlapping spectral bands and to denoise the spectra. Figures 4.19 thru 4.24 show principal component plots of the 435 spectra and the wavelet coefficients identified by the pattern recognition GA for the different symmlet mother wavelets at various levels of decomposition. From an examination of these principal component plots, it is evident that symmlet 6 at the $10^{th}$ level decomposition gave the best results for the training set. Therefore, this wavelet was used to develop a search prefilter to detect carboxylic acids.

Figure 4.25 shows the projection of the spectra from the first prediction set (Table 4.4) onto a principal component map developed from the 435 training set spectra and the 53 wavelet coefficients identified by the pattern recognition GA. All of the training set samples were correctly classified. Three carboxylic acids in the prediction set did not lie in a region of the principal component map occupied by spectra possessing the same class label. Discriminant analysis was also used to develop a classifier from the training set data (435 spectra; each is described by 53 wavelet coefficients). LDA, QDA, K-NN, RDA, and back propagation neural networks were applied to this binary classification

problem. The results are summarized in Table 4.5 for both the training set and the first prediction set.

**10sym4**

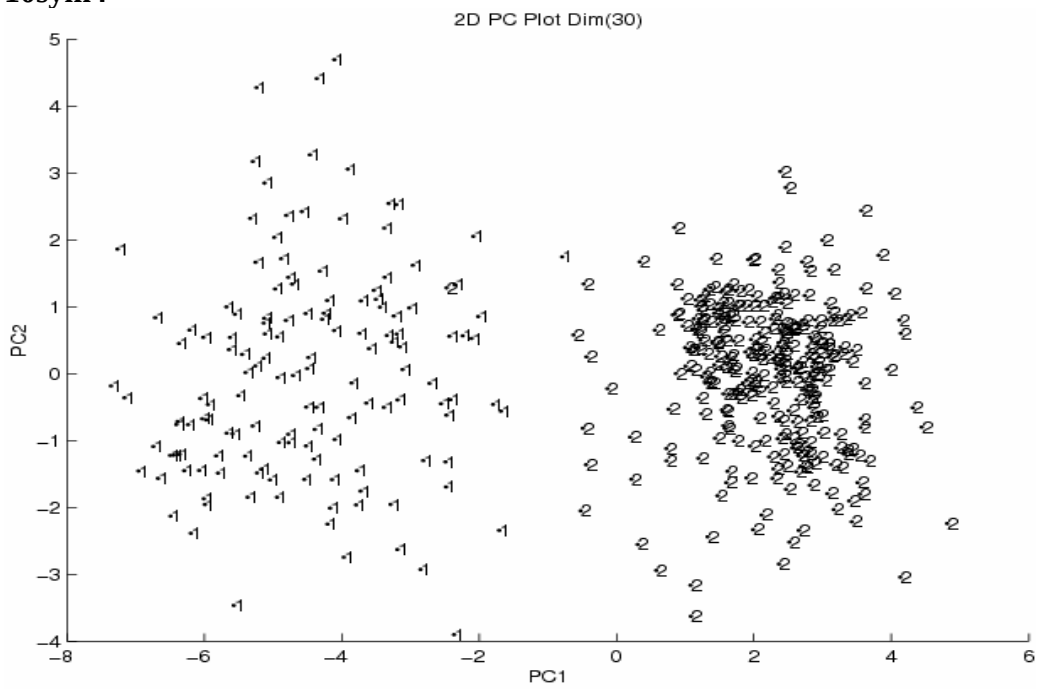



Figure 4.19. Plot of the two largest principal components of the 435 spectra and the 30 wavelet coefficients (10sym4) identified by the pattern recognition GA using PCKaNN with the Hopkins statistic as the fitness function. (1 = carboxylic acid and 2 = noncarboxylic acid)

**6sym6**

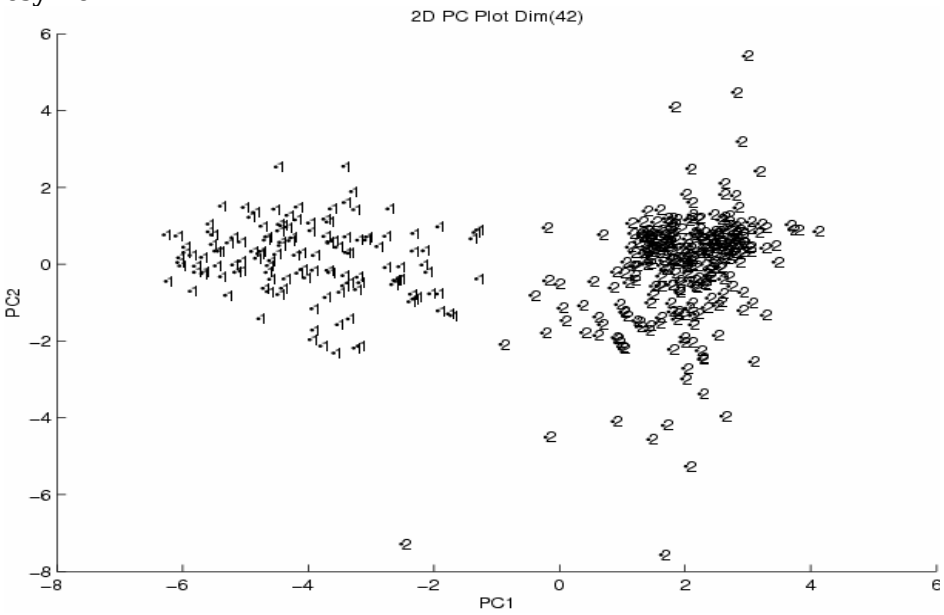



Figure 4.20. Plot of the two largest principal components of the 435 spectra and the 42 wavelet coefficients (6sym6) identified by the pattern recognition GA using PCKaNN with the Hopkins statistic as the fitness function. (1 = carboxylic acid and 2 = noncarboxylic acid)

**8sym6**

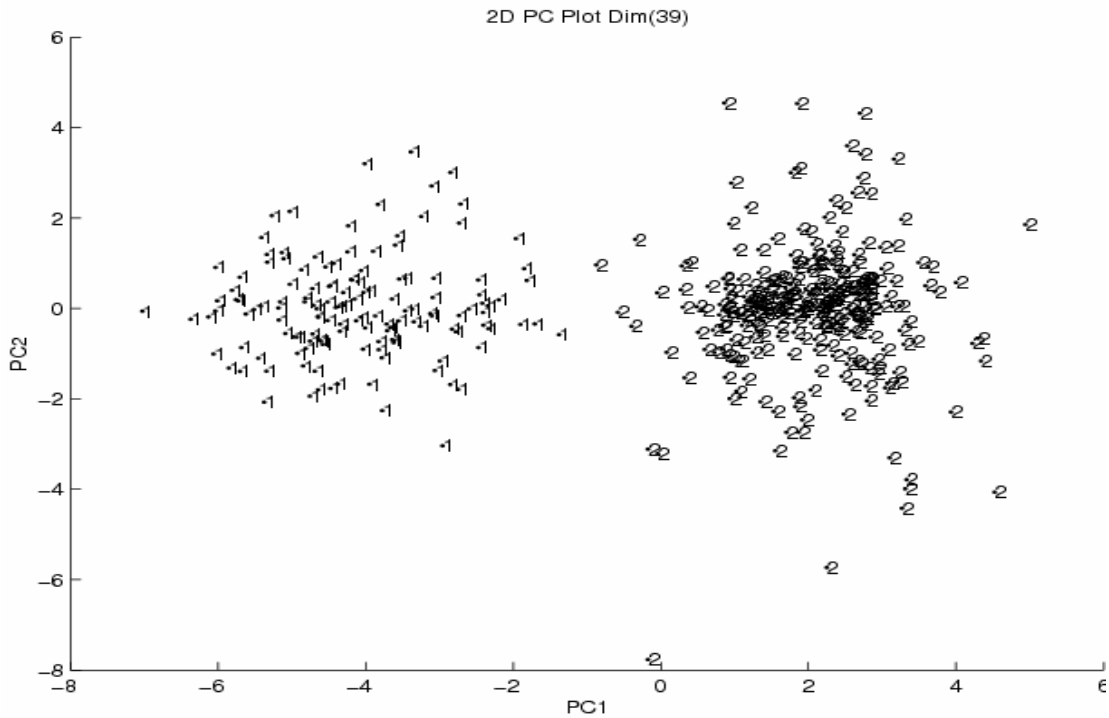



Figure 4.21. Plot of the two largest principal components of the 435 spectra and the 39 wavelet coefficients (8sym6) identified by the pattern recognition GA using PCKaNN with the Hopkins statistic as the fitness function. (1 = carboxylic acid and 2 = noncarboxylic acid)

**10sym6**

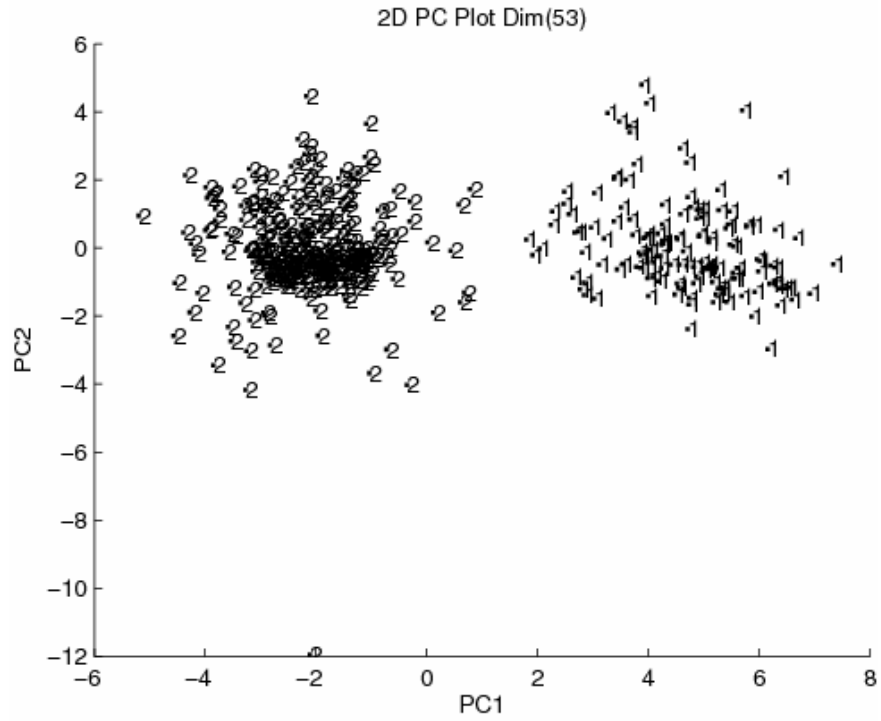



Figure 4.22. Plot of the two largest principal components of the 435 spectra and the 53 wavelet coefficients (10sym6) identified by the pattern recognition GA using PCKaNN with the Hopkins statistic as the fitness function. (1 = carboxylic acid and 2 = noncarboxylic acid)

**6sym8**

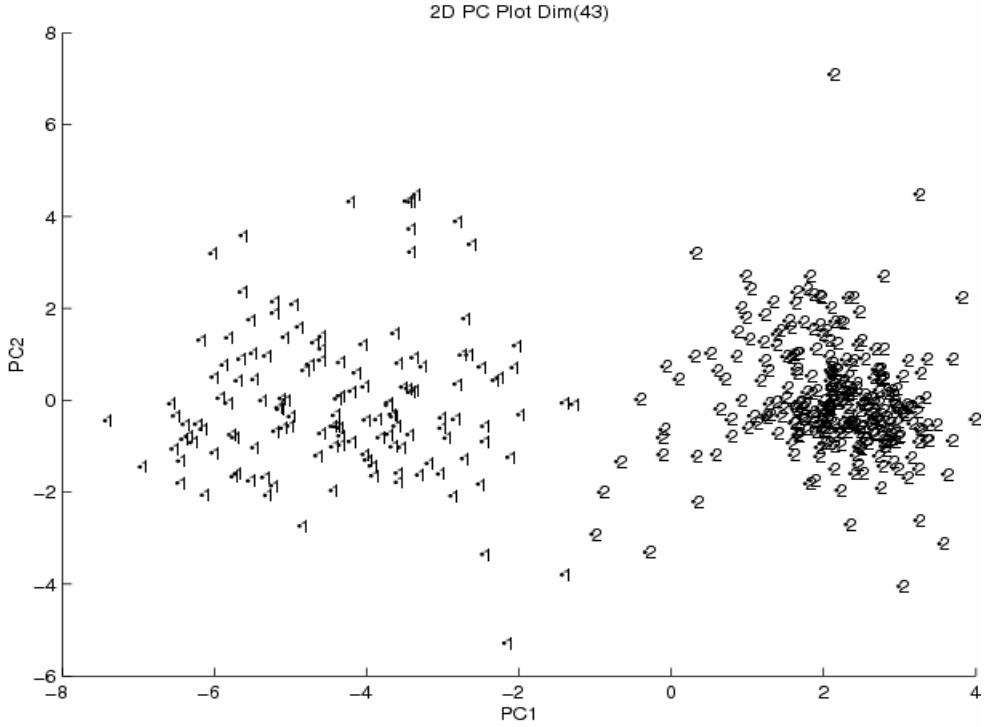



Figure 4.23. Plot of the two largest principal components of the 435 spectra and the 43 wavelet coefficients (6sym8) identified by the pattern recognition GA using PCKaNN with the Hopkins statistic as the fitness function. (1 = carboxylic acid and 2 = noncarboxylic acid)

**8sym8**

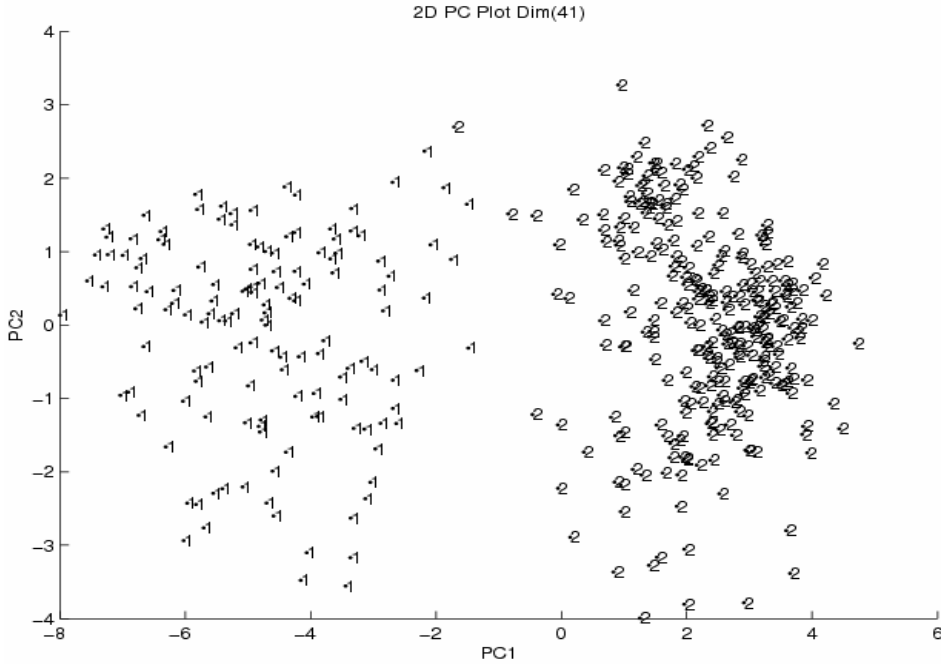



Figure 4.24. Plot of the two largest principal components of the 435 spectra and the 41 wavelet coefficients (8sym8) identified by the pattern recognition GA using PCKaNN with the Hopkins statistic as the fitness function. (1 = carboxylic acid and 2 = noncarboxylic acid)

**10sym6-Tset/Pset**

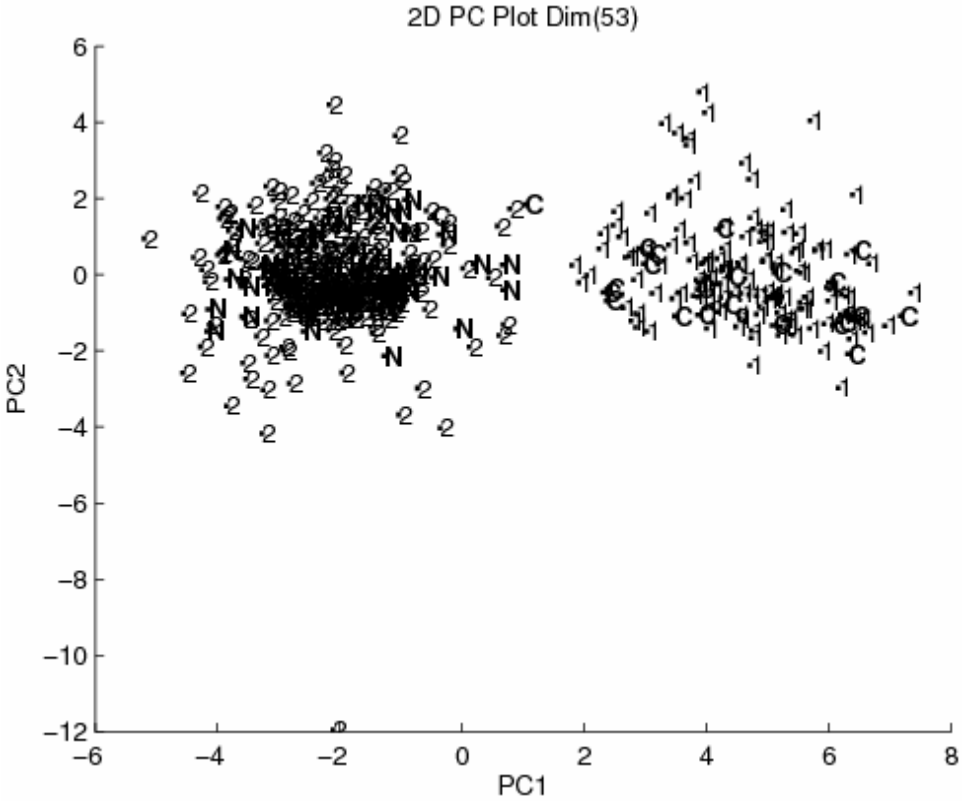



Figure 4.25. Plot of the two largest principal components of the 435 spectra and the 53 wavelet coefficients (10sym6) identified by the pattern recognition GA. 1 = carboxylic acid and 2 = noncarboxylic acid (training set) using PCKaNN with the Hopkins statistic as the fitness function. C = carboxylic acid and N = noncarboxylic acid (prediction set).

**Table 4.5. Discriminant Analysis Results for 10symmlet 6**

| | **LDA** | **QDA** | **RDA (auto)** | **BPN** | **KNN1** | **KNN3** | **KNN5** | **KNN7** |
|---|---|---|---|---|---|---|---|---|
| **Tset # wrong** | 0 | 1 | 0 | 0 | 7 | 5 | 3 | 2 |
| **Pset1 # wrong** | 4 | 5 | 3 | 3 | 4 | 4 | 3 | 3 |
| **Pset2 # wrong** | 33 | 62 | 38 | 31 | 36 | 35 | 34 | 34 |

The three prediction set samples misclassified by PCA were also the same three samples misclassified by LDA, QDA, RDA, back propagation neural networks, and K-NN. Spectra of these three misclassified compounds are shown in Figures 4.26 and 4.27. The spectrum of valeric acid appears to be distorted because of background correction problems, whereas the spectrum of o-toulic acid is of low quality because there does not appear to be a sufficient amount of sample used to collect the spectrum. Cyclopentaneacetic acid also suffers from spectral distortions due to problems associated with background correction.

To further test the predictive ability of the 53 wavelet coefficients and the discriminants associated with them, a second prediction set consisting of 264 carboxylic acids and 72 noncarboxylic acids was used to validate the proposed carboxylic acid search prefilter. The noncarboxylic acids were phosphates, alkenes, alkynes, and alkanes. The results from PCA are not shown because PCA does not scale up well when the number of samples in the prediction set is comparable to the number in the training set. Table 4.5 summarizes the results obtained for the second prediction set using LDA, QDA, RDA, back propagation neural networks, and K-NN. All of the misclassified spectra in the second prediction set are atypical carboxylic acids. This is apparent when examining the spectra of these compounds individually. In some cases a spectrum showed lots of $CO_2$, and in other cases they looked like Raman spectra. Many of the troublesome spectra are very noisy, and often the amount of sample used to generate the spectrum did not appear to be sufficient. Some of the spectra are of poor quality and in other cases the spectra are mislabeled. Suitable background correction is often an issue in these troublesome spectra as well as spectral distortions. Examples of mislabeled or

low quality IR spectra that were misclassified in the second prediction set are shown in Figures 4.28 and 4.29.

From the two studies described in this chapter, one can conclude that substructure specific search prefilters can be developed for IR library matching using the wavelet packet transform. The wavelet packet tree when combined with the genetic algorithm for pattern recognition analysis constitutes a general approach for analyzing and extracting information from spectroscopic data.

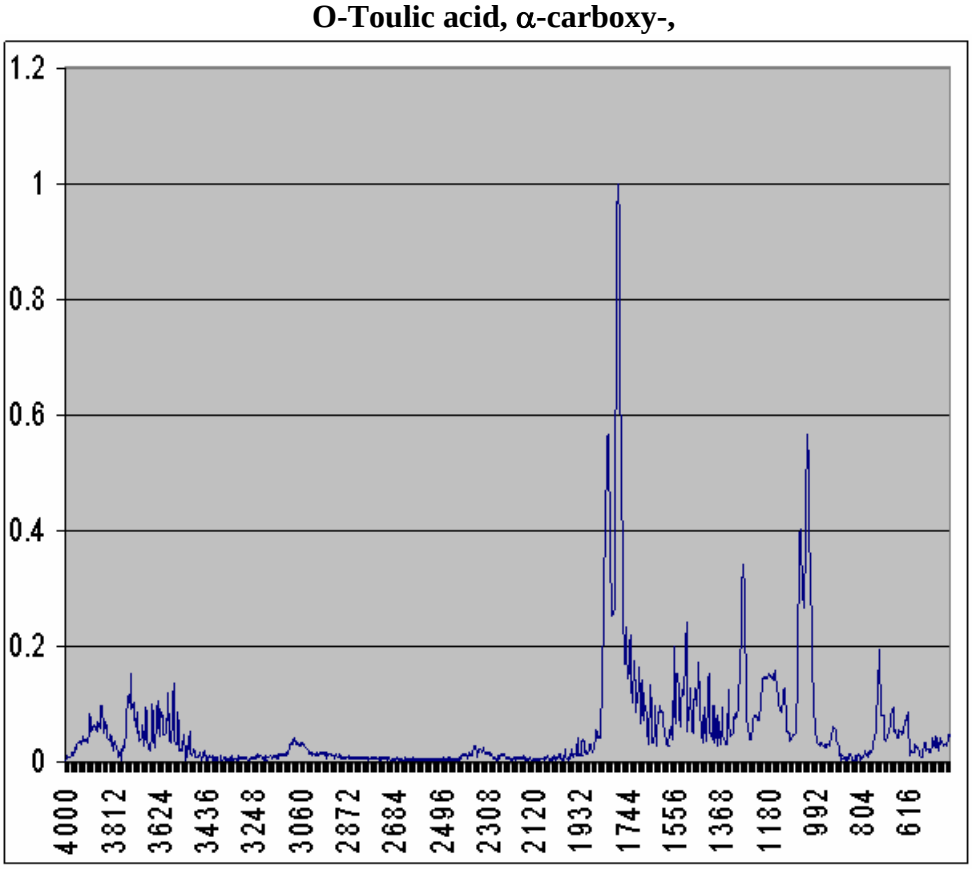
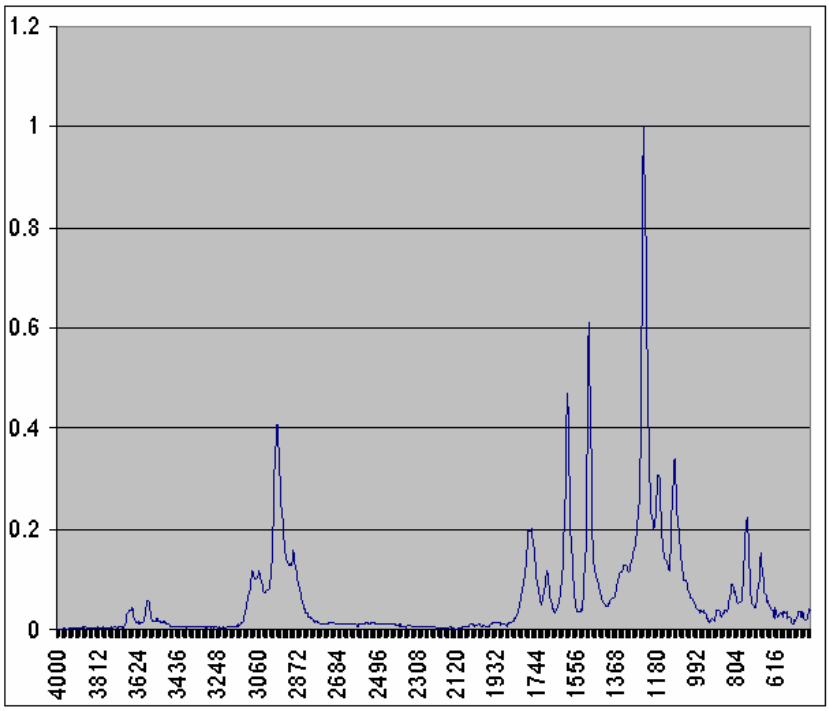



Figure 4.26. Low quality IR spectrum due to an insufficient amount of sample

**Cyclopentaneacetic acid, 1-hydroxy-a-phenoxy-,**




**Valeric acid, 5-amino**

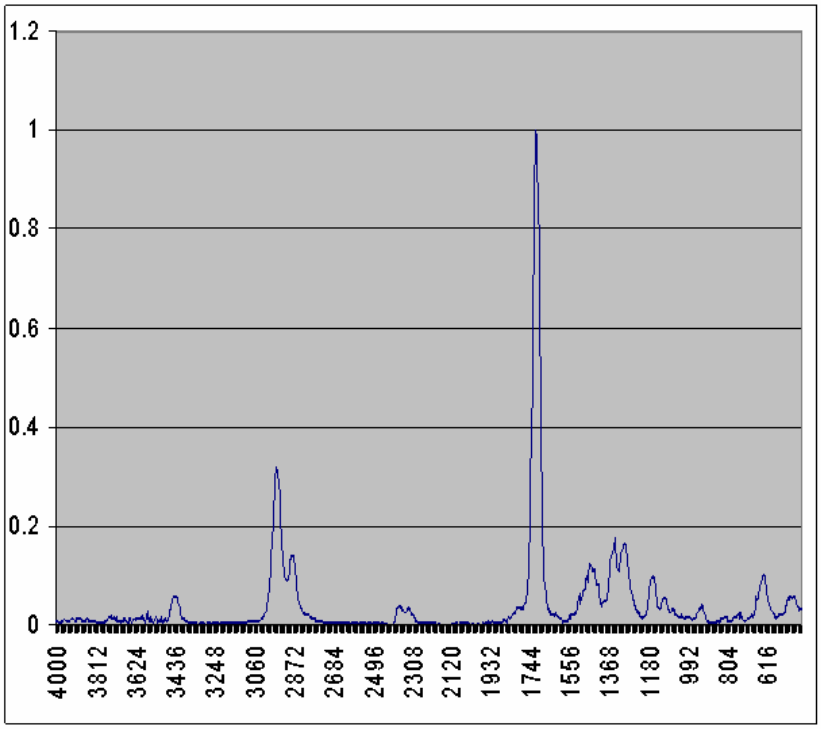
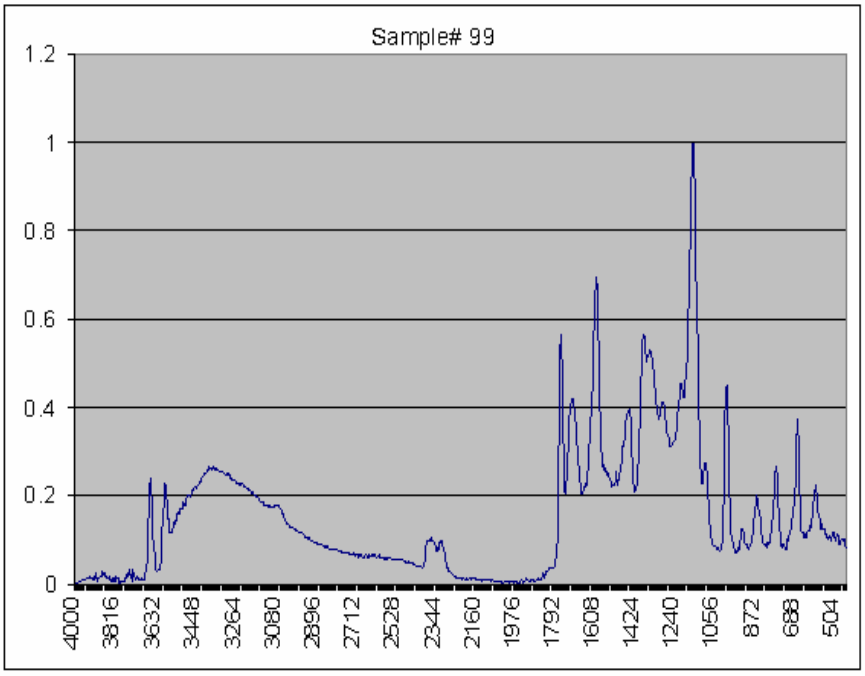



Figure 4.27. Spectra of both valeric acid and cyclopentaneacetic acid suffer from spectral distortions due to problems associated with background correction.

**Benzoic acid, 3, 4-dihydroxy**




**Acetic acid, o-hydroxyphenyl-,**

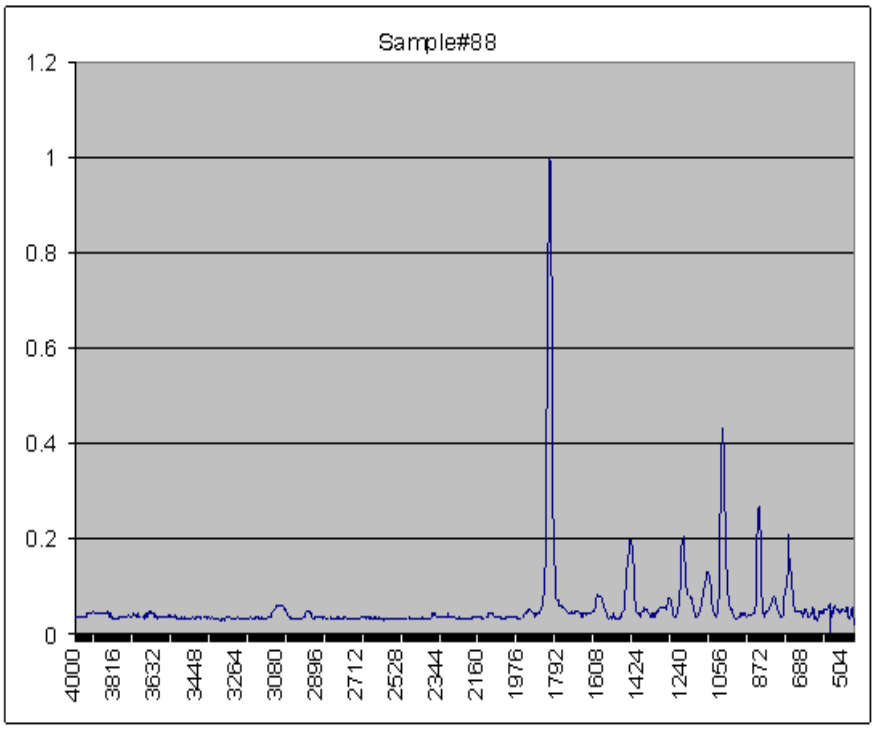
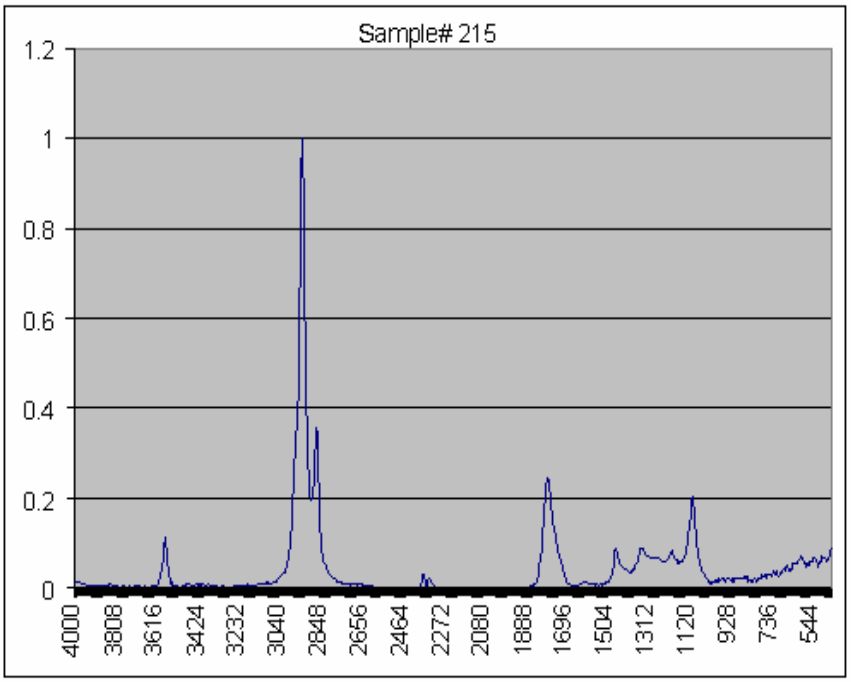



Figure 4.28. o-hydroxyphenyl acetic acid spectrum appears to be a Raman spectrum whereas the spectrum of 3, 4-dihydroxy benzoic acid has lots of $CO_2$ in it.

**Acetic acid, dodecylthio-,**




**Leucine, N-chloroacetyl-, L-minus-,**

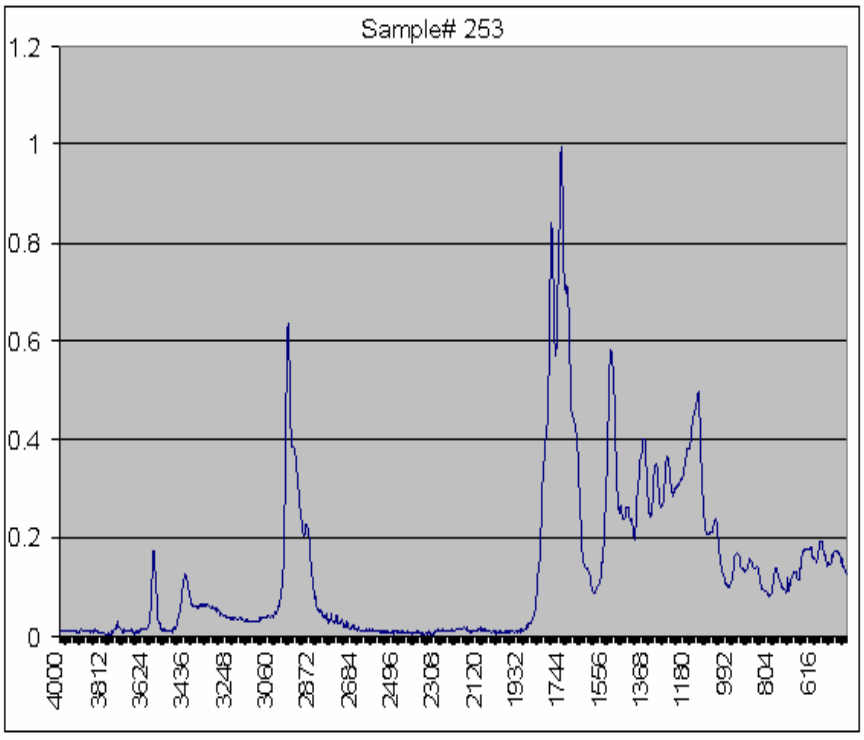



Figure 4.29. IR spectrum of N-chloroacetyl-, L-minus Leucine is noisy and is of low quality due to a small amount of sample. The dodecylthio-acetic acid spectrum is probably mislabeled.

## 4.5 Conclusions

Search prefilters can eliminate dissimilar spectra from a library search affording the user an opportunity to take advantage of more powerful but also more time-consuming search algorithms. Most infrared library search systems compare spectra by summing the squares of the difference between two spectra at every wave number. This generally makes for a fairly reliable identity search. If no compound that is identical to the unknown is present in the library, the results of the search then tend not to be useful. To improve the performance of a similarity search, the cross correlation function can be used to provide the best match between an unknown and the spectra in a hit list generated by a set of search prefilters. The cross correlation function has been shown to be able to differentiate between similar but nonidentical spectra and to correctly identify unknown spectra [72]. Although the cross correlation function is not computationally and statistically scalable as compared to more conventional search algorithms used in IR searching, it is suitable as a post searching method to rank probable matches that have been selected by a faster algorithm (i.e., a set of search prefilters). Furthermore, correlation based searching appears to be very sensitive to changes in peak shape and relative peak position making it sensitive to structural differences. Other advantages of the cross correlation function include the ability to compensate for variability in signal amplitude associated with larger peaks, insensitivity to instrumental noise, and the ability to correct for wave number imprecision. By combining search prefilters with library search algorithms that are more powerful but also more computationally intensive than the Euclidean distance, similarity searching will be feasible.

# CHAPTER 5

## Summary

In the preceding chapters, a basic methodology for analyzing complex multivariate data sets was described. A spectrum was represented as a point in a high dimensional measurement space. Pattern recognition methods were then used to investigate the properties of this vector space. The techniques found most useful in the studies reported here were graphical in nature. As such, they do not attempt to fit the data to a model; rather relationships are sought which provide definitions of similarity between diverse groups of data.

In a typical pattern recognition study, mapping and display methods such as PCA are first used to assess the structure of the data space. These methods also provide information about trends present in the data. Classification methods can then be used to further quantify these relationships. Linear discriminants have been found to be especially well suited for analyzing spectral data.

The basic premise underlying the pattern recognition methodology described in this thesis is that all data analysis methods will work well when the problem is simple. By identifying the appropriate features, a “hard” problem can be reduced to a “simple” one. Therefore, feature selection is an important step in a pattern recognition study. To ensure identification of all relevant features, it is best that a multivariate approach to feature

selection be employed. This approach should take into account the existence of redundancies in the data.

Feature selection can also lead to an understanding of the essential features that play an important role in governing the behavior of the system that is under investigation. It can identify those measurements, which are informative and those measurements, which are not informative, or perhaps it may reveal that all the measurements are informative since they are all correlated with each other. For all these reasons, it is the author's view that feature selection should be the principal focus of any new research on methodology involving supervised learning.

Pattern recognition methods operate with well defined criteria and attempt to extract useful information from raw data. If the limitations of these methods are not fully understood, the danger of misinterpretation and misuse of costly measurements are significant. It is the author's opinion that these techniques should be used to extend the ability of human pattern recognition. Hence, the approach suggested here relies heavily on graphics for the presentation of results. Although the computer can assimilate more numbers at a given time than can the scientist or engineer, it is the scientist or engineer who in the end must make the decisions and judgments.

# References


1. H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, J. Machine Learning Research, 2002, 2, 419-444.

2. L. M. Malmquist, R. R. Olsen, A. B. Hansen, O. Andersen, and J. H. Christensen, Journal of Chromatography, A, 2007, 1164(1-2), 262-270.

3. W. Chu and B.-H. Juang (Editors), Pattern Recognition in Speech and Language Processing, CRC Press, Electrical Engineering & Applied Signal Processing Series, Boca Raton, FL 2003.

4. P. C. Jurs, B. R. Kowalski, and T. L. Isenhour, Analytical Chemistry, 1969, 41(1), 21-27.

5. P. C. Jurs, B. R. Kowalski, and T. L. Isenhour, and C. N. Reilley, Analytical Chemistry, 1969, 41(6), 690-5.

6. B. R. Kowalski, P. C. Jurs, T. L. Isenhour, and C. N. Reilley, Analytical Chemistry, 1969, 41(6), 695-700.

7. P. C. Jurs, B. R. Kowalski, T. L. Isenhour, and C. N. Reilley, Analytical Chemistry, 1969, 41(14), 1949-53.

8. T. L. Isenhour and P. C. Jurs, Computers in Chemistry and Instrumentation, 1973, 1, 285-330.

9. E. Widjaja, Geok Hong Lim, and A. An, Analyst, 2008, 133(4), 493-498.

10. P. C. Chen, and C. C. Joyner, Journal of Physical Chemistry A (2006), 110(26), 7989-7993.

11. Y. Chen, Yi; Shang-Bin Zhu, Ming-Yong Xie, Shao-Ping Nie, Wei Li Liu, Gong Chan, Wang Xiao-Feng, and Yuan-Xing Wang, Analytica Chimica Acta, 2008, 623(2), 146-156.

12. Peishan Xie, Sibao Chen, Yi-Zeng Liang, Xianghong Wang, Runtao Tian and R. Upton, Journal of Chromatography, A (2006), 1112(1-2), 171-180.

13. M. R. Viant, E. S. Rosenblum, and R. S. Tjeerdema, Environmental Science and Technology 2003, 37(21), 4982-4989.

14. J. C. Lindon, E. Holmes, and J. K. Nicholson, Progress in Nuclear Magnetic Resonance Spectroscopy, 2001, 39(1), 1-40.

15. B. I. Guyon and A. Elisseeff, Journal of Machine Learning Research, 2003, 3, 1157-1182.

16. J. T. Tou, and R. C. Gonzalez, Pattern Recognition Principles, Addison Wesley, Reading, MA, 1974.

17. R. G. Brereton (Eds.), Multivariate Pattern Recognition in Chemometrics, Elsevier, Amsterdam, 1992.

18. K. Varmuza, Pattern Recognition in Chemistry, Springer-Verlag, Berlin, 1980.

19. A. M. van Nederkassel, M. Daszykowski, P. H. C. Eilers, Y. Vander Heyden, *Journal Chromatog*. *A*. 2006, 1118(2), 199-210.

20. O. M. Kvalheim, F. Brakstad, and Y-zeng Liang, 1994, Anal. Chem., 66, 43-51.

21. R. J. O. Torgrip, I. A. Lewis, K. M. Aberg, E. Alm, I. Schuppe-Koistinen, and J. Lindberg, Metabolomics, 2008, 4(2), 114-121.

22. I.P. Joliffe, *Principal Component Analysis*, Springer-Verlag Press, New York, 1986.

23. J. Edward Jackson, *A User's Guide to Principal Components*, Wiley Interscience, New York, 1991.

24. M. A. Sharaf, D. L. Illman, and B. R. Kowalski, Chemometrics, John Wiley & Sons, New York 1986, p. 216.

25. G. Golub, and C. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, 1971.

26. G. J. McLachlan, Discriminant Analysis and Statistical Pattern Recognition, John Wiley & Son, NY 1992.

27. W. L. Grogan, and W. W. Wirth, W.W. *Proceedings of the Biological Society of Washington1981,* 94, 1279-1305.

28. R. A. Fisher, Ann. Eugen., 1936, 7, 179-186.

29. Mike James, Classification Algorithms, John Wiley & Sons, NY, 1985, p.206-209.

30. B. K. Lavine, P. C. Jurs, D. R. Henry, R. K. Vander Meer, J. A. Pino, and J. E. McMurry, Chemolab, 1988, 3, 79-89.

31. P. C. Jurs, B. K. Lavine, and T. R. Stouch, "Pattern Recognition Studies of Complex Chromatographic Data Sets," NBS Journal of Research, 90, 543 (1985)

32. M. Stone, *J. R. Stat. Soc*., 1974, 36, 111-121.

33. I. E. Frank and J. H. Friedman, *J. Chem*., 1989, 3, 463-475.

34. B. R. Kowalski and S. Wold, "Pattern Recognition in Chemistry," in Classification, Pattern Recognition and Reduction of Dimensionality, P. R. Krishnaiah and L. N. Kanal, Eds., North Holland, Amsterdam, 1982.

35. S. Wold and M. Sjostrom, SIMCA, A Method for Analyzing Chemical Data in Terms of Similarity and Analogy, in Chemometrics, Theory and Application, B. R. Kowalski, Ed., American Chemical Society, Washington, DC, 1977.

36. M. Barker and W. Rayens, *J. Chem*. 2003, 17(3), 166-173.

37. P. T Funke, E. R. Malinowski, D. E. Martire, and L. Z. Pollara, Separation Science, 1966, 1(6), 661-76.

38. D.L.Duewer and B.R.Kowalski, Anal.Chem, 1975, 47, 526-532.

39. T. L. Isenhour and P. C. Jurs, Anal. Chem. 1971, 43, 20A-35A.

40. B. R. Kowalski, and C. F. Bender, J. Am. Chem. Soc. 1973, 95, 686-695.

41. M. Mitchell, Introduction to Genetic Algorithms, MIT Press, Cambridge, MA, 1998.

42. Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, 3rd Edition, Springer Verlag, Berlin, 1995

43. J. H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, 6th Printing, MIT Press, Cambridge, MA, 2001.

44. D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Publishing Company, Reading, MA 1989.

45. W. D. Hillis, Physica D., 1990, 42, 228-234.

46. B. K. Lavine and A. J. Moores, "Genetic Algorithms for Pattern Recognition Analysis and Fusion of Sensor Data," in Pattern Recognition, Chemometrics, and Imaging for Optical Environmental Monitoring, K. Siddiqui and D. Eastwood (Eds.), Proc. Soc. Photo-Opt. Instrum. Eng., 1999, pp. 103-112.

47. B. K. Lavine, A. J. Moores, H. T. Mayfield, and A. Faruque, Microchem. J., 1999, 61, 69-79.

48. B. K. Lavine, J. Ritter, A. J. Moores, M. Wilson, A. Faruque, and H. T. Mayfield, Anal. Chem., 2000, 72(2), 423-430.

49. B. K. Lavine, A. Vesanen, D. M. Brzozowski, and H. T. Mayfield "Authentication of Fuel Standards using Gas Chromatography/Pattern Recognition Techniques," Anal Letters, 2001, 34(2), 281- 294

50. B. K. Lavine, C. E. Davidson, and A. J. Moores, Chemometrics & Intelligent Laboratory Instrumentation, 2002, 60(1), 161-171.

51. B. K. Lavine, C. E. Davidson, and A. J. Moores, Vibrational Spectroscopy, 2002, 28(1), 83-95.

52. B. K. Lavine, C. E. Davidson, Robert K. Vander Meer, S. Lahav, V. Soroker, and A. Hefetz, Chemometrics & Intelligent Laboratory Instrumentation, 2003, 66(1), 51-62.

53. B. R. Kowalski and C. F. Bender, J. Am. Chem. Soc., 1972, 94, 5632-5639.

54. A. K. Jain and R. C. Dubes, Algorithms for Clustering Data, Prentice Hall, Englewood Cliffs, NJ 1988.

55. B. K. Lavine and W. T. Rayens, Journal of Chemometrics, 2008, submitted.

56. P. J. Huber, Annals of Statistics, 1985, 13(2), 435-475.

57. I. A. Hyvarinen, J. Karhunen and E. Oja, Independent Component Analysis, John Wiley & Sons, New York, 2001.

58. J. Friedman**,** J. R. Statistic. Soc., A**.** 1987, 150(1), 26-27.

59. C.E. Davidson, Genetic Algorithms for Data Mining and Multivariate Data Analysis, PhD Thesis, Clarkson University, December 2003.

60. B. K. Lavine, C. E. Davidson, and W. T. Rayens, Combinatorial Chemistry & High Throughput Screening, 2004, 7, 115-131.

61. V. Vapnik, Statistical Learning Theory, Wiley, NY, 1998.

62. L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Wadsworth and Brooks/Cole: Monterrey, CA, 1984.

63. A. J. Panshin and C. de Zeeuw, Textbook of Wood Technology: Structure, Identification, Properties, and use of the Commercial Woods of the United States and Canada, McGraw-Hill, New York (1980).

64. B. K. Lavine, C. E. Davidson, A. J. Moores, and P. R. Griffiths, "Raman Spectroscopy and Genetic Algorithms for the Classification of Wood Types," Applied Spectroscopy, 2001, 55(8), 960-966.

65. M. Ruprecht and J. T. Clerc, J. Chem. Inf. Comput. Sci., 1985, 25, 241-244.

66. S. R. Lowry, D. A. Huppler, and C. R. Anderson, J. Chem. Inf. Comput. Sci., 1985, 25, 235-241.

67. J.C.W.Bink and H. A. Van'T Klooster, Anal. Chim. Acta., 1983, 150, 53-59.

68. L. Domokos, I. Frank, G.Matolcsy, and G. Jalsovszky, Anal. Chim. Acta., 1983, 154, 181-189.

69. Amara Graps, IEEE, 1995, 2(2), 50-61..

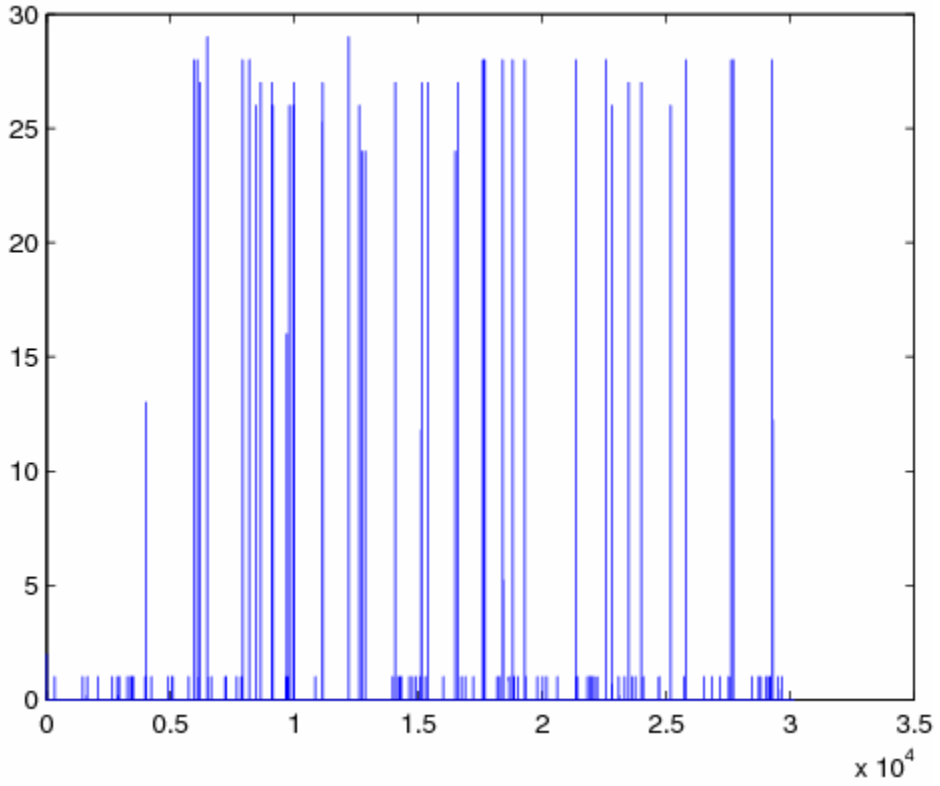
70. Kyoji Kawagoe, Tomohiro Ueda: A Similarity Search Method of Time Series Data with Combination of Fourier and Wavelet Transforms in Proceedings of the Ninth International Symposium on Temporal Representation and Reasoning, IEEE, 2002, pp. 86-92.

71. MATLAB R 2006a Wavelet Toolbox (Natick, MA)

72. L. A. Powell and G. M. Hieftje, Anal. Chim. Acta., 1978, 100, 313-327

# Appendix I

## Gene Expressions Identified by the Pattern Recognition GA




Frequency histogram of the best features selected by the pattern recognition GA during each generation from all of the runs performed in the 90%/10% validation study to simulate the ability of a classifier to predict the class membership (reoccurrence versus no reoccurrence) of an unknown biopsy sample using segmented cross validation.

**List of the Features Most Frequently Selected by the Pattern Recognition GA**

| ID | Description of Genes Most Frequently Selected |
|---|---|
| 4017 | Consensus includes gb:AK025007.1 /DEF=Homo sapiens cDNA: FLJ21354 fis, clone COL02773. /FEA=mRNA /DB_XREF=gi:10437440 /UG=Hs.283707 Homo sapiens cDNA: FLJ21354 fis, clone COL02773 |
| 5941 | gb:NM_013316.1 /DEF=Homo sapiens CCR4-NOT transcription complex, subunit 4 (CNOT4), mRNA. /FEA=mRNA /GEN=CNOT4 /PROD=CCR4-NOT transcription complex, subunit 4 /DB_XREF=gi:7019466 /UG=Hs.20423 CCR4-NOT transcription complex, subunit 4 /FL=gb:U71267.1 gb:NM_013316.1 |
| 6093 | Consensus includes gb:AV702789 /FEA=EST /DB_XREF=gi:10719119 /DB_XREF=est:AV702789 /CLONE=ADBAGG04 /UG=Hs.164595 ESTs |
| 6189 | gb:NM_013300.1 /DEF=Homo sapiens protein predicted by clone 23733 (HSU79274), mRNA. /FEA=mRNA /GEN=HSU79274 /PROD=protein predicted by clone 23733 /DB_XREF=gi:9558740 /UG=Hs.150555 protein predicted by clone 23733 /FL=gb:U79274.1 gb:NM_013300.1 |
| 6476 | Consensus includes gb:AC004542 /DEF=Homo sapiens PAC clone RP3-430N8 from 22q12.1-qter /FEA=CDS /DB_XREF=gi:3041846 /UG=Hs.35276 KIAA0852 protein |
| 7921 | Consensus includes gb:L24521.1 /DEF=Human transformation-related protein mRNA, 3 end. /FEA=mRNA /PROD=transformation-related protein /DB_XREF=gi:403459 /UG=Hs.300705 Human transformation-related protein mRNA, 3 end |
| 8184 | Consensus includes gb:AA424065 /FEA=EST /DB_XREF=gi:2103026 /DB_XREF=est:zv80e10.s1 /CLONE=IMAGE:759978 /UG=Hs.115467 ESTs |
| 8456 | Consensus includes gb:AI809870 /FEA=EST /DB_XREF=gi:5396436 /DB_XREF=est:wf59c03.x1 /CLONE=IMAGE:2359876 /UG=Hs.66170 HSKM-B protein |
| 8640 | Consensus includes gb:AK000822.1 /DEF=Homo sapiens cDNA FLJ20815 fis, clone ADSE01038, highly similar to AJ007398 Homo sapiens mRNA for PBK1 protein. /FEA=mRNA /DB_XREF=gi:7021134 /UG=Hs.85963 DKFZP564M182 protein |
| 9097 | Consensus includes gb:BF110363 /FEA=EST /DB_XREF=gi:10940053 /DB_XREF=est:7n52a10.x1 /CLONE=IMAGE:3568050 /UG=Hs.80248 RNA-binding protein gene with multiple splicing |
| 9126 | Consensus includes gb:AW029203 /FEA=EST /DB_XREF=gi:5887959 /DB_XREF=est:wx07c09.x1 /CLONE=IMAGE:2542960 /UG=Hs.191952 ESTs |
| 9682 | Consensus includes gb:AL390736 /DEF=Human DNA sequence from clone RP11-209J19 on chromosome 13 Contains ESTs, STSs and GSSs. Contains the gene for the GW112 protein with two isoforms (GW112 and KIAA4294) /FEA=mRNA /DB_XREF=gi:11182238 /UG=Hs.273321 differentially expressed in hematopoietic lineages |
| 9742 | Consensus includes gb:AA528138 /FEA=EST /DB_XREF=gi:2270207 /DB_XREF=est:nj15d05.s1 /CLONE=IMAGE:986409 /UG=Hs.179520 Homo sapiens, clone MGC:10702, mRNA, complete cds |

9792 Consensus includes gb:AI090487 /FEA=EST /DB_XREF=gi:3429546 /DB_XREF=est:qa64d12.x1 /CLONE=IMAGE:1691543 /UG=Hs.168325 ESTs, Moderately similar to ALU1_HUMAN ALU SUBFAMILY J SEQUENCE CONTAMINATION WARNING ENTRY H.sapiens

9960 Consensus includes gb:BF221850 /FEA=EST /DB_XREF=gi:11129027 /DB_XREF=est:7p37f06.x1 /CLONE=IMAGE:3648131 /UG=Hs.122365 ESTs

9987 gb:M65254.1 /DEF=Protein phosphatase 2A 65 kDa regulatory subunit-beta mRNA, complete cds. /FEA=mRNA /GEN=SNRPEP1 /PROD=protein phosphatase-2A regulatory subunit-beta /DB_XREF=gi:189429 /UG=Hs.108705 protein phosphatase 2 (formerly 2A), regulatory subunit A (PR 65), beta isoform /FL=gb:NM_002716.1 gb:AF163473.1 gb:M65254.1 gb:AF087438.1

11132 Consensus includes gb:AA908777 /FEA=EST /DB_XREF=gi:3048182 /DB_XREF=est:ol06b06.s1 /CLONE=IMAGE:1522643 /UG=Hs.50158 ESTs

12189 gb:NM_021977.1 /DEF=Homo sapiens solute carrier family 22 (extraneuronal monoamine transporter), member 3 (SLC22A3), mRNA. /FEA=mRNA /GEN=SLC22A3 /PROD=solute carrier family 22 (extraneuronalmonoamine transporter), member 3 /DB_XREF=gi:11415037 /UG=Hs.81086 solute carrier family 22 (extraneuronal monoamine transporter), member 3 /FL=gb:NM_021977.1

12627 gb:NM_005518.1 /DEF=Homo sapiens 3-hydroxy-3-methylglutaryl-Coenzyme A synthase 2 (mitochondrial) (HMGCS2), mRNA. /FEA=mRNA /GEN=HMGCS2 /PROD=3-hydroxy-3-methylglutaryl-Coenzyme A synthase 2(mitochondrial) /DB_XREF=gi:5031750 /UG=Hs.59889 3-hydroxy-3-methylglutaryl-Coenzyme A synthase 2 (mitochondrial) /FL=gb:NM_005518.1

12726 Consensus includes gb:AF143884.1 /DEF=Homo sapiens clone IMAGE:121558 mRNA sequence. /FEA=mRNA /PROD=unknown /DB_XREF=gi:4895028 /UG=Hs.145643 Homo sapiens clone IMAGE:121558 mRNA sequence

12867 Consensus includes gb:AB033044.1 /DEF=Homo sapiens mRNA for KIAA1218 protein, partial cds. /FEA=mRNA /GEN=KIAA1218 /PROD=KIAA1218 protein /DB_XREF=gi:6330582 /UG=Hs.114012 KIAA1218 protein

14059 gb:AF128846.1 /DEF=Homo sapiens indolethylamine N-methyltransferase (INMT) mRNA, INMT-1 allele, complete cds. /FEA=mRNA /GEN=INMT /PROD=indolethylamine N-methyltransferase /DB_XREF=gi:6580814 /UG=Hs.204038 indolethylamine N-methyltransferase /FL=gb:NM_006774.2 gb:AF128846.1 gb:AF128847.1

15130 Consensus includes gb:BE220399 /FEA=EST /DB_XREF=gi:8907717 /DB_XREF=est:hv71g09.x1 /CLONE=IMAGE:3178912 /UG=Hs.323836 ESTs, Weakly similar to alternatively spliced product using exon 13A H.sapiens

15401 gb:NM_001275.2 /DEF=Homo sapiens chromogranin A (parathyroid secretory protein 1) (CHGA), mRNA. /FEA=mRNA /GEN=CHGA /PROD=chromogranin A /DB_XREF=gi:10800418 /UG=Hs.172216 chromogranin A (parathyroid secretory protein 1) /FL=gb:NM_001275.2 gb:BC001059.1 gb:J03483.1 gb:J03915.1

16485 Consensus includes gb:AI627965 /FEA=EST /DB_XREF=gi:4664765 /DB_XREF=est:ty83c12.x1 /CLONE=IMAGE:2285686 /UG=Hs.301732 hypothetical protein MGC5306

16595 Consensus includes gb:AK022874.1 /DEF=Homo sapiens cDNA FLJ12812 fis, clone NT2RP2002498. /FEA=mRNA /DB_XREF=gi:10434520 /UG=Hs.108779 DKFZP586E1519 protein

17621 Consensus includes gb:N63953 /FEA=EST /DB_XREF=gi:1211782 /DB_XREF=est:yz81b03.s1 /CLONE=IMAGE:289421 /UG=Hs.243662 ESTs

17676 gb:NM_003272.1 /DEF=Homo sapiens transmembrane 7 superfamily member 1 (upregulated in kidney) (TM7SF1), mRNA. /FEA=mRNA /GEN=TM7SF1 /PROD=transmembrane 7 superfamily member 1(upregulated in kidney) /DB_XREF=gi:4507544 /UG=Hs.15791 transmembrane 7 superfamily member 1 (upregulated in kidney) /FL=gb:AF027826.1 gb:NM_003272.1

18409 gb:NM_003122.1 /DEF=Homo sapiens serine protease inhibitor, Kazal type 1 (SPINK1), mRNA. /FEA=mRNA /GEN=SPINK1 /PROD=serine protease inhibitor, Kazal type 1 /DB_XREF=gi:4507178 /UG=Hs.181286 serine protease inhibitor, Kazal type 1 /FL=gb:NM_003122.1

18800 Consensus includes gb:AA868380 /FEA=EST /DB_XREF=gi:2963825 /DB_XREF=est:ak41e02.s1 /CLONE=IMAGE:1408538 /UG=Hs.126914 KIAA1430 protein

19282 gb:AF130059.1 /DEF=Homo sapiens clone FLB5634 PRO1477 mRNA, complete cds. /FEA=mRNA /PROD=PRO1477 /DB_XREF=gi:11493424 /UG=Hs.99858 ribosomal protein L7a /FL=gb:AF130059.1

21349 Consensus includes gb:BF345728 /FEA=EST /DB_XREF=gi:11293323 /DB_XREF=est:602019377F1 /CLONE=IMAGE:4154971 /UG=Hs.297962 ESTs

21351 Consensus includes gb:AW297731 /FEA=EST /DB_XREF=gi:6704367 /DB_XREF=est:UI-H-BW0-aiy-a-04-0-UI.s1 /CLONE=IMAGE:2730894 /UG=Hs.123310 ESTs

22560 gb:NM_018977.1 /DEF=Homo sapiens neuroligin 3 (NLGN3), mRNA. /FEA=mRNA /GEN=NLGN3 /PROD=neuroligin 3 /DB_XREF=gi:9506786 /UG=Hs.47320 neuroligin 3 /FL=gb:AF217411.1 gb:NM_018977.1

22815 Consensus includes gb:AI671488 /FEA=EST /DB_XREF=gi:4851219 /DB_XREF=est:wc30h04.x1 /CLONE=IMAGE:2316727 /UG=Hs.65082 ESTs

23490 Cluster Incl. U79256:Human clone 23719 mRNA sequence /cds=UNKNOWN /gb=U79256 /gi=1710209 /ug=Hs.80305 /len=1196

23990 Consensus includes gb:AI922972 /FEA=EST /DB_XREF=gi:5659022 /DB_XREF=est:wn66h07.x1 /CLONE=IMAGE:2450461 /UG=Hs.196073 ESTs

25180 gb:AF098641.1 /DEF=Homo sapiens CD44 isoform RC (CD44) mRNA, complete cds. /FEA=mRNA /GEN=CD44 /PROD=CD44 isoform RC /DB_XREF=gi:3832517 /UG=Hs.306278 Homo sapiens CD44 isoform RC (CD44) mRNA, complete cds /FL=gb:AF098641.1

25779 Consensus includes gb:AI025103 /FEA=EST /DB_XREF=gi:3240716 /DB_XREF=est:ov40d05.x1 /CLONE=IMAGE:1639785 /UG=Hs.54699 ESTs

27616 gb:NM_003155.1 /DEF=Homo sapiens stanniocalcin 1 (STC1), mRNA. /FEA=mRNA /GEN=STC1 /PROD=stanniocalcin 1 /DB_XREF=gi:4507264 /UG=Hs.25590 stanniocalcin 1 /FL=gb:U46768.1 gb:U25997.1 gb:NM_003155.1

27708 gb:NM_006157.1 /DEF=Homo sapiens nel (chicken)-like 1 (NELL1), mRNA. /FEA=mRNA /GEN=NELL1 /PROD=nel (chicken)-like 1 /DB_XREF=gi:5453763 /UG=Hs.21602 nel (chicken)-like 1 /FL=gb:D83017.1 gb:NM_006157.1

29281 Consensus includes gb:AU156625 /FEA=EST /DB_XREF=gi:11018146 /DB_XREF=est:AU156625 /CLONE=PLACE1003936 /UG=Hs.296738 Homo sapiens cDNA FLJ13489 fis, clone PLACE1003936

# Appendix II

## Results of Segmented Cross Validation for Reoccurrence of Cancer

**Segmented Cross Validation (90% Training Set/10% Validation Set)**

| Classification method | Average Tset % classification | | | Average Pset % classification | | |
|---|---|---|---|---|---|---|
| | Normal | Modified Hopkins | Hopkins | Normal | Modified Hopkins | Hopkins |
| **LDA** | 96 | 97 | 97 | 97 | 97 | 98 |
| **QDA** | 95 | 97 | 96 | 96 | 97 | 97 |
| **RDA(auto)** | 97 | 98 | 97.25 | 98 | 99 | 98 |
| **1-NN** | 92 | 94 | 95 | 96 | 97 | 95 |

**Segmented Cross Validation (50% Training Set/50% Validation Set)**

| Classification method | Average Tset % classification | | | Average Pset % classification | | |
|---|---|---|---|---|---|---|
| | Normal | Modified Hopkins | Hopkins | Normal | Modified Hopkins | Hopkins |
| **LDA** | 99 | 99 | 100 | 40 | 51 | 43 |
| **QDA** | 100 | 100 | 100 | 0 | 0 | 0 |
| **RDA(auto)** | 100 | 100 | 98 | 50 | 51 | 54 |
| **1-NN** | 53 | 58 | 46 | 59 | 53 | 61 |

**Pset1-Normal**

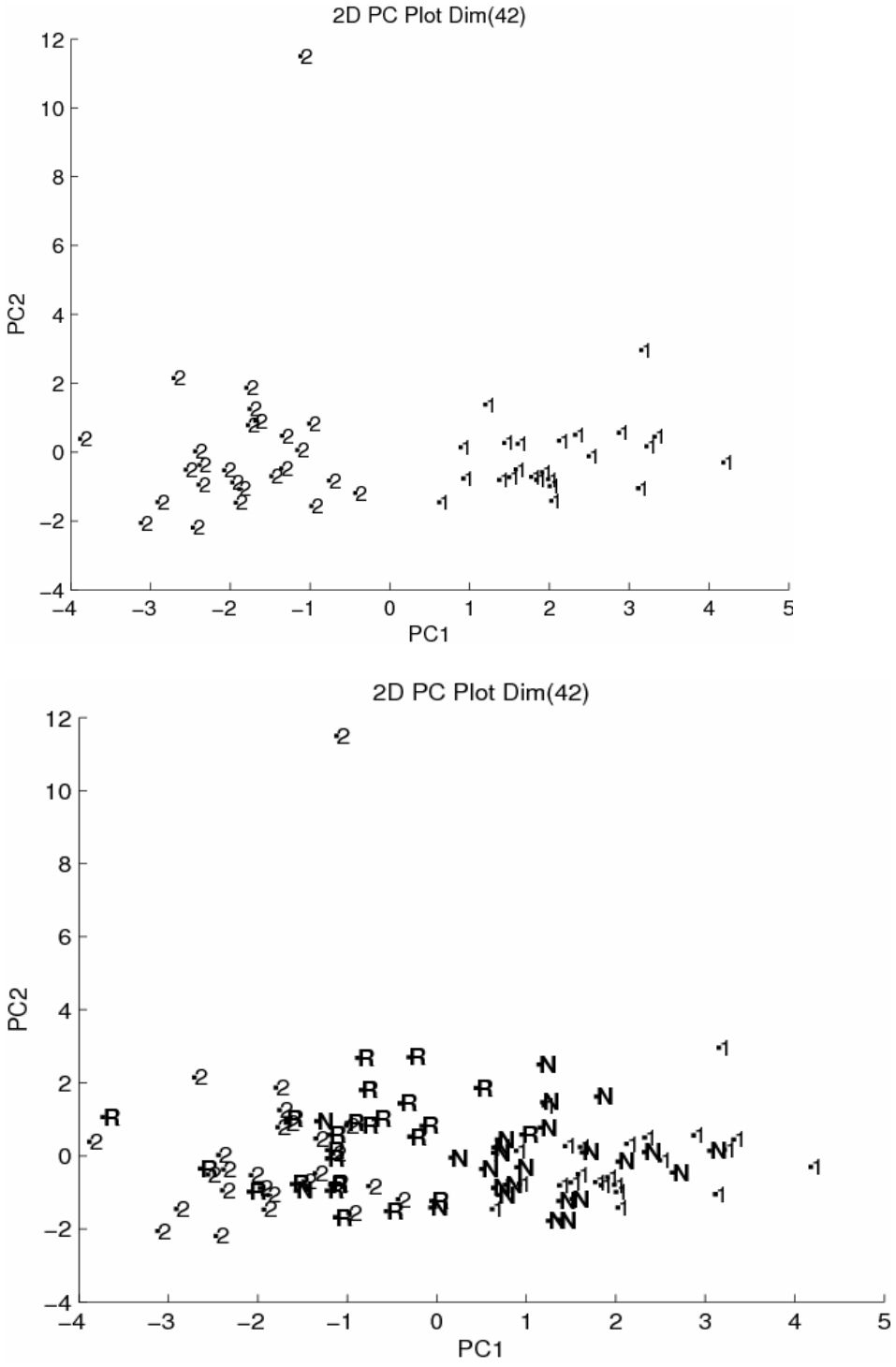



Figure A2.1. A plot of the two largest principal components developed from the 50 biopsy samples and 42 features identified by PCKaNN for the first training set/prediction set pair from the 50%/50% segmented cross validation study. 1 = no reoccurrence and 2 = reoccurrence (training set). R = reoccurrence and N = no reoccurrence (validation set).

**Pset1-hopkins**

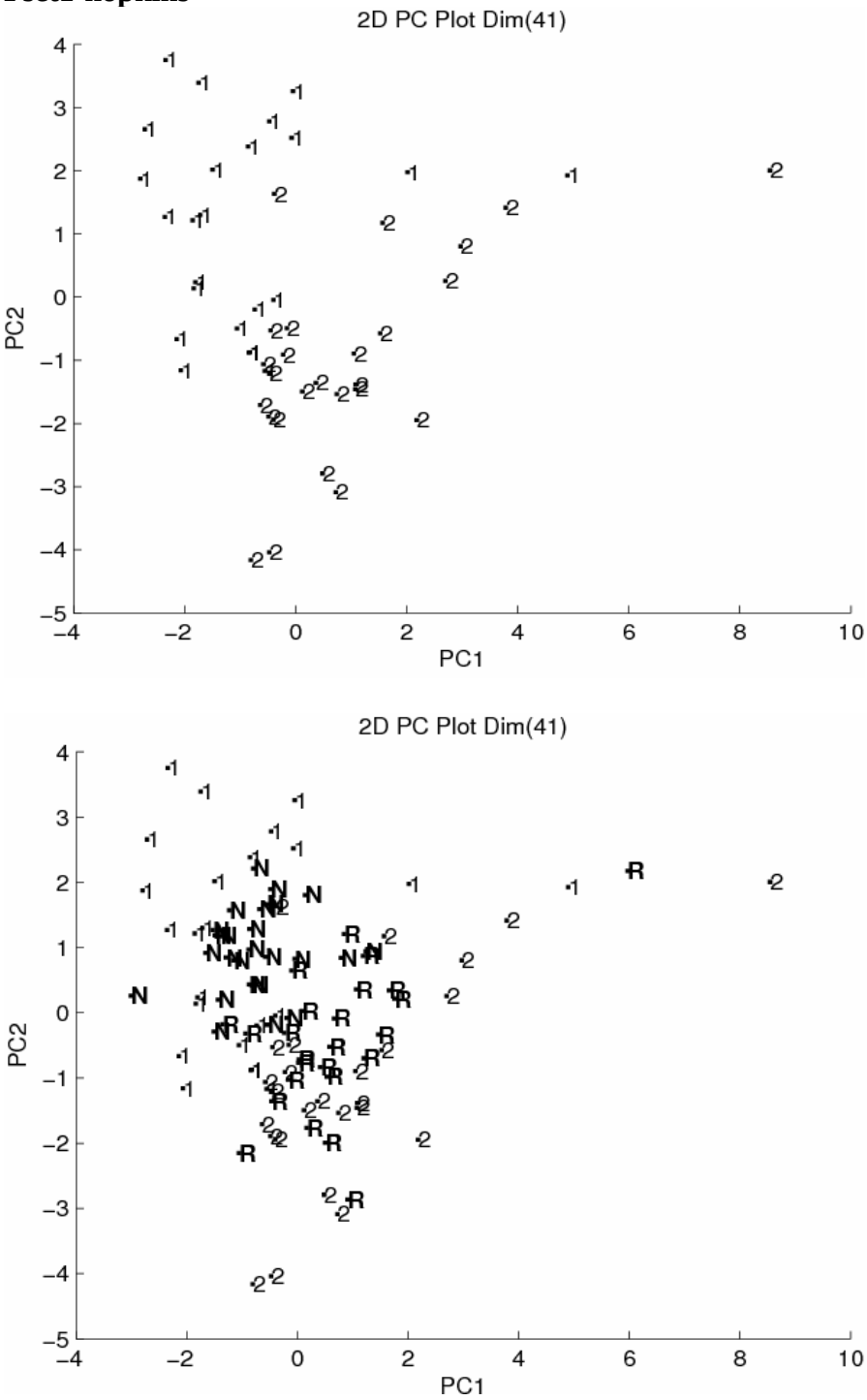



Figure A2.2. A plot of the two largest principal components developed from the 50 biopsy samples and 41 features identified by PCKaNN with the Hopkins statistic for the first training set/prediction set pair from the 50%/50% segmented cross validation study. 1 = no reoccurrence and 2 = reoccurrence (training set). R = reoccurrence and N = no reoccurrence (validation set).

**Pset1-Modified Hopkins**

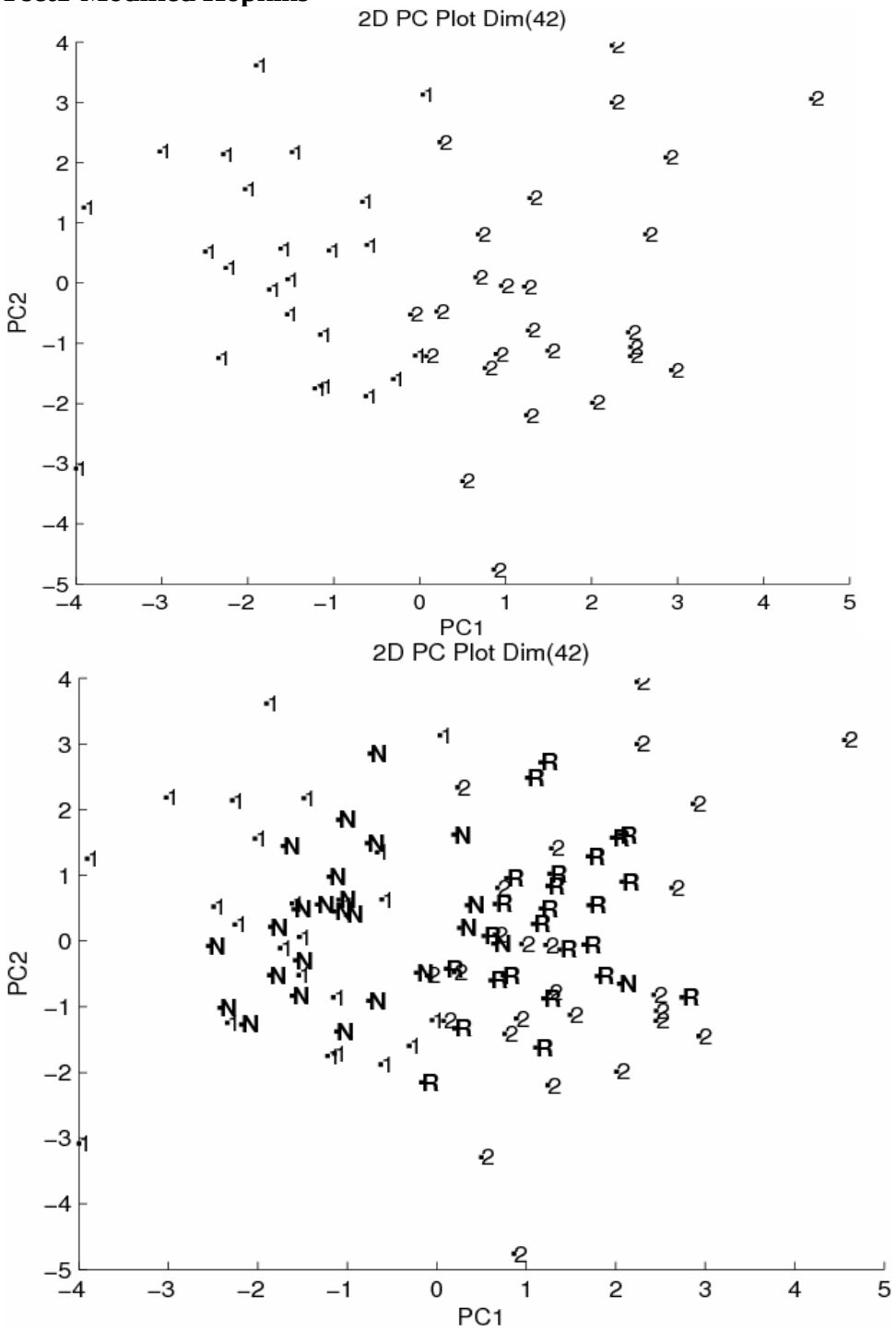



Figure A2.3. A plot of the two largest principal components developed from the 50 biopsy samples and 42 features identified by PCKaNN with the modified Hopkins statistic for the first training set/prediction set pair form the 50%/50% segmented cross validation study. 1 = no reoccurrence and 2 = reoccurrence (training set). R = reoccurrence and N = no reoccurrence (validation set).

**Pset2-Normal**

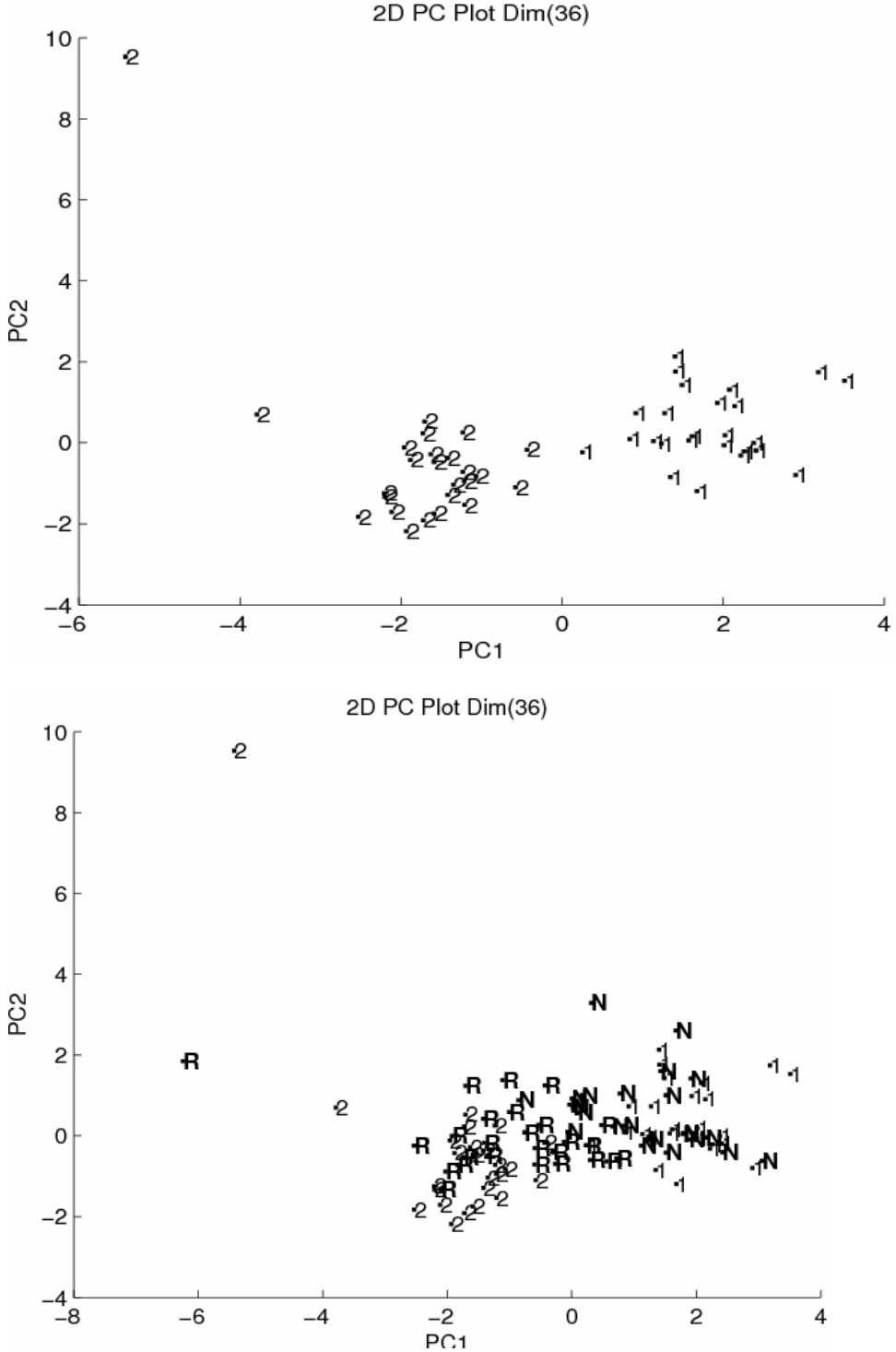



Figure A2.4. A plot of the two largest principal components developed from the 50 biopsy samples and 36 features identified by PCKaNN for the second training set/prediction set pair from the 50%/50% segmented cross validation study. 1 = no reoccurrence and 2 = reoccurrence (training set). R = reoccurrence and N = no reoccurrence (validation set).

**Pset2-Hopkins**

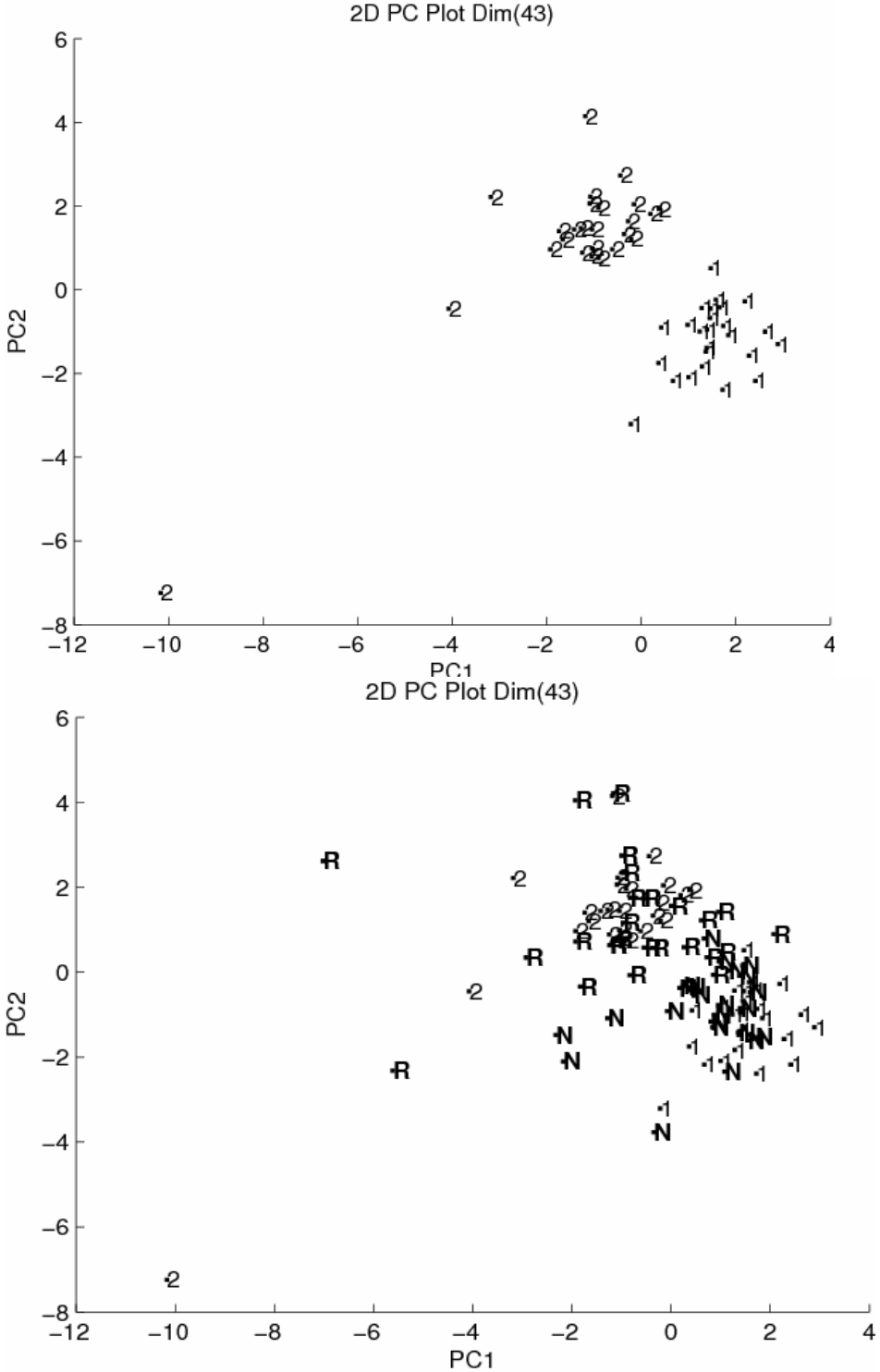



Figure A2.5. A plot of the two largest principal components developed from the 50 biopsy samples and 43 features identified by PCKaNN and the Hopkins statistic for the second training set/prediction set pair from the 50%/50% segmented cross validation study. 1 = no reoccurrence and 2 = reoccurrence (training set). R = reoccurrence and N = no reoccurrence (validation set).

**Pset2-Modified Hopkins**

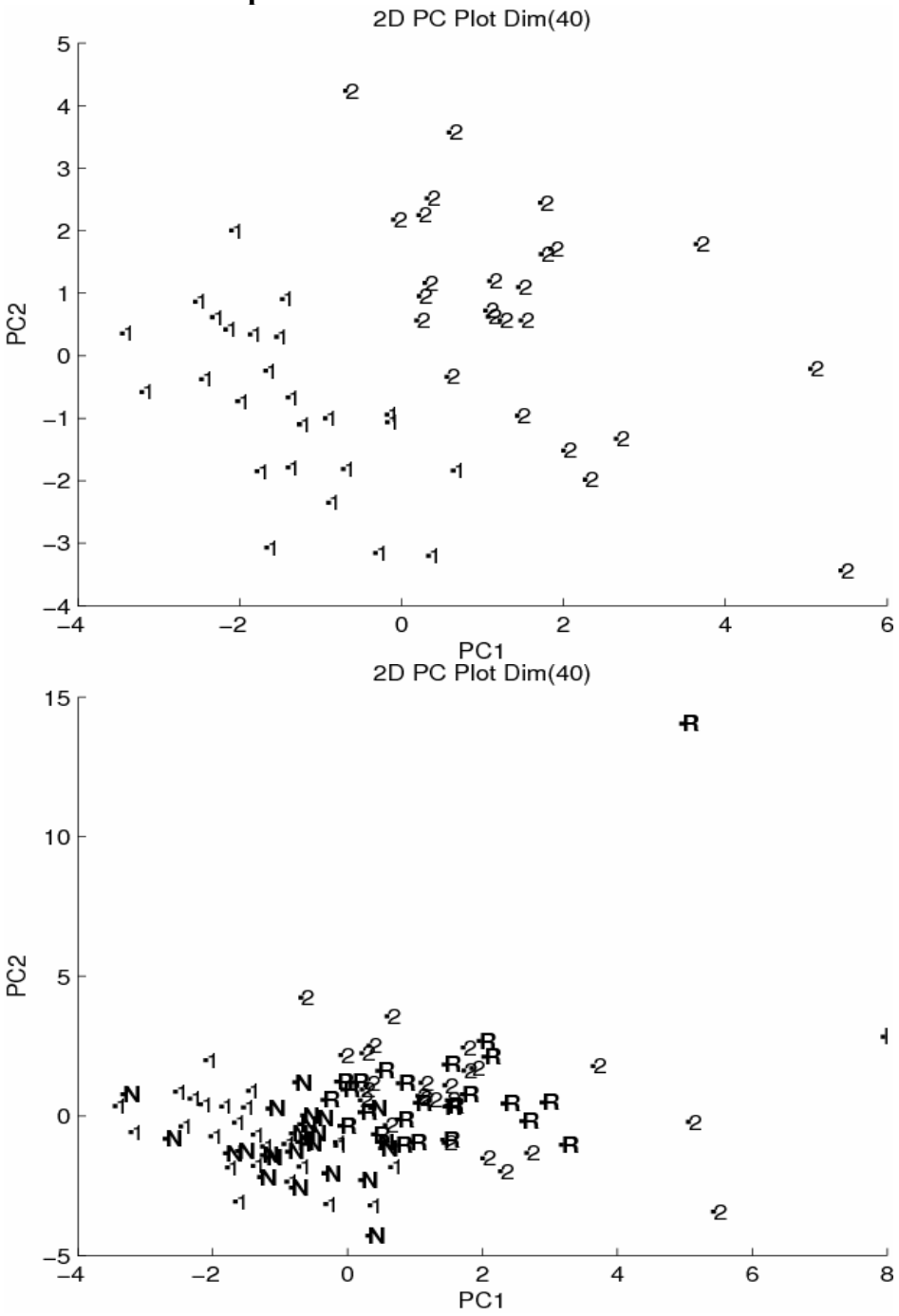
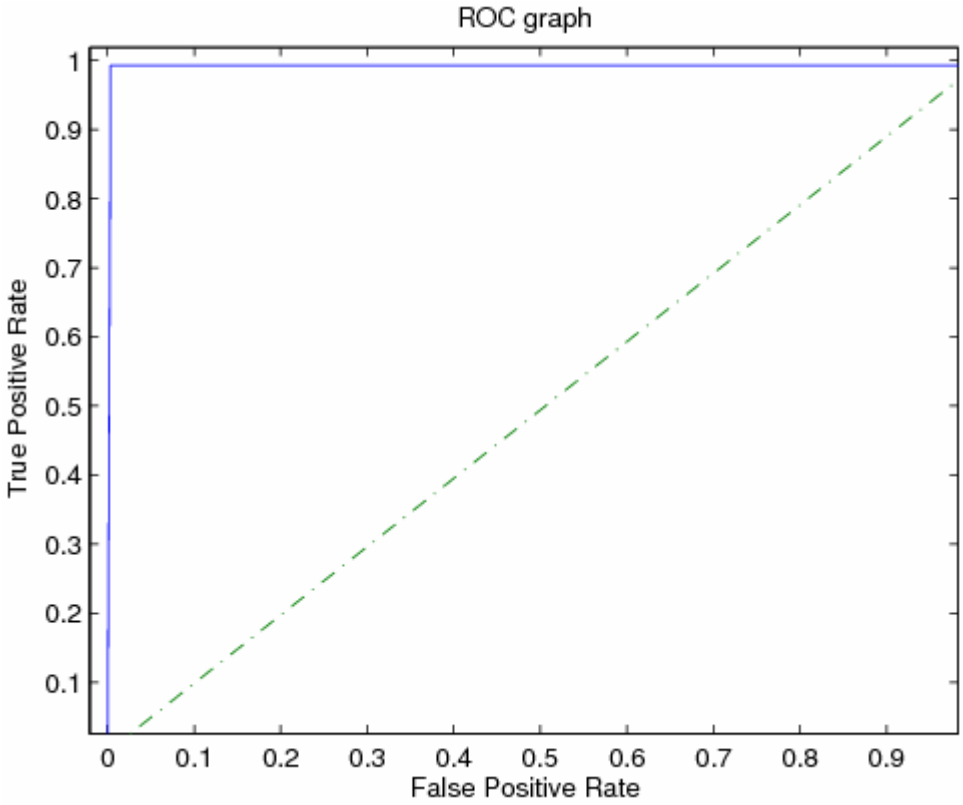



Figure A2.6. A plot of the two largest principal components developed from the 50 biopsy samples and 40 features identified by PCKaNN and the modified Hopkins statistic for the second training set/prediction set pair from the 50%/50% segmented cross validation study. 1 = no reoccurrence and 2 = reoccurrence (training set). R = reoccurrence and N = no reoccurrence (validation set).

# Appendix III

## Receiver Operator Curves for Classifiers Developed from the 53 Coefficients of the 10symmlet6 Wavelet for the Carboxylic Acid Functional Group




ROC curve (solid line) is shown for the discriminant developed by LDA with the dashed line indicative of the response obtained by a random classifier for this data. ROC curves for QDA, RDA, and backpropagation are similar in shape and form to the one obtained for LDA.

$$TPR = \frac{d}{b + d}$$

True positive rate (TPR) where b is the number of samples from class B assigned to class

A, and d is the number of samples from class B assigned to class B by the classifier

$$FPR = \frac{c}{c + a}$$

False positive error rate (FPR) where c is the number of samples from class A assigned to class B and a is the number of samples from class A that are assigned to class A by the classifier

Name: Kadambari Nuguru Date of Degree: May, 2009

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: Pattern Recognition assisted Infrared Library Searching

Pages in Study: 156 Candidate for the Degree of Master of Science

Major Field: Electrical and Computer Engineering

**Scope and Method of Study:**
The development of a genetic algorithm (GA) for pattern recognition analysis of infrared spectral data is proposed. The GA selects spectral features that optimize the separation of the different functional groups in a plot of the two or three largest principal components of the data. Because the largest principal components capture the bulk of the variance in the data, the features chosen by the GA primarily convey information about differences between classes. Hence, the principal component analysis routine embedded in the fitness function of the GA acts as an information filter, significantly reducing the size of the search space, since it restricts the search to feature sets whose principal component plots show clustering of the spectra on the basis of chemical structure. In addition, the algorithm focuses on those classes and or samples that are difficult to classify as it trains using a form of boosting to modify class and sample weights. Samples that consistently classify correctly are not as heavily weighted as samples that are more difficult to classify. Over time, the algorithm learns its optimal parameters in a manner similar to a neural network. The proposed GA integrates aspects of artificial intelligence and evolutionary computations to yield a "smart" one -pass procedure for feature selection and pattern recognition.

**Findings and Conclusions:**
Using the pattern recognition GA to select spectral features, a search prefilter based on the response function to the simple binary classification problem, carboxylic acids versus other compounds including carbonyl compounds, has been developed that allows for the specific detection of carboxylic acids from IR spectra. Carboxylic acids have highly characteristic features but there are also complications that confound the interpretation of their spectra. The wavelet packet transform has been used to denoise and deconvolute the spectra by decomposing each spectrum into wavelet coefficients that represent both high and low frequency components of the signal. This decomposition process is iterated through successive wavelet packets until the required level of signal decomposition is achieved. Using a symmlet 6 mother wavelet at the tenth level decomposition to deconvolve spectral features, the genetic algorithm for pattern recognition analysis was able to identify wavelet coefficients characteristic of the carboxylic acid functional group. Classifiers developed from these wavelet coefficients have been successfully validated.

ADVISER'S APPROVAL: Dr. Alan Cheville