6-12-2006

# Some Significant Results in the Classification Analysis of the Spectroscopic Evaluation of Cervical Cancer

C Shen

# Some Significant Results in the Classification Analysis of the Spectroscopic Evaluation of Cervical Cancer

by

Chenghong Shen

Under the Direction of Dr. Yu-Sheng Hsu

## ABSTRACT

Cervical Cancer is the second most common type of cancer in women worldwide (500,000 cases/year) and one of the leading causes of cancer-related mortality in women in developing countries (230,000 cases/year).

The Spectrx *LightTouch*™ device uses light to detect chemical and structural changes in cervical tissue. Light responds differently when exposed to normal cells and cancerous cells.

The purpose of this research is to find the best model that can be used to diagnose the early cervical cancerous conditions. To achieve this goal, we first tried to reduce the number of variables. We use statistical and non-statistical methods to search for useful explanatory variables. Partial Least Square, Logistic Regression, CART, MARS, SVM have been used to build models. Bootstrap was adopted to estimate the threshold of PLS model. Comparison of the results indicates that PLS produces relatively better model in terms of the performances and to control over model threshold.

INDEX WORDS: Cervix Cancer, Sensitivity, Specificity, AUC, Partial Least Squares, CART, MARS, Logistic regression, SVM, Spectroscopic data

# Some Significant Results in the Classification Analysis of the Spectroscopic Evaluation of Cervical Cancer

by

Chenghong Shen

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Science

In the College of Arts and Sciences

Georgia State University

May, 2006

# Some Significant Results in the Classification Analysis of the Spectroscopic Evaluation of Cervical Cancer

by

Chenghong Shen

| | |
|---|---|
| Major Professor: | Yu-Sheng Hsu |
| Committee: | Gengsheng Qin |
| | Yichuan Zhao |

Electronic Version Approved:

Office of Graduate Studies
College of Arts and Sciences
Georgia State University
May 2006

## ACKNOWLEDGEMENTS

Throughout the process of finishing this thesis, I greatly benefited from the support and advice of many people. To them I would like to express my deepest thanks and sincere gratitude.

I would like to heartedly thank Dr. Yu-sheng Hsu for his gracious guidance and support. I appreciate the suggestions and review of the draft of the thesis by my committee members. I must also thank Dr. Mark Faupel, Chief Executive Officer of Spectrx, Dr. Shabbir Bambot, Senior director of Spectrx, who supervised me during my internship with Spectrx.

Finally, I thank my wife who is always giving me support throughout my internship and the preparation of this thesis.

# TABLE OF CONTENTS

## Appendix VIII SAS code for data manipulation (point analysis)

## LIST OF TABLES

## LIST OF GRAPHS

# Chapter I Introduction

Spectrx, Inc. is a medical technology company providing innovative detection, monitoring and treatment solutions for the diabetes and cancer detection healthcare markets. They believe that their technology for detecting early signs of malignancy could become an important weapon in the war on cancer. Experts agree that early detection and treatment offer the best hope for surviving a cancer disease. The technology not only identifies disease before it becomes malignant, but may be used to guide treatment; therefore preserving more healthy tissue.

Their cervical cancer detection device uses proprietary biophotonic technology to create an image of the cervix that highlights the location and severity of disease at the point of care. Unlike Pap or HPV tests, our test does not require a tissue sample or laboratory analysis, and the results are available immediately. To date, more than 1,000 women have been tested with prototype devices that have consistently provided more accurate results than Pap tests alone.

Fluorescence and reflectance spectroscopy have been shown to be valuable in cancer diagnosis by some investigators. As an example, the latest report from the Richard Kortum Lab indicates that Variability between normal tissues in different patients is higher than variability between tissues with disease grades. In addition, reflectance spectra of cervical pre-cancer showed consistent differences from that of normal tissue at all source detector separations; reflectance intensity of pre-cancer was lower than that of normal tissue on average. In conclusion, Spectral patterns in diffuse reflectance spectra

can be used for the discrimination of normal cervical tissue from low grade and high grade intraepithelial lesions.

Spectrx, Inc. collected data from about 771 patients through clinical trials. The device collects data from 56 spatial points on the surface of the cervix of each patient. For each point, one reflectance spectrum wavelength and several fluorescence spectrum wavelengths were gathered. It was later found that only three of the fluorescence spectrum could effectively predict cancer. The dataset has 771 observations and around 10,000 initial variables for each observation. The purpose of the research is to find the model that can best diagnose cancerous conditions. The model should have 99% sensitivity with diagnosis specificity as high as possible. In this study we have fit Partial Least Squares, Logistic Regression, CART, MARS and Support Vector Machine models. In Chapter II, we will present data manipulation and variable selection process. Two different variables construction methods are to be introduced. In Chapter III – Chapter IV, the methodologies and the results will be demonstrated. All the SAS code used in the research is included in the Appendices.

# CHAPTER II

## Data Manipulation and Variable Selection

There are 572 patients in the training dataset. For each patient, there is an ASCII file containing the reflectance and fluorescence spectra. Each file has 56 rows corresponding to 56 spatial points on the surface of cervix. There is another ASCII file associated with each patient which includes the information about the validity of each point. The format of the spectra data file is described in the following table.

Table-1 data structure of the data file

| Columns | Description | No.of wavelength elements |
|---|---|---|
| 1 | Interrogation point number | N/A |
| 2-63 | Reflectance spectrum | 62 |
| 69-126 | Fluorescence spectrum 1 | 58 |
| 132-184 | Fluorescence spectrum 2 | 53 |
| 189-228 | Fluorescence spectrum 3 | 39 |

Among the 572 subjects, 62 subjects are known to be ‘instrumental data unclean’. This can be caused by ambient light, poor contact with cervix, poor centering on cervix, excess movement, excess mucus or excess blood. Therefore we excluded the 62 patients from the training dataset.

Table-2 Pap and pathology variables information

| Variable description | Code/Value description |
|---|---|
| Pathology (whole cervix gold standard) | 0 = Normal<br>1 = Non-dysplastic changes<br>2 = CIN 1<br>2.5= CIN 1/2<br>3 = CIN 2<br>3.2 = CIN 2/3<br>3.5 = CIN 3+ |
| Pap (cytology) | 0 = Normal<br>1 = Benign changes<br>2 = ASCUS, not favoring neoplasia<br>2.8 =ASCUS, favor neoplasia<br>3 = LSIL<br>3.2 = AGUS<br>3.5 = ASC-H/cannot rule out HSIL<br>4 = HSIL<br>4.2 = Invasive cancer |

In addition to the spectral data, Spectrx Inc. also collect Pap smear (cytology), biopsy (pathology) result, patient age and menopause information through clinical trial. Pap test is microscopic examination of cells collected from the cervix. Pap is used to detect changes that may be cancerous or may lead to cancer, and it can also show non-cancerous conditions, such as infection or inflammation. Spectrx Inc. is authorized by FDA (Food and Drug Administration) to use Pap smear result as a predictor. Biopsy is conducted by a pathologist and the diagnosis is generally considered the final word. It is a procedure that involves removal and examination of a sample of tissue from the cervix

for diagnostic purposes. Spectrx uses the biopsy (pathology) result as the response variable for all models.

FDA stated that patient with pathology diagnosis 2 or 2.5 can be classified as either disease or non-disease. So in our model building process, CIN1 and CIN1/2 patients were excluded. We define pathology value larger than 3 to be disease and pathology value smaller than 3 to be non-disease. So the problem becomes a typical two class classification problem.

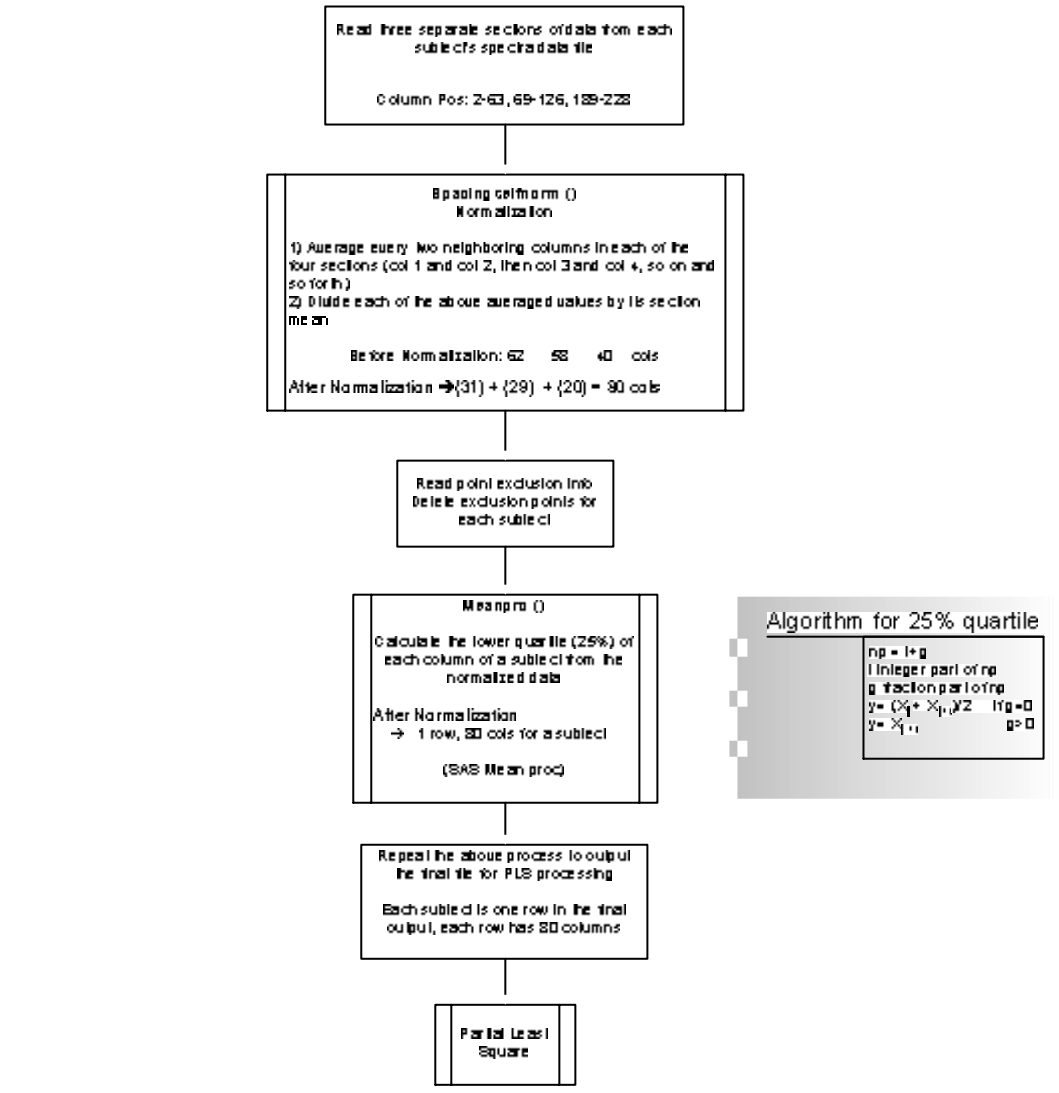
Spectrx Inc. published the key characteristics of the best model they had developed in their paper. The data manipulation and modeling procedures illustrated in Graph-1 have been applied as we studied the details of the model (hereinafter referred to as PLS 25$^{th}$ percentile model).

1. Spectrx statisticians found that Fluorescence 2 variables have little discriminating capability, hence eliminate Fluorescence 2. However we later discovered that Fluorescence 2 variables are also predictive;
2. Earlier studies indicated that 10 nm binning (which is to average 2 neighboring spectra variables) can improve the PLS model performance. We will verify the statement by comparing10nm binning with 15nm binning (3 variables) and 20nm binning (4 variables). After 10 nm binning, there are 80 spectral variables in total;
3. Study also shows that after self-normalization (divide the spectral variable by group mean), the performance of the PLS model improves. Our spectra graph shows that after binning, all the 56 cervix points will have closer intensity across the wavelength;

2006-1-2

# Model1 Var Selection

Author: Chenghong Shen



4. Spectrx claimed that 25$^{th}$ percentile of the 56 locations' spectral data can best discriminate CIN2 or worse patients.

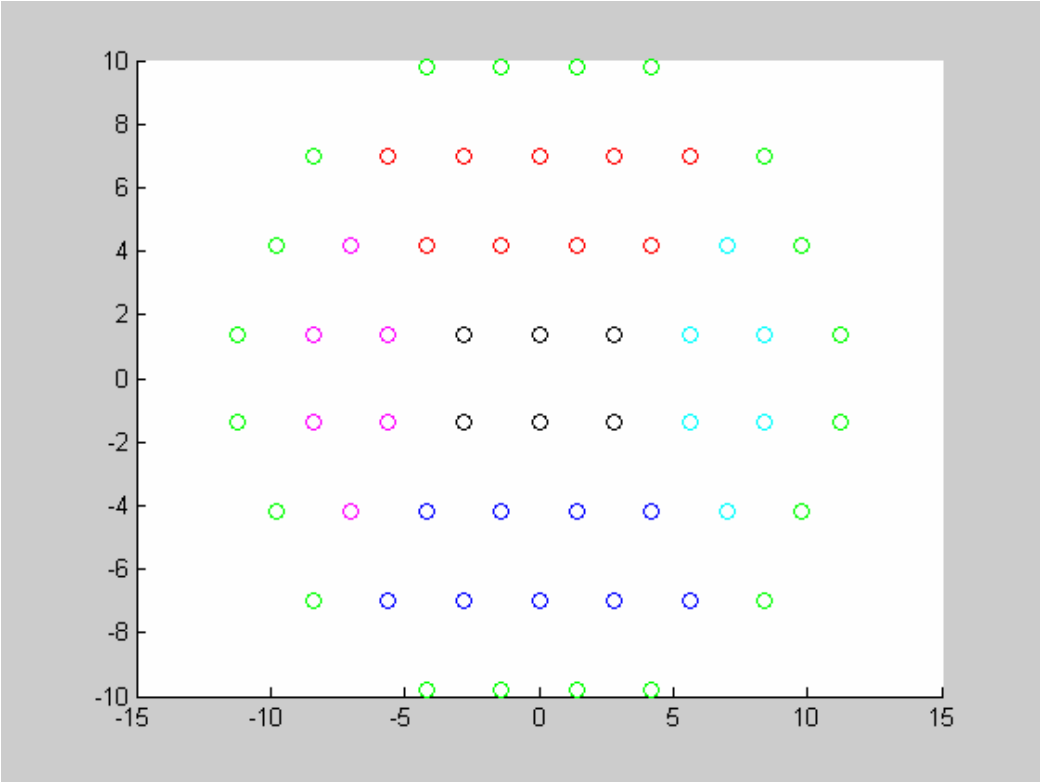
In order to verify the above result in 4, we pre-selected 5$^{th}$ percentile, 10$^{th}$ percentile, 15$^{th}$ percentile, 25$^{th}$ percentile, median, 75% percentile, 95$^{th}$ percentile, descriptive statistics MAX, MIN, Range, Q3-Q1, STD, Skewness, Kurtosis, Mean, etc. and their linear combinations such as Mean + 25$^{th}$ percentile in the variable selection process. In addition, the variables selected by MARS as well as variables in other transformed forms are also included.

Additionally we took a fresh look at the variable selection process. The best way to discriminate cancer patients is to find the difference between normal and disease in the data. It has been shown that usually only a few points in the 56 points would be cancerous for a cancer or pre-cancer patient. It was also proved that cancer usually doesn't start on the peripheral region of the cervix. Thus we divided the whole cervix into six groups, one is peripheral group the rests are central groups. Then we calculated the difference between each of the center groups with the peripheral group. The detailed data manipulation process is as follows.

1. Obtain the mean of the 80 spectral variables for each of the groups. Denote the 6 averages as $A_0, A_1, ..., A_5$;

2. Find max $|A_0 - A_i| = X_{1\max} X_{2\max} ... X_{80\max}$, i=1…5. There are still 80 spectral variables in total;

3. Use the variables obtained by the above process (hereinafter referred as difference variable) as the covariates of the model




Graph-2 Cervix surface divided into 6 groups

Appendix I and II record SAS code for data manipulation, variable pre-selection and difference variable construction as described in here.

Moreover, spectra graph was plotted to assist the variable selection. Red curves represent cancerous points. The SAS code to draw the spectra plot is shown in Appendix III. The following patterns can be observed from the spectra graph.

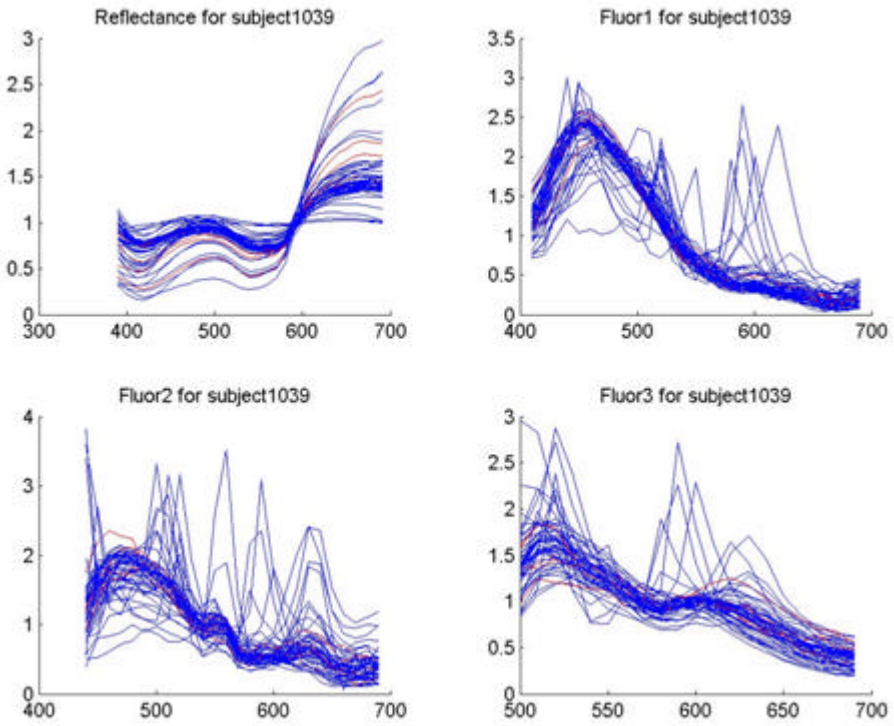
1. Cancer spectra are different in intensity from non-cancer spectra, particularly the difference can be observed in reflectance spectrum. The difference variables are considered to be predictive;

2. In all reflectance and fluorescence spectra, there exists a middle point. Cancer curves are more often below normal curves when wavelength is smaller than the middle point. Cancer curves are more often above normal curves when wavelength is larger than the middle point;

3. The difference in intensity between cancer and non-cancer curves is not homogeneous at different values of wavelength.

4. There is much variability among different points and patients, and sometimes no obvious difference between cancer and non-cancer curves can be observed.

5. Fluorescence2 should also be informative. There is no reason to exclude Flurescence2 from our study.




Graph-3 Reflectance and Fluorescence spectra (intensity vs. wavelength)

# CHAPTER III

## Methodology

### I Variable selection and Model building

The response variable we described in Chapter II is binary, which is either 1 (disease) or 0 (non-disease). Therefore, it is not appropriate to apply multiple ordinary linear regression models on the data. A linear model can be written as: $\tilde{Y} = \tilde{X}\tilde{b} + \tilde{e}$, where

$\tilde{b}$ is the regression coefficient(s)

$\tilde{X}$ is the matrix of all the covariates

$\tilde{e}$ is the error term.

The OLS regression should not be directly applied here primarily because of the following reasons:

1. The response variable is discrete and not normally distributed.
2. The variances of error terms are heteroskedastic. It can be derived that

   $Var(e_i) = P_i(1 - P_i)$, where $P_i$ is the probability of $Y_i = 1$, $i = 1,..., n$.

Therefore, we used Partial Least Squares, Logistic Regression, CART, MARS and Support Vector Machine, etc. to accommodate the binary data.

Partial Least Squares (PLS) is a method for constructing predictive models when the factors are many and highly collinear. PLS balances objectives of explaining response variation and explaining predictor variation. A PLS model can be show as

$Y = t_1 q_1 + t_2 q_2 + ... + t_n q_n + E_n$, where t are the latent variables or scores; q are the loading vectors.

Note that the scores are chosen so that the relationship between successive pairs of scores is as strong as possible. In general, PLS is doing a PCR on the explanatory variables, a PCR on the response variable and then doing a Multiple Linear Regression to relate the X block principal components to the Y block principal components. PLS uses the response y to construct its directions, its solution is a nonlinear function of y.

It can be shown that PLS seeks directions that have high variance and have high correlation with the response in contrast to principal components. In particular, the mth principal component direction $v_m$ solves:

$$\max_{\substack{||a||=1 \\ V_l^T Sa = 0, l=1,..., m-1}} Var(Xa),$$

where S is the sample covariance matrix of $x_j$. The conditions $V_l^T Sa = 0$ ensures that $z_m = Xa$ is uncorrelated with all the previous linear combinations $z_l = Xv_l$. The mth PLS direction $\hat{J}_m$ solves:

$$\max_{\substack{||a||=1 \\ \hat{J}_l^T Sa = 0, l=1,..., m-1}} Corr^2(y, Xa) Var(Xa)$$

We used PROC PLS in SAS package to build the Partial Least Squares model and discovered that the PLS model has less shrinkage than the logistic regression model. In the variable selection process, PLS is used to evaluate the performance of the candidate models. The SAS codes to calculate the sensitivity, specificity, the area under the ROC curve and cross validation are shown in Appendix IV.

Variable selection is very important in any model building process especially in case that the number of variables is around 10,000. After the initial data manipulation and variable pre-selection, we still have around 5000 variables left. Partial Least Squares can be very useful in variable reduction. Like principal components, the PLS model extract latent factors which are essentially linear combination of the original variables. The first 10 factors often count at least 90% of the total variation. Therefore, the first 10 latent factors contain almost all the information from the original variables.

We calculated Area under Curve (AUC), sensitivity and specificity to evaluate model performance. It was observed that model performance in terms of AUC is closely related to the number of variables and the variables chosen. A variable which is significant (accounts for the variation of Y in PLS) can possibly degrade the performance. There is no mature algorithm so far to select variables based on AUC. Therefore, we have adopted both statistical methods and non-statistical (manual) methods in variable selection.

It was verified that a model based on the 25th percentile of the Reflectance spectrum and the three Fluorescence spectra variables has very good performance in terms of AUC. Other statistics like mean, median or any combination of the descriptive statistics didn't have performance as good as $25^{th}$ percentile.

Moreover, a comparison between the $25^{th}$ percentile variables and the difference variables were made. After binning, the Reflectance and Fluorescence 1 and 3 have 80 variables of $25^{th}$ percentile in total. Similarly 80 difference variables were obtained. We divided the $25^{th}$ percentile data into four even groups and do the same for the difference variables. This resulted in 8 groups. Four groups out of eight were randomly selected to

build a PLS model. The procedure created 70 different PLS models. It was discovered that the four groups exclusively from the 25$^{th}$ percentile data had the model with best performance. Further research shows that although the 25$^{th}$ percentile data is more predictive then the difference variables, mixing the two types of variables together can improve the model performance. APPENDIX V records the SAS code for the two types of variable comparison described in here.

Besides the above approaches, we used MARS to select variables. MARS (Multivariate Adaptive Regression Splines) was developed in the early 1990s by world-renowned Stanford physicist and statistician Jerome Fredman. Salford Systems (San Diego, CA) produced MARS software package based on his original code. MARS automates model development and deployment, including separating relevant from irrelevant predictor variables, transforming predictor variables exhibiting a nonlinear relationship with the target, determining interactions between predictor variables, handling missing values with new nested variable techniques and conducting extensive self-tests to protect against overfitting.

MARS essentially builds flexible models by fitting piecewise linear regressions. The nonlinearity of a model is approximated through the use of separate regression slopes in distinct intervals of the predictor variable space. The slope of the regression line is allowed to change from one interval to the other as the two "knot" points are crossed. The optimal MARS model is selected in a two-stage process. In the first stage, MARS constructs an overly large model by adding "basis functions". Basis functions represent either single variable transformations or multivariable interaction terms. The process continues until a user-specified maximum number of basis functions is reached. In the

second stage, basis functions are backward eliminated in order of least contribution to the model until the best model is found.

Although the MARS model doesn't perform well here, the variables obtained from MARS (basis functions), which are essentially variable transformations and interactions can be used by PLS algorithm to improve model performance.

In addition to MARS, CART can also be used for variable selection and model construction. CART (Classification and Regression Tree) is a nonparametric technique that can select from among a large number of variables and their interactions that are most important in determining the outcome variable to be explained. Salford Systems (San Diego, CA) developed a very user friendly CART package.

CART starts building the tree from the root. At each node, it always tries to find the best variable to split. The goodness of split criteria was derived from an impurity function and the best splitter seeks to maximize the average "purity" of two child nodes. CART software implements two splitting rules, one uses the Gini index of diversity as a measure of a specific node impurity function $f(p_1,...,p_j) = -\sum p_j \log p_j$, and it can be shown that $i(t) = \sum_{i \neq j} p(i \mid t)\, p(j \mid t)$. The other is the towing rule: At a node t, with s splitting into $t_L$ and $t_R$, s was chosen by maximizing the following formula:

$$P_L P_R [\sum | p(j \mid t_L) - p(j \mid t_R) |]^2 / 4,$$

where $P_L$ and $P_R$ are the probabilities split into the left or the right. $p(j \mid t_L)$ and $p(j \mid t_R)$ are conditional probabilities split into the left or the right at a node t.
The tree building process is stopped until a maximum tree is reached or there is external limit on the number of levels in the tree. Then CART starts the pruning process. A child

is pruned away if the resulting change in the predicted misclassification cost is less than the complexity parameter times the change in tree complexity. Such a process continues until the optimal tree is obtained.

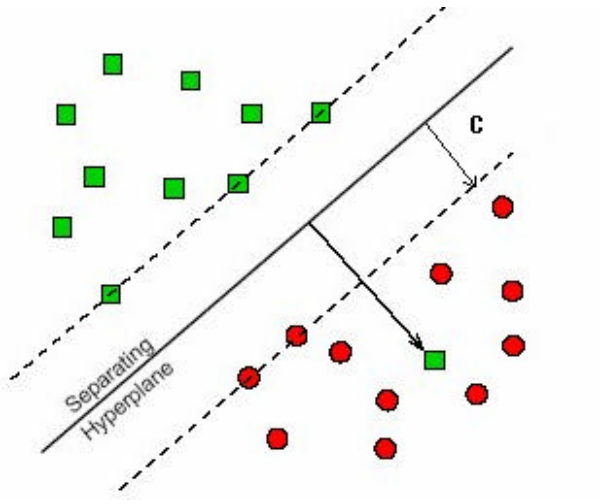
According to FDA's requirement, the algorithm should have sensitivity 99% (or 95%) with specificity not lower than 20%. The drawback of using CART in our problem is that the sensitivity function in terms of the choice of trees is discrete. There is notmuch room for 99% (or 95%) sensitivity trees. It is very hard to find a tree which performs well for the validation.

Although CART model doesn't meet FDA's requirement, CART has its use in variable selection. CART analysis needs not to be the final stage of an analysis. It may serve as a method for selecting variables to be included in PLS or logistic regression model. CART software can generate a variable importance diagram which ranks the importance of all the variables as splitter. Breiman, Frieman ans Stone (1984) cautioned against placing too much emphasis on these rankings. They pointed out that rankings can be quite sensitive to random fluctuations in the data. However, attention should be paid to variables that are strong competitors near the top of the tree.

In this study, we used statistical and non-statistical methods to transform the original variables. Since Pap was found to be the most important predictor, we focused on the transformation of Pap. An analytical approach is to use power transformation. That is, to determine what value of $X^p$ yields the best model for Pap. The power function can be shown as $F(x) = X^p$, p={-2,-1,-0.5,0,0.5,1,2,3}, where p=0 denotes the log of variable. We fit the 8 models and found the one that resulted in largest AUC.

In refer to table-2, AGUS (Atypical Glandular Cells of Undermined Significance) maybe more similar to ASCUS (Atypical Squamous Cells of Undermined Significance) than to LSIL (Low grade intraepithelial Lesion). This can be justified biologically. Therefore we tried to adjust the Pap test value in different ways to find a model with best performance.

As mentioned previously, Support Vector Machine was also used to build classification models. OSU SVM is a Support Vector Machine (SVM) toolbox for the MATLAB numerical environment. The software is used for this thesis research. Suppose our training data consists of N pairs $(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)$, with $x_i \in IR^p$ and $y_i \in \{-1,1\}$.




$x^T b + b_0 = 0$

Graph-4 Support Vector Classifier

Define a hyperplane by $\{x : f(x) = x^T b + b_0 = 0\}$, where $b$ is a unit vector: $\| b \|=1$. A classification rule (decision function) induced by $f(x)$ is $G(x) = sign[x^T b + b_0]$. If the two classes are separable, we are able to find the

hyperplane that creates the biggest margin between the training points for class 1 and -1. The optimization problem can be phrased as

$$\max_{b,b_0,\|b\|=1} C, \quad \text{subject to } y_i(x_i^T b + b_0) \geq C, i = 1,\ldots N,$$

Or

$$\min_{b,b_0} \|b\|, \text{ subject to } y_i(x_i^T b + b_0) \geq 1, i = 1,\ldots N,$$

Note that C=1/ $\|b\|$

Suppose that the classes overlap in feature space. One way to deal with it is to still maximize C (minimize $\|b\|$), but allow for some points to be on the wrong side of the margin. Define the slack variables $x = (x_1, x_2, \ldots, x_N)$. The constraint is

$$y_i(x_i^T b + b_0) \geq C(1 - x_i),$$

$$\forall i, x_i \geq 0, \sum_{i=1}^{N} x_i \leq const$$

The value $x_i$ in the constraint $y_i(x_i^T b + b_0) \geq C(1 - x_i)$ is the proportional amount by which the prediction $f(x_i) = x_i^T b + b_0$ is on the wrong side of its margin. Therefore, we can bound the total proportional amount by which prediction fall on the wrong side of their margin by bounding $\sum x_i$. So the optimization problem becomes

$$\min \|b\| \text{ subject to } \begin{array}{l} y_i(x_i^T b + b_0) \geq 1 - x_i \ \forall i, \\ x_i \geq 0, \sum x_i \leq const \end{array}$$

We can see that points well inside their class boundary do not play a big role in shaping the boundary. Points on the wrong side of the boundary are support vectors. In addition, points on the right side of the boundary but close to it (within the margin) are also support vectors.

The above formula is actually a convex optimization problem and it can be re-expressed as

$$\min_{b,b_0} \frac{1}{2} \| b \|^2 + g \sum_{i=1}^{N} x_i$$

where $g$ represents the constant ( $\sum x_i \leq cons\tan t$ )

Solve the above function, $\hat{b}_0$ and $\hat{b}$ are obtained. The decision function can be written as $\hat{G}(x) = sign[\hat{f}(x)] = sign[x^T \hat{b} + \hat{b}_0]$. The tuning parameter is $g$ . By changing $g$ , we were able to obtain different levels of sensitivity and specificity for classifying cancer.

The support vector classifier discussed above finds linear boundary in the input feature space. We can extend this idea to produce nonlinear boundaries by constructing a linear boundary in a large, transformed version of the feature space. Linear boundaries in a transformed space can achieve better two class separation, and then we translate back to nonlinear boundaries in the original space. In this thesis, we used two kernel functions to build SVM models, one is linear kernel, the other is polynomial kernel. The SAS code to build SVM models is shown in Appendix VI.

**II Bootstrap estimate of model threshold**

For PLS model at a given threshold of the response variable, which is the value above which a person is said to be positive for cervical cancer, estimates of the true positive fraction (TPF) and false positive fraction (FPF) can be obtained. There are ways to determine an optimal value for a threshold but there are no derived estimates of standard errors to produce similar confidence intervals around the threshold estimate.

FDA has requested that the medical device should have 99% sensitivity with specificity above 20%. Therefore, we need to estimate an optimal threshold for the PLS model. To obtain estimate of the standard error for a threshold we proposed to use the method of bootstrap. By resampling with replacement, "new" ROC curves can be generated. For each "new" ROC we will obtain an estimate of the threshold value that produces 99% sensitivity.

The estimates of the threshold obtained from the resampling process will constitute an empirical sampling distribution for the threshold value. Range intervals for the threshold can then be established using the [$a$ , 1] of the distribution of the threshold value. For our study $a$ is set to 0.05. Therefore in order to ensure 99% sensitivity, the optimal threshold is $5^{th}$ percentile of the threshold distribution. The SAS code to calculate threshold using bootstrap is shown in Appendix VII.

The bootstrap method was first proposed by B. Efron in 1979 after the initial development of computer technology. In our study 1000 bootstrap samples were produced to constitute empirical sampling distribution for the threshold. We also calculated average specificity and standard error of the threshold value for the empirical sample to evaluate a model performance.

In standard bootstrap, since the data is sampled with replacement the probability of any observation in the original dataset will not be selected at each pick is $1-\frac{1}{n}$, where n is the sample size. Then the probability that this observation will not be selected in a bootstrap sample is $(1-\frac{1}{n})^n$. As $n->\infty$, $(1-\frac{1}{n})^n$ -> 0.368; the proportion of

observations from the original dataset appearing in the bootstrap sample is 0.632. Therefore the result from the standard bootstrap is biased.

Because the sample size is small we believe that the optimal threshold should be larger than the one we estimated from standard bootstrap. In addition the average specificity will also be larger. The sensitivity of a small sample will drop sharply merely because of one or two mis-classified disease patients when the sensitivity is close to one. So a lot of bootstrap samples can not attain sensitivity in the range of [0.98, 1). Their sensitivity is either 1 or below 0.98. Our simulation study indicated that with the standard bootstrap procedure, a larger sample size will result in higher specificity and optimal threshold.

In order to correct the bias, we used 632 bootstrap to estimate the threshold. The estimate is

$$Threshold632 = 0.632 * T_S + 0.368 * T_{Origin}, \quad \text{where}$$

$T_S$ = estimate of the threshold from the standard bootstrap

$T_{Origin}$ = estimate of the threshold from the original sample

## III Model Validation

Model validation is used to evaluate how well a model can be applied to any new data. We employed conventional cross-validation as well as K-folder cross validation in the research. The conventional cross-validation is to randomly split the data into two parts, say 60/40, the larger part for training and the smaller for validation. Since the split is random, the results can differ from one split to the other. K-folder cross-validation is a

technique to train and validate data on the same dataset. In this study, we divide the training dataset into k=10 approximately equal sized subsets. Besides, we ensure that patients with a certain Pap value are evenly allocated to each subset. Therefore the 10 subsets are ‘equivalent’ in size and content. Next we build the model 10 times and each time leave out one subset for validation.

FDA has also requested that the performance (sensitivity and specificity) of the final model should be irrelevant to the patients’ Pap test result. That is to say, if we divide patients into groups based on their Pap test result, then at a fixed threshold, the model should always attain 99% sensitivity with at least 20% specificity in each Pap group.

# CHAPTER IV

## Results and Conclusion

In this study, we went through data manipulation, variable reduction and search for useful variables. Then we employed different classification methodologies to find the best model. In particular Partial Least Square, CART, MARS, Support Vector Machine and Logistic Regression were used for building classification models.

It was verified that 10 nm binning which is averaging 2 neighboring variables is better than 15nm binning (averaging 3 variables), 20nm binning (averaging 4 variables) and no binning. Therefore the data manipulation method used here is effective in achieving better disease classification.

Table-3 Comparison of different binning scheme

| $25^{th}$ percentile PLS model | AUC (Train) | AUC (Validation) |
|---|---|---|
| No binning | 0.83680 | 0.72680 |
| 2 variable binning | 0.85644 | 0.72708 |
| 3 variable binning | 0.84511 | 0.72122 |
| 4 variable binning | 0.81653 | 0.71508 |

We used Partial Least Squares model to evaluate the effectiveness of new variables and to eliminate redundant ones.

Table-4 Effectiveness of pre-selected variables (descriptive statistics)

| Variable Used | AUC training | AUC Validation |
|---|---|---|
| 1% | 0.665 | 0.522 |
| 5% | 0.81 | 0.48 |
| 10% | 0.819 | 0.488 |
| 25% | **0.856** | **0.518** |
| 50% | 0.827 | 0.525 |
| 75% | 0.779 | 0.512 |
| 95% | 0.698 | 0.445 |
| MAX | 0.675 | 0.489 |
| MIN | 0.67 | 0.572 |
| Range | 0.641 | 0.489 |
| Q3-Q1 | 0.762 | 0.457 |
| STD | 0.66 | 0.476 |
| Skewness | 0.659 | 0.569 |
| Kurtosis | 0.688 | 0.411 |
| Coeff of Variation | 0.656 | 0.486 |
| USS(sum of square) | 0.749 | 0.518 |
| Mean | 0.789 | 0.527 |
| LCLM | 0.816 | 0.532 |
| UCLM | 0.781 | 0.515 |
| STDERR | 0.652 | 0.483 |

From Table-4, 25$^{th}$ percentile seems to be the most useful variable. Next is median and 10$^{th}$ percentile. We only kept 25$^{th}$ percentile variables in our model building since adding any other additional variable listed in Table-4 did not improve the validation result.

As described in Chapter II, we divided the 25$^{th}$ percentile variables and difference variables into 8 even groups and randomly selected four groups out of eight to build a PLS model. The procedure created 70 different combinations of covariate groups. Table-5 lists some of the combinations that perform on top of the list. The original 25$^{th}$ percentile covariate group has the best performance in terms of AUC (0.828/0.773).

Table-5 Performance of some combinations of 25$^{th}$ percentile and difference variables

| Variables | Train(AUC) | Valid(AUC) |
|---|---|---|
| p25ra1-p25ra20 p25ra21-p25ra40 p25ra41-p25ra60 p25ra87-p25ra106 | 0.827735645 | 0.772681954 |
| p25ra21-p25ra40 diff1-diff20 diff41-diff60 diff61-diff80 | 0.746084901 | 0.643868395 |
| p25ra1-p25ra20 p25ra21-p25ra40 diff41-diff60 diff61-diff80 | 0.741160248 | 0.650448654 |
| p25ra21-p25ra40 diff21-diff40 diff41-diff60 diff61-diff80 | 0.737909977 | 0.652642074 |
| diff1-diff20 diff21-diff40 diff41-diff60 diff61-diff80 | 0.736334088 | 0.596011964 |
| p25ra21-p25ra40 diff1-diff20 diff21-diff40 diff41-diff60 | 0.736038609 | 0.640478564 |
| p25ra1-p25ra20 p25ra21-p25ra40 diff1-diff20 diff41-diff60 | 0.7347582 | 0.640079761 |
| p25ra1-p25ra20 p25ra21-p25ra40 diff21-diff40 diff41-diff60 | 0.73091697 | 0.647258225 |
| p25ra21-p25ra40 diff1-diff20 diff21-diff40 diff61-diff80 | 0.729439575 | 0.632901296 |
| p25ra87-p25ra106 diff1-diff20 diff41-diff60 diff61-diff80 | 0.728159165 | 0.656031904 |
| p25ra21-p25ra40 p25ra41-p25ra60 p25ra87-p25ra106 diff41-diff60 | 0.726287797 | 0.660219342 |
| p25ra21-p25ra40 p25ra41-p25ra60 p25ra87-p25ra106 diff21-diff40 | 0.725499852 | 0.664007976 |
| p25ra21-p25ra40 p25ra41-p25ra60 p25ra87-p25ra106 diff61-80 | 0.724908894 | 0.665403789 |
| p25ra21-p25ra40 p25ra41-p25ra60 p25ra87-p25ra106 diff1-diff20 | 0.72382547 | 0.666600199 |
| p25ra41-p25ra60 p25ra87-p25ra106 diff41-diff60 diff61-diff80 | 0.723628484 | 0.64885344 |
| p25ra1-p25ra20 p25ra41-p25ra60 p25ra87-p25ra106 diff61-80 | 0.723234512 | 0.662811565 |
| p25ra41-p25ra60 p25ra87-p25ra106 diff21-diff40 diff41-diff60 | 0.723136019 | 0.649850449 |
| p25ra41-p25ra60 diff1-diff20 diff41-diff60 diff61-diff80 | 0.722939033 | 0.620937188 |

However we discovered that mixing the 25$^{th}$ percentile variables and difference variables together does improve model performance. We have developed several PLS mixed models as illustrated in Table-6. Notice that Mixed Model 1.9 has quite significant

performance improvement over the original 25$^{th}$ percentile model in terms of AUC. 10-folder cross validation was employed to evaluate model performance. Note that the MARS variable is primarily the transformation of Pap. The significance of the MARS variable indicates the non-linear relationships between Pap and the response variable. In table-6, Mars variables are defined as

BF1 = (PREFERRE = 3 OR PREFERRE = 4);

BF3 = (PREFERRE = 0 OR PREFERRE = 1 OR PREFERRE = 3);

BF5 = max(0, P25RA32 - 0.418);

where PREFERRE is the Pap test result.

In addition to spectroscopic data, Spectrx Inc. also collected patient personal information, medical history data such as age and menopause. We discovered that age and menopause are related to cancer. Menopause can be used as a predictor while age is not. Table-7.1 and 7.2 make a comparison of models with menopausal and without menopausal. Menopausal improves the performance of Mixed1.9 + Pap in terms of AUC on validation (0.83705 vs. 0.83133). Surprisingly Menopausal doesn't improve the model performance of 25$^{th}$ percentile model in terms of AUC but it significantly increases the training specificity. For example, at 99.2% sensitivity the specificity jumps from 0.23 to 0.398. This can be explained by a ROC curve as illustrated in Graph-5.

Table-6 List of Partial Least Square models

| **Model** | **AUC (Train)** | **AUC (Val)** | **Dallas D2** |
|---|---|---|---|
| ***Model with Pap*** | | | |
| 25$^{th}$ percentile+ PAP | 0.90918 | 0.8268 | 0.82667 |
| Mixed1.0+Pap | 0.9118 | 0.82474 | 0.82667 |
| Mixed1.1+Pap | 0.9131 | 0.82288 | 0.80667 |
| Mixed1.2+Pap | 0.91603 | 0.82318 | 0.82333 |
| Mixed1.3+Pap | 0.91227 | 0.80215 | 0.87333 |
| Mixed1.4+Pap | 0.86483 | 0.81519 | 0.96667 |
| Mixed1.5+Pap | 0.8634 | 0.81516 | 0.94667 |
| Mixed1.6+Pap | 0.87142 | 0.81579 | 0.96 |
| Mixed1.7+Pap | 0.92451 | 0.8097 | 0.89333 |
| Mixed1.8+Pap | 0.92059 | 0.8118 | 0.88 |
| Mixed1.9+Pap | 0.91942 | 0.83133 | 0.84 |
| Mixed1.9+Pap+Mars | 0.92431 | 0.81855 | 0.87 |
| ***Model without Pap*** | | | |
| 25 percentile | 0.86017 | 0.75494 | 0.70667 |
| Mixed1.5 | 0.85548 | 0.73062 | 0.64667 |
| Mixed1.6 | 0.86879 | 0.74416 | 0.71667 |
| Mixed1.7 | 0.87521 | 0.72473 | 0.72 |
| Mixed1.8 | 0.87504 | 0.72214 | 0.73 |
| Mixed1.9 | 0.88369 | 0.76226 | 0.71 |
| | | | |
| PAP only | 0.8 | 0.78794 | 0.93333 |

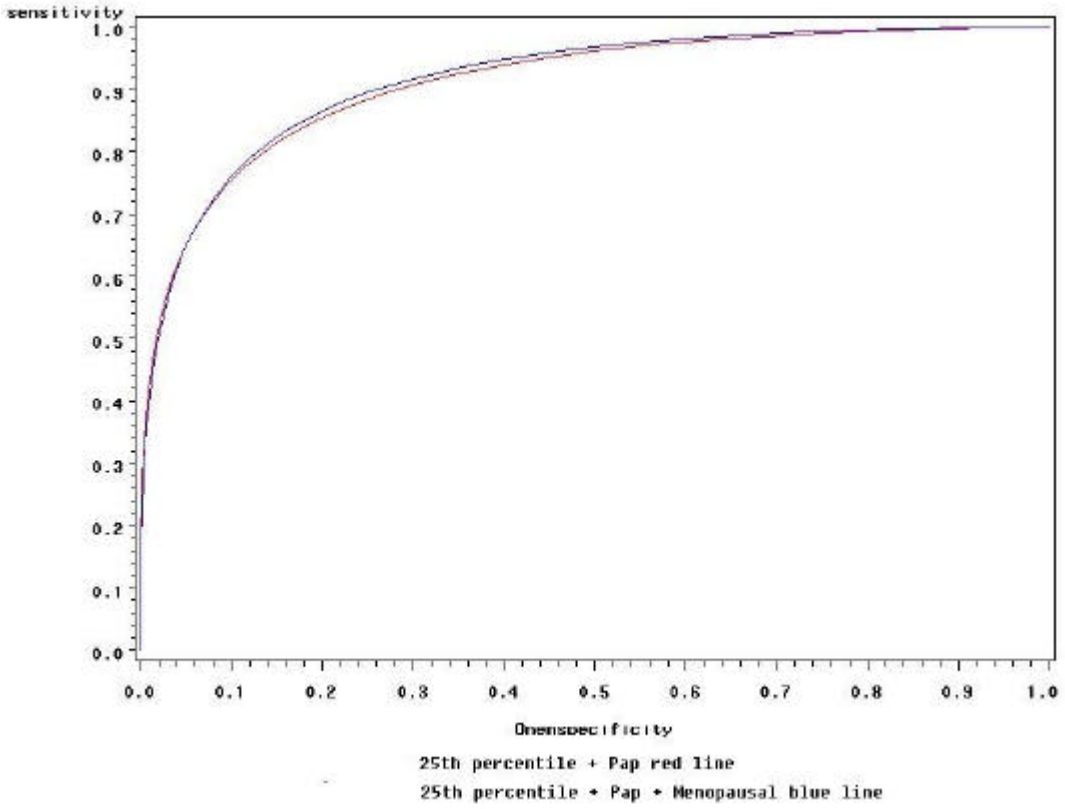
Table-7.1 Model performance with or without menopausal and age

| Model | AUC (Train) | AUC (Validation) |
|---|---|---|
| | | |
| $25^{th}$ percentile + Pap | 0.90918 | 0.82680 |
| $25^{th}$ percentile+ Pap + Menopausal | 0.90804 | 0.82437 |
| $25^{th}$ percentile | 0.86017 | 0.75494 |
| $25^{th}$ percentile+ Menopausal | 0.86157 | 0.75524 |
| | | |
| Mixed1.9+Pap | 0.91942 | 0.83133 |
| Mixed1.9+menopausal+Pap | 0.91739 | 0.83705 |
| Mxed1.9 | 0.88369 | 0.76226 |
| Mixed1.9+menopausal | 0.87501 | 0.76752 |
| Mixed1.9+Age | 0.87878 | 0.71537 |

Table-7.2 Model performance in terms of sensitivity and specificity

| Model | Sensitivity(train) | Specificity(train) |
|---|---|---|
| $25^{th}$ percentile + pap | 0.992 | 0.23 |
| | 0.955 | 0.5265 |
| $25^{th}$ percentile+pap+Menopaussal | 0.992 | 0.398 |
| | 0.955 | 0.544 |
| | | |
| M1.9+Pap | 0.992 | 0.323 |
| | 0.955 | 0.633 |
| M1.0+Pap+Menopausal | 0.992 | 0.31 |
| | 0.955 | 0.5664 |

From Graph-5, at high sensitivity the ROC curve for $25^{th}$ percentile + Pap + Menopausal model is higher than the model without Menopausal, while at low sensitivity it is the opposite. This can explain why the two models have almost equal AUC, but very different performance at high sensitivity.




Graph-5 ROC curve of 25$^{th}$ percentile model

We re-categorized the Pap smear test as shown in Table-8.1. The performance comparison is presented in Table-8.2. Compared with the untransformed Pap model, 25$^{th}$ Percentile + Category 6 has better performance in terms of AUC. Apart from Pap re-categorization, we applied power transformation, which is the simplest form of fractional polynomials to Pap. It can be seen from Table-9 that 25$^{th}$ percentile +Pap with power transformation is better than the original model without Pap transformation.

Table-8.1 New Categories of Pap smear test

| Cytology | NewCat 1 | NewCat 2 | NewCat 3 | NewCat 4 | NewCat 5 | NewCat 6 |
|---|---|---|---|---|---|---|
| 0 (Normal) | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 (Benigh changes) | 1 | 1 | 0 | 0 | 0 | 0 |
| 2 (ASCUS, not favor neoplasia) | 2 | 2 | 2 | 2 | 2 | 2 |
| 2.8(ASCUS,favor neoplasia) | 3 | 3 | 4 | 3 | 4 | 4 |
| 3 (LSIL) | 3 | 2 | 4 | 2 | 4 | 4 |
| 3.2 (AGUS) | 2 | 2 | 2 | 2 | 2 | 2 |
| 3.5 (ASC_H/cannot rule out HSIL) | 3 | 4 | 6 | 3 | 7 | 7 |
| 4 (HSIL) | 4 | 4 | 7 | 4 | 7 | 7 |
| 4.2 (Invasive Cancer) | 4 | 4 | 7 | 4 | 7 | 8 |

Table-8.2 Performance Comparison of models using new Pap categories

| Model | AUC Training | AUC Validation | Dallas D2 |
|---|---|---|---|
| $25^{th}$ percentile+Pap | 0.90918 | 0.82680 | 0.82667 |
| $25^{th}$ percentile+ Cate2 | 0.89414 | 0.82793 | 0.84667 |
| $25^{th}$ percentile+ Cate3 | 0.91230 | 0.83442 | 0.85 |
| $25^{th}$ percentile+ Cate4 | 0.90232 | 0.82304 | 0.81 |
| $25^{th}$ percentile+ Cate5 | 0.91134 | 0.83445 | 0.85333 |
| $25^{th}$ percentile +Cate6 | 0.91140 | 0.83475 | 0.85333 |

Table-9 Performance comparison of models w and w/o power transformation

| Model | AUC (Train) | AUC (Valid) | Dallas D2 |
|---|---|---|---|
| $25^{th}$ percentile+Pap | 0.90918 | 0.82680 | 0.82667 |
| $25^{th}$ percentile +Pap power 3 | 0.90149 | 0.84247 | 0.85333 |

In this study, we used bootstrap to estimate the thresholds of PLS models. Average specificity of the 1000 bootstrap samples is one important parameter to evaluate a model performance. Our research indicates that the threshold for the Normal (Pap=0) group is significantly different from other group. Therefore two different thresholds can be applied for Normal and other Pap groups separately. Additionally, FDA doesn't prohibit using more than two thresholds and even two or more model for different Pap groups.

Table-10 Estimate of threshold of Partial Least Square models

| Model | Bootstrap Threshold | Threshold (Remove Sen =1) | No. of samples with Sen =1 | Ave spec. | Ave spec. (remove sen=1) |
|---|---|---|---|---|---|
| $25^{th}$ percentile + Pap (Pap ? 0) | 0.178 | 0.236 | 300 | 0.339 | 0.368 |
| Mixed1.3+Pap (Pap ? 0) | 0.101 | 0.199 | 130 | 0.373 | 0.389 |
| Mixed1.6+Pap (Pap ? 0) | 0.222 | 0.264 | 131 | 0.352 | 0.362 |
| Mixed1.7+Pap (Pap ? 0) | 0.077 | 0.194 | 136 | 0.424 | 0.450 |
| Mixed1.8+Pap (Pap ? 0) | 0.041 | 0.23 | 130 | 0.416 | 0.446 |
| Mixed1.9+Pap (Pap ? 0) | 0.081 | 0.231 | 136 | 0.418 | 0.442 |
| Mixed1.9+Mars+Pap=2,2.8,3,3.2,3.5 | 0.136 | 0.234 | 418 | 0.377 | 0.447 |
| Mixed1.9+Mars+Pap=0,1 | -0.056 | N/A | 1000 | N/A | 0.393 |
| Mixed1.9+Mars+Pap=4, 4.2 | 0.356 | 0.423 | 441 | 0.355 | 0.443 |

All models listed in Table-10 has average specificity larger than 20%, therefore FDA's requirement that a model should have average specificity at least 20% can be easily met. In addition, all mixed models listed in Table-10 have average specificity above 30%. This further proves that mixing $25^{th}$ percentile variables and difference variables results in models with much better performance.

We used Mixed Model1.9 + Pap to evaluate sample size's impact on the estimate of model threshold. Our simulation study shows that as sample size increases, the

Table-11 Sample size's impact on model threshold estimation

| Model | Threshold | Average Specificity | No. of samples with sens<1 (total:1000) |
|---|---|---|---|
| 100% entire sample size | 0.081 | 0.418 | 864 |
| 75% entire sample size | 0.081 | 0.383 | 612 |
| 50% entire sample size | 0.3 | 0.53 | 579 |

number of bootstrap samples with sensitivity 100% decreases. Since we were using a small sample of 399 patients (CIN1 excluded) to do the bootstrap, it can be concluded that the current estimate is biased and the true threshold must be higher than our estimate. Table-10 also shows the estimates after excluding all the 100% sensitivity bootstrap samples.

In order to correct the bias of the estimate, we employed 632 bootstrap to improve the result. As an example, the threshold for Mixed1.9+Mars variable +Pap =2, 2.8,3, 3.2,3.5 from standard bootstrap estimation is 0.234, the threshold of original sample is 0.234. Therefore, 632 bootstrap estimate = 0.632*0.136+ 0.368*0.234 = 0.172
Similarly the 632 bootstrap estimate of the original $25^{th}$ percentile + Pap model is as follows,

632 bootstrap estimate = 0.632*0.178 + 0.368*0.274 = 0.213, where 0.178 and 0.274 are the standard bootstrap estimate and the threshold on the original sample respectively. The performance comparison of the two models in different Pap groups of the original sample is as follows. A portion of table-2 is also pasted here to show all available Pap groups.

Table-12.1 25$^{th}$ percentile + Pap model performance in different pap groups

| Group | threshold | Sens | Spec | TP | FN | TN | FP | Total |
|---|---|---|---|---|---|---|---|---|
| 0 | 0.21333 | 0.28571 | 0.72727 | 2 | 5 | 56 | 21 | 84 |
| 1 | 0.21333 | 1 | 0.72973 | 2 | 0 | 27 | 10 | 39 |
| 2 | 0.21333 | 1 | 0.38776 | 13 | 0 | 19 | 30 | 62 |
| 2.8 | 0.21333 | 1 | 0.08333 | 5 | 0 | 1 | 11 | 17 |
| 3 | 0.21333 | 0.98039 | 0.08475 | 50 | 1 | 5 | 54 | 110 |
| 3.2 | 0.21333 | 0 | 0.5 | 0 | 0 | 2 | 2 | 4 |
| 3.5 | 0.21333 | 1 | 0.16667 | 7 | 0 | 1 | 5 | 13 |
| 4 | 0.21333 | 1 | 0.08333 | 57 | 0 | 1 | 11 | 69 |
| 4.2 | 0.21333 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |

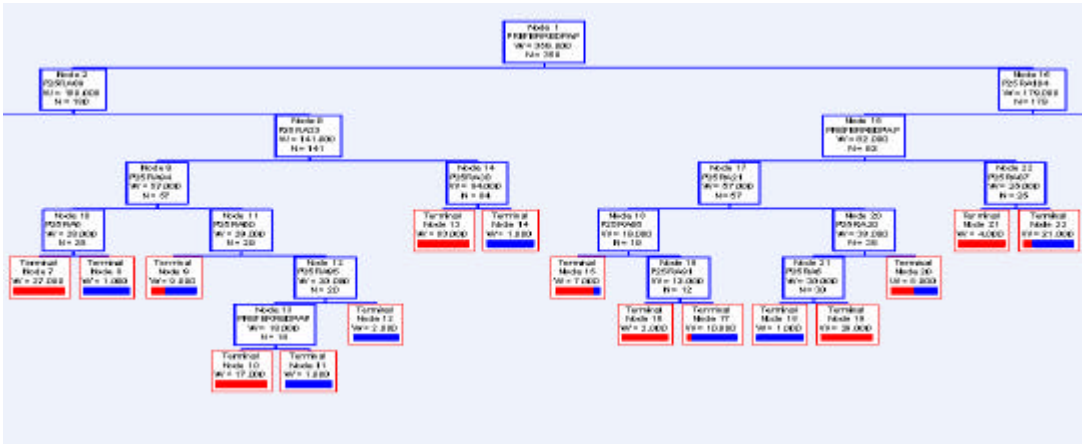
Table-12.2 Mixed1.9+Mars variable +Pap=2,2.8,3,3.2,3.5

| Group | Threshold | Sen | Spec | TP | FN | TN | FP | Total |
|---|---|---|---|---|---|---|---|---|
| 0 | 0.172 | 0.57143 | 0.55844 | 4 | 3 | 43 | 34 | 84 |
| 1 | 0.172 | 1 | 0.81081 | 2 | 0 | 30 | 7 | 39 |
| **2** | **0.172** | **1** | **0.32653** | **13** | **0** | **16** | **33** | **62** |
| **2.8** | **0.172** | **1** | **0.33333** | **5** | **0** | **4** | **8** | **17** |
| **3** | **0.172** | **0.98039** | **0.13559** | **50** | **1** | **8** | **51** | **110** |
| **3.2** | **0.172** | **0** | **0.75** | **0** | **0** | **3** | **1** | **4** |
| **3.5** | **0.172** | **1** | **0.33333** | **7** | **0** | **2** | **4** | **13** |
| 4 | 0.172 | 1 | 0 | 57 | 0 | 0 | 12 | 69 |
| 4.2 | 0.172 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |

| Pap groups |
|---|
| 0 (Normal) |
| 1 (Benigh changes) |
| 2 (ASCUS, not favor neoplasia) |
| 2.8 (ASCUS, favor neoplasia) |
| 3 (LSIL) |
| 3.2 (AGUS) |
| 3.5 (ASC_H/cannot rule out HSIL) |
| 4 (HSIL) |
| 4.2 (Invasive Cancer) |

Table 12.1 and 12.2 show that Mixed1.9+Mars variable +Pap performs much better in the middle Pap groups (Pap=1,2, 2.8, 3, 3.2, 3.5) than the original 25$^{th}$ percentile + Pap model. Our strategy is to use three thresholds for Mixed1.9+Mars +Pap. One for middle Pap groups, one for normal and the other for HSIL and invasive cancer. The SAS code for creating tables-12 is shown in Appendix VIII.

In the CART analysis, we included all the spectral variables and Pap. 10-folder cross validation option was chosen to build and evaluate the model performance. In order to have high sensitivity, a large tree was built. Therefore the model shrinkage is large.



Graph-6 CART TREE

Notice that Pap is on the root node since CART found it is the most effective predictor. The result of the CART analysis is as follows.

Table-13 Result for CART analysis

| Model | Sens (Train) | Spec (Train) | Sens (Valid) | Spec (Valid) |
|---|---|---|---|---|
| Exclude CIN1 | 0.99248 | 0.89381 | 0.6391 | 0.67699 |
| Exclude CIN1 | 0.87218 | 0.76991 | 0.73684 | 0.69912 |
| Include CIN1 | 0.95423 | 0.92920 | 0.64437 | 0.50 |

CIN1: 284 disease, 226 non-disease
No-CIN1 133 disease, 226 non-disease

The shrinkage on sensitivity is over 30% in order that the training sensitivity is 99% (or 95%). In order to improve CART performance, we did the following, 1) Set Pap and response variable as categorical variable 2) Allow linear combination of variables for splitting 3) Use PLS model outputs as predictor variables 4) Incorporating difference variables

Table-14 CART models performance comparison

| Model(CIN1 excluded) | Sens(Train) | Spec(Train) | Sens (Valid) | Spec (Valid) |
|---|---|---|---|---|
| P25, Pap | 0.99248 | 0.89381 | 0.6391 | 0.67699 |
| P25, diff, Pap | 0.96479 | 0.96460 | 0.66549 | 0.59292 |
| P25, diff, Pap, pred1 | 0.94737 | 0.83628 | 0.82707 | 0.78319 |
| P25,diff, Pap, pred2<br>P25,diff, Pap, pred1,pred2 | 0.90977<br>0.95489 | 0.85841<br>0.83786 | 0.75940<br>0.81955 | 0.75664<br>0.80531 |

Pred1: Model output for Mixed1.3+ Pap
Pred2: model output for Mixed1.3

From Table-14, the last model has least shrinkage on specificity, but the shrinkage on sensitivity is still above 10%. In addition, Spectrx's target of sensitivity on the validation set no less than 99% (or 95%), is not met.

As stated in Chapter III, we used CART's variable importance diagram to incorporate more variables for PLS model and at the same time exclude some low importance variables. Mixed1.9 was benefited from CART's variable importance diagram. A partial diagram is illustrated below.

Table-15 CART variable importance diagram

| | | |
|---|---|---|
| PAP | 47.79 | ||||||||||||||||||| |
| P25RA33 | 27.01 | |||||||||| |
| P25RA89 | 18.6 | ||||||| |
| P25RA88 | 17.67 | ||||||| |
| P25RA73 | 8.79 | ||| |
| Diff25 | 7.94 | || |
| P25RA72 | 7.29 | || |
| P25RA71 | 5.76 | | |
| Diff34 | 5.52 | | |
| Diff33 | 5.52 | | |
| Diff28 | 5.52 | | |
| Diff27 | 5.52 | | |
| Diff26 | 5.52 | | |
| Diff55 | 5.3 | | |
| Diff54 | 5.3 | | |
| P25RA93 | 5.22 | | |
| P25RA96 | 5.22 | | |
| Diff4 | 4.49 | | |
| Diff18 | 4.27 | | |
| Diff23 | 4.06 | | |
| Diff24 | 4.06 | | |
| Diff69 | 3.79 | | |
| Diff60 | 3.79 | | |
| P25RA58 | 3.56 | | |
| P25RA57 | 3.56 | | |

From Table-14, Pap is the most important variable and it is on top of the list. The rest are spectral variables that can be selectively incorporated into the PLS model.

In this study, we used linear kernel and polynomial kernel to build SVM model. Cost of constrain violation is equivalent to $g$ explained in Chapter III. Tolerance of termination criterion (epsilon) is set as 2. $g$ is the tuning parameter. By adjusting $g$ we obtained different pairs of sensitivity and specificity. From table-15, SVM models have almost no shrinkage. Linear kernel performs better than polynomial kernel.

Table-16 Performance of SVM models

| Kernel | Model | Cost of constrain violation ($g$) | Training | | Validation | | Dallas D2 | |
|---|---|---|---|---|---|---|---|---|
| | | | Sens | Spec | Sens | Spec | Sens | Spec |
| Linear | Pap only | 0.15 | 0.92647 | 0.41007 | 0.9661 | 0.4881 | 1 | 0.46667 |
| | P25+Pap | 0.24 | 0.91176 | 0.47482 | 0.9661 | 0.55952 | 1 | 0.56667 |
| | P25 | 0.41 | 0.95588 | 0.15827 | 0.98305 | 0.27381 | 0.9 | 0.1 |
| | P25+diff+Pap (full) | 0.15 | 0.89706 | 0.4964 | 0.91525 | 0.63095 | 1 | 0.6 |
| | P25+diff(full) | 0.34 | 0.72059 | 0.61871 | 0.72881 | 0.55952 | 0.9 | 0.16667 |
| | P25+Pap+diff(reduced) | 0.15 | 0.91176 | 0.5036 | 0.94915 | 0.58333 | 1 | 0.56667 |
| | P25+diff(reduced) | 0.34 | 0.91176 | 0.33813 | 0.94915 | 0.36905 | 0.9 | 0.1 |
| Polynomial | Pap only | 0.22 | 0.92647 | 0.41007 | 0.9661 | 0.4881 | 1 | 0.5 |
| | P25+Pap | 0.42 | 0.89706 | 0.4964 | 0.91525 | 0.63095 | 1 | 0.28571 |
| | P25 | 0.43 | 0.97059 | 0.11511 | 1 | 0.2619 | 1 | 0 |

# CHAPTER V

# FUTURE STUDY

One important application of the medical device is to locate cancer on the surface of the cervix. The device collects data from 56 spatial points on the surface of the cervix of each patient. Therefore we should do a point level analysis of the disease. Every point will be an observation in this new analysis. A cervix map can be drawn from the point level diagnosis. Some work has already been done on this. We implemented similar data manipulation procedure, such as binning and normalizing. Then we got a very large dataset on which PLS and Logistic regression model were applied.

Table-17 Point analysis model performance

| Model | AUC Train | AUC Validation |
|---|---|---|
| Logistic full model | 0.84416 | 0.79609 |
| Logistic model (stepwise) | 0.83577 | 0.80703 |
| PLS full model | 0.83493 | 0.80009 |
| PLS reduced model | 0.83378 | 0.79479 |

From the table, Logistic model and PLS model have very similar results however PLS model has slightly less shrinkage. The SAS code for point analysis data manipulation is shown in Appendix IX.

Boosting is a procedure that combines the outputs of many "weak" classifiers to produce a powerful classifier. Freund and Schapire (1997) proposed a popular boosting algorithm called " AdaBoost.M1". The details are as follows.

1. Initialize the observation weights $w_i = 1/N, i = 1,2,...,N$.

2. For m=1 to M:

(a) Fit a classifier $Gm(x)$ to the training data using weights $w_i$.

(b) Compute $err_m = \frac{\sum_{i=1}^{N} w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^{N} w_i}$

(c) Compute $a_m = \log((1-err_m)/err_m)$

(d) Set $w_i \leftarrow w_i . \exp[a_m . I(y_i \neq G_m(x_i))], i = 1,2,..., N.$

3. Output G(x) = sign $[\sum_{m=1}^{M} a_m G_m(x)]$

At each step, those observations that were misclassified by a classifier have their weights increased, whereas the weights are decreased for those that were correctly classified. Much has been written about the success of AdaBoost in producing accurate classifiers. In particular tree-based classifiers were most explored. Table-6 lists some PLS classifiers we developed. Future study should focus on PLS based classifiers as well as PLS-TREE-SVM combined classifiers.

Yuchun Tang (2005) successfully used Granular Computing and GSVM (Granular Support Vector Machine) for cancer related gene subsets extraction on microarray expression data. The idea of "granular" has been used in the cervical cancer study by Qu. However the usefulness of the method is limited due to the small sample size we had. Future researchers can consider employing granular computing to achieve better cancer classification. Granular computing represents information in the form of some aggregates such as subsets, subspaces, classes, or clusters of a universe and then solves the targeted problem in each information granule. Granular Support Vector Machine (GSVM) is a granular computing-based learning model. It combines the

principals from statistical learning theory (SVM) and granular computing theory in a systematic and formal way.

The future study can also use BARS (Bayesian Adaptive Regression Splines) to solve the classification problem. BARS is a Bayesian version of MARS model of Friedman (1991). It is a MCMC-based algorithm that samples from a suitable approximate posterior distribution on a knot set. BARS can be viewed as a powerful engine for searching for optimal knot set.

**References:**

[1] Nena M. Marin, Andrea Melbourne. (2005). *Diffuse reflectance patterns in cervical spectroscopy*. Gynecologic Oncology. Vol99, issue3, supplement 1, S116-S120.

[2] Trevor Hastie, Robert Tibshirani, Jerome Friedman, 2000, *The element of Statistical Learning*, 67-68

[3] Randall D. Tobias, *An Introduction to Partial Least Squares Regresión*, SAS Institute., Cary, NC

[4] SAS Institute Inc., *2002-2005 SAS OnlineDoc*, version 9.1.3

[5] Herve Abdi. *Partial Least Squares (PLS) Regresión*, 2003, The University of Texas at Dallas

[6] Salford Systems, *MARS User Guide*, 1-3

[7] Jerome H. Friedman (1991), Multivariate Adptive Regressio n Splines, The Annals of Statistics, Vol. 19, No.1, 1-141

[8] Breiman,L.,Friedman, J.h., Olshen, R.A. and Stone, C.J., 1984, *Classification and regression tree*,. Chapman & Hall (wadsworth, Inc.): New York, 1984

[9] Yisehac Yohannes, John Hoddinott, 1999, *Classification and Regression Trees: An Introduction*

[10] Roger J. Lewis, M.D., Ph.D., *An Introduction to Classification and Regression Tree (CART) Analysis*, 2000 Annual Meeting of the Society for Academic Emergency Medicine.

[11] Bradley Efron, Robert J. Tibshirani, 1993, *An Introduction to the Bootstrap*, Chapman & Hall

[12] Hosmer,D.W, Lemeshow,S.(2000). *Applied Logistic Regression(2nd edition)*. John Wiley & Sons, Inc. pp.160-164.

[13] Jerome Friedman, 1999, *Additive Logistic Regression: A statistical view of boosting.* The annals of statistics, Vol. 28, No. 2, 337-407

[14] Kai Qu, 2004, Some Contribution in the Classification Analysis of the Spectroscopic Evaluation of Cervical Cancer, Graduate Thesis

[15] Yuchun Tang, Yichuan Zhao, 2005, *Granular SVM-RFE Feature Selection Algorithm for Reliable Cancer-Related Gene Subsets Extraction on Microarray Expression Data*, Special Issue on Bioinformatics

[16] Sam Behseta, Robert E. Kass, 2005, *Testing equality of two functions using BARS*, Statistics in Medicine

[17] Mark L. Faupel, 2004, *Spectroscopic Evaluation of Cervical neoplasia*

## APPENDIX I: SAS CODE FOR INTIAL DATA MANIPULATION, VARIABLE REDUCTION AND PRE-SELECTION

```
/* Initial data manipulation and variable pre-selection
   File: test_model1_new_mani.sas
   Last update: 5/25/2005

   Created by Fan Xu
   Updated by Chenghong Shen

*/



%include 'c:\spectrx\fan\missing_mac.sas';
libname After 'c:\spectrx\Fan\Aftertrain\data3';

option nonotes;
options nonumber nodate;
%macro readdata(path1 = , path2 = , path3 = , file = , spacing = 10, dataout =,
                subselect = 1, pointselect = 0, disq = no, extype = manual, spectype = orig);

 data demo;
        infile "&path1&file" expandtabs lrecl = 10000 missover;
   input sub_id$ available unclean datec$ whole1 sitepath qa1 PriorPap PriorPaptype
                DayofPap DayofPaptype PreferredPap PreferredPaptype scjvisible
colpoadequacy Age Race
                menstrual Menopause Gravida Para Abort Birthcontrol Priorsurgery1
DaysPriorsurgery1
                Priorsurgery2 DaysPriorsurgery2 Priorsurgery3 DaysPriorsurgery3
                Priorsurgery4 DaysPriorsurgery4 Priorsurgery5 DaysPriorsurgery5 height weight
                smoking Cigarettesperday;
        d_id = substr(sub_id, 1, 1);
        year = substr(datec, 1, 4); month = substr(datec, 5, 2); day = substr(datec, 7, 2);
        date = mdy(month, day, year);
        %nmissing(varlist = available unclean whole1 sitepath qa1 PriorPap PriorPaptype
                DayofPap DayofPaptype PreferredPap PreferredPaptype scjvisible
colpoadequacy Age Race menstrual Menopause Gravida Para Abort Birthcontrol Priorsurgery1
DaysPriorsurgery1 Priorsurgery2 DaysPriorsurgery2 Priorsurgery3 DaysPriorsurgery3
Priorsurgery4 DaysPriorsurgery4 Priorsurgery5 DaysPriorsurgery5 height weight  smoking
Cigarettesperday, missing = -1 -2);
        if available and &subselect;
 run;

 proc sort data = demo; by sub_id; run;

 data _null_; set demo end = last;
   call symput('sub'||left(_n_), trim(left(sub_id)));
   if last then call symput('nsub', _n_);
 run;

 proc sort data = demo; by sub_id; run;

 data coordinates;
        infile 'c:\spectrx\fan\nci\hybrid\data3\HybridInterrogationPointCoordsmm.txt' expandtabs;
```

```
        input point x y;
run;

%do i = 1 %to &nsub;
  %put Read Data File For Subject #&i out of %left(&nsub) &&sub&i;

  Data org;
        infile "&path2.&&sub&i.._spectra_&spectype..txt" expandtabs lrecl = 100000;
        /*INFILE "&path2.&&sub&i.._spectra_autopeakrowdetect_notiszero2.txt"
                        EXPANDTABS LRECL=100000;*/
        %if %upcase(&disq) = NO %then %do;
           input point rf_1-rf_63 f1_1-f1_63 f2_1-f2_57 f3_1-f3_45;
                array rf rf_1-rf_63; array f1 f1_1-f1_63; array f2 f2_1-f2_57; array f3 f3_1-f3_45;
                %if &spacing = 5 %then %do;
                        %let t1 = 63; %let t2 = 63; %let t3 = 57; %let t4 = 45;
                %end;
                %else %if &spacing = 10 %then %do;
                        %let t1 = 31; %let t2 = 31; %let t3 = 28; %let t4 = 22;
                %end;
                %else %if &spacing = 20 %then %do;
                        %let t1 = 15; %let t2 = 15; %let t3 = 14; %let t4 = 11;
                %end;
        %end;

        %else %do;
           input point rf_1-rf_63 b1-b4 f1_1-f1_59 b5-b8 f2_1-f2_53 b9-b12 f3_1-f3_41;
                array rf rf_1-rf_63; array f1 f1_1-f1_59; array f2 f2_1-f2_53; array f3 f3_1-f3_41;
                %if &spacing = 5 %then %do;
                        %let t1 = 63; %let t2 = 59; %let t3 = 53; %let t4 = 41;
                %end;
                %else %if &spacing = 10 %then %do;
                        %let t1 = 31; %let t2 = 29; %let t3 = 26; %let t4 = 20;
                %end;
                %else %if &spacing = 20 %then %do;
                        %let t1 = 15; %let t2 = 14; %let t3 = 13; %let t4 = 10;
                %end;
        %end;
        %let t = %eval(&t1 + &t2 + &t3 + &t4);

        %spacingselfnorm ;
        sub_id = "&&sub&i";
  run;

  data pointcat;
        infile "&path3.&&sub&i.._excl_&extype..txt" expandtabs;
        input point reject;
  run;

  data org; merge org pointcat coordinates; by point; run;

  %meanpro(datain = org, dataout = m&i);

 %end;

  data model; merge demo %mf;
        by sub_id;
```

```
  run;


  data &dataout; set model;
          CIN31 = (whole1 = 3.5);
          CIN32 = (whole1 >= 3.2);
          high = (whole1 >= 3);
          highlow = (whole1 >= 2);
          low = whole1 in (2 2.5);
          nandb = whole1 in (0 1);
          nc = whole1 = 1;
          normal = whole1 = 0;
  run;

%mend;

%macro mf;
  %do j = 1 %to &nsub;
    m&j
  %end;
%mend;

%macro spacingselfnorm;

          array nrf nrf_1-nrf_&t1; array nf1 nf1_1-nf1_&t2; array nf2 nf2_1-nf2_&t3; array nf3 nf3_1-
nf3_&t4;
          array rnrf rnrf_1-rnrf_&t1; array rnf1 rnf1_1-rnf1_&t2; array rnf2 rnf2_1-rnf2_&t3; array
rnf3 rnf3_1-rnf3_&t4;

          %if &spacing = 5 %then %do;
                    do i = 1 to &t1; nrf(i) = rf(i); end;
                    do i = 1 to &t2; nf1(i) = f1(i); end;
                    do i = 1 to &t3; nf2(i) = f2(i); end;
                    do i = 1 to &t4; nf3(i) = f3(i); end;
          %end;
          %else %if &spacing = 10 %then %do;
                    do i = 1 to &t1; nrf(i) = (rf(2 * i - 1) + rf( 2 * i)) / 2; end;
                    do i = 1 to &t2; nf1(i) = (f1(2 * i - 1) + f1(2 * i)) / 2; end;
                    do i = 1 to &t3; nf2(i) = (f2(2 * i - 1) + f2(2 * i)) / 2; end;
                    do i = 1 to &t4; nf3(i) = (f3(2 * i - 1) + f3(2 * i)) / 2; end;
          %end;
          %else %do;
                    do i = 1 to &t1; nrf(i) = (rf(4 * i - 3) + rf( 4 * i - 2) + rf(4 * i - 1) + rf(4 * i)) / 4; end;
                              do i = 1 to &t2; nf1(i) = (f1(4 * i - 3) + f1(4 * i - 2) + f1(4 * i - 1) + f1(4 * i)) /
          4; end;
                    do i = 1 to &t3; nf2(i) = (f2(4 * i - 3) + f2(4 * i - 2) + f2(4 * i - 1) + f2( 4 * i)) / 4; end;
                    do i = 1 to &t4; nf3(i) = (f3(4 * i - 3) + f3(4 * i - 2) + f3(4 * i - 1) + f3( 4 * i)) / 4; end;
          %end;

          avgnrf = mean(of nrf_1-nrf_&t1); stdnrf = std(of nrf_1-nrf_&t1);
          avgnf1 = mean(of nf1_1-nf1_&t2); stdnf1 = std(of nf1_1-nf1_&t4);
          avgnf2 = mean(of nf2_1-nf2_&t3); stdnf2 = std(of nf2_1-nf2_&t4);
          avgnf3 = mean(of nf3_1-nf3_&t4); stdnf3 = std(of nf3_1-nf3_&t4);

          do i = 1 to &t1; rnrf(i) = nrf(i) / avgnrf; end;
          do i = 1 to &t2; rnf1(i) = nf1(i) / avgnf1; end;
```

```
        do i = 1 to &t3; rnf2(i) = nf2(i) / avgnf2; end;
        do i = 1 to &t4; rnf3(i) = nf3(i) / avgnf3; end;

%mend;


%macro meanpro(datain = , dataout = );

  proc means data = &datain noprint;
        var nrf_1-nrf  &t1 nf1  1-nf1  &t2 nf2  1-nf2  &t3 nf3  1-nf3  &t4
              rnrf_1-rnrf_&t1 rnf1_1-rnf1_&t2 rnf2_1-rnf2_&t3 rnf3_1-rnf3_&t4 avgnrf avgnf1-
              avgnf3 stdnrf stdnf3;
    output out = &dataout

    VAR = va1-va&t var1-var&t
    USS = us1-us&t usr1-usr&t
        LCLM = lclm1-lclm&t lclmr1-lclmr&t
        UCLM = uclm1-uclm&t uclmr1-uclmr&t
        STDERR = stderr1-stderr&t stderrr1-stderrr&t

        p10 = pta1-pta&t ptra1-ptra&t ptavgnrf ptavgnf1-ptavgnf3
        p25 = p25a1-p25a&t p25ra1-p25ra&t
    mean = ma1-ma&t mra1-mra&t mavgnrf mavgnf1-mavgnf3 mstdnrf mstdnf3
        cv = ca1-ca&t cra1-cra&t cavgnrf cavgnf1-cavgnf3
        p50 = p50a1-p50a&t p50ra1-p50ra&t p50avgnrf p50avgnf1-p50avgnf3
        p90 = pna1-pna&t pnra1-pnra&t pnavgnrf pnavgnf1-pnavgnf3
        qrange = qa1-qa&t qra1-qra&t qavgnrf qavgnf1-qavgnf3
        skewness = wa1-wa&t wra1-wra&t
        kurtosis = ka1-ka&t kra1-kra&t
        std = sa1-sa&t sra1-sra&t
        max = xa1-xa&t xra1-xra&t
    min = na1-na&t nra1-nra&t
    range = ra1-ra&t rra1-rra&t
        p1 = p1a1-p1a&t p1ra1-p1ra&t
        p5 = p5a1-p5a&t p5ra1-p5ra&t
        p75 = p75a1-p75a&t p75ra1-p75ra&t
        p95 = p95a1-p95a&t p95ra1-p95ra&t
        p99 = p99a1-p99a&t p99ra1-p99ra&t ;
        where reject in (&pointselect);
        by sub_id;

  run;

  proc univariate data = &datain noprint;
        var nrf_1-nrf  &t1 nf1  1-nf1  &t2 nf2  1-nf2  &t3 nf3  1-nf3  &t4
              rnrf  1-rnrf  &t1 rnf1_1-rnf1_&t2 rnf2_1-rnf2_&t3 rnf3_1-rnf3_&t4 avgnrf avgnf1-
            avgnf3 stdnrf stdnf3;
    output out = dataout  p23
        pctlpts=23  pctlpre = p23a1-p23a&t p23ra1-p23ra&t pctlname=
        ;
    where reject in (&pointselect);
        by sub_id;

  run;

  proc univariate data = &datain noprint;
```

```
        var nrf_1-nrf_&t1 nf1_1-nf1_&t2 nf2_1-nf2_&t3 nf3_1-nf3_&t4
                rnrf_1-rnrf_&t1 rnf1_1-rnf1_&t2 rnf2_1-rnf2_&t3 rnf3_1-rnf3_&t4 avgnrf avgnf1-
        avgnf3 stdnrf stdnf3;
    output out = dataout_p27
        pctlpts=27  pctlpre = p27a1-p27a&t p27ra1-p27ra&t pctlname=
        ;
    where reject in (&pointselect);
        by sub_id;

  run;

  proc univariate data = &datain noprint;
        var nrf_1-nrf_&t1 nf1_1-nf1_&t2 nf2_1-nf2_&t3 nf3_1-nf3_&t4
                rnrf_1-rnrf_&t1 rnf1_1-rnf1_&t2 rnf2_1-rnf2_&t3 rnf3_1-rnf3_&t4 avgnrf avgnf1-
avgnf3 stdnrf stdnf3;
    output out = dataout_p20
        pctlpts=20  pctlpre = p20a1-p20a&t p20ra1-p20ra&t pctlname=
        ;
    where reject in (&pointselect);
        by sub_id;

  run;

  proc univariate data = &datain noprint;
        var nrf_1-nrf_&t1 nf1_1-nf1_&t2 nf2_1-nf2_&t3 nf3_1-nf3_&t4
                rnrf_1-rnrf_&t1 rnf1_1-rnf1_&t2 rnf2_1-rnf2_&t3 rnf3_1-rnf3_&t4 avgnrf avgnf1-
        avgnf3 stdnrf stdnf3;
    output out = dataout_p15
        pctlpts=15  pctlpre = p15a1-p15a&t p15ra1-p15ra&t pctlname=
        ;
    where reject in (&pointselect);
        by sub_id;

  run;

  proc univariate data = &datain noprint;
        var nrf_1-nrf_&t1 nf1_1-nf1_&t2 nf2_1-nf2_&t3 nf3_1-nf3_&t4
                rnrf_1-rnrf_&t1 rnf1_1-rnf1_&t2 rnf2_1-rnf2_&t3 rnf3_1-rnf3_&t4 avgnrf avgnf1-
        avgnf3 stdnrf stdnf3;
    output out = dataout_p40
        pctlpts=40  pctlpre = p40a1-p40a&t p40ra1-p40ra&t pctlname=
        ;
    where reject in (&pointselect);
        by sub_id;

  run;

  proc univariate data = &datain noprint;
        var nrf_1-nrf_&t1 nf1_1-nf1_&t2 nf2_1-nf2_&t3 nf3_1-nf3_&t4
                rnrf_1-rnrf_&t1 rnf1_1-rnf1_&t2 rnf2_1-rnf2_&t3 rnf3_1-rnf3_&t4 avgnrf avgnf1-
avgnf3 stdnrf stdnf3;
    output out = dataout_p30
        pctlpts=30  pctlpre = p30a1-p30a&t p30ra1-p30ra&t pctlname=
        ;
    where reject in (&pointselect);
        by sub_id;
```

```
  run;

  proc univariate data = &datain noprint;
            var nrf_1-nrf_&t1 nf1_1-nf1_&t2 nf2_1-nf2_&t3 nf3_1-nf3_&t4
                     rnrf_1-rnrf_&t1 rnf1_1-rnf1_&t2 rnf2_1-rnf2_&t3 rnf3_1-rnf3_&t4 avgnrf avgnf1-
           avgnf3 stdnrf stdnf3;
     output out = dataout_p45
            pctlpts=45  pctlpre = p45a1-p45a&t p45ra1-p45ra&t pctlname=
            ;
     where reject in (&pointselect);
            by sub_id;

  run;

  data &dataout;
           merge &dataout dataout_p15 dataout_p23 dataout_p27 dataout_p30 dataout_p40
dataout  p45;
           by sub_id;
  run;



%mend;


%readdata (path1 = c:\spectrx\fan\Aftertrain\,

    /* Data for training  */
    path2 = c:\spectrx\workdir\DATA2\,
    path3 = c:\spectrx\workdir\manual2\,
           file = HybridFINAL  ClinicalData  dm  2.txt, spacing = 10, dataout = After.All, disq = yes,
    subselect = (unclean = 0 and whole1~= .));
```

## APPENDIX II: SAS CODE FOR CREATING DIFFERENCE VARIABLES

```
/* This is to implment the idea of dividing the cervical
   surface into 6 different areas and get substraction between
   peripheral and  each of the  five different areas

   file: test_diff_model_new_mani.sas
   Last update: 07/05/2005

   Created by: Chenghong Shen

*/

%include 'c:\spectrx\fan\missing_mac.sas';
libname After 'c:\spectrx\Fan\Aftertrain\diff';

option nonotes;
options nonumber nodate;
%macro readdata(path1 = , path2 = , path3 = , file = , spacing = 10, dataout =,
        subselect = 1, pointselect = 0, disq = no, extype = manual, spectype = orig);

data demo;
        infile "&path1&file" expandtabs lrecl = 10000 missover;
input sub_id$ available unclean datec$ whole1 sitepath qa1 PriorPap PriorPaptype
        DayofPap DayofPaptype PreferredPap PreferredPaptype scjvisible colpoadequacy Age
Race menstrual Menopause Gravida Para Abort Birthcontrol Priorsurgery1 DaysPriorsurgery1
Priorsurgery2 DaysPriorsurgery2 Priorsurgery3 DaysPriorsurgery3
Priorsurgery4 DaysPriorsurgery4 Priorsurgery5 DaysPriorsurgery5 height weight  smoking
Cigarettesperday;
        d_id = substr(sub_id, 1, 1);
        year = substr(datec, 1, 4); month = substr(datec, 5, 2); day = substr(datec, 7, 2);
        date = mdy(month, day, year);
        %nmissing(varlist = available unclean whole1 sitepath qa1 PriorPap PriorPaptype
                DayofPap DayofPaptype PreferredPap PreferredPaptype scjvisible
colpoadequacy Age Race menstrual Menopause Gravida Para Abort Birthcontrol Priorsurgery1
DaysPriorsurgery1 Priorsurgery2 DaysPriorsurgery2 Priorsurgery3 DaysPriorsurgery3
Priorsurgery4 DaysPriorsurgery4 Priorsurgery5 DaysPriorsurgery5 height weight smoking
Cigarettesperday, missing = -1 -2);
        if available and &subselect;
run;

proc sort data = demo; by sub_id; run;

data   null  ; set demo end = last;
   call symput('sub'||left(  n  ), trim(left(sub_id)));
   if last then call symput('nsub', _n_);
run;

proc sort data = demo; by sub_id; run;

data coordinates;
        infile 'c:\spectrx\fan\nci\hybrid\data3\HybridInterrogationPointCoordsmm.txt' expandtabs;
        input point x y;
run;
```

```
%do i = 1 %to &nsub;
  %put Read Data File For Subject #&i out of %left(&nsub) &&sub&i;

Data org;
        infile "&path2.&&sub&i.._spectra_&spectype..txt" expandtabs lrecl = 100000;
        /*INFILE "&path2.&&sub&i.._spectra_autopeakrowdetect_notiszero2.txt"
                        EXPANDTABS LRECL=100000;*/
        %if %upcase(&disq) = NO %then %do;
           input point rf_1-rf_63 f1_1-f1_63 f2_1-f2_57 f3_1-f3_45;
           array rf rf_1-rf_63; array f1 f1_1-f1_63; array f2 f2_1-f2_57; array f3 f3_1-f3_45;
                %if &spacing = 5 %then %do;
                        %let t1 = 63; %let t2 = 63; %let t3 = 57; %let t4 = 45;
                %end;
                %else %if &spacing = 10 %then %do;
                        %let t1 = 31; %let t2 = 31; %let t3 = 28; %let t4 = 22;
                %end;
                %else %if &spacing = 20 %then %do;
                        %let t1 = 15; %let t2 = 15; %let t3 = 14; %let t4 = 11;
                %end;
        %end;

        %else %do;
           input point rf_1-rf_63 b1-b4 f1_1-f1_59 b5-b8 f2_1-f2_53 b9-b12 f3_1-f3_41;
              array rf rf_1-rf_63; array f1 f1_1-f1_59; array f2 f2_1-f2_53; array f3 f3_1-f3_41;
                %if &spacing = 5 %then %do;
                        %let t1 = 63; %let t2 = 59; %let t3 = 53; %let t4 = 41;
                %end;
                %else %if &spacing = 10 %then %do;
                        %let t1 = 31; %let t2 = 29; %let t3 = 26; %let t4 = 20;
                %end;
                %else %if &spacing = 20 %then %do;
                        %let t1 = 15; %let t2 = 14; %let t3 = 13; %let t4 = 10;
                %end;
        %end;
        %let t = %eval(&t1 + &t2 + &t3 + &t4);

        %spacingselfnorm ;
        sub_id = "&&sub&i";
run;

data pointcat;
        infile "&path3.&&sub&i.._excl_&extype..txt" expandtabs;
        input point reject;
run;

data org;
        set org;
        keep sub_id point rnrf_1-rnrf_31 rnf1_1-rnf1_29 /*rnf2_1-rnf2_26 */ rnf3_1-rnf3_20;
run;


data org; merge org pointcat coordinates; by point; run;

data Peripheral;
         set org;
   where point in (1 2 3 4 5 11 12 19 20 28 29 37 38 45 46 52 53 54 55 56) and reject in (0);
```

```
        *keep rf_1-rf_63 b1-b4 f1_1-f1_59 b5-b8 f2_1-f2_53 b9-b12 f3_1-f3_41;
run;

data Central;
        set org;
        where point in (23 24 25 32 33 34) and reject in (0);
        *keep rf_1-rf_63 b1-b4 f1_1-f1_59 b5-b8 f2_1-f2_53 b9-b12 f3_1-f3_41;
run;

data Topleft;
        set org;
        where point in (6 7 8 13 14 15 21 22) and reject in (0);
        *keep rf_1-rf_63 b1-b4 f1_1-f1_59 b5-b8 f2_1-f2_53 b9-b12 f3_1-f3_41;
run;

data Topright;
        set org;
        where point in (9 10 16 17 18 26 27) and reject in (0);
        *keep rf_1-rf_63 b1-b4 f1_1-f1_59 b5-b8 f2_1-f2_53 b9-b12 f3_1-f3_41;
run;

data Bottomleft;
        set org;
        where point in (30 31 39 40 41 47 48) and reject in (0);
        *keep rf_1-rf_63 b1-b4 f1_1-f1_59 b5-b8 f2_1-f2_53 b9-b12 f3_1-f3_41;
run;

data Bottomright;
  set org;
        where point in (35 36 42 43 44 49 50 51) and reject in (0);
        *keep rf_1-rf_63 b1-b4 f1_1-f1_59 b5-b8 f2_1-f2_53 b9-b12 f3_1-f3_41;
run;

data Left;
        set org;
        where point in (13 21 22 30 31 39) and reject in (0);
run;

data Right;
        set org;
        where point in (18 26 27 35 36 44) and reject in (0);
run;

data Top;
        set org;
        where point in (6 7 8 9 10 14 15 16 17) and reject in (0);
run;

data Bottom;
        set org;
        where point in (40 41 42 43 47 48 49 50 51) and reject in (0);
run;


%meanpro(datain = Peripheral, dataout = m_Peri&i);
```

```
%meanpro(datain = Central, dataout = m_Central&i);
%meanpro(datain = Topleft, dataout = m_TopL&i);
%meanpro(datain = Topright, dataout = m_TopR&i);
%meanpro(datain = Bottomleft, dataout = m_BottomL&i);
%meanpro(datain = Bottomright, dataout = m_BottomR&i);

%meanpro(datain = Left, dataout = m_Left&i);
%meanpro(datain = Right, dataout = m_Right&i);
%meanpro(datain = Top, dataout = m_Top&i);
%meanpro(datain = Bottom, dataout = m_Bottom&i);

data P_Central;
        set m_Peri&i m_Central&i;
run;

%getdiff(datain = p_central, dataout = p_central_diff);

data P_left;
        set m_Peri&i m_Left&i;
run;

%getdiff(datain = p_left, dataout = p_left_diff);

data P_Right;
        set m_Peri&i m_Right&i;
run;

%getdiff(datain = p_right, dataout = p_right_diff);

data P_Top;
        set m_Peri&i m_Top&i;
run;

%getdiff(datain = p_top, dataout = p_top_diff);

data P_Bottom;
        set m_Peri&i m_Bottom&i;
run;

%getdiff(datain = p_bottom, dataout = p_bottom_diff);

data P_Topleft;
        set m_Peri&i m_TopL&i;
run;

%getdiff(datain = p_Topleft, dataout = p_Topleft_diff);

data P_Topright;
        set m_Peri&i m_TopR&i;
run;

%getdiff(datain = p_Topright, dataout = p_Topright_diff);

data P_Bottomleft;
        set m_Peri&i m_BottomL&i;
run;
```

```
 %getdiff(datain = p_Bottomleft, dataout = p_Bottomleft_diff);

 data P_Bottomright;
        set m_Peri&i m_BottomR&i;
 run;

 %getdiff(datain = p_Bottomright, dataout = p_Bottomright_diff);

 data CombineDiff;
   set p_central_diff p_left_diff p_right_diff p_top_diff
       p_bottom_diff p_Topleft_diff p_Topright_diff p_Bottomleft_diff p_Bottomright_diff;
 run;

 proc means data = CombineDiff noprint;
                var r1-r80;
                output out= datasub&i

                max= m1-m80;
                by sub_id;
 run;


%end;

data model;
        merge demo %mf;
        by sub_id;
run;


data &dataout; set model;
        CIN31 = (whole1 = 3.5);
        CIN32 = (whole1 >= 3.2);
        high = (whole1 >= 3);
        highlow = (whole1 >= 2);
        low = whole1 in (2 2.5);
        nandb = whole1 in (0 1);
        nc = whole1 = 1;
        normal = whole1 = 0;
run;


%mend;


%macro mf;
  %do j = 1 %to &nsub;
    datasub&j
  %end;
%mend;

%macro spacingselfnorm;

        array nrf nrf_1-nrf_&t1; array nf1 nf1_1-nf1_&t2; array nf2 nf2_1-nf2_&t3; array nf3 nf3_1-
nf3_&t4;
```

```
        array rnrf rnrf_1-rnrf_&t1; array rnf1 rnf1_1-rnf1_&t2; array rnf2 rnf2_1-rnf2_&t3; array
rnf3 rnf3_1-rnf3_&t4;

        %if &spacing = 5 %then %do;
                do i = 1 to &t1; nrf(i) = rf(i); end;
                do i = 1 to &t2; nf1(i) = f1(i); end;
                do i = 1 to &t3; nf2(i) = f2(i); end;
                do i = 1 to &t4; nf3(i) = f3(i); end;
        %end;
        %else %if &spacing = 10 %then %do;
                do i = 1 to &t1; nrf(i) = (rf(2 * i - 1) + rf( 2 * i)) / 2; end;
                do i = 1 to &t2; nf1(i) = (f1(2 * i - 1) + f1(2 * i)) / 2; end;
                do i = 1 to &t3; nf2(i) = (f2(2 * i - 1) + f2(2 * i)) / 2; end;
                do i = 1 to &t4; nf3(i) = (f3(2 * i - 1) + f3(2 * i)) / 2; end;
        %end;
        %else %do;
                do i = 1 to &t1; nrf(i) = (rf(4 * i - 3) + rf( 4 * i - 2) + rf(4 * i - 1) + rf(4 * i)) / 4; end;
                do i = 1 to &t2; nf1(i) = (f1(4 * i - 3) + f1(4 * i - 2) + f1(4 * i - 1) + f1( 4 * i)) / 4;
                end;
                do i = 1 to &t3; nf2(i) = (f2(4 * i - 3) + f2(4 * i - 2) + f2(4 * i - 1) + f2( 4 * i)) / 4;
                end;
                do i = 1 to &t4; nf3(i) = (f3(4 * i - 3) + f3(4 * i - 2) + f3(4 * i - 1) + f3( 4 * i)) / 4;
                end;
        %end;

        avgnrf = mean(of nrf_1-nrf_&t1); stdnrf = std(of nrf_1-nrf_&t1);
        avgnf1 = mean(of nf1  1-nf1  &t2); stdnf1 = std(of nf1  1-nf1  &t4);
        avgnf2 = mean(of nf2  1-nf2  &t3); stdnf2 = std(of nf2  1-nf2  &t4);
        avgnf3 = mean(of nf3_1-nf3_&t4); stdnf3 = std(of nf3_1-nf3_&t4);

        do i = 1 to &t1; rnrf(i) = nrf(i) / avgnrf; end;
        do i = 1 to &t2; rnf1(i) = nf1(i) / avgnf1; end;
        do i = 1 to &t3; rnf2(i) = nf2(i) / avgnf2; end;
        do i = 1 to &t4; rnf3(i) = nf3(i) / avgnf3; end;

%mend;


%macro meanpro(datain = , dataout = );

  proc means data = &datain noprint;
          var rnrf  1-rnrf  31 rnf1_1-rnf1_29 /*rnf2_1-rnf2_26 */ rnf3_1-rnf3_20;
    output out = &dataout
    mean =  x1-x80;
          /*p10 =   pta1-pta80;
          p25 =   p25a1-p25a80;
    mean =  &datain.1-&datain.80;*/
          by sub_id;

  run;

%mend;


%macro getdiff(datain = , dataout = );
        proc means data = &datain noprint;
```

```
                var x1-x80;
                output out= &dataout

        range= r1-r80;
                by sub_id;
        run;

%mend;

%readdata (path1 = c:\spectrx\fan\Aftertrain\,

    /* Data for training  */
    path2 = c:\spectrx\workdir\DATA2\,
    path3 = c:\spectrx\workdir\manual2\,
        file = HybridFINAL  ClinicalData  dm  2.txt, spacing = 10, dataout = After.All, disq = yes,
    subselect = (unclean = 0 and whole1~= .));
```

## APPENDIX III MATLAB CODE FOR DRAWING SPECTRA PLOTS

```
% This is a program to plot spectra chart in matlab
%   file:plotspectra.m
%
%   Created by: Chenghong Shen
%   Last update: 09/04/2005

function export_plot_chart

SPECTRAGRAPHPATH=(['C:\Spectrx\workdir\graph\']);
SPECTRADATAPATH=(['C:\Spectrx\workdir\DATA2\']);
AUTOMATEDEXCLUDED=(['C:\Spectrx\workdir\manual2\']);

%Overall 80 coefficients
%This model uses P25 of Reflectance, 340nm and 460nm Fluorescence
%measurements.  It was trained on the Hybrid data without CIN1s
constant=2.103443657;
threshold=0.11
coeff=[-0.986547475
   -0.482468666
   0.162711119
   0.642237349
   0.871837088
   0.092322399
   -0.459286846
   -0.853936994
   1.15876166
   0.335249267
   0.915496024
   0.596199987
   -0.376393612
   -0.398701107
   0.082313383
   0.16729728
   -1.386010807
   -3.552869049
   -0.329499821
   4.487880515
   3.664079111
   -2.980851504
   -1.853662088
   0.915418514
   -0.287330533
   -0.848340018
   -0.041982258
   -0.231096379
   -0.241955144
   0.281856024
   1.027978808

   -0.579192242
   -0.048065736
   0.295798978
   0.098515583
```

```
-0.009505274
0.551944505
0.184169494
-0.880223812
-1.570954955
-2.719168833
0.710060575
0.90152155
3.117555062
4.054090538
2.716968137
1.260520388
-4.809582946
-2.687065794
-0.73753886
-0.851781249
1.495971633
0.555296171
0.189944767
-2.440063316
-0.296869935
-4.968964119
-1.458079717
2.868168143
2.88932203

0.521407989
0.175938844
-0.803212261
0.018519374
0.764413486
1.360873394
-0.532511395
-3.748217301
-4.745703179
5.404546642
-0.239467757
3.426820237
-2.085249695
-0.567166879
-1.824066766
-1.262546158
1.567865173
1.789646247
0.011337185
1.980897515];

d=dir([SPECTRAGRAPHPATH,'*reflectance.txt']);
%e=dir([AUTOMATEDEXCLUDED,'*.txt']);

for i=1:length(d)

   % sub_id reflectance.txt is in the format of wavelength intensity ...

   data=csvread([SPECTRAGRAPHPATH d(i).name],1,0);
```

```
id=d(i).name(1,1:4);
disp(id)

% Get predicted value for this subject using Fan's Model1
% Apply coefficents and calculate the predicted values
clear refl_data_summed f1_data_summed f3_data_summed
buffer=textread([strcat(SPECTRADATAPATH,id,'_spectra_orig.txt') ]);
excluded=textread([strcat(AUTOMATEDEXCLUDED,id,'_excl_manual.txt')]);

% POINT 1-63, 1-4, 1-59, 5-8, 1-53, 9-12, 1-41
nrf_data=buffer(:,2:64);
nf1_data=buffer(:,69:127);
nf2_data=buffer(:,132:184);
nf3_data=buffer(:,189:229);


for x=1:31
   nrf_data_summed(:,x)=(nrf_data(:,(2*x-1))+nrf_data(:,(2*x)))/2;
end

for x=1:29
   nf1_data_summed(:,x)=(nf1_data(:,(2*x-1))+nf1_data(:,(2*x)))/2;
end

for x=1:26
   nf2_data_summed(:,x)=(nf2_data(:,(2*x-1))+nf2_data(:,(2*x)))/2;
end

for x=1:20
   nf3_data_summed(:,x)=(nf3_data(:,(2*x-1))+nf3_data(:,(2*x)))/2;
end


%Mean normalize spectra
for j=1:56
   rnrf_data_summed(j,:)=nrf_data_summed(j,:)./mean(nrf_data_summed(j,:));
   rnf1_data_summed(j,:)=nf1_data_summed(j,:)./mean(nf1_data_summed(j,:));
   rnf2_data_summed(j,:)=nf2_data_summed(j,:)./mean(nf2_data_summed(j,:));
   rnf3_data_summed(j,:)=nf3_data_summed(j,:)./mean(nf3_data_summed(j,:));
end


%Remove excluded points from data set

r=1;
excludedpoints=0;
for q=1:56;
   if excluded(q,2)>0
      % if (excluded(q,2)/round(excluded(q,2))==1)
         excludedpoints(r)=q;
         r=r+1;
      % end
```

```
    end
end


if max(excludedpoints)>0
    rnrf_data_summed(excludedpoints,:)=[];
    rnf1_data_summed(excludedpoints,:)=[];
    rnf2_data_summed(excludedpoints,:)=[];
    rnf3_data_summed(excludedpoints,:)=[];
end

%Generate 25th Percentile of subject data
rnrf_p25=Model_percentile(rnrf_data_summed,25);
rnf1_p25=Model_percentile(rnf1_data_summed,25);
rnf2_p25=Model_percentile(rnf2_data_summed,25);
rnf3_p25=Model_percentile(rnf3_data_summed,25);


% Form final prediction variables
Final = [rnrf_p25 rnf1_p25 rnf2_p25 rnf3_p25];
FinalVar = [Final(1:60) Final(87:106)];


predicted=constant+FinalVar*coeff
if ((predicted-threshold)>0 )
    disease = 'Cancer'
else
    disease = 'Not Cancer'
end

head=csvread([strcat(SPECTRAGRAPHPATH,id,'_headinfo.txt') ], 1, 0);

[a,b] = size(head)
subplot(2,2,1)
hold on


for j=1:b
    if j==1
        m=b
    else
        m=j-1
    end

    if ( head(2,j)>=3 & head(2,j)<=4 )
        plot(data(:,1), data(:,m+1),'r');
    elseif(head(2,j)==2 || head(2,j)==2.5)
        plot(data(:,1), data(:,m+1),'y');
    elseif(head(2,j)<2.5)
        plot(data(:,1), data(:,m+1));
    end
end
```

```
title(strcat('Reflectance for subj', id, ' (', disease, ')'));


% sub_id fluro1.txt consists of wavelengh intensity ...
data=csvread([strcat(SPECTRAGRAPHPATH,id,'_Fluor1.txt') ],1,0);

subplot(2,2,2)
hold on
for j=1:b
   if j==1
      m=b
   else
      m=j-1
   end

   if ( head(2,j)>=3 & head(2,j)<=4 )
      plot(data(:,1), data(:,m+1),'r');
   elseif(head(2,j)==2 || head(2,j)==2.5)
      plot(data(:,1), data(:,m+1),'y');
   elseif(head(2,j)<2.5)
      plot(data(:,1), data(:,m+1));
   end
end

title(strcat('Fluor1 for subj', id, ' (', disease, ')'));

% sub_id fluro2.txt consists of wavelengh intensity ...
data=csvread([strcat(SPECTRAGRAPHPATH,id,'_Fluor2.txt') ],1,0);
subplot(2,2,3)
hold on
for j=1:b
   if j==1
      m=b
   else
      m=j-1
   end

   if ( head(2,j)>=3 & head(2,j)<=4 )
      plot(data(:,1), data(:,m+1),'r');
   elseif(head(2,j)==2 || head(2,j)==2.5)
      plot(data(:,1), data(:,m+1),'y');
   elseif(head(2,j)<2.5)
      plot(data(:,1), data(:,m+1));
   end
end

title(strcat('Fluor2 for subj', id, ' (', disease, ')'));

% sub_id fluro3.txt consists of wavelengh intensity ...
data=csvread([strcat(SPECTRAGRAPHPATH,id,'_Fluor3.txt') ],1,0);
subplot(2,2,4)
hold on
for j=1:b
   if j==1
```

```
        m=b
    else
        m=j-1
    end

    if ( head(2,j)>=3 & head(2,j)<=4 )
        plot(data(:,1), data(:,m+1),'r');
    elseif(head(2,j)==2 || head(2,j)==2.5)
        plot(data(:,1), data(:,m+1),'y');
    elseif(head(2,j)<2.5)
        plot(data(:,1), data(:,m+1));
    end
  end

  title(strcat('Fluor3 for subj', id, ' (', disease, ')'));
  h=figure


end
```

## APPENDIX IV SAS CODE FOR PLS REGRESSION, LOGISTIC REGRESSION, CROSS VALIDATION, 10-FOLDER CROSS-VALIDATION AS WELL AS FINDING AUC, SENSITIVITY AND SPECIFICITY

```
/* This is a macro to train and valid a PLS model

   file name: PLStrainvalid_mac.sas
   last updated: Sep. 16, 2005

   created by: Fan Xu
   updated by  Chenghong Shen  */

%include 'c:\spectrx\fan\macros\rocest_mac.sas';
%macro PLStrainvalid(train = train, valid = valid, var = , print = yes,
        response = whole, qtow = no, roc = no, stat = max);

        /*options nonotes;*/
        data comb; set &train &valid(in = v);
                if v then &response = .;
                valid = v;
        run;


        ods listing close;

        ods output
          PercentVariation  = pctvar
          ParameterEstimates = solution;

        ODS TRACE OFF;
        proc pls data = comb /*noprint*/ /*cv = split(10)*/ cv=one /*details*/  /*nefac = 10*/;
                model &response = &var /SOLUTION;
                output out = lout predicted = pred /*xscore= t*/;
        run;

        ods listing;

        data pred1; set lout; if valid; keep pred; run;
        data p; merge pred1 &valid; run;

        data l; set lout; if ~valid; run;

   /*
   /  Just reformat the coefficients.
   /-----------------------------------------------------*/
   data solution; set solution;
     *&response log  RAI 8.5;
     *if (RowName = 'Intercept') then delete;
     *rename RowName = Predictor &response = B;
   run;

   %if %upcase(&qtow) = YES %then %do;
                proc means data = l noprint;
                        class sub_id;
```

```
                    var &response high highlow pred pap;
                    %if %upcase(&stat) = MAX %then %do;
                        output out = l max = &response high highlow pred pap;
                    %end;
                    %else %do;
                        output out = l max = &response high highlow maxpred pap &stat
                        = b1-b3 pred;
                    %end;
                run;
                proc means data = p noprint;
                    class sub_id;
                    var &response high highlow pred pap;
                    %if %upcase(&stat) = MAX %then %do;
                        output out = p max = &response high highlow pred pap;
                    %end;
                    %else %do;
                        output out = p max = &response high highlow maxpred pap &stat
                        = b1-b3 pred;
                    %end;
                run;
                data l; set l; where   type   = 1; run;
                data p; set p; where _type_ = 1; run;
        %end;

        %if %upcase(&roc) = YES %then %do;
                %rocest(datain = l, tests = pred, gold = high, dataout = train);
                %rocest(datain = l, tests = pred, gold = highlow, dataout = trainhl);

                %rocest(datain = p, tests = pred, gold = high, dataout = valid);
                %rocest(datain = p, tests = pred, gold = highlow, dataout = validhl);

                %if %upcase(&print) = YES %then %do;
                    title "Training CIN2+ AUC";
                    proc print data = train; run;
                    title "Training CIN1+ AUC";
                    proc print data = trainhl; run;
                    title "Validation CIN2+ AUC";
                    proc print data = valid; run;
                    title "Validation CIN1+ AUC";
                    proc print data = validhl; run;
                %end;
        %end;


%mend;



/* This is a macro modified from Delong's example
   ref: "Comparing the Areas Under Two or More Correlated
   Receiver Operating Characteristics Curves: A Nonparametric
   Approach" by Delong, et. al, Biometrics 44, 837-845
     &datain     set to the name of the data set.
     &tests      set to the names of the screen tests.
     &gold       set to a zero/one indicator for gold standard
```

```
    File: rocest_mac.sas
*/

%macro rocest(datain = , tests = , gold = , dataout = roc, select = 1, negative = no);

data temp; set &datain;
           if &select;
run;

data comp; set temp(keep = &tests &gold);
 if &gold ^= 0 and &gold ^= 1 then do;
   *put 'goldicator is not zero/one';
   delete;
 end;
 %if %upcase(&negative) = YES %then %do;
           &tests = - &tests;
 %end;

proc iml;

start mwcomp(psi,z);
 *;
 * program to compute the man-whitney components  ;
 * z is (nn by 2);
 *  z[,1] is the column of data values;
 *  z[,2] is the column of goldicator variables;
 *   z[i,2]=1 if the observation is from the x population;
 *   z[i,2]=0 if the observation is from the y population;
 *
 * psi is the returned vector of u-statistic components;

 rz  = ranktie( z[,1] );                  * average ranks;
 nx  = sum( z[,2] );                      * num. of x's ;
 ny  = nrow(z)-nx;                        * num of y's  ;
 loc = loc( z[,2]=1 );                    * x goldexes   ;
 psi = j(nrow(z),1,0);
 psi[loc] = (rz[loc] - ranktie(z[loc,1]))/ny;   * x components ;
 loc = loc( z[,2]=0 );                    * y goldexes    ;
 psi[loc] = ( nx+ranktie(z[loc,1])-rz[loc])/nx; * y components ;
 free rz loc nx ny;                       * free space   ;
 finish;

start mwvar(t,v,nx,ny,z);
 *;
 * compute man-whitney statistics and variance;
 * input z, n by (k+1);
 *  z[,1:k] are the different variables;
 *  z[,k+1] are goldicator values,
 *    1 if the observation is from population x and ;
 *    0 if the observation is from population y;
 * t is the k by k vector of estimated statistics;
 *  the (i,j) entry is the MannWhitney statistic for the
 *  i-th column when used with the j-th column. The only
 *  observations with nonmissing values in each column are
 *  used. The diagonal elements are, hence, based only on the
 *  single column of values.
```

```
* v is the k by k estimated variance matrix;
* nx is the matrix of x population counts on a pairwise basis;
* ny is the matrix of y population counts on a pairwise basis;

k   = ncol(z)-1;
gold = z[,k+1];
v   = j(k,k,0); t=v; nx=v; ny=v;

* The following computes components after pairwise deletion of
*  observations with missing values. If either there are no missing
*  values or it is desired to use the components without doing
*  pairwise deletion first, the nested do loops could be evaded.
*;
do i=1 to k;
  do j=1 to i;
     who = loc( (z[,i]^=.)#(z[,j]^=.) );    * nonmissing pairs;
     run mwcomp(psii,(z[,i]||gold)[who,]);   * components;
     run mwcomp(psij,(z[,i]||gold)[who,]);
     inow = gold[who,];                      * x's and y's;
     m = inow[+];                          * current x's;
     n = nrow(psii)-m;                     * current y's;
     nx[i,j] = m; ny[i,j] = n;
     mi = (psii#inow)[+] / m;              * means;
     mj = (psij#inow)[+] / m;
     t[i,j] = mi; t[j,i] = mj;
     psii = psii-mi; psij = psij-mj;       * center;
     v[i,j] = (psii#psij#inow)[+]     / (m#(m-1))
           + (psii#psij#(1-inow))[+] / (n#(n-1));
     v[j,i] = v[i,j];
  end;
end;
free psii psij inow gold who;
finish;

/* start of execution of the IML program */

use comp var {&tests &gold};
read all into data [colname=names];

run mwvar(t,v,nx,ny,data);                * estimates and variances;

vname = names[1:(ncol(names)-1)];
manwhit = vecdiag(t);
* print 'Area Under ROC Curve', manwhit[ rowname=vname];
* print 'Estimated Variance Matrix', v [colname=vname rowname=vname];
create &dataout from manwhit;
append from manwhit;
close &dataout;


quit;


%mend;
```

```
/* This is a macro to carry out the n-folder cross
   validation for the PLS Model based on AUC

   It takes only 1 vector of independent  variables.

   file name: plsnfolder_mac.sas
   last updated: Aug 11, 2005

   Created by: Fan Xu
   Modified by: Chenghong Shen
   sample usuage:
                libname hybrid2 'c:\spectrx\fan\nci\hybrid\data2';
                data model; set hybrid2.model7_15_02; run;

                %include 'c:\spectrx\fan\nci\plsnfolder1  mac.sas';
                %nfolder(datain = model, var1 = ptra, folder = 10, n1 = 106, print = yes);

*/

%include 'c:\spectrx\fan\macros\rocest_mac.sas';
options nonotes;
%macro nfolder(datain = model, folder = n, response = high, var1 =, n1 = , select = stepwise,
print = no);
  option nonotes;

  %foldermark(datain = &datain, folder = &folder);

  %do i = 1 %to &folder;

        %put &i out of &folder running...;
        data oneout; set mark;
                if group = &i then whole = .;
        run;

        proc pls data = oneout /*nfac = 9*/ /*cv = split(10)*/ cv=one noprint;
                /*model whole = &var1.1-&var1.60 &var1.87-&var1.106;*/
                model whole = &var1;
                output out = regout predicted = pred;
        run;

        data pred1; set regout; if group = &i ; keep pred; run;
        data pred2; set mark; if group = &i ; keep whole high highlow; run;
        data pred; merge pred1 pred2; run;

        %if &i = 1 %then %do;
                data valid; set _null_; run;
        %end;

        data valid; set valid pred; run;
  %end;

        proc pls data = mark /*nfac = 9*/ /*cv = split(10)*/ cv=one noprint;
                /*model whole = &var1.1-&var1.60 &var1.87-&var1.106;*/
                model whole = &var1;
                output out = regout predicted = pred;
        run;
```

```
  %if %upcase(&print) = YES %then %do;
        title 'Training Performance';
        %rocest(datain = regout, tests = pred, gold = high);
        proc print data = roc; run;
        %rocest(datain = regout, tests = pred, gold = highlow);
        proc print data = roc; run;
        title 'Cross-Validation Performance';
        %rocest(datain = valid, tests = pred, gold = high);
        proc print data = roc; run;
        %rocest(datain = valid, tests = pred, gold = highlow);
        proc print data = roc; run;
  %end;
%mend;

%macro foldermark(datain = , folder = );

        proc sort data = &datain; by whole1; run;

        data mark; set &datain;
                by whole1;
                if first.whole1 then obs = 0;
                else obs + 1;
                if last.whole1 then do;

                        if whole1=3.2 then do;
                                call symput('groupobs3_2', round(obs / &folder));
                        end;

                        if whole1=3.5 then do;
                                call symput('groupobs3_5', round(obs / &folder));
                        end;

                        if whole1=3 then do;
                                call symput('groupobs3', round(obs / &folder));
                        end;

                        if whole1=2.5 then do;
                                call symput('groupobs2_5', round(obs / &folder));
                        end;

                        if whole1=2 then do;
                                call symput('groupobs2', round(obs / &folder));
                        end;

                        if whole1=1 then do;
                                call symput('groupobs1', round(obs / &folder));
                        end;

                        if whole1=0 then do;
                                call symput('groupobs0', round(obs / &folder));
                        end;

                end;
        run;
```

```
        data mark; set mark;
                if whole1 = 0 then group = int(obs / &groupobs0) + 1;
                if whole1 = 1 then group = int(obs / &groupobs1) + 1;
                *if whole1 = 2 then group = int(obs / &groupobs2) + 1;
                if whole1 = 3 then group = int(obs / &groupobs3) + 1;
                *if whole1 = 2.5 then group = int(obs / &groupobs2_5) + 1;
                if whole1 = 3.2 then group = int(obs / &groupobs3_2) + 1;
                if whole1 = 3.5 then group = int(obs / &groupobs3_5) + 1;
                if group > &folder then group = &folder;
        run;

%mend;



/* This is a macro to carry out the n-folder cross
   validation for the PLS Model based on sensitivity and specificity

   It takes only 1 vector of independent variables.

   file name: plsnfolder_sens_spec.sas
   last updated: Jan 12, 2006
   Created by: Chenghong Shen
   sample usuage:

                libname hybrid2 'c:\spectrx\fan\nci\hybrid\data2';
                data model; set hybrid2.model7_15_02; run;

                %include 'c:\spectrx\fan\nci\plsnfolder1_sens_spec.sas';
                %nfolder(datain = model, var1 = ptra, folder = 10, n1 = 106, print = yes);

*/

%include 'c:\spectrx\fan\macros\rocest_mac.sas';
%include 'c:\spectrx\workdir\programs\sensitivity_specificity.sas';
options nonotes;
%macro nfolder_sens_spec(datain = model, folder = n, response = high, var1 =, n1 = , select =
stepwise, print = no);
  option nonotes;

  %foldermark(datain = &datain, folder = &folder);

  %do i = 1 %to &folder;

        %put &i out of &folder running...;
        data oneout; set mark;
                if group = &i then whole = .;
        run;

        proc pls data = oneout /*nfac = 9*/ /*cv = split(10)*/ cv=one noprint;
                /*model whole = &var1.1-&var1.60 &var1.87-&var1.106;*/
                model whole = &var1;
                output out = regout predicted = pred;
        run;
```

```
        data pred1; set regout; if group = &i ; keep pred; run;
        data pred2; set mark; if group = &i ; keep whole high highlow; run;
        data pred; merge pred1 pred2; run;

        %if &i = 1 %then %do;
                data valid; set _null_; run;
        %end;

        data valid; set valid pred; run;
%end;


        proc pls data = mark /*nfac = 9*/ /*cv = split(10)*/ cv=one noprint;
                /*model whole = &var1.1-&var1.60 &var1.87-&var1.106;*/
                model whole = &var1;
                output out = regout predicted = pred;
        run;


%if %upcase(&print) = YES %then %do;
        title 'Training Performance';
        /*%rocest(datain = regout, tests = pred, gold = high);
        proc print data = roc; run; */
  %sens_spec(datain = regout);
  proc print data=sensspec;
                var sen spec;
        run;
        title 'Cross-Validation Performance';
        %sens_spec(datain = valid);
        proc print data=sensspec;
                var sen spec;
        run;
        /*%rocest(datain = valid, tests = pred, gold = high);
        proc print data = roc; run; */

%end;
%mend;

%macro foldermark(datain = , folder = );

        proc sort data = &datain; by whole1; run;

        data mark; set &datain;
                by whole1;
                if first.whole1 then obs = 0;
                else obs + 1;
                if last.whole1 then do;

                        if whole1=3.2 then do;
                                call symput('groupobs3_2', round(obs / &folder));
                        end;

                        if whole1=3.5 then do;
                                call symput('groupobs3_5', round(obs / &folder));
                        end;
```

```
                        if whole1=3 then do;
                                call symput('groupobs3', round(obs / &folder));
                        end;

                        if whole1=2.5 then do;
                                call symput('groupobs2_5', round(obs / &folder));
                        end;

                        if whole1=2 then do;
                                call symput('groupobs2', round(obs / &folder));
                        end;

                        if whole1=1 then do;
                                call symput('groupobs1', round(obs / &folder));
                        end;

                        if whole1=0 then do;
                                call symput('groupobs0', round(obs / &folder));
                        end;

                end;
        run;


        data mark; set mark;
                if whole1 = 0 then group = int(obs / &groupobs0) + 1;
                if whole1 = 1 then group = int(obs / &groupobs1) + 1;
                *if whole1 = 2 then group = int(obs / &groupobs2) + 1;
                if whole1 = 3 then group = int(obs / &groupobs3) + 1;
                *if whole1 = 2.5 then group = int(obs / &groupobs2_5) + 1;
                if whole1 = 3.2 then group = int(obs / &groupobs3_2) + 1;
                if whole1 = 3.5 then group = int(obs / &groupobs3_5) + 1;
                if group > &folder then group = &folder;
        run;

%mend;


/* This is a macro to carry out the n-folder cross
   validation for the logistic regression model.

   It is modified from nfolder_mac.sas. It takes 3
   sets of variables.

   file name: nfolder_mac.sas
   last updated: 12/9/2005

   Creasted by: Fan Xu
   Modified by: Chenghong Shen
   sample usuage:

*/

%include 'c:\spectrx\fan\macros\rocest_mac.sas';
```

```
%macro nfolder(datain = model, folder = n, response = whole, var1 =, var2 = , var3 = , n = ,
select = stepwise,
        print = no, sig = 0.01, pap = no);
  option nonotes;

  %foldermark(datain = &datain, folder = &folder);

  /*proc princomp data = mark noprint out = prin prefix;
                var &var1;
  run;*/

  %do i = 1 %to &folder;

        %put &i out of &folder running...;
        data oneout; set mark;
                if group = &i then &response = .;
        run;

        proc logistic data = oneout descending noprint;
                %if %upcase(&pap) = YES %then %do;
                model &response = &var1 /*pm1-pm&n ps1-ps&n pt1-pt&n*/ preferredPap
                %end;
                %if %upcase(&pap) = NO %then %do;
                model &response = &var1 /*pm1-pm&n ps1-ps&n pt1-pt&n*/
                %end;
                        %if %upcase(&select) = STEPWISE %then %do;
                                / fast selection = stepwise sle = &sig sls = &sig;
                        %end;
                        %else %do;

                                ;
                        %end;
        output out = lout pred = pred;
    run;

        data pred1; set lout; if group = &i ; keep pred; run;
        data pred2; set mark; if group = &i ; keep &response; run;
        data pred; merge pred1 pred2; run;

    %if &i = 1 %then %do;
        data valid; set _null_; run;
    %end;

        data valid; set valid pred; run;
%end;

  proc logistic data = mark noprint descending ;
                %if %upcase(&pap) = YES %then %do;
                model &response = &var1 /* pm1-pm&n ps1-ps&n pt1-pt&n */  preferredPap
                %end;
                %if %upcase(&pap) = NO %then %do;
                model &response = &var1 /*pm1-pm&n ps1-ps&n pt1-pt&n */
                %end;
                        %if %upcase(&select) = STEPWISE %then %do;
                                /  selection = stepwise sle = &sig sls = &sig;
```

```
                        %end;
                        %else %do;

                        ;
                        %end;
        output out = lout pred = pred;
run;

%if %upcase(&print) = YES %then %do;
        %rocest(datain = lout, tests = pred, gold = &response);
        title 'Training Performance';
        proc print data = roc; run;
        %rocest(datain = valid, tests = pred, gold = &response);
        title 'Cross-Validation Performance';
        proc print data = roc; run;
%end;
%mend;

%macro foldermark(datain = , folder = );

        proc sort data = &datain; by whole1; run;

        data mark; set &datain;
                by whole1;
                if first.whole1 then obs = 0;
                else obs + 1;
                if last.whole1 then do;

                        if whole1=3.2 then do;
                                call symput('groupobs3_2', round(obs / &folder));
                        end;

                        if whole1=3.5 then do;
                                call symput('groupobs3_5', round(obs / &folder));
                        end;

                        if whole1=3 then do;
                                call symput('groupobs3', round(obs / &folder));
                        end;

                        if whole1=2.5 then do;
                                call symput('groupobs2_5', round(obs / &folder));
                        end;

                        if whole1=2 then do;
                                call symput('groupobs2', round(obs / &folder));
                        end;

                        if whole1=1 then do;
                                call symput('groupobs1', round(obs / &folder));
                        end;

                        if whole1=0 then do;
                                call symput('groupobs0', round(obs / &folder));
                        end;
```

```
                end;
        run;

        data mark; set mark;
                if whole1 = 0 then group = int(obs / &groupobs0) + 1;
                if whole1 = 1 then group = int(obs / &groupobs1) + 1;
                if whole1 = 3 then group = int(obs / &groupobs3) + 1;
                if whole1 = 3.2 then group = int(obs / &groupobs3_2) + 1;
                if whole1 = 3.5 then group = int(obs / &groupobs3_5) + 1;
                if group > &folder then group = &folder;
        run;

%mend;




/* This is a macro to calculate the sensitivity and specifity

   last updated: Jan 12, 2006
   Created by: Chenghong Shen

*/

libname cv 'c:\spectrx\cv';

%macro sens_spec(datain= );   /* datain includes the model predicted value and response
variable, etc */

data all;
        set &datain;
run;

data cin1out (KEEP= whole);
set all;
run;

data predicted (KEEP=pred);
set all;
run;


PROC IML;

START Sens_Spec;
/*USE INPUTS;
READ ALL VAR _ALL_ into X;*/

USE cin1out;
READ ALL VAR _ALL_ into whole;

USE predicted;
READ ALL VAR _ALL_ into Y;
```

```
Z=NROW(WHOLE);
N=NROW(Y);

/*  Calculate all the response variables */
/*  X is a 80 column matrix, coeff` is a 80 row matrix */

o=J(1500,4,0);

DO J= 1 TO 1500 by 1;

        CUTOFF= -0.2+ J * 0.001;
        R = J(N, 1, CUTOFF);

        Diff = Y - R;

        ALL = WHOLE[LOC(WHOLE= 1),];
        COUNT_ALL = NROW(ALL);

        DIF = DIFF[LOC(WHOLE=1),];
        indices = LOC(DIF>0);

        if nrow(indices) > 0 then
                do;
                TEST  P = DIF[LOC(DIF>0),];
                COUNT_P = NROW(TEST_P);
                        SENS = COUNT_P/COUNT_ALL;
                end;
        else SENS = 0;


        /* FIND SPECIFICITIES */

        ALL = WHOLE[LOC(WHOLE= 0),];
        COUNT_ALL = NROW(ALL);

        DIF = DIFF[LOC(WHOLE=0),];

        indices = LOC(DIF<=0);

        if nrow(indices) > 0 then
                do;
                TEST  N = DIF[LOC(DIF<=0),];
                COUNT_N = NROW(TEST_N);
                        SPEC = COUNT_N/COUNT_ALL;
                end;
        else SPEC = 0;


        /* PUT CUTOFF SENS SPEC INTO MATRIX O FOR OUTPUT */
        O[J,1]=CUTOFF;
        O[J,2]=SENS;
        O[J,3]=1-SPEC;
        O[J,4]=SPEC;

END;
```

```
CREATE SSPEC FROM O;
APPEND FROM O;

FINISH;

RUN Sens_Spec;
quit iml;
run;

data SENSPEC  M1 (rename=(col1=cutoff col2=sen col3=one_minus_spec col4=spec));
        set SSPEC;
        if col2>=0.98 and col2<=1 and col4>=0;
run;

data sensspec;
        set SENSPEC_M1 end=last;
        if last;
run;
%mend;
```

```
/* This is a program to define the data set for the final training. It is the data set of cases that has
the pap test but without CIN1 cases. It creates coefficients for the original 25th percentile model
and Mixed Model 1.9, it also evaluates different model results

   file name: final train  cross  validation_diff_CategoryPap_mars.sas
   last updated:  Dec 6th, 2005
   by: Chenghong Shen

*/

libname cv 'c:\spectrx\cv';
libname After 'c:\spectrx\Fan\Aftertrain\diff';
libname Dallas 'c:\spectrx\Fan\Dallas\diff';
libname Spectrx 'c:\spectrx\data';
%include 'c:\spectrx\fan\Training\readtv  mac.sas';
%include 'c:\spectrx\workdir\programs\plstrainvalid_mac_new.sas';
%include 'c:\spectrx\workdir\programs\plsnfolder1_mac.sas';

data train;

        set cv.Train510cv;
        if preferredpap eq 3 or preferredpap eq 4
        then
                BF1 = 1;
        else
                BF1 = 0;

        if (preferredpap eq  0 or preferredpap eq  1 or preferredpap eq  3) then
                BF3 = 1;
        else
```

```
                BF3 = 0;

        if P25RA32 - 0.418>0 then
                BF5 = P25RA32 - 0.418;
        else
                BF5 = 0;



    BF7 = max(0, 0.362 - P25RA106 );
    BF9 = max(0, 0.035 - M40 );
    BF10 = max(0, P25RA57 - 0.134);
    BF13 = max(0, 0.275 - P25RA83 );
    BF15 = max(0, 2.229 - P25RA38 );

    if age >45 then
        age=0;
    else
        age=1;

run;

data train62;
        set cv.Train62cv;
        if preferredpap eq 3 or preferredpap eq 4 then
                BF1 = 1;
        else
                BF1 = 0;

   if (preferredpap eq 0 or preferredpap eq 1 or preferredpap eq 3) then
                BF3 = 1;
        else
                BF3 = 0;

        if P25RA32 - 0.418>0 then
                BF5 = P25RA32 - 0.418;
        else
                BF5 = 0;

        if age >45 then
                age=0;
        else
                age=1;

run;

data d2;
        set cv.D2cv;
        *preferredPap = PriorPap;
        if preferredpap eq 3 or preferredpap eq 4 then
                BF1 = 1;
        else
                BF1 = 0;

        if (preferredpap eq 0 or preferredpap eq 1 or preferredpap eq 3) then
                BF3 = 1;
```

```
        else
                BF3 = 0;

        if P25RA32 - 0.418>0 then
                BF5 = P25RA32 - 0.418;
        else
                BF5 = 0;

    BF7 = max(0, 0.362 - P25RA106 );
    BF9 = max(0, 0.035 - M40 );
    BF10 = max(0, P25RA57 - 0.134);
    BF13 = max(0, 0.275 - P25RA83 );
    BF15 = max(0, 2.229 - P25RA38 );

run;


/* 25th percentile model */
%nfolder(datain = train, var1 = p25ra1-p25ra60 p25ra87-p25ra106 preferredpap, folder = 10, n1
= 106, print = yes);
%PLStrainvalid(train =  train, valid = train62, var = p25ra1-p25ra60 p25ra87-p25ra106
preferredPap, roc = yes);

/* Mixed model 1.3 */
%nfolder(datain = train, var1 = p25ra1-p25ra19 p25ra24-p25ra37 p25ra41-p25ra60 p25ra61-
p25ra68 p25ra79-p25ra86 p25ra87-p25ra106 m1-m14 m28-m30 preferredpap, folder = 10, n1 =
106, print = yes);
%PLStrainvalid(train = train, valid = d2, var = p25ra1-p25ra19 p25ra24-p25ra37 p25ra41-
p25ra60 p25ra61-p25ra68 p25ra79-p25ra86 p25ra87-p25ra106 m1-m14 m28-m30
/*preferredpap*/, roc = yes);

/* Mixed model 1.4 */
%PLStrainvalid(train = cv.Train510cv, valid = cv.d2cv, var = p25ra1-p25ra19 p25ra24-p25ra37
p25ra41-p25ra60 p25ra61-p25ra68 p25ra79-p25ra86 p25ra87-p25ra94 p25ra97-p25ra106 m1-
m14 m28-m30 Preferredpap, roc = yes);

/* Mixed model 1.5 */
%nfolder(datain = cv.Train510cv, var1 = p25ra1-p25ra19 p25ra24-p25ra37 p25ra41-p25ra60
p25ra87-p25ra94 p25ra97-p25ra106 m28-m31 m1-m15, folder = 10, n1 = 106, print = yes);
%PLStrainvalid(train = cv.Train510cv, valid = cv.D2cv, var = p25ra1-p25ra19 p25ra24-p25ra37
p25ra41-p25ra60 p25ra87-p25ra94 p25ra97-p25ra106 m28-m31 m1-m15 preferredpap, roc =
yes);


/* Mixed model 1.6 */
%nfolder(datain = cv.Train510cv, var1 = p25ra1-p25ra37 p25ra41-p25ra60 p25ra87-p25ra106
m28-m30 m1-m10 Preferredpap, folder = 10, n1 = 106, print = yes);
%PLStrainvalid(train = cv.Train510cv, valid = train62, var = p25ra1-p25ra37 p25ra41-p25ra60
p25ra87-p25ra106 m28-m30 m1-m10 Preferredpap, roc = yes);

/* Mixed Model 1.7 */
%nfolder(datain = train, var1 = p25ra1-p25ra21 p25ra24-p25ra37 p25ra41-p25ra60 p25ra61-
p25ra68 p25ra79-p25ra86 p25ra87-p25ra106 m1-m14 m20-m30 /*preferredpap*/ m33-m34,
folder = 10, n1 = 106, print = yes);
%PLStrainvalid(train = train, valid = d2, var = p25ra1-p25ra21 p25ra24-p25ra37 p25ra41-
p25ra60 p25ra61-p25ra68 p25ra79-p25ra86 p25ra87-p25ra106 m1-m14 m20-m30 m33-m34
```

```
/*preferredpap*/, roc = yes);

/* Mixed Model 1.8 */
%nfolder(datain = train, var1 = p25ra10-p25ra21 p25ra24-p25ra37 p25ra41-p25ra60 p25ra61-
p25ra68 p25ra79-p25ra86 p25ra87-p25ra106 m1-m14 m20-m30 m33-m34 preferredpap /*m16-
m18*/ /*m36-m39*/ /*m54-m55*/, folder = 10, n1 = 106, print = yes);
%PLStrainvalid(train = train, valid = d2, var = p25ra10-p25ra21 p25ra24-p25ra37 p25ra41-
p25ra60 p25ra61-p25ra68 p25ra79-p25ra86 p25ra87-p25ra106 m1-m14 m20-m30 m33-m34
preferredpap  /*m16-m18*/ /*m36-m39*/ /*m54-m55*/, roc = yes);

/* Mixed model 1.9 */
%nfolder(datain = train, var1 = p25ra10-p25ra21 p25ra24-p25ra37 p25ra41-p25ra60 p25ra87-
p25ra106 m1-m14 m20-m30 m33-m34 BF1 BF3 BF5 /*preferredpap*/, folder = 10, n1 = 106, print
= yes);
%PLStrainvalid(train = Train, valid =cv.D2cv, var = p25ra10-p25ra21 p25ra24-p25ra37 p25ra41-
p25ra60 p25ra87-p25ra106 m1-m14 m20-m30 m33-m34 preferredpap BF1 BF3 BF5, roc = yes);
```

## APPENDIX V: C AND SAS CODE FOR RANDOMLY SELECT FOUR GROUPS OUT OF EIGHT GROUPS AND COMPARE MODEL PERFORMANCE

```
/*================================================================*/
/* From the Combinatorial Object Server:                          */
/* Generates the k-combinations of [n] by transpositions          */
/* No input error checking.  Assumes 0 <= k <= n <= MAX.          */
/* Outputs both the bitstring and the transposition (x,y) (meaning */
/* that x leaves the subset and y enters).                        */
/* Algorithm is CAT (Constant Amortized Time).                    */
/* Original version of this program by Frank Ruskey (1995) can be  */
/* found at http://sue.uvic.ca/~cos/inf/comb/CombinationsInfo.html */
/*================================================================*/

#include <stdio.h>

void NEG( int, int);
void GEN( int, int);
void PrintIt();

    int n, k;
    int a;

void main(void)
{
    printf( "Enter n,k: ");
    scanf( "%d,%d", &n, &k );

    a = (1 << k) - 1;
    a <<= (n-k);

    PrintIt();
    GEN( n, k);
}


void PrintIt()
{
    int i, d;
    for (i = 1, d=1; i <= n; i++, d <<= 1)
        printf( "%d ", (a&d) ? 1 : 0);
    printf( "\n");
}


void swap ( int x, int y)
{
    a |= (1<<(n-x)); a &= ~(1<<(n-y));
    PrintIt();
}

void GEN( int n, int k)
{
    if (k > 0 && k < n)
        {
```

```
        GEN( n-1, k);

        if (k == 1)
           swap( n, n-1 );
        else
           swap( n, k-1 );

        NEG( n-1, k-1);
        }
}

void NEG( int n, int k)
{
   if (k > 0 && k < n)
        {
        GEN( n-1, k-1 );

        if (k == 1)
           swap( n-1, n);
        else
           swap( k-1, n);

        NEG( n-1, k);
        }
}



/* This is a program to randomly select four groups out of eight
   groups and compare model performance

   last updated: 7/18/2005
   Created by: Chenghong Shen

*/
libname After 'c:\spectrx\Fan\Aftertrain\diff';
libname Dallas 'c:\spectrx\Fan\Dallas\diff';
%include 'c:\spectrx\workdir\programs\readtv_mac.sas';
%include 'c:\spectrx\workdir\programs\plstrain_mixed_model_detect.sas';

data cin1out; set After.a; if whole1 not in (2 2.5); whole = (whole1 > 2); run;
data dcin1out; set Dallas.a; if whole1 not in (2 2.5); whole = (whole1 > 2); run;

%readtv(path = c:\spectrx\workdir\,
        file = HybridFINAL  TrainValRandomSplit.txt, datain = cin1out,
        train = trandom46, valid = vrandom46);


data EightSelectFour;
        /* file generated from the above c code */
        infile "c:\temp\8.txt" dlm=' ' MISSOVER LRECL=100;
   input x1-x8;
        call symput('pos1'||left(  n  ), x1);
        call symput('pos2'||left(_n_), x2);
        call symput('pos3'||left(  n  ), x3);
        call symput('pos4'||left(_n_), x4);
```

```
        call symput('pos5'||left(_n_), x5);
        call symput('pos6'||left(_n_), x6);
        call symput('pos7'||left(_n_), x7);
        call symput('pos8'||left(_n_), x8);

run;

%readin;

%macro readin();
  %let nsub=2;

  %do i=1 %to 8;
        %let var&i= ;
    %end;

    data output; set _null_ ; run;

    %do i=1 %to 70;
            %put &i;
            %if &&pos1&i eq 1 %then %do;
        %let var1=p25ra1-p25ra20;
            %end;

            %if &&pos2&i eq 1 %then %do;
                %let var2=p25ra21-p25ra40;
            %end;

            %if &&pos3&i eq 1 %then %do;
                %let var3=p25ra41-p25ra60;
            %end;

            %if &&pos4&i eq 1 %then %do;
                %let var4=p25ra87-p25ra106;
            %end;

            %if &&pos5&i eq 1 %then %do;
                %let var5=m1-m20;
            %end;

            %if &&pos6&i eq 1 %then %do;
                %let var6=m21-m40;
            %end;

            %if &&pos7&i eq 1 %then %do;
                %let var7=m41-m60;
            %end;

            %if &&pos8&i eq 1 %then %do;
                %let var8=m61-m80;
            %end;

            %let var=&var1 &var2 &var3 &var4 &var5 &var6 &var7 &var8;

        %put &var;
            %PLStrainvalid(train = trandom46, valid = vrandom46, var = &var);
```

```
                        %do j=1 %to 8;
                                %let var&j= ;
                        %end;
        %end;
        proc sort data=output out=sortoutput;
                by DESCENDING ht hv;
        run;
%mend readin;
```

## APPENDIX VI MATLAB CODE FOR BUILDING SVM MODEL

```
% This is a program to use OSU SVM toolbox
% to build svm model
%   file: osu_svm_p25_diff.m
%
%   created by: Chenghong Shen
%   last update: 12/14/2005


sens_spec=[];
sens_specV=[];
sens_specD=[];
data = load('C:\Spectrx\workdir\programs\SVM\data\train60.txt');

m1=data(:,2:15);
m2=data(:,21:31);
m3=data(:,34:35);
m=[m1 m2 m3];
m=data(:,2:81);

s1=data(:,91:102);
s2=data(:,105:118);
s3=data(:,122:141);
s4=data(:,142:161);
Sample=[s1 s2 s3 s4];
%Sample=data(:,82:161);   % 1-81 are the covariates including Pap
pap=data(:,1 );
Samples=[Sample m];
%Samples=[Sample pap m];
%Samples=[pap];
Samples = Samples';

Label =data(:,162);     % 162 is the response variable
Labels = Label';


data = load('C:\Spectrx\workdir\programs\SVM\data\valid40.txt');
m1=data(:,2:15);
m2=data(:,21:31);
m3=data(:,34:35);
m=[m1 m2 m3];
m=data(:,2:81);

s1=data(:,91:102);
s2=data(:,105:118);
s3=data(:,122:141);
s4=data(:,142:161);
ValidSample=[s1 s2 s3 s4];
%ValidSample=data(:,82:161);   % 1-81 are the covariates including Pap

Validpap=data(:,1);
%validSamples=[ValidSample Validpap m];
%validSamples=[Validpap];
```

```
validSamples=[ValidSample m];
ValidSamples = validSamples';

ValidLabels =data(:,162);     % 162 is the response variable
ValidLabels = ValidLabels';

data = load('C:\Spectrx\workdir\programs\SVM\data\d2.txt');
m1=data(:,1:14);
m2=data(:,20:30);
m3=data(:,33:34);
m=[m1 m2 m3];
m=data(:,1:80);

s1=data(:,90:101);
s2=data(:,104:117);
s3=data(:,121:140);
s4=data(:,141:160);
DallasSample=[s1 s2 s3 s4];
%DallasSample=data(:,81:160);   % 1-81 are the covariates including Pap
Dallaspap=data(:,162);
%DallasSamples=[DallasSample Dallaspap m];
%DallasSamples=[Dallaspap];
DallasSamples=[DallasSample m];
DallasSamples = DallasSamples';

DallasLabels =data(:,161);     % 161 is the response variable
DallasLabels = DallasLabels';


% Use a linear support vector machine classifier

for j=0.01:0.01: 2;

c=j;

Parameters = [0 1 1 1 c 40 2 0 0.5 0.1 1]; %linear
%Parameters = [0 1 1 1 c 40 0.001 0 0.5 0.1 1]; %linear
%Parameters = [1 2 1 1 c 40 0.001 0 0.5 0.1 1]; %Polynomial
%Parameters = [2 1 1 1 c 40 0.001 0 0.5 0.1 1]; %RBF
% [AlphaY, SVs, Bias, Parameters, nSV, nLabel] = u_LinearSVC(Samples, Labels,1);
[AlphaY, SVs, Bias, Para, nSV, nLabel] = SVMTrain(Samples, Labels, Parameters);

% Training
[ClassRate, DecisionValue, Ns, ConfMatrix, PreLabels]= SVMTest(Samples, Labels, AlphaY, SVs,
Bias,Parameters, nSV, nLabel);

PreLabel = PreLabels';
jointpap = [Label PreLabel pap];

% Validation
[ClassRateV, DecisionValueV, NsV, ConfMatrixV, PreLabelsV]= SVMTest(ValidSamples, ValidLabels,
AlphaY, SVs, Bias,Parameters, nSV, nLabel);
```

```
% Validation on Dallas data
[ClassRateD, DecisionValueD, NsD, ConfMatrixD, PreLabelsD]= SVMTest(DallasSamples, DallasLabels,
AlphaY, SVs, Bias,Parameters, nSV, nLabel);

s=size(PreLabels, 2);   % number of training subjects
sv=size(PreLabelsV, 2); % number of validation subjects
sd=size(PreLabelsD, 2); % number of Dallas subjects

total_disease =0;
true_disease =0;
total_no_disease=0;
true_no_disease=0;

total_diseaseV =0;
true_diseaseV =0;
total_no_diseaseV =0;
true_no_diseaseV =0;

total_diseaseD =0;
true_diseaseD =0;
total_no_diseaseD =0;
true_no_diseaseD =0;

% Calculate sensitivity and specificity
for i=1:s;
   if Labels(1,i)==1
      total_disease = total_disease +1;
      if PreLabels(1,i) == 1
         true_disease = true_disease +1;
      end;
   else
      total_no_disease=total_no_disease +1;
      if PreLabels(1,i) == 0
         true_no_disease = true_no_disease +1;
      end;

   end;
end;

sens = true_disease/total_disease;
spec = true_no_disease/total_no_disease;

senspec = [sens spec c];
sens_spec = cat(1,sens_spec,senspec);


for j=1:sv;
   if ValidLabels(1,j)==1
      total_diseaseV = total_diseaseV +1;
      if PreLabelsV(1,j) == 1
         true_diseaseV = true_diseaseV +1;
      end;
   else
      total_no_diseaseV=total_no_diseaseV +1;
```

```
      if PreLabelsV(1,j) == 0
         true_no_diseaseV = true_no_diseaseV +1;
      end;

   end;
end;

sensV = true_diseaseV/total_diseaseV;
specV = true_no_diseaseV/total_no_diseaseV;

senspecV = [sensV specV c];
sens_specV = cat(1,sens_specV,senspecV);
for j=1:sd;
   if DallasLabels(1,j)==1
      total_diseaseD = total_diseaseD +1;
      if PreLabelsD(1,j) == 1
         true_diseaseD = true_diseaseD +1;
      end;
   else
      total_no_diseaseD=total_no_diseaseD +1;
      if PreLabelsD(1,j) == 0
         true_no_diseaseD = true_no_diseaseD +1;
      end;

   end;
end;

sensD = true_diseaseD/total_diseaseD;
specD = true_no_diseaseD/total_no_diseaseD;

senspecD = [sensD specD c];
sens_specD = cat(1,sens_specD,senspecD);

end;

oneminusspec= 1-sens_spec(:,2);
oneminusspecV= 1-sens_specV(:,2);
```

## APPENDIX VII SAS CODE FOR ESTIMATING THRESHOLD USING BOOTSTRAP

```
/* This program is to generate bootstrap samples and estimate the CI for
   the threshold for 25th percentile + Pap model
   file: boot_estimate_of_CI_M1_PAP

   Last update: 8/06/2005
   Created by: Chenghong Shen

*/

%include 'c:\spectrx\workdir\programs\jackboot.sas';
libname Spectrx 'c:\spectrx\data';
libname After 'c:\spectrx\Fan\Aftertrain\diff';
libname Dallas 'c:\spectrx\Fan\Dallas\diff';

options nonotes;
data cin1out; set Spectrx.all; if whole1 not in (2 2.5); whole = (whole1 > 2); run;
data spectrx.final_M1_PAP; set _null_; run;

data dcin1out; set Dallas.All_d2_dayofpap; if whole1 not in (2 2.5) and DayofPap ne -1 and d_id
eq 'A2'; whole = (whole1 > 2); PreferredPap=DayofPap; run;      /* Dallas D2 */
data dcind1out; set Dallas.All  d1  dayofpap; if whole1 not in (2 2.5) and DayofPap ne -1 and d_id
eq 'A1'; whole = (whole1 > 2); PreferredPap=DayofPap; run;    /* Dallas D1 with 10 subjects */

/* Dallas D1 w/o 10, 9 subjects + 39 */
data dcind1  2out; set dcind1out;
if sub  id not in ('7PTA1039'  '7PTA1038' '7PTA1037' '7PTA1036' '7PTA1035' '7PTA1034'
'7PTA1033'
'7PTA1032' '7PTA1031' '7PTA1030');
run;

data all;
set cin1out dcin1out dcind1_2out;
run;


/* This is to generate n bootstrap samples
     Only empty Analyze macro is constructed  */
%macro analyze(data=,out=);
          %bystmt;
%mend;
%boot(data=all, samples=1000, random=123);


%macro cutoffpoint(n=);
          %do i=1 %to &n;
             %put sample no. &i;

                    data sample;
                              set Bootdata;
                              if _sample_ eq &i;
                    run;
```

```
data response (KEEP= whole);
        set sample;
run;

DATA vars (KEEP= /*preferredpap*/ p25ra1-p25ra60 p25ra87-p25ra106);
        set sample;
RUN;

data pap (KEEP= preferredpap);
        set sample;
run;

PROC IML;
START Sens_Spec;
USE vars;
READ ALL VAR _ALL_ into X;

USE response;
READ ALL VAR _ALL_ into whole;

use pap;
READ ALL VAR _ALL_ into pap;


Z=NROW(WHOLE);
M = COEFF`;
P = PAPINDEX`;

/*  Calculate all the response variables */
/*  X is a 80 column matrix. coeff` is a 80 row matrix */
Y= 0.394286442 + X * M + pap*P;
N = NROW(Y);
*PRINT N;

dim = 400;
o=J(dim,4,0);

DO J= 1 TO dim by 1;

        CUTOFF= J * 0.001;
        R = J(N, 1, CUTOFF);

        Diff = Y - R;

        ALL = WHOLE[LOC(WHOLE= 1),];
        COUNT_ALL = NROW(ALL);

        DIF = DIFF[LOC(WHOLE=1),];

        indices = LOC(DIF>0);

        if nrow(indices) > 0 then
        do;
        /* Larger than cutoff point, disease */
        TEST  P = DIF[LOC(DIF>0),];
        COUNT_P = NROW(TEST_P);
```

```
                SENS = COUNT_P/COUNT_ALL;
        end;
        else SENS = 0;


        /* FIND SPECIFICITIES */

        ALL = WHOLE[LOC(WHOLE= 0),];
        COUNT_ALL = NROW(ALL);


        DIF = DIFF[LOC(WHOLE=0),];
        indices = LOC(DIF<=0);

        if nrow(indices) > 0 then
        do;
        TEST  N = DIF[LOC(DIF<=0),];
        COUNT  N = NROW(TEST  N);
                SPEC = COUNT_N/COUNT_ALL;
        end;
        else SPEC = 0;


        /* PUT CUTOFF SENS SPEC INTO MATRIX O FOR OUTPUT */
        O[J,1]=CUTOFF;
        O[J,2]=SENS;
        O[J.3]=1-SPEC;
        O[J,4]=SPEC;

END;

CREATE SSPEC FROM O;
APPEND FROM O;


FINISH;

PAPINDEX = {
        0.123336215
};

COEFF={
        /*0.123336215*/
        -0.263321304
        -0.421427229
        -0.030666347
        0.350462983
        0.533249093
        -0.107802855
        0.006297154
        -0.272240931
        0.285892776
        -0.460624146
        0.564781485
        0.681043586
```

-0.052878897
0.34686853
0.782423438
0.151901304
-0.98682887
-3.463493797
-0.314715562
3.420185656
2.667903765
-2.117923684
-0.749258404
0.48957531
-0.70483027
-0.717909078
-0.097937986
-0.19452495
-0.041618972
0.473301996
0.713851244
-0.60192158
-0.008711535
0.488308635
0.046022813
-0.338552561
0.743530392
0.328000122
-0.596114962
-1.547837405
-2.086067819
-0.227861586
0.718820729
2.63918207
3.842851069
2.286832769
1.349088544
-3.532176579
-3.12362593
-0.938392025
-0.915668932
1.974233522
0.900094561
0.031135391
-2.290136299
-0.385669265
-5.685983435
-2.321450806
4.081305301
3.753938972
0.509121533
-0.035598535
-1.019971325
0.04482162
0.824830757
1.28954536
-0.047046472
-3.279756773

```
                    -4.473488307
                    5.73688439
                    -0.500921936
                    3.218421303
                    -1.046098159
                    -0.548420617
                    -1.855921384
                    -0.741903744
                    1.13208846
                    1.368937269
                    -0.514304886
                    1.186800966

                };

                RUN Sens_Spec;
                quit iml;
                run;

                data SENSPEC (rename=(col1=cutoff col2=sen col3=one_minus_spec
                 col4=spec));
                        set SSPEC;
                        *k=round(100*col2);
                        SampleID = &i;
                if col2>0.986 /*and col2<0.994*/;
                run;

                data SENS  SPEC:
                        set SENSPEC end=last;
                        if last;
                run;

                data spectrx.final  M1  PAP;
                        set spectrx.final_M1_PAP SENS_SPEC;
                run;
        %end;
%mend;

%cutoffpoint(n=1000);

proc means data=spectrx.final_M1_PAP;
        var cutoff sen one _minus_spec spec sampleID;
         output out = result
         p5 = p5cutoff p5_sen p5_one_minus p5_spec
         mean = avecutoff ave  sen ave  one  minus ave  spec
         max = maxcutoff max  sen max  one  minus max  spec
         min = mincutoff min_sen min_one_minus min_spec;
         *by sampleID;
run;
```

```
/* This is a program to calculate sensitivity and specificity of model
Mixed1.9 +Pap in different Pap groups at a specified threshold
  file: classify_of_disease_m1.9
  Last update: 11/18/2005

  Created by: Chenghong Shen

*/

libname cv 'c:\spectrx\cv';

data input;
      set cv.Train510cv cv.D2cv;
run;

data normal benign ascus_n_neoplasia ascus_neoplasia lsil agus asc hsil
cancer;
      set input;
      if preferredpap in (0) then  output  normal;
      if preferredpap in (1)  then output benign;
      if preferredpap = 2 then output ascus_n_neoplasia;
      if preferredpap = 2.8 then output ascus_neoplasia;
      if preferredpap = 3 then output lsil;
      if preferredpap =3.2 then output agus;
      if preferredpap =3.5 then output asc;
      if preferredpap =4 then output hsil;
      if preferredpap =4.2 then output cancer;
run;


%macro count_patients(datain=  );

data cin1out (KEEP= whole);
      set &datain;
      *if whole1 not in (2 2.5);
run;

DATA INPUTS (KEEP=  p25ra10-p25ra21 p25ra24-p25ra37 p25ra41-p25ra60
p25ra87-p25ra106);
      set &datain;
      *if whole1 not in (2 2.5);
RUN;

data pap (KEEP= preferredpap);
      set &datain;
      *if whole1 not in (2 2.5);
run;

data mdata (keep = m1-m14 m20-m30 m33-m34);
      set &datain;
run;

PROC IML;
```

```
START Sens_Spec;
USE INPUTS;
READ ALL VAR _ALL_ into X;

USE cin1out;
READ ALL VAR _ALL_ into whole;

use pap;
READ ALL VAR _ALL_ into pap;

USE mdata;
read all var _all_ into diff_data;


M = COEFF`;
P = PAPINDEX`;
D = M_DATA`;

/*  Calculate all the response variables */
/*  X is a 80 column matrix, coeff` is a 80 row matrix */
/* The model is mixed Model1.6 +PAP */

Y= -0.515142174  + X * M + diff_data * D  + pap*P;
N = NROW(Y);
*PRINT Y;

o=J(1,8,0);


CUTOFF= 0.23;
R = J(N, 1, CUTOFF);

Diff = Y - R;

a=LOC(WHOLE=1);
if nrow(a)>0 then
    do;

    ALL = WHOLE[LOC(WHOLE=1),];
    COUNT_ALL = NROW(ALL);

    DIF = DIFF[LOC(WHOLE=1),];
    indices = LOC(DIF>0);

    if nrow(indices) > 0 then
            do;
                N_POSITIVE = NROW(DIF);
            TEST_P = DIF[LOC(DIF>0),];

                  /* COUNT_P is true positive */
                  COUNT_P = NROW(TEST_P);
                  /* COUNT_FN is false negative  */
                  COUNT_FN=COUNT_ALL - COUNT_P;
                  SENS = COUNT_P/COUNT_ALL;
            end;
    else
            do;
```

```
                SENS = 0;

                COUNT_P=0;
                COUNT_FN=COUNT_ALL;
        end;

end;
else
        do;
                SENS = 0;
                COUNT_P=0;
                COUNT_FN=0;

        end;

/* FIND SPECIFICITIES */

a=LOC(WHOLE=0);
  if nrow(a)>0 then
  do;

  ALL = WHOLE[LOC(WHOLE=0),];
  COUNT_ALL = NROW(ALL);
  *print count_all;

  DIF = DIFF[LOC(WHOLE=0),];

  indices = LOC(DIF<=0);

  if nrow(indices) > 0 then
        do;
        TEST_N = DIF[LOC(DIF<=0),];
        COUNT_N = NROW(TEST_N);
                COUNT_FP = COUNT_ALL-COUNT_N;

                SPEC = COUNT_N/COUNT_ALL;
        end;
  else
        do;
                SPEC = 0;
                COUNT_N=0;
                COUNT_FP = COUNT_ALL;
        end;

end;
  else
        do;
                SPEC = 0;
                COUNT_N=0;
                COUNT_FP=0;
        end;

  total = count_p + count_fn + count_n + count_fp;

  /* PUT CUTOFF SENS SPEC INTO MATRIX O FOR OUTPUT */
  O[1,1]=CUTOFF;
  O[1,2]=SENS;
```

```
      O[1,3]=SPEC;
      O[1,4]=COUNT_P;
      O[1,5]=COUNT_FN;
      O[1,6]=COUNT_N;
      O[1,7]=COUNT_FP;
      O[1,8]=total;

*END;


CREATE output_&datain FROM O;
APPEND FROM O;

FINISH;

PAPINDEX = {
                  0.125242983

            };

M_DATA={
-1.25377434
-0.04349457
0.398587181
0.974020696
0.628338215
-0.398300846
-1.136834931
-0.928446538
0.324550239
0.446623347
0.423307534
0.557267802
0.56290349
1.169726716
-1.152740775
0.953544524
1.08446455
-0.601697854
-0.045307415
-0.07313421
-0.168501509
0.018037026
-0.059059567
-0.106282541
-0.094116204
0.105199239
0.128857791

};


COEFF={

-0.240051027
0.856451773
0.979423668
```

```
0.023714956
0.453055858
1.288089915
0.271415093
-0.872126917
-4.265628314
-1.245263917
3.522494656
1.640683023
-0.232903711
-1.362671333
-1.182066208
-0.142103353
-0.143789807
0.072661232
0.659934136
1.00548919
-0.673499706
-0.076893887
0.602591255
0.180517186
-0.506365442
0.118540394
-3.475703024
-1.050030293
0.207077066
2.927393365
4.837768785
2.95374269
1.706366089
-3.901217367
-3.650382921
-0.813856632
-0.620491817
2.01647467
0.384289642
-0.103000692
-2.164024083
0.944074211
-6.034518636
-4.273246713
4.226157558
3.648600284
0.704088782
-0.04238803
-1.425621527
0.03693343
1.049837729
2.215755903
0.056270115
-3.511440849
-5.306506974
6.32501039
-1.299033447
3.708766413
-1.068075404
-0.314722398
```

```
-1.872428141
-0.634846616
1.489399271
1.624486866
-0.567343592
1.321689613

            };

RUN Sens_Spec;
quit iml;
run;

data output_&datain(rename=(col1=threshold col2=sensitivity
col3=specificity col4=true_positive col5=false_negative
col6=true_negative col7=fase_positive col8=total) );
      set output_&datain;
run;

%mend;

%count_patients(datain= normal );
%count_patients(datain= benign );
%count_patients(datain= ascus_n_neoplasia );
%count_patients(datain= ascus_neoplasia );
%count_patients(datain= lsil );
%count_patients(datain= agus );
%count_patients(datain= asc );
%count_patients(datain= hsil );
%count_patients(datain= cancer );

data all_19;
      set output_normal output_benign output_ascus_n_neoplasia
output_ascus_neoplasia output_lsil output_agus output_asc output_hsil
output_cancer;
run;
```

## APPENDIX IX SAS CODE FOR DATA MANIPULATION (POINT ANALYSIS)

```
/* This is a program to manipulate data for point analysis
   file: point_analysis_mani.sas

  created by: Chenghong Shen
  last update: 12/05/2005

*/

%include 'c:\spectrx\fan\missing_mac.sas';
libname After 'c:\spectrx\PointAnalysis\';

options nonotes;
options nonumber nodate;

data After.M1;
        set _null_;
run;

%macro getpointdata(path1 = , path2 = , path3 = , path4 = , path5 = ,
      file = , spacing = 10, dataout =,
                subselect = 1, pointselect = 0, disq = no, extype = manual, spectype = orig,
                /*sub_id=, point_start=, point_end=, reflec=, fluore1=, fluore2=, fluore3=*/);

        data demo;
                infile "&path1&file" expandtabs lrecl = 10000 missover;
                input sub_id$ available unclean datec$ whole1 sitepath qa1 PriorPap
PriorPaptype DayofPap DayofPaptype PreferredPap PreferredPaptype scjvisible colpoadequacy
Age Race menstrual Menopause Gravida Para Abort Birthcontrol Priorsurgery1
DaysPriorsurgery1 Priorsurgery2 DaysPriorsurgery2 Priorsurgery3 DaysPriorsurgery3
Priorsurgery4 DaysPriorsurgery4 Priorsurgery5 DaysPriorsurgery5 height weight smoking
Cigarettesperday;
        d_id = substr(sub_id, 1, 1);
        year = substr(datec, 1, 4); month = substr(datec, 5, 2); day = substr(datec, 7, 2);
        date = mdy(month, day, year);
        %nmissing(varlist = available unclean whole1 sitepath qa1 PriorPap PriorPaptype
                DayofPap DayofPaptype PreferredPap PreferredPaptype scjvisible
colpoadequacy Age Race
                menstrual Menopause Gravida Para Abort Birthcontrol Priorsurgery1
DaysPriorsurgery1 Priorsurgery2 DaysPriorsurgery2 Priorsurgery3 DaysPriorsurgery3
Priorsurgery4 DaysPriorsurgery4 Priorsurgery5 DaysPriorsurgery5 height weight smoking
Cigarettesperday, missing = -1 -2);
        if available and &subselect;
 run;

 proc sort data = demo; by sub_id; run;

 data _null_; set demo end = last;
    call symput('sub'||left(_n_), trim(left(sub_id)));
    if last then call symput('nsub', _n_);
 run;

 proc sort data = demo; by sub_id; run;

 data coordinates;
```

```
        infile 'c:\spectrx\fan\nci\hybrid\data3\HybridInterrogationPointCoordsmm.txt' expandtabs;
        input point x y;
run;

%do i = 1 %to &nsub;
 %put Read Data File For Subject #&i out of %left(&nsub) &&sub&i;

 /* Read the point analysis data */

 %if %sysfunc(fileexist("&path4.&&sub&i.._pointgold.txt")) %then %do;
 data pointgold;

        infile "&path4.&&sub&i..  pointgold.txt" expandtabs lrecl = 100000;
        input point pathology1 pathology2;

  if pathology1>pathology2 then pathology=pathology1;
        else pathology=pathology2;

        if pathology=0.5 then pathology=0;

        drop pathology1 pathology2;

 run;


 data pointcat;
        infile "&path3.&&sub&i.._excl_&extype..txt" expandtabs;
        input point reject;
 run;

 Data org;

        infile "&path2.&&sub&i.._spectra_&spectype..txt" expandtabs lrecl = 100000;

        input point rf_1-rf_63 b1-b4 f1_1-f1_59 b5-b8 f2_1-f2_53 b9-b12 f3_1-f3_41;
        array rf rf_1-rf_63; array f1 f1_1-f1_59; array f2 f2_1-f2_53; array f3 f3_1-f3_41;


        %spacingselfnorm;
        sub_id = "&&sub&i";

 run;

 data org_merge;
        merge org pointgold pointcat;
        if reject in (&pointselect);
        by point;
 run;


data After.M1;
        set After.M1 org_merge;
run;
%end;
%end;
```

```
%mend;


%macro spacingselfnorm;
        %let t1 = 31; %let t2 = 29; %let t3 = 26; %let t4 = 20;
    array nrf nrf_1-nrf_&t1; array nf1 nf1_1-nf1_&t2; array nf2 nf2_1-nf2_&t3; array nf3 nf3_1-
nf3_&t4;
        array rnrf rnrf_1-rnrf_&t1; array rnf1 rnf1_1-rnf1_&t2; array rnf2 rnf2_1-rnf2_&t3; array
rnf3 rnf3_1-rnf3_&t4;

        %if &spacing = 5 %then %do;
                do i = 1 to &t1; nrf(i) = rf(i); end;
                do i = 1 to &t2; nf1(i) = f1(i); end;
                do i = 1 to &t3; nf2(i) = f2(i); end;
                do i = 1 to &t4; nf3(i) = f3(i); end;
        %end;
        %else %if &spacing = 10 %then %do;
                do i = 1 to &t1; nrf(i) = (rf(2 * i - 1) + rf( 2 * i)) / 2; end;
                do i = 1 to &t2; nf1(i) = (f1(2 * i - 1) + f1(2 * i)) / 2; end;
                do i = 1 to &t3; nf2(i) = (f2(2 * i - 1) + f2(2 * i)) / 2; end;
                do i = 1 to &t4; nf3(i) = (f3(2 * i - 1) + f3(2 * i)) / 2; end;
        %end;
        %else %do;
                do i = 1 to &t1; nrf(i) = (rf(4 * i - 3) + rf( 4 * i - 2) + rf(4 * i - 1) + rf(4 * i)) / 4; end;
                do i = 1 to &t2; nf1(i) = (f1(4 * i - 3) + f1(4 * i - 2) + f1(4 * i - 1) + f1(4 * i)) / 4; end;
                do i = 1 to &t3; nf2(i) = (f2(4 * i - 3) + f2(4 * i - 2) + f2(4 * i - 1) + f2(4 * i)) / 4; end;
                do i = 1 to &t4; nf3(i) = (f3(4 * i - 3) + f3(4 * i - 2) + f3(4 * i - 1) + f3(4 * i)) / 4; end;
        %end;


        avgnrf = mean(of nrf  1-nrf  &t1); stdnrf = std(of nrf  1-nrf  &t1);
        avgnf1 = mean(of nf1  1-nf1  &t2); stdnf1 = std(of nf1  1-nf1  &t4);
        avgnf2 = mean(of nf2  1-nf2  &t3); stdnf2 = std(of nf2  1-nf2  &t4);
        avgnf3 = mean(of nf3_1-nf3_&t4); stdnf3 = std(of nf3_1-nf3_&t4);

        do i = 1 to &t1; rnrf(i) = (nrf(i) / avgnrf); end;
        do i = 1 to &t2; rnf1(i) = (nf1(i) / avgnf1); end;
        do i = 1 to &t3; rnf2(i) = (nf2(i) / avgnf2); end;
        do i = 1 to &t4; rnf3(i) = (nf3(i) / avgnf3); end;


%mend;

%getpointdata(path1 = c:\spectrx\fan\Aftertrain\,

    /* Data for training  */
    path2 = c:\spectrx\workdir\DATA2\,
    path3 = c:\spectrx\workdir\manual2\,
        path4 = c:\spectrx\workdir\point_analysis\,
        path5 = c:\spectrx\workdir\graph\,
    /*sub  id =4124,
    point  start =29,
        point  end =33,
        reflec =1,
        fluore1 =1,
```

```
        fluore2 =1,
        fluore3 =1, */

file = HybridFINAL_ClinicalData_dm_2.txt, spacing = 10, dataout = All, disq = yes,
subselect = (unclean = 0 and whole1~= .));
```