11-17-2008

Infrared Spectroscopy in Combination with Advanced Statistical Methods for Distinguishing Viral Infected Biological Cells

Tian Tang

Infrared Spectroscopy In Combination With Advanced Statistical Methods For Distinguishing Viral Infected Biological Cells

by

TIAN TANG

Under the Direction of Yu-Sheng Hsu

ABSTRACT

Fourier Transform Infrared (FTIR) microscopy is a sensitive method for detecting difference in the morphology of biological cells. In this study FTIR spectra were obtained for uninfected cells, and cells infected with two different viruses. The spectra obtained are difficult to discriminate visually. Here we apply advanced statistical methods to the analysis of the spectra, to test if such spectra are useful for diagnosing viral infections in cells. Logistic Regression (LR) and Partial Least Squares Regression (PLSR) were used to build models which allow us to diagnose if spectral differences are related to infection state of the cells. A three-fold, balanced cross-validation method was applied to estimate the shrinkages of the area under the receiving operator characteristic curve (AUC), and specificities at sensitivities of 95%, 90% and 80%. AUC, sensitivity and specificity were used to gauge the goodness of the discrimination methods. Our statistical results shows that the spectra associated with different cellular states are very effectively discriminated. We also find that the overall performance of PLSR is better than

that of LR, especially for new data validation. Our analysis supports the idea that FTIR microscopy is a useful tool for detection of viral infections in biological cells.

INDEX WORDS: Wilcoxon Rank Sum Test, Logistic Regression, Partial Least Square Regression, Area under the ROC Curve, Sensitivity and specificity, Cross-validation, Infrared spectroscopy

Infrared Spectroscopy In Combination With Advanced Statistical Methods For Distinguishing Viral Infected Biological Cells

by

TIAN TANG

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

2008

Infrared Spectroscopy In Combination With Advanced Statistical Methods For Distinguishing Viral Infected Biological Cells

by

TIAN TANG

Major Professor: Dr. Yu-Sheng Hsu

Committee: Dr. Gary Hastings
Dr. Jiawei Liu

Electronic Version Approval:

Office of Graduate Studies
College of Arts and Sciences
Georgia State University
December 2008

ACKNOWLEDGEMENTS

I would like to express my deep and sincere gratitude to all those who gave me help to complete this thesis.

Firstly, I am deeply grateful to my supervisor, Dr. Yu-Sheng Hsu. Throughout my thesis-writing period, he uses his enthusiasm and his inspiration to provide me encouragement, good teaching, and a lot of great ideas. I would have been lost without his help.

Secondly, I wish to express my warm and sincere thanks to Dr. Gary Hastings, a professor in the Department of Physics & Astronomy. He provided me all of the original data and the source. I must also thank Jing Guo, a PhD student of Dr. Hasting. She helped me to understand the biological and physical background of the study and provided me many valuable materials to be applied in this thesis.

I would also like to thank Dr. Jiawei Liu for taking the time to read this thesis and provide useful comments.

Last but not least, I would like to give my special thanks to my husband, Chen Zhu. His patient love enabled me to never give up and finally complete this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF GRAPHS

Chapter I

Introduction

Patients can be benefit from the early detection of infectious disease, because more effective treatments can be performed at the early stage. Unfortunately, current methods of disease detection, such as detection of pathogen-specific macromolecules or host antibody production, require days before a diagnosis can be made. As a result, it would be more desirable to obtain a method which can detect infection before the onset of symptoms. Fortunately, scientists have already begun exploring the application of Fourier transform infrared (FTIR) spectroscopy in Biomedicine (Cohenford et al., 1997; wong et al., 1991; Jackson et al., 1998; Mantsch et al., 1996). Infrared (IR), a kind of electromagnetic radiation, with a longer wavelength than UV and visible radiation, can penetrate to a greater depth and be absorbed with less scattering by the tissue. In addition, many of the vibration bands in the IR region are well resolved; thus, during development of the disease, subtle changes in the molecular structure could be detected (Yazdi et al., 1996; Benedetti et al., 1997; Chiriboga et al, 1998; Yang et al., 1995). These features of IR techniques show that FTIR could be applied as an accurate and sensitive method for the diagnosis and study of different diseases.

To investigate the effectiveness of FTIR spectroscopy for early detection of infections by viruses, we use Herpes family of viruses and Adenoviruses in our study. Herpes family of viruses, which contains several members like Herpes simplex types 1 and 2 (HSV1, HSV2), and Varicella zoster (VZV) viruses, is involved in many severe infections (disorders) in animals and humans. Adenoviruses, a group of viruses which infect the membranes (tissue linings) of the

respiratory tract, the eyes, the intestines, and the urinary tract, are responsible for 5-10% of upper respiratory infections in children and many infections in adults as well. We use HSV1 virus (HSV1) and Adenoviruses (Adeno) in our study.

Various studies have been done to investigate the possibility of developing FTIR microscopy as a diagnostic method. Salmn et al. (2002) have applied Cluster analysis to show that FTIR microscopic signatures can be used to differentiate normal cells from herpes-infected cells. According to Alam et al. (2004), activated murine (mouse) macrophage cells can be distinguished from live cells before activation using Principal Components Analysis (PCA) coupled with Linear Discriminate Analysis (LDA) and K-Nearest Neighbor (K-NN) models. Burattini et al. (2008) have applied two multivariate statistical analysis methods - Hierarchical Cluster Analysis (HCA) and PCA - to compare the spectral behavior of *S. cerevisiae* in model wine medium and base wine, before and after 5 days of autolysis. It was proven by this study that FTIR microspectroscopy is a rapid and accurate tool to simultaneously probe the major biochemical events associated with the autolytic process. However, most of the studies were only focused on the differentiation between normal cells and infected cells. In this study, we will use Logistic Regression (LR) and Partial Least Square Regression (PLSR) to perform the diagnosis of two different viruses (HSV1 and Adeno). In addition, we also compare normal cells (Mock) and viruses-infected cells.

In this study, monkey kidney (Vero) cells were grown at 37℃ in an RPMI medium supplemented with 10% new-born calf serum (NBCS) and the antibiotics penicillin, streptomycin and neomycin. HSV1 and Adeno were used for infecting the cells. FTIR measurements were performed in transmission mode with a liquid nitrogen-cooled MCT detector of FTIR microscope, coupled to the FTIR spectrometer. The spectra were obtained in the

wavenumber range of 700-2000 cm^{-1} in the mid-IR region. A spectrum was taken as an average of 64 scans to increase the signal to noise ratio, and the spectral resolution was at 2 cm^{-1}. All of the FTIR measurements included in our study were all taken at 24 hours postinfection (24 hp.i).

The data used to build models were all obtained on March 28, 2008, and that used to validate these models were obtained on April 16, 2008.The former data include 79 HSV1, 94 Adeno, and 69 Mock samples. The latter data include 79 HSV1, 84 Adeno, and 80 Mock samples.

The thesis is organized as follows: In Chapter II, we will introduce the whole process and the methodologies used in the thesis, including variable pre-selection and stabilization, LR and PLSR, Area under the ROC Curve (AUC), sensitivity and specificity, cross-validation and how to use new data to validate an existing model. In Chapter III, we will respectively present the results of comparison between Mock and viruses-infected cells, or between two different kinds of viruses-infected cells. Chapter IV discusses possible future studies. All SAS code involved in the thesis are attached as Appendices.

\

Chapter II

Methodology

2.1 Data Standardization

In this study, monkey kidney (Vero) cells were infected by HSV1 and Adeno viruses, and the absorbances of spectra on the wavenumber range of 800-1500 cm^{-1} are studied. For each observation point for mock or infected cells, 728 FTIR measurements were taken respectively. At the first step we would like to standardize all observations, because it will make the data easy to compare. The standardized data obtained by subtracting the mean and then being divided by the standard deviation of each cell. That is, for each point in the same batch, the standardized data is

$$y_i = \frac{x_i - \overline{x}}{s_x} \quad ,$$

where $x_i, i = 1,2,\cdots,728$ are the 728 absorbance at one point, $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ is the mean, and $s_x = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2}$ is the standard deviation.

2.2 Variable Pre-selection by Wilcoxon Rank Sum Test

For every two kinds of cells, we can use the standardized data to draw a graph, in which every cell is shown as a curve connecting 728 standardized FTIR measurements. Because of the overlapping between these curves, it is difficult to differentiate between two kinds of cells with visual judgment. However, if we use Wilcoxon Rank Sum Test (WRST) to calculate the

standardized test statistic for the set of spectra taken on every specific wavenumber, we may find some wavenumber ranges which can discriminate two kinds of cells we want to compare.

In statistics, Wilcoxon Rank Sum Test, or Mann-Whitney test, is one of the non-parametric tests for assessing whether two samples of observations come from the distribution with same mean. Being different with two-sample t-test, which tests for differences in means, the WRST test is more robust against outliers, and is more sensitive to the distributions.

We assume that we have independent random samples $x_1, x_2, \cdots, x_m$ and $y_1, y_2, \cdots, y_n$, of sizes m and n respectively, from each population. We then rank the pooled sample from lowest to highest. All sequences of ties are assigned an average rank. The Wilcoxon test statistic W is the sum of the ranks from population X. For large samples, the distribution of W can be approximated by a Normal distribution $N(\mu, \sigma)$. The mean and standard deviation μ and σ are given by

$$\mu = \frac{m(m+n+1)}{2}$$

and

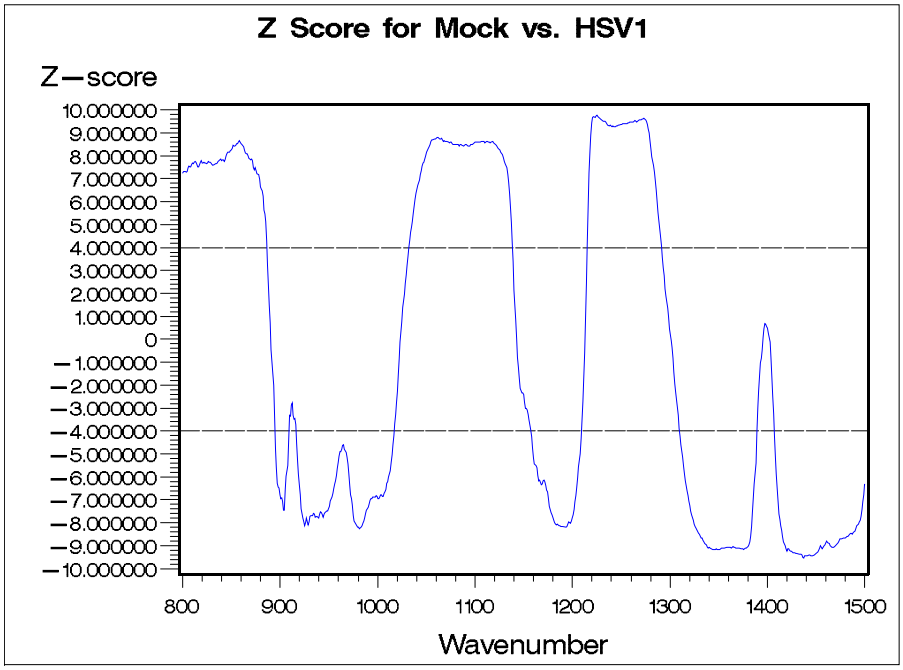
$$\sigma = \sqrt{\frac{mn(N+1)}{12}}$$

where $N = m + n$.

We test the null hypothesis H_o: No difference in means. A two-sided alternative is H_a: there is a difference in means. In this case, the p-value is given by

$$P\left(Z > |z|\right),$$

where $z = \frac{w - \mu}{\sigma}$.



Graph-1 Z-score for Mock vs. HSV1

For any two different kinds of cells, A and B, with sample sizes m and n respectively, we can obtain the Z-score of a group of measurements with sample size $m+n$ for each specific wavenumber by WRST. Because 728 FTIR measurements were taken in the wavenumber range of 800-1500 cm^{-1} for each cell, we will have 728 Z-scores. As can be seen in Graph-1, these 728 Z-scores can be connected by a smoothed curve. We can then apply Bonferroni method to obtain the critical value $z_{0.05/(2\times728)}$, which is approximately equal to 4, since here we have 728 dependent multiple significant tests. For those Z-scores which are larger than 4 or smaller than -4, the data on corresponding wavenumbers are significant at level of 0.05 simultaneous, which

means that we can differentiate the two kinds of cells on those significant wavenumbers. As shown in Graph-1, we can easily find the significant ranges of wavenumbers by using reference lines $y = \pm 4$. That is, the ranges which are above the upper line or below the bottom line are significant.

2.3 Model Building with Logistic Regression and Partial Least Square Regression

After finding out which parts of wavenumbers are significant, we would like to build a Logistic Regression or Partial Least Squares Regression models. In order to stabilize the data and reduce the noise, we take the average of every neighboring five spectrums in the ranges selected from the WRST. Those averages are used as independent or predictor variables in the regression. The response variable is binary, which usually denoted by either 1 (disease) or 0 (non-disease). For example, when we would like to diagnose viruses-infected cells from Mock, the response variable should equal to 1 if the data come from viruses-infected cells and equal to 0 if the data come from Mock. Therefore, instead of applying Multiple Ordinary Linear Regression (MOLR) models, we use two popular statistical methods, LR and PLSR to discriminate cells.

LR is a type of predictive model that can be used when the target variable is categorical. LR model yields the probability of occurrence of an event by fitting data to a logistic curve. In other words, LR is estimates $p(Y|X)$, where Y is discrete, and $X = (X_1, X_2, \cdots, X_n)$ is any vector containing discrete or continuous variables.

The relationship between the predictor and response variables is not a linear function in LR. Instead, the LR finds a linear combinations of X, which is the logit transformation of the probability of success g, i.e.

$$\log it\ g = \log \frac{g}{1-g} = \alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n ,$$

or equivalently

$$g = P(Y = 1 | X) = \frac{e^{(\alpha+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_n x_n)}}{1+e^{(\alpha+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_n x_n)}},$$

where α is the constant of the equation, and $\beta_i, i = 1, \cdots, n$ are the coefficients of the predictor variables.

Variable selection is important in any model building process especially in case that the number of variables is large. After the Variable Pre-selection by WRST and data stabilization by taking the average of every neighboring five spectrums, we still have around 100 variables. We can use stepwise regression method in LR to select variables.

PLSR is a method for constructing predictive models when the predictor variables are many, and are highly correlated. In PLSR, we extract linear combinations of the predictors, called factors, or latent variables, which can reach two goals-explaining response variation and explaining predictor variation.

For Principal Component Regression (PCR), we use principal components U_1 (the first component), U_2 (the second component), $\cdots$, U_n (the *n*th component) as predictor variables, where principal components are linear combinations of X, such that their variances are maximized and are all independent. However, PLSR uses variable combinations U_1 (the first factor), U_2 (the second factor), $\cdots$, U_n (the *n*th factor), such that

$$\max_{\substack{\|\alpha_i\|=1 \\ \alpha_i^T S\alpha_l=0, l=1,\cdots,i-1}} Corr^2(Y,U_i)Var(U_i),$$

where $U_i = X\alpha_i, i = 1,\cdots,n$, S is the sample covariance matrix, X is a $I \times J$ matrix which contains all the values of J predictor variables collected on I observations, and Y is a $I \times 1$ matrix storing the I observations described by the dependent variable. The conditions $\alpha_i^T S\alpha_l = 0, l = 1,\cdots,i-1$ ensure that $U_i = X\alpha_i$ is uncorrelated with all the previous linear combinations $U_l = X\alpha_l, l = 1,\cdots i-1$.

A PLSR model can be shown as

$$Y = f_1U_1 + f_2U_2 + \cdots + f_nU_n + E_n,$$

where $U_i, i = 1,\cdots,n$ are factors, and $f_i, i = 1,\cdots n$ are the coefficients of them.

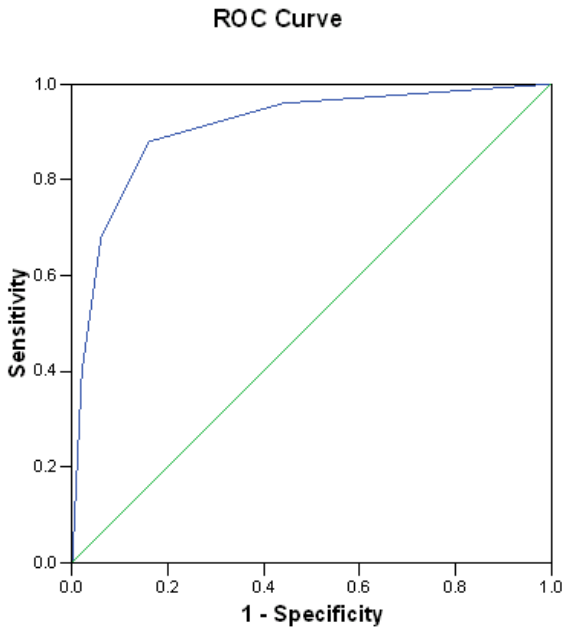
Variable reduction is also used in PLSR. It extracts latent factors which are linear combinations of the original predictor variables. There are many ways to select the number of factors included in the PLSR model. We simply use the number of factors which count about 95% of the total variation.

2.4 Area under the Curve, Sensitivity and Specificity

After building a model, we then need to evaluate its diagnostic performance, the ability to correctly classify two categories. Usually we can use sensitivity and specificity, and the area under the Receiver Operating Characteristic (ROC) curve to make the evaluation.

Sensitivity and specificity are closely related to the concepts of type I and type II errors. Sensitivity measures the proportion of correct identifications among actual positives, such as the

probability of a positive test among patients with disease; and the specificity measures the proportion of correct identifications among all negatives, such as the probability of a negative test among patients without disease.



Graph-2 ROC Curve

A complete description of classification is given by the area under the ROC curve, which is a plot of the sensitivity against 1-specificity for the different possible cut-off points of a diagnostic model. Each point on the ROC curve represents a sensitivity and specificity pair corresponding to a particular decision threshold. As shown in Graph-2, when the sensitivity increases, the corresponding specificity will decrease. If the objective is to choose an optimal cut-off point for the purpose of discrimination, one might select a cut-off point that maximizes both sensitivity and specificity. An area of 1 represents the high accuracy of discrimination, and

an area of 0.5 represents very low accuracy. A rough guide for classifying the accuracy of discrimination is the traditional academic point system. That is, Area Under the Curve (AUC) between 0.90 and 1 represents excellent discrimination; AUC between 0.80 and 0.90 represents good discrimination; AUC between 0.70 and 0.80 represents fair discrimination; AUC between 0.60 and 0.70 represents poor discrimination; and AUC between 0.50 and 0.60 represents no discrimination.

In this study, we consider AUC and the specificities corresponding to the sensitivities 95%, 90% and 80%.

2.5 Three-fold Balanced Cross-validation

Most model fitting procedures often yield over-fitting problem. In other words, the goodness of the procedure obtained from the sample is frequently over-rated. This is what we usually referred as the shrinkage. Calculating the shrinkages of AUC and the specificities corresponding to the sensitivities 95%, 90% and 80% is certainly necessary for the next step of this study.

Cross-validation, a method of estimating sampling error, can be used to assess the shrinkage of the AUC and specificities of the model we built. In *K*-fold cross-validation, the original sample is randomly divided into *K* approximately equal size subsets. Of the *K* subsets, a single subset is retained as the validation data, and the remaining *K-1* subsets as a whole are used as training data which is used to build the model. The cross-validation process is then repeated *K* times, with each of the *K* subsets used exactly once as the validation data. The *K* results then can be averaged to produce a single estimation.

We employ three-fold balanced cross-validation to examine the accuracy of the AUC and specificity found in the models. The original data are randomly divided into three balanced subsets in which not only the three subsets have approximately equal size but also each subset

has almost same number of observations in both categories. AUC and specificities which correspond to the sensitivities 95%, 90% and 80% are calculated by validation data and training data respectively. We then obtain the shrinkages by subtracting AUC and specificities for validation data from the ones for training data. An average of shrinkage for AUC or specificities can be obtained by one cross-validation process. In the end, we can acquire the average of the n averages of shrinkage for AUC or specificities by repeating the process for *n* times. In our study, we repeat 100 times. The average shrinkage of AUC or specificities then can be used to subtract from original sample estimates to obtain the final estimations.

2.6 New Data Validation

The new data validation can be applied to evaluate the model we built from the data set. In the process of new data validation, we apply a completely new data set to the final model which we built from the old data set, and obtain the AUC and specificities respectively. Because we do not change the coefficient and the variables of the model, the new data validation shows the shrinkages of the final model. Small shrinkages of AUC and specificities imply that the diagnostic performance of the model is very stable.

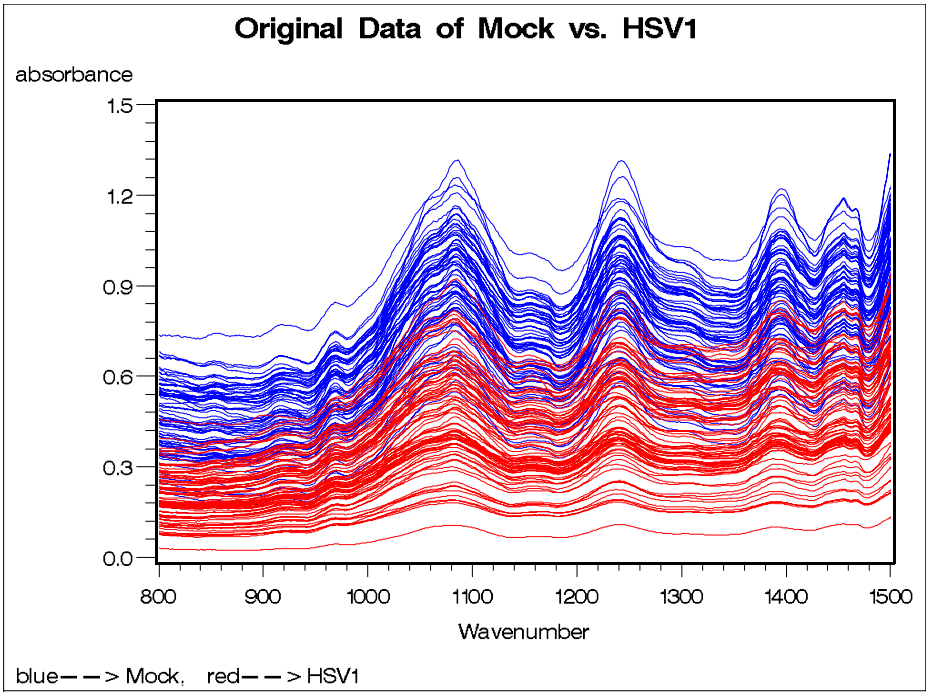
Chapter III

Results and Conclusion

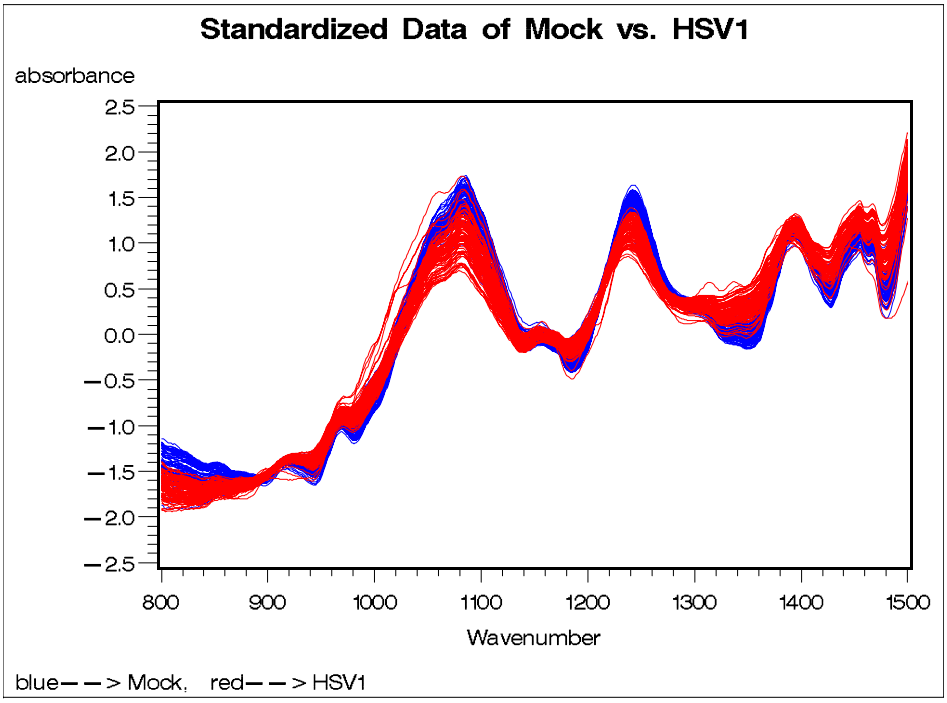
3.1 Mock versus HSV1

The original data for Mock and HSV1 are shown in Graph-3, in which the blue curves represent Mock cells and the red curves represent HSV1-infected cells. As can be seen in the graph, most of the blue curves are above the red curves in the range of 800-1500 cm-1; that is, we cannot find the wavenumber ranges in which the two kinds of cells can be easier to be differentiated. Graph-4 shows us the standardized data for Mock and HSV, in which blue curves also represent Mock cells and red curves represent HSV1-infected cells. In the graph, these two kinds of curves overlap each other a lot; however, we can still find some overall trend. For instance, in the region of 800-880 cm-1, some of the blue curves are above red curves, and in some of regions, such as 1310-1380 cm-1, most of the red curves are below the blue curves.

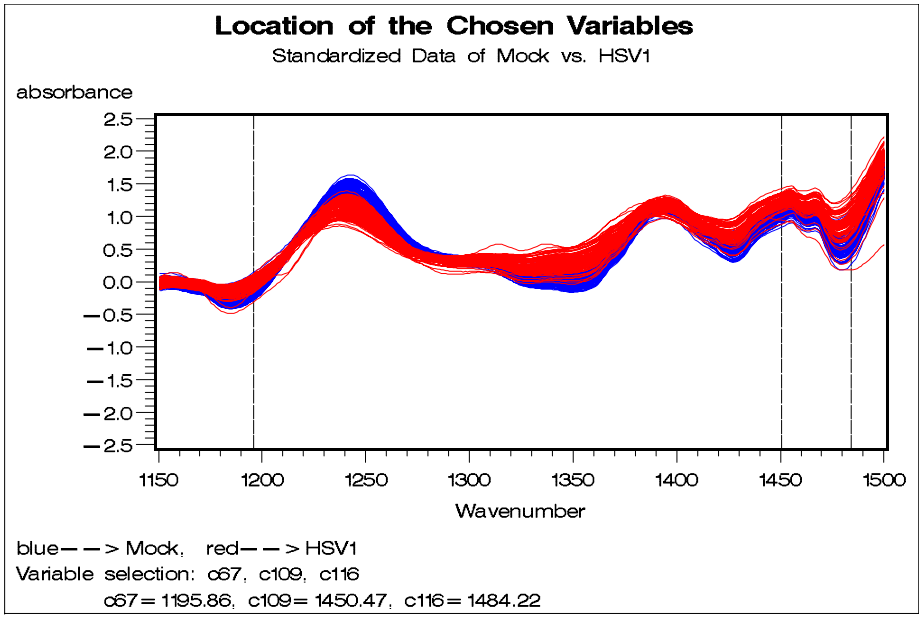
Graph-1 shows us the Z-score of Mock and HSV1 obtained by WRST for each spectrum. Also, we draw two horizontal lines on 4 and -4, which indicate the threshold of multiple statistical significance. As shown in this picture, seven ranges, 800-885 cm^{-1}, 918-1014 cm^{-1}, 1036-1136 cm^{-1}, 1160-1207 cm^{-1}, 1216-1288 cm^{-1}, 1312-1388 cm^{-1}, and 1410-1500 cm^{-1}, are above the top line or below the bottom line; that is, they are significant in the study. For this reason, we focus on the data in these ranges. These significant wavenumber ranges include 595 variables, which can be stabilized into 119 variables (c1-c119) by taking the average of every neighboring five variables.



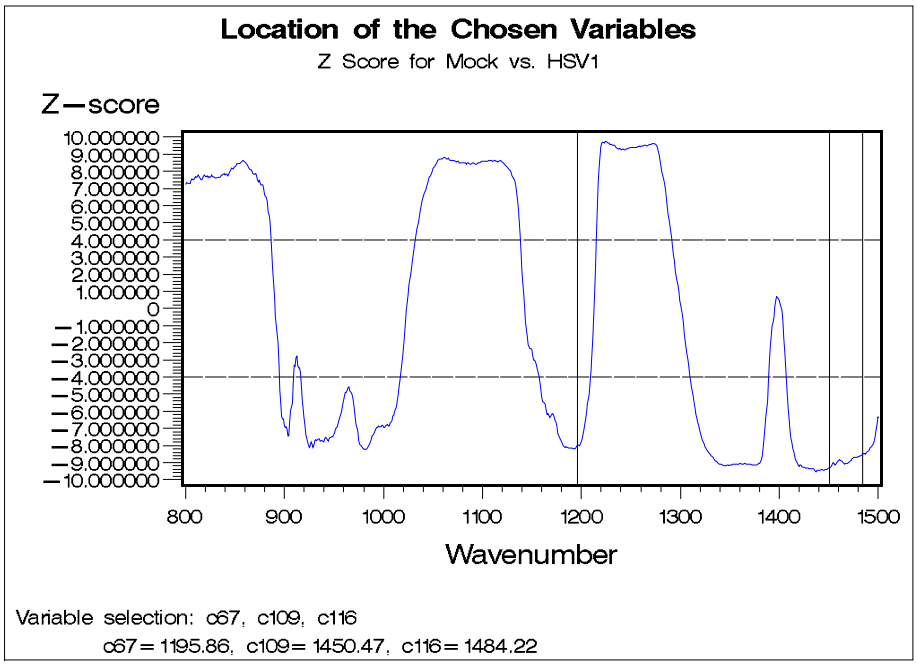
Graph-3 Original data of Mock vs. HSV1



Graph-4 Standardized data of Mock vs. HSV1



Graph-5 Location of chosen variables (Standardized data of Mock vs. HSV1)



Graph-6 Location of the chosen variables (Z-score for Mock vs. HSV1)

After stepwise selection, we choose c67, c109, and c116 to build the LR model. The wavenumbers corresponding to these three variables are 1195.86, 1450.47 and 1484.22 cm^{-1} respectively, and their locations are shown in Graph-5 and Graph-6. Graph-5 shows that the variables we chose are located at the wavenumbers where the two kinds of lines are partially separated, and Graph-6 shows that the z-scores of these variables are below the bottom line and also very small. In other words, these selected variables are very significant in the wavenumber ranges. Table-1 shows us the estimates of coefficients and p-values of these variables for the LR model. As can be seen in the table, the p-values of the three variables as well as the intercept are all very small (<0.01), which means that they are quite significant in the model. The final LR model is

$$p(Y=1|\underset{\sim}{X}) = \frac{e^{g(x)}}{1+e^{g(x)}} \ ,$$

and

$$g(x) = -39.5016+24.2216\times c67+50.1860\times c109-19.3781\times c116 \, .$$

Table-1 LR for Mock vs. HSV1

Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	-39.5016	8.2330	23.0204	<.0001
c109	1	50.1860	10.4437	23.0919	<.0001
c116	1	-19.3781	5.2112	13.8277	0.0002
c67	1	24.2216	6.2265	15.1326	0.0001

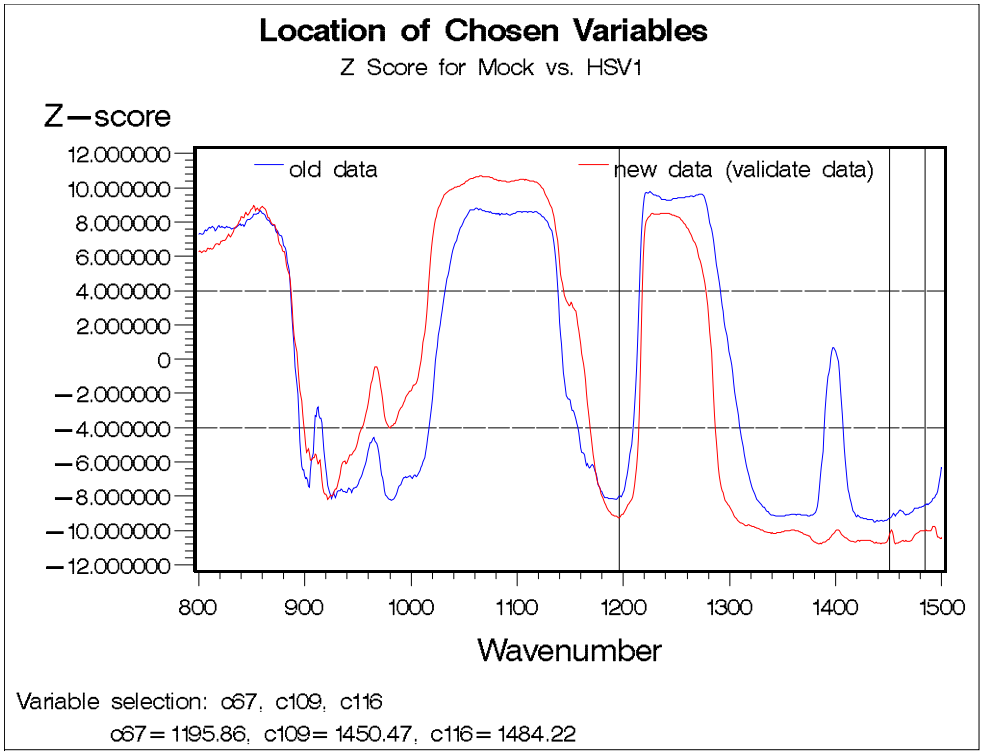
As shown in Table-2, the AUC of the final LR model is equal to 0.970, which represents excellent discrimination, and the specificities for sensitivities of 95%, 90%, and 80% are equal to 0.899, 0.957, and 1 respectively, which are large enough to exhibit excellent discrimination of

the final model. After estimating the shrinkage from the cross-validation method, the AUC is equal to 0.963, which still discriminate well, and for the specificities corresponding to the sensitivities of 95%, 90%, and 80%, which equal to 0.820, 0.938 and 0.989, are very close to the ones calculated by the final model. After the new data validation process, we obtain that the AUC is 0.935, still representing excellent discrimination, and the corresponding specificities are 0.950, 0.950 and 0.950 respectively, which also show no big difference between the ones obtained from the old data. In addition, Graph-7 shows the z-score plot for both of the old data (for build model) and new data (for new data validation). We can see from the graph that there is no huge difference between the two z-score curves, and the variables in the final model are located at very significant wavenumbers for both of the data sets.

Table-2 AUC and specificities corresponding to sensitivities 95%, 90% and 80% for Mock vs. HSV1

Mock vs. HSV1			
Logistic regression			
	The old data	After the shrinkage of Cross-validation	The new validate data
Area under the curve (AUC)	0.970	0.963	0.935
Specificity for 95% Sensitivity	0.899	0.820	0.950
Specificity for 90% Sensitivity	0.957	0.938	0.950
Specificity for 80% Sensitivity	1	0.989	0.950
PLS regression (Number of Factors=5) Percent Variation Accounted for by Partial Least Squares Factors (Model effects)=93.4			
Area under the curve (AUC)	1	0.983	0.989
Specificity for 95% Sensitivity	1	0.999	0.974
Specificity for 90% Sensitivity	1	1	0.975
Specificity for 80% Sensitivity	1	1	0.988

We also use the 119 variables to build the PLSR model. The first 5 factors, which count about 93.4% of the total variation, contain almost all the information from the original 119 variables. The coefficients of the variables for the final model are shown in Table-3. As can be seen in Table-2, the AUC, the specificities corresponding to 95%, 90%, and 80% sensitivities are all equal to 1, which demonstrate super discrimination of the final PLSR model. The AUC and the specificities obtained after estimating the shrinkage from the cross-validation method are equal to 0.983, 0.999, 1 and 1, which also shows superexcellent discrimination of the PLSR method. After the new data validation process, the AUC and the specificities are 0.989, 0.974, 0.975 and 0.988 respectively, displaying the excellent discrimination of the final PLSR model for a new data set.



Graph-7 Location of chosen variables (Z-score for Mock vs. HSV1 for both of the old data and new data)

Table-3 The coefficients of PLSR model for Mock vs. HSV1

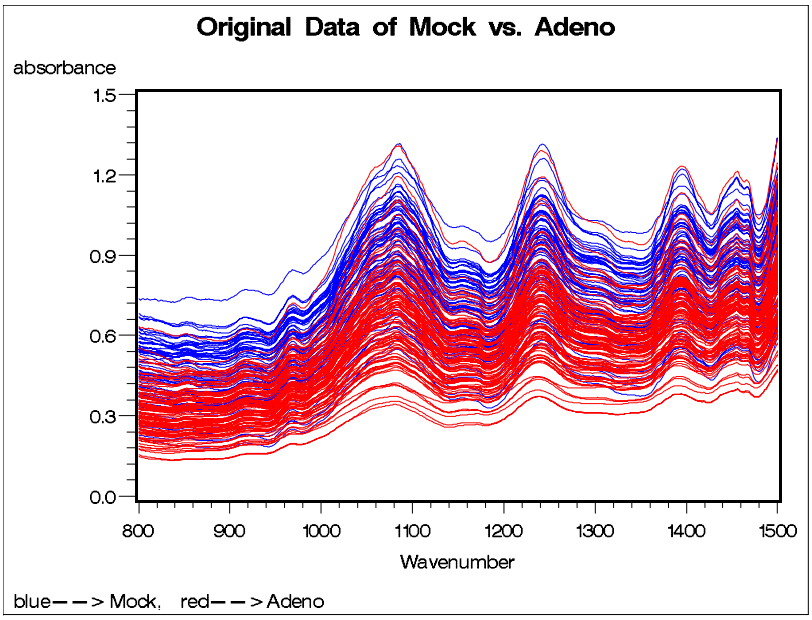
Name	Value	Name	Value	Name	Value	Name	Value
Intercept	-0.06865	c30	0.266898	c60	0.247545	c90	0.064058
c1	-0.00691	c31	0.305447	c61	0.234613	c91	0.153956
c2	-0.01992	c32	0.236876	c62	0.165753	c92	0.155152
c3	-0.0414	c33	0.05054	c63	0.195147	c93	0.096227
c4	0.030196	c34	-0.02684	c64	0.265544	c94	0.023825
c5	0.00548	c35	0.027028	c65	0.249903	c95	-0.03997
c6	0.021325	c36	0.091258	c66	0.226271	c96	-0.07118
c7	0.035525	c37	0.168579	c67	0.126771	c97	-0.02619
c8	-0.0076	c38	0.13993	c68	-0.00556	c98	0.030176
c9	-0.06493	c39	-0.3061	c69	-0.25607	c99	0.073497
c10	-0.09595	c40	-0.28802	c70	-0.52126	c100	0.122499
c11	-0.18185	c41	-0.31271	c71	-0.40015	c101	0.116806
c12	-0.2573	c42	-0.3487	c72	-0.23889	c102	0.135698
c13	-0.3033	c43	-0.46259	c73	-0.09827	c103	0.166782
c14	-0.18447	c44	-0.4513	c74	0.033029	c104	0.15481
c15	-0.11778	c45	-0.29368	c75	0.088522	c105	0.218434
c16	-0.04112	c46	-0.06974	c76	0.056291	c106	0.361255
c17	0.000876	c47	0.233254	c77	-0.04852	c107	0.399425
c18	0.060578	c48	0.415644	c78	-0.09456	c108	0.544927
c19	0.258256	c49	0.541232	c79	-0.17445	c109	0.597024
c20	0.326353	c50	0.560327	c80	-0.22963	c110	0.574578
c21	0.239051	c51	0.49739	c81	-0.33707	c111	0.299601
c22	0.131728	c52	0.304341	c82	-0.43793	c112	0.344346
c23	0.089904	c53	0.175074	c83	-0.47945	c113	0.306861
c24	0.067392	c54	0.08123	c84	-0.51094	c114	-0.08596
c25	0.052189	c55	-0.02151	c85	-0.39061	c115	-0.25838
c26	-0.02497	c56	-0.08273	c86	-0.32194	c116	-0.34026
c27	-0.10927	c57	-0.09028	c87	-0.23766	c117	-0.32037
c28	-0.06554	c58	-0.11332	c88	-0.12525	c118	-0.31033
c29	0.091715	c59	-0.11526	c89	-0.04176	c119	-0.20219

3.2 Mock versus Adeno

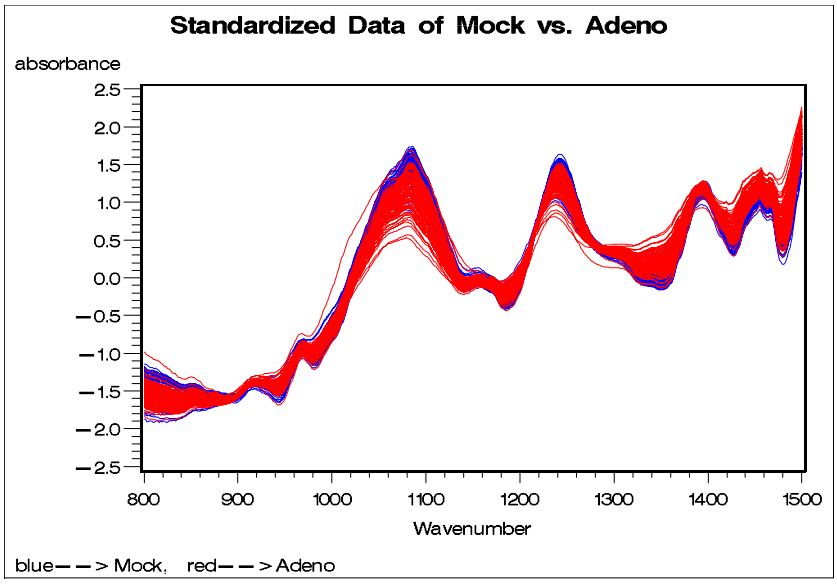
In order to avoid iterant and verbose, we do not repeat the results for other comparisons as detailed as what we did for Mock versus HSV1.

Graph-8 to Graph-10 shows the original data, the standardized data, and the Z-score data respectively. As shown in Graph-10, six ranges, 925-953 cm^{-1}, 1021-1136 cm^{-1}, 1173-1206 cm^{-1}, 1219-1271 cm^{-1}, 1311-1392 cm^{-1}, and 1410-1500 cm^{-1}, are significant in the study. They include

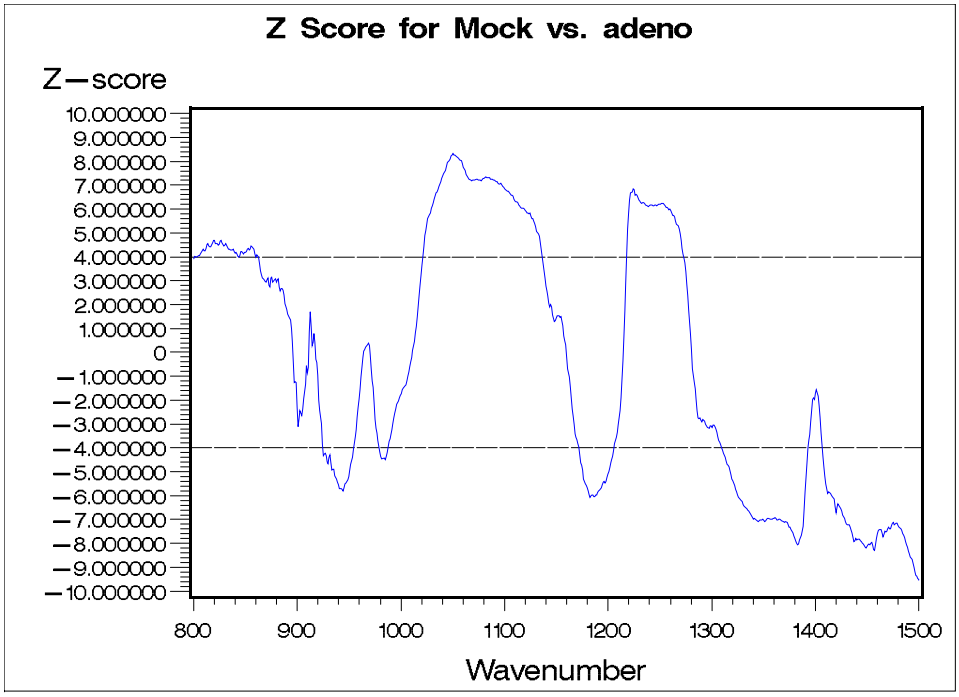
420 variables, which can be stabilized into 84 variables (c1-c84) by taking the average of every five neighboring variables.



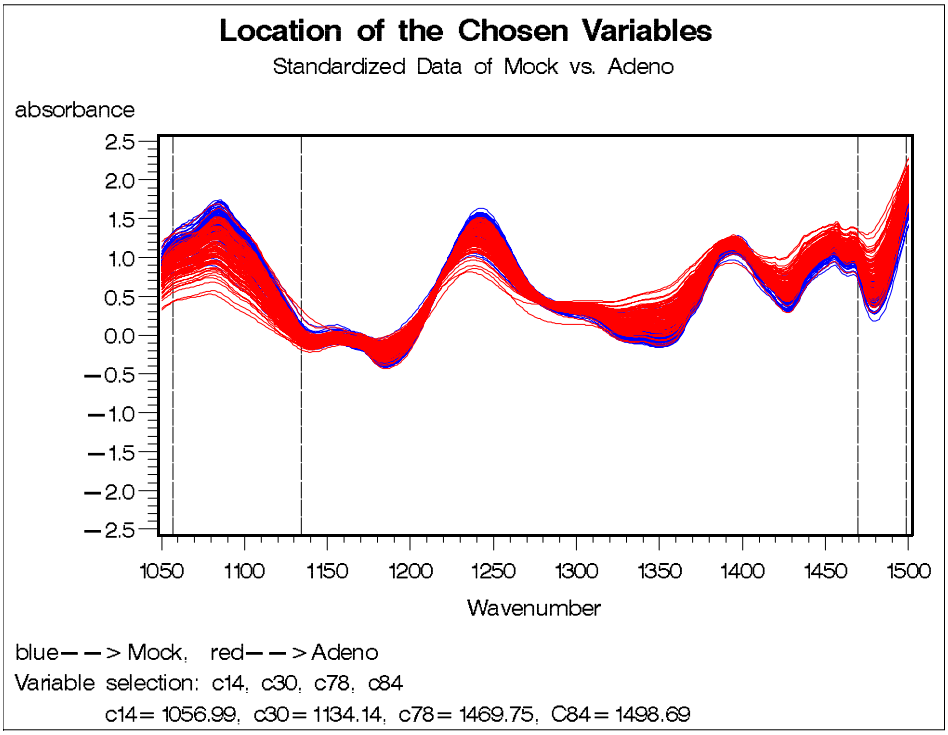
Graph-8 Original data of Mock vs. Adeno



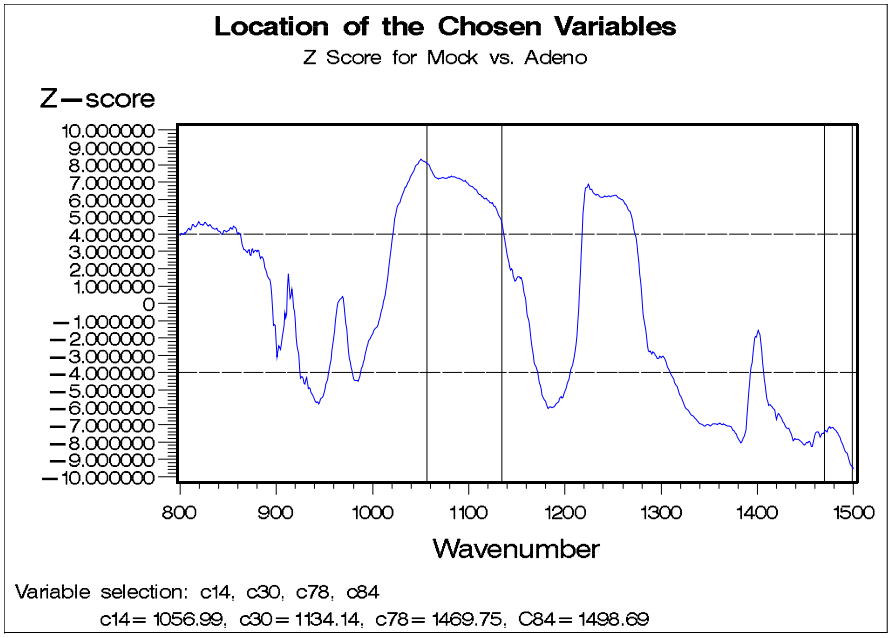
Graph-9 Standardized data of Mock vs. Adeno



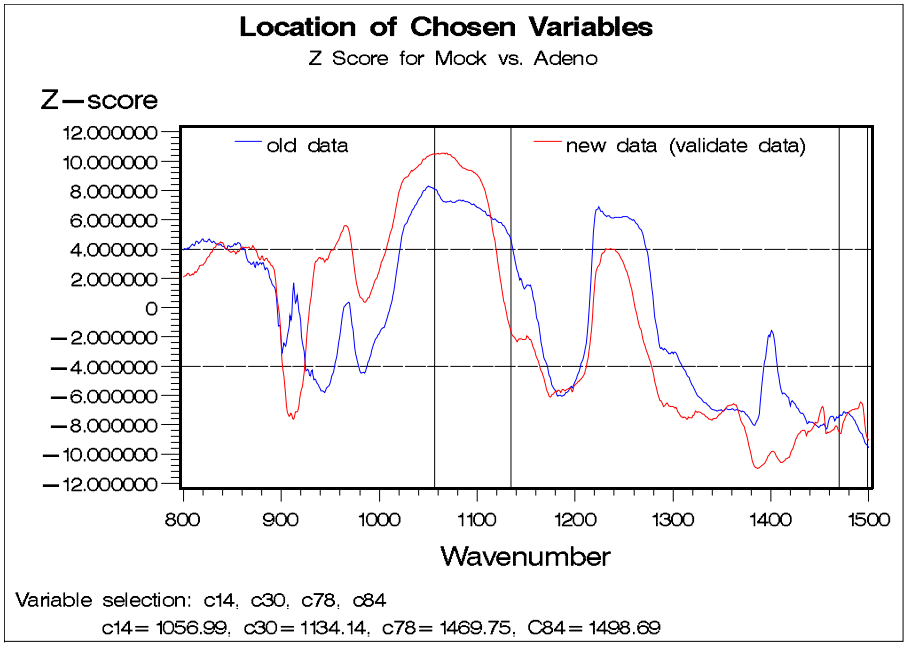
Graph-10 Z-score for Mock vs. Adeno



Graph-11 Location of the chosen variables (Standardized data of Mock vs. Adeno)



Graph-12 Location of the chosen variables (Z-score for Mock vs. Adeno)



Graph-13 Location of chosen variables (Z-score for Mock vs.

Adeno for both of the old data and new data)

In the process of stepwise selection, c14, c30, c78 and c84 were chosen to build the LR model. The wavenumbers corresponding to these four variables are 1056.99, 1131.14, 1469.75 and 1498.69 cm^{-1} respectively, and their locations are shown in Graph-11 and Graph-12. As can be seen in the table-4, the small p-values of the three variables as well as the intercept indicate that the variables we chose are quite significant in the model. The final LR model is

$$p(Y=1|\underset{\sim}{X}) = \frac{e^{g(x)}}{1+e^{g(x)}} \ ,$$

and

$$g(x) = 62.7717 - 69.6928 \times c14 + 54.3219 \times c30 - 121.5 \times c78 + 70.5545 \times c84 \ .$$

Table-4 LR for Mock vs. Adeno

Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	62.7717	23.1063	7.3802	0.0066
c84	1	70.5545	15.6892	20.2232	<.0001
c78	1	-121.5	29.7660	16.6510	<.0001
c14	1	-69.6928	16.4212	18.0122	<.0001
c30	1	54.3219	13.1841	16.9766	<.0001

As shown in Table-5, AUC, specificities for sensitivities 95%, 90%, and 80% of final LR model are all large enough to exhibit excellent discrimination. After estimating the shrinkage from the cross-validation method, the AUC and specificities are very close to the ones calculated by the final model. After the new data validation process, the AUC and specificities do not have big difference with the ones obtained from the old data. Graph-13 shows the z-score plot for both of the old data (for build model) and new data (for new data validation). We can see from the graph that there is some difference between the two z-score curves, and variable c30 is on the border of a significant wavenumbers range for old data but even not significant for new data.

Table-5 AUC and specificities corresponding to sensitivities 95%, 90% and 80% for Mock vs. Adeno

Mock vs. Adeno			
Logistic regression			
	The old data	After the shrinkage of Cross-validation	The new validate data
Area under the curve (AUC)	0.992	0.974	0.993
Specificity for 95% Sensitivity	1	0.951	0.965
Specificity for 90% Sensitivity	1	0.991	0.980
Specificity for 80% Sensitivity	1	0.996	1
PLS regression (Number of Factors=4) Percent Variation Accounted for by Partial Least Squares Factors (Model effects)=95.8			
	The old data	After the shrinkage of Cross-validation	The new validate data
Area under the curve (AUC)	0.982	0.955	0.994
Specificity for 95% Sensitivity	0.958	0.853	0.965
Specificity for 90% Sensitivity	0.986	0.939	0.993
Specificity for 80% Sensitivity	1	0.971	1

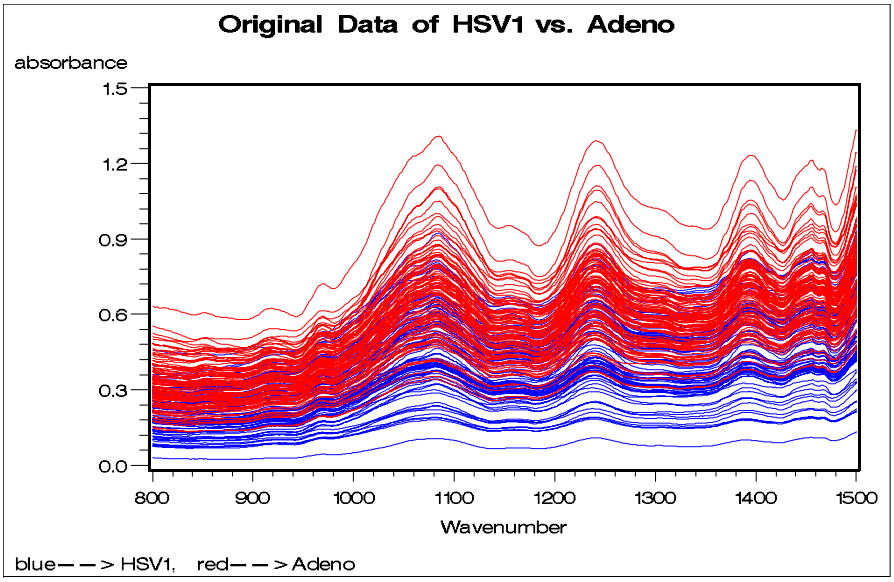
For PLSR model, the first 4 factors count about 95.8% of the total variation. Table-6 shows the coefficients of the variables for the final model. As can be seen in Table-5, the AUC, the specificities corresponding to 95%, 90%, and 80% sensitivities are equal to 0.982, 0.958, 0.986, and 1, which indicates superexcellent discrimination of the final PLSR model. The AUC and the specificities obtained after the shrinkage of the cross-validation equal to 0.955, 0.853, 0.939 and 0.971, also showing excellent discrimination of the PLSR method. After the new data validation process, the AUC and the specificities are 0.994, 0.965, 0.993 and 1, again displaying the excellent discrimination of the final PLSR model for a new data set.

Table-6 The coefficients of PLSR model for Mock vs. Adeno

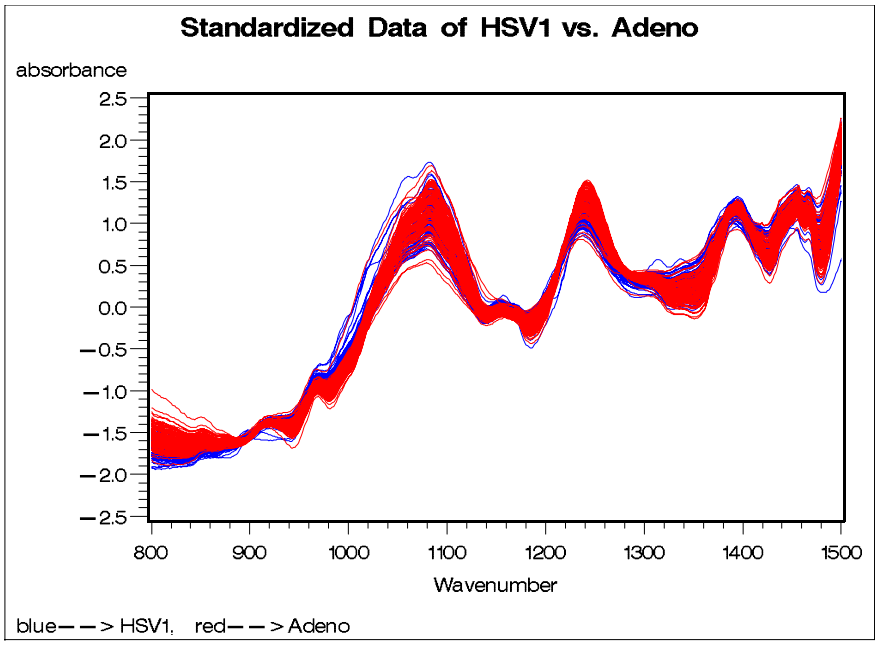
Name	Value	Name	Value	Name	Value	Name	Value
Intercept	0.059213	c22	-0.08519	c43	-0.2922	c64	-0.00393
c1	0.063366	c23	-0.11195	c44	-0.28043	c65	-0.013
c2	0.023038	c24	-0.12662	c45	-0.18709	c66	-0.01416
c3	-0.01617	c25	-0.13175	c46	-0.05161	c67	-0.05976
c4	0.018954	c26	-0.17537	c47	0.1236	c68	-0.13699
c5	-0.01622	c27	-0.21926	c48	0.226979	c69	-0.29996
c6	-0.01442	c28	-0.17704	c49	0.302109	c70	-0.47864
c7	-0.01649	c29	-0.05857	c50	0.317921	c71	-0.39647
c8	-0.05506	c30	0.073756	c51	0.281883	c72	-0.29478
c9	-0.1051	c31	0.122552	c52	0.152077	c73	-0.2114
c10	-0.13255	c32	0.10891	c53	0.065426	c74	-0.13744
c11	-0.19636	c33	0.011424	c54	-0.0025	c75	-0.11277
c12	-0.25363	c34	-0.02184	c55	-0.08078	c76	-0.14446
c13	-0.291	c35	0.024757	c56	-0.13808	c77	-0.21753
c14	-0.21912	c36	0.078894	c57	-0.15431	c78	-0.23818
c15	-0.18485	c37	0.143541	c58	-0.18278	c79	-0.28386
c16	-0.14262	c38	0.141187	c59	-0.189	c80	-0.31866
c17	-0.12462	c39	-0.19302	c60	0.063388	c81	-0.38273
c18	-0.09285	c40	-0.18759	c61	0.034288	c82	-0.43554
c19	-0.01309	c41	-0.20106	c62	-0.03439	c83	-0.45446
c20	0.034067	c42	-0.22015	c63	-0.0385	c84	-0.47054
c21	-0.01983						

3.3 HSV1 versus Adeno

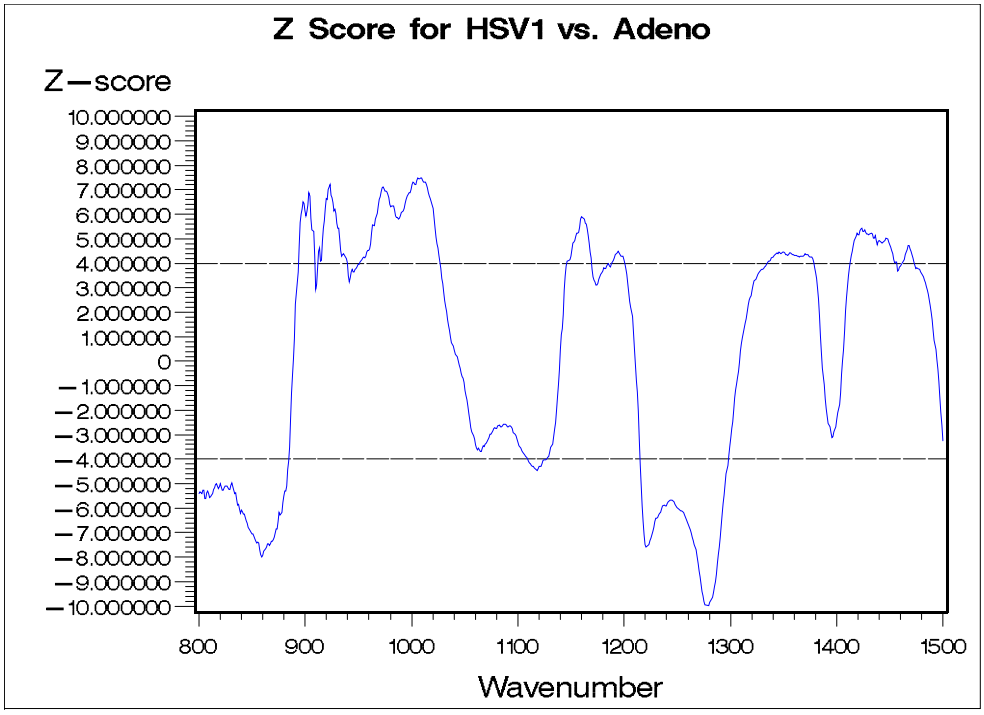
Graph-14 to Graph-16 shows the original data, the standardized data, and the Z-score data respectively. Graph-16 shows that seven ranges, 800-881 cm^{-1}, 915-938 cm^{-1}, 950-1026 cm^{-1}, 1146-1169 cm^{-1}, 1216-1297 cm^{-1}, 1336-1378 cm^{-1}, and 1413-1455 cm^{-1}, are significant in the study. They include 390 variables, which can be stabilized into 78 variables (c1-c78).



Graph-14 Original data of HSV1 vs. Adeno



Graph-15 Standardized data of HSV1 vs. Adeno



Graph-16 Z-score for HSV1 vs. Adneo

Variables c12, c19, c23 and c56 were chosen to build the LR model by stepwise selection. The wavenumbers corresponding to these four variables are 854.46, 921.97, 951.87 and 1275.91 cm^{-1} respectively, and their locations are shown in Graph-17 and Graph-18. Table-7 shows that the p-values of the four variables and the intercept are very small, indicating that these variables are significant in the model. The final LR model is

$$p(Y=1|\underset{\sim}{X}) = \frac{e^{g(x)}}{1+e^{g(x)}} \ ,$$

and

$$g(x) = 61.4356 + 76.4351 \times \text{c12-55.3450} \times \text{c19+88.3002} \times \text{c23+61.4356} \times \text{c56} \, .$$

Table-7 LR for HSV1 vs. Adeno

Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	132.3	42.4598	9.7107	0.0018
c56	1	61.4356	13.1783	21.7331	<.0001
c23	1	88.3002	18.5274	22.7140	<.0001
c12	1	76.4351	16.9445	20.3484	<.0001
c19	1	-55.3450	17.9974	9.4566	0.0021

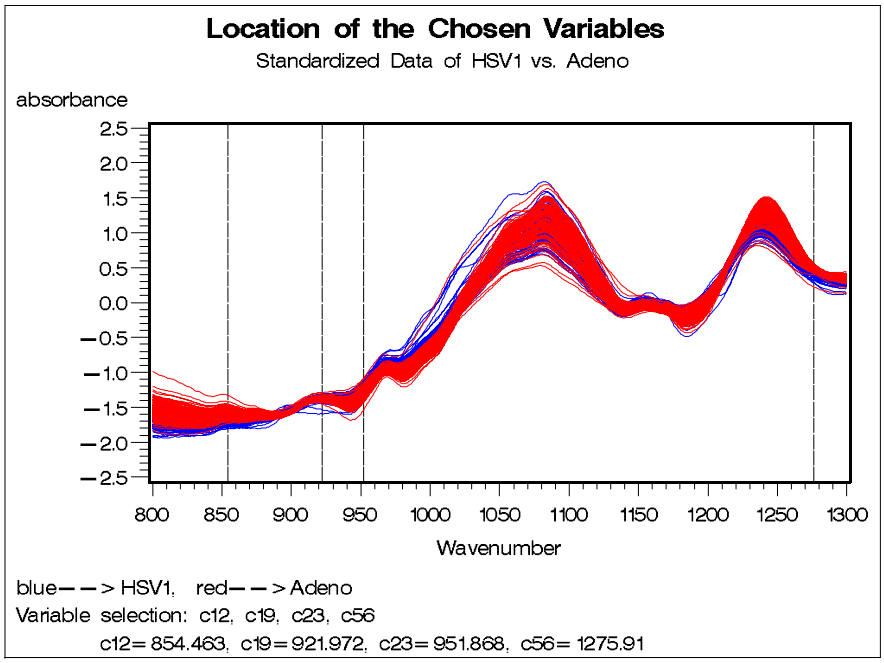
Table-8 AUC and specificities corresponding to sensitivities

95%, 90%and 80% for HSV1 vs. Adeno

HSV1 vs. Adeno			
Logistic regression			
	The old data	After the shrinkage of Cross-validation	The new validate data
Area under the curve (AUC)	0.978	0.969	0.882
Specificity for 95% Sensitivity	0.984	0.955	0.646
Specificity for 90% Sensitivity	0.987	0.977	0.759
Specificity for 80% Sensitivity	0.987	0.981	0.848
PLS regression (Number of Factors=6) Percent Variation Accounted for by Partial Least Squares Factors (Model effects)=94.9			
Area under the curve (AUC)	1	0.992	0.913
Specificity for 95% Sensitivity	1	0.997	0.603
Specificity for 90% Sensitivity	1	0.998	0.734
Specificity for 80% Sensitivity	1	0.999	0.911

As can be seen in Table-8, AUC, specificities for sensitivities 95%, 90%, and 80% of final LR model are all large enough to exhibit excellent discrimination. After estimating the shrinkage from the cross-validation method, the AUC and specificities are very close to the ones calculated by the final model. After the new data validation process, the AUC is 0.882, still having good discrimination, and specificities at sensitivities of 95%, 90% and 80% are 0.646, 0.759, 0.848 respectively. All the specificities decrease almost 20%, which means that the discrimination of

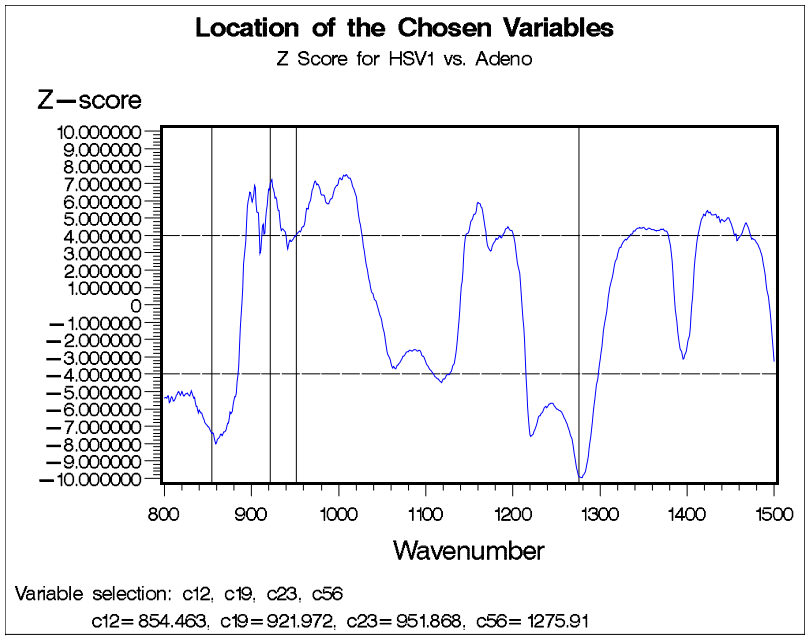
the final model is not as good as other comparisons, but still not bad. Graph-19 shows the z-score plot for both of the old data and new data. As shown in the graph, there is some difference between the two z-score curves, but the chosen variables are all at or near the peaks of the significant wavenumber ranges.



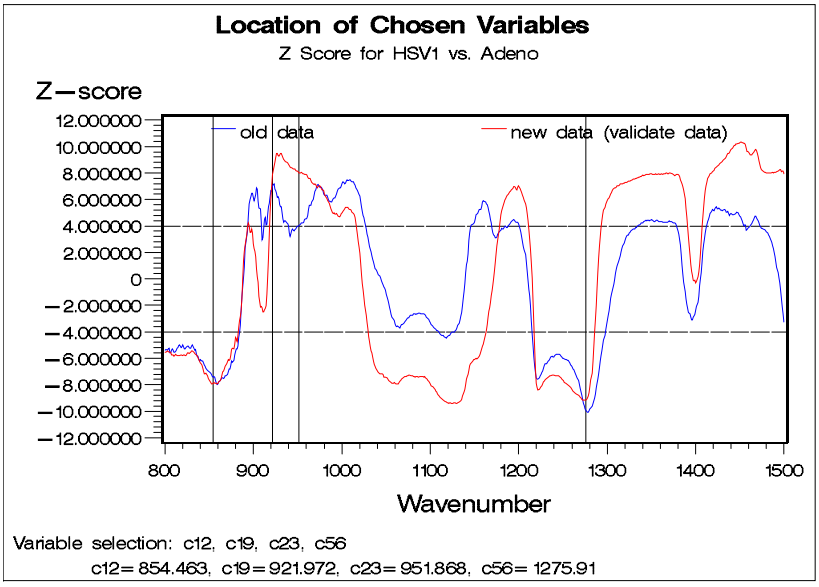
Graph-17 Location of the chosen variables (Standardized data of HSV1 vs. Adeno)

For PLSR model, the first 6 factors count about 94.9% of the total variation. Table-9 shows the coefficients of the variables for the final model. As shown in Table-8, the AUC, and the specificities corresponding to 95%, 90%, and 80% sensitivities are all equal to1, indicating excellent discrimination of the final PLSR model. The AUC and the specificities obtained after the cross-validation equal to 0.992, 0.997, 0.998 and 0.999, with the shrinkages less than 1%, also showing excellent discrimination of the PLSR method. However, the AUC and the specificities obtained in the new data validation process are equal to 0.913, 0.603, 0.734, and

0.911. Except the specificity for the sensitivity 80%, other two specificities all decreased more than 25%.



Graph-18 Location of the chosen variables (Z-score for HSV1 vs. Adeno)



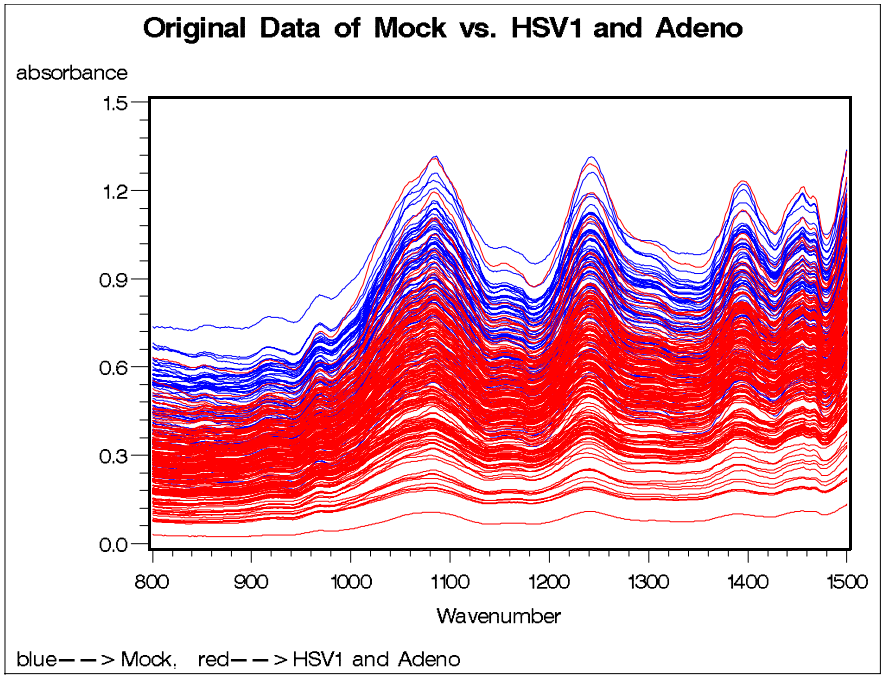
Graph-19 Location of chosen variables (Z-score for HSV1 vs. Adeno for both of the old data and new data)

Table-9 The coefficients of PLSR model for HSV1 vs. Adeno

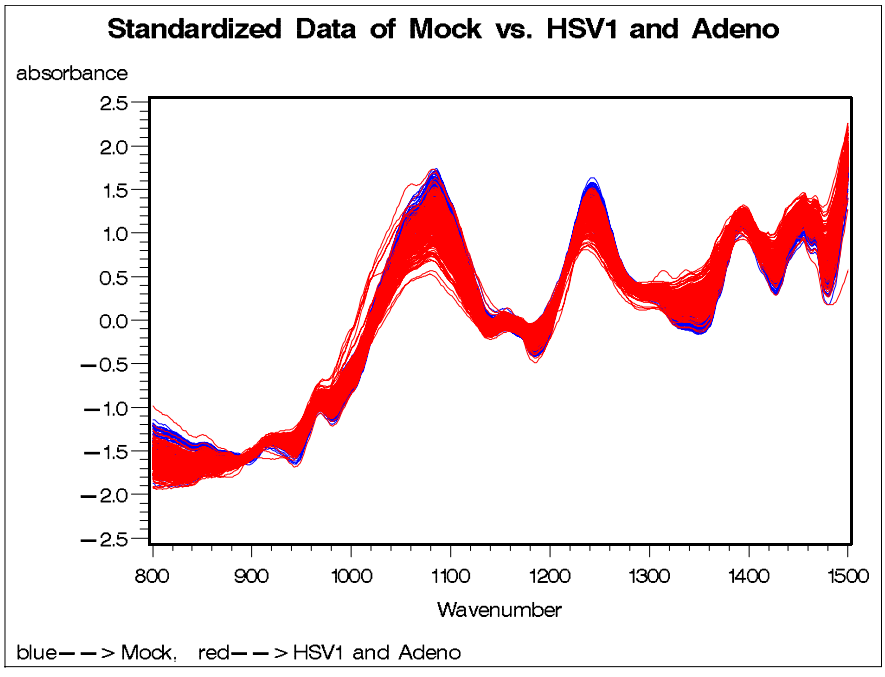
Name	Value	Name	Value	Name	Value
Intercept	0.630386	c27	-1.22813	c54	0.071945
c1	0.54999	c28	-1.01347	c55	0.655478
c2	0.145039	c29	-0.38594	c56	1.141181
c3	-0.11883	c30	0.240592	c57	1.389502
c4	-0.74925	c31	0.412829	c58	1.440889
c5	-0.80763	c32	0.057044	c59	1.358339
c6	-0.91433	c33	-0.39651	c60	1.145496
c7	-0.8985	c34	-0.43939	c61	-0.65315
c8	-0.51796	c35	-0.43988	c62	-0.75794
c9	-0.18848	c36	-0.17699	c63	-0.61421
c10	0.014216	c37	-0.19892	c64	-0.33556
c11	0.354223	c38	0.000236	c65	-0.10041
c12	0.642011	c39	-0.87575	c66	0.192073
c13	0.950055	c40	-1.36407	c67	0.172209
c14	0.690383	c41	-1.58419	c68	-0.06312
c15	0.467151	c42	-1.57394	c69	-0.1944
c16	0.078162	c43	-1.13158	c70	-1.17825
c17	-0.24328	c44	0.805104	c71	-1.24443
c18	-1.23137	c45	0.49141	c72	-1.10589
c19	-1.4359	c46	0.019265	c73	-0.58697
c20	-1.11172	c47	-0.48155	c74	-0.33911
c21	-0.53226	c48	-1.01785	c75	-0.16616
c22	0.055645	c49	-1.33556	c76	-0.40004
c23	0.802684	c50	-1.37611	c77	-0.52508
c24	0.552887	c51	-1.0543	c78	-0.32266
c25	0.086736	c52	-0.77339		
c26	-0.58358	c53	-0.31777		

3.4 Mock versus HSV1 and Adeno

Graph-20 to Graph-22 shows the original data, the standardized data, and the Z-score data respectively. Graph-22 reveals that eight ranges, 800-885cm^{-1}, 921-959 cm^{-1}, 973-1006 cm^{-1}, 1027-1137 cm^{-1}, 1165-1207 cm^{-1}, 1217-1279 cm^{-1}, 1310-1391 cm^{-1}, and 1410-1500 cm^{-1}, are significant in the study. The 570 variables included in these ranges can be stabilized into 114 variables (c1-c114).



Graph-20 Original data of Mock vs. HSV1 and Adeno



Graph-21 Standardized data of Mock vs. HSV1 and Adeno

Variables c39, c66, c98, c103 and c106 were chosen to build the LR model by stepwise selection. The wavenumbers corresponding to these five variables are 1053.13, 1219.01, 1421.53, 1445.64, and 1460.11 cm^{-1} respectively, and their locations are shown in Graph-23 and Graph-24. Table-10 shows that the five variables and the intercept are significant in the model because of their small p-values. The final LR model is

$$p(Y=1|\underset{\sim}{X}) = \frac{e^{g(x)}}{1+e^{g(x)}} \ ,$$

and

$$g(x) = 114.9\text{-}54.8103\times c39\text{-}64.6512\times c66\text{-}156.2\times c98\text{+}434.7\times c103\text{-}342.8\times c106 \ .$$

Table-10 LR for Mock vs. HSV1 and Adeno

The LOGISTIC Procedure
Analysis of Maximum Likelihood Estimates

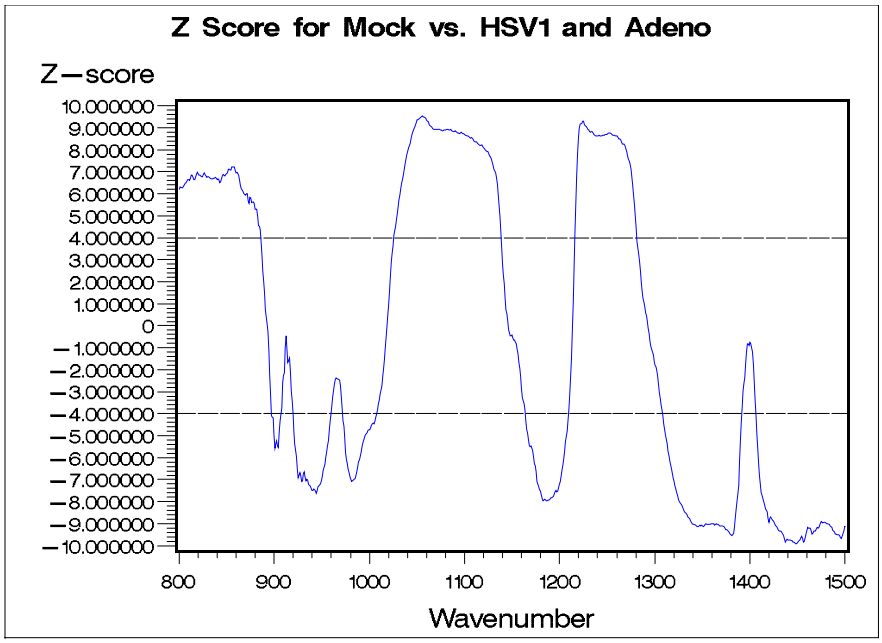
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	114.9	39.2517	8.5689	0.0034
c103	1	434.7	95.4416	20.7484	<.0001
c66	1	-64.6512	17.2193	14.0968	0.0002
c106	1	-342.8	77.0594	19.7885	<.0001
c98	1	-156.2	36.6402	18.1772	<.0001
c39	1	-54.8103	16.4698	11.0752	0.0009

As shown in Table-11, AUC, specificities for sensitivities 95%, 90%, and 80% of final LR model are all large enough to exhibit excellent discrimination. After the cross-validation, the AUC and specificities do not have much difference with the ones calculated by the final model. After the new data validation process, the AUC is 0.689, representing poor discrimination, and specificities for 95%, 90% and 80% are 0.208, 0.375, 0.465 respectively, also indicating the poor discrimination of the final model. Graph-25 shows the z-score plot for both of the old data and new data. As illustrated in the graph, there is not much difference between the two z-score curves,

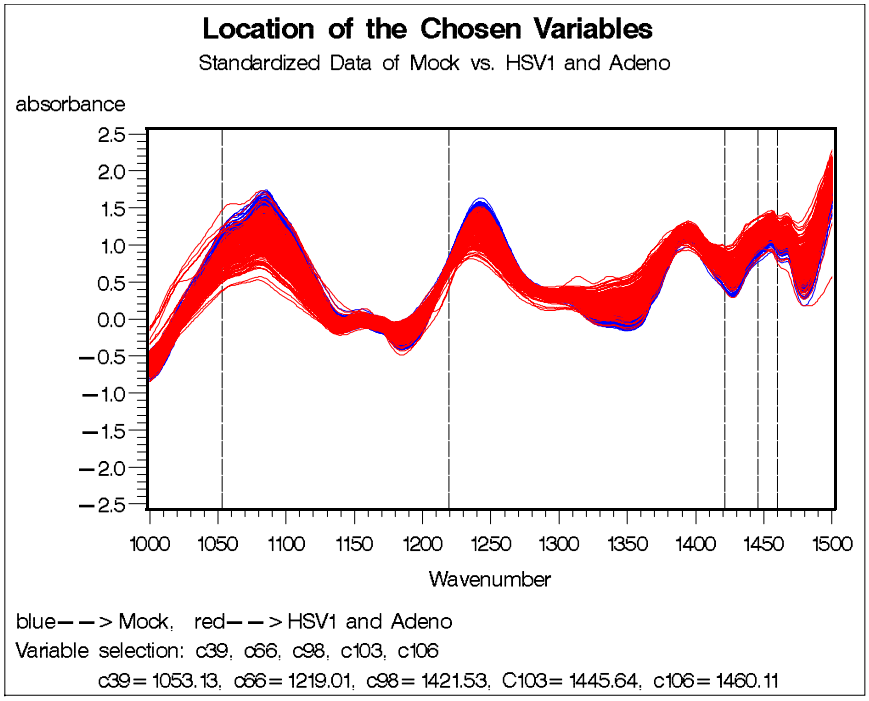
and except c66, the other variables in the final model are all at peaks of the significant wavenumber ranges.

Table-11 AUC and Specificities corresponding to sensitivities 95%, 90% and 80% for Mock vs. HSV1 and Adeno

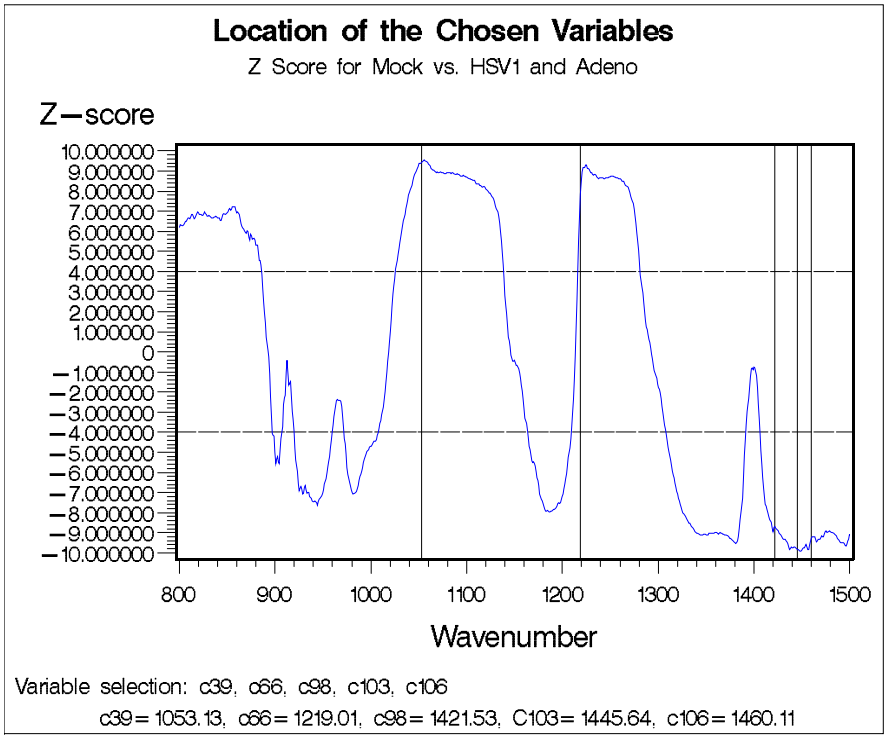
Mock vs. HSV1 and Adeno			
Logistic regression			
	The old data	After the shrinkage of Cross-validation	The new validate data
Area under the curve (AUC)	0.993	0.980	0.689
Specificity for 95% Sensitivity	0.971	0.939	0.208
Specificity for 90% Sensitivity	1	0.975	0.375
Specificity for 80% Sensitivity	1	0.991	0.465
PLS regression (Number of Factors=5) Percent Variation Accounted for by Partial Least Squares Factors (Model effects)=94.1			
Area under the curve (AUC)	0.990	0.973	0.986
Specificity for 95% Sensitivity	0.942	0.877	0.938
Specificity for 90% Sensitivity	0.986	0.956	0.975
Specificity for 80% Sensitivity	1	0.990	0.975



Graph-22 Z-score for Mock vs. HSV1 and Adeno

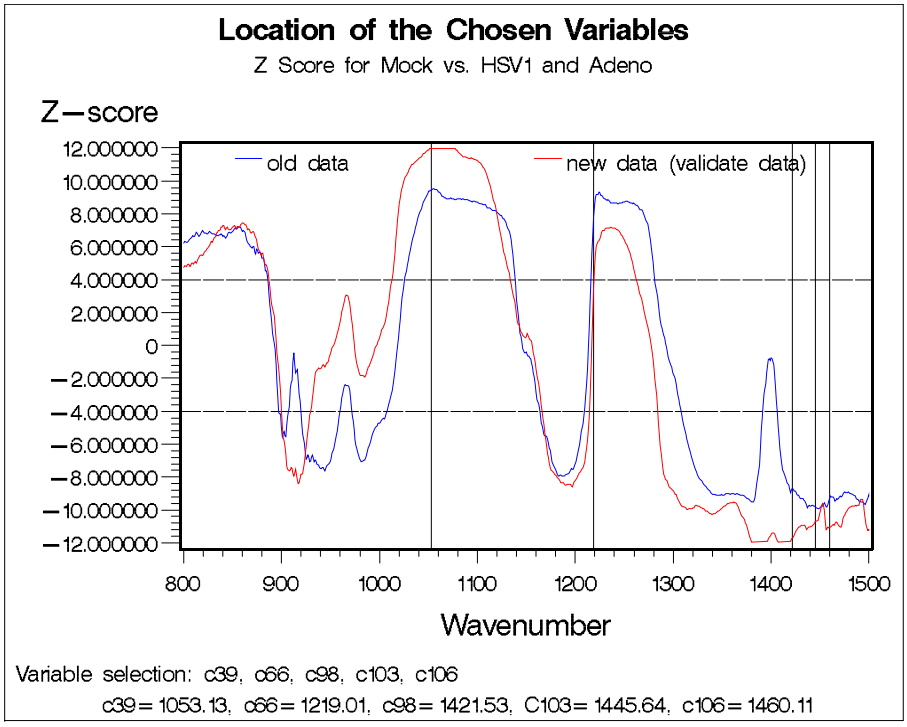


Graph-23 Location of the chosen variables (Standardized data of Mock vs. HSV1 and Adeno)



Graph-24 Location of the chosen variables (Z-score for Mock vs. HSV1 and Adeno)

For PLSR model, the first 5 factors count about 94.1% of the total variation. Table-10 shows the coefficients of the variables for the final model. As shown in Table-11, the AUC, and the specificities corresponding to 95%, 90%, and 80% sensitivities are equal to 0.990, 0.942, 0.986, and 1 respectively, which indicate excellent discrimination of the final PLSR model. The AUC and the specificities obtained after the cross-validation equal to 0.973, 0.877, 0.956 and 0.990, also excellent. In the new data validation process, the AUC and the specificities are equal to 0.986, 0.938, 0.975, and 0.975. Except the specificity for the sensitivity 80%, the other two specificities are all even better than the ones for old data.



Graph-25 Location of the chosen variables (Z-score for Mock vs. HSV1 and Adeno for both of the old data and new data)

Table-12 The coefficients of PLSR model for Mock vs. HSV1 and Adeno

Name	Value	Name	Value	Name	Value	Name	Value
Intercept	-0.13981	c29	0.609728	c58	0.16886	c87	0.32165
c1	0.154775	c30	0.417297	c59	0.169211	c88	0.24649
c2	0.119852	c31	0.247996	c60	0.250045	c89	0.181776
c3	0.002784	c32	0.222505	c61	0.184691	c90	0.214342
c4	-0.00218	c33	0.246129	c62	0.137461	c91	0.258647
c5	-0.07481	c34	-0.21835	c63	-0.02702	c92	0.354402
c6	-0.08741	c35	-0.18454	c64	-0.17601	c93	0.456695
c7	-0.05832	c36	-0.25622	c65	-0.5328	c94	0.498446
c8	-0.02393	c37	-0.39814	c66	-0.76264	c95	0.574397
c9	-0.02944	c38	-0.56275	c67	-0.45028	c96	-0.17141
c10	0.014947	c39	-0.59602	c68	-0.19763	c97	-0.25905
c11	-0.04394	c40	-0.54714	c69	-0.06725	c98	-0.22491
c12	-0.17582	c41	-0.29206	c70	0.023534	c99	-0.19465
c13	-0.2184	c42	-0.01903	c71	-0.02048	c100	0.031599
c14	-0.09096	c43	0.144072	c72	-0.08303	c101	0.34478
c15	-0.00996	c44	0.237376	c73	-0.15483	c102	0.348941
c16	0.005894	c45	0.128903	c74	-0.10733	c103	0.472
c17	-0.03749	c46	0.075283	c75	-0.08483	c104	0.509104
c18	-0.03409	c47	0.097596	c76	-0.08529	c105	0.488474
c19	0.141338	c48	0.128893	c77	-0.12139	c106	-0.01766
c20	0.116559	c49	0.210816	c78	-0.11699	c107	-0.12171
c21	-0.04582	c50	0.354062	c79	-0.50705	c108	-0.29004
c22	-0.0501	c51	0.463249	c80	-0.44816	c109	-0.75886
c23	-0.02279	c52	0.414076	c81	-0.37489	c110	-0.88392
c24	-0.02982	c53	0.364218	c82	-0.19692	c111	-0.70103
c25	-0.14742	c54	0.272629	c83	0.009252	c112	-0.19129
c26	-0.4068	c55	0.193636	c84	0.18898	c113	0.392587
c27	0.193778	c56	0.131911	c85	0.344585	c114	1.081963
c28	0.506901	c57	0.243508	c86	0.382148		

Chapter IV

Discussion

Based on the high values of AUC and specificities at sensitivities of 95%, 90% and 80%, both of LR and PLSR methods are considered to have excellent discrimination for any two different kinds of cells that we studied. However, after comparing LR and PLSR, we find that the overall performance of PLSR is better than that of LR, especially for the new data validation. For instance, in the comparison of Mork and virus-infected (HSV1 and Adeno) cells, the AUC and specificities in new data validation process of the PLSR model are all much larger than that of the LR model. This can be explained by the fact that a PLSR model includes combinations of all variables while a LR model only uses several selected variables. When some of the significant wavenumber ranges of new data shift even a little bit, the variables of the LR model might not be significant anymore. Therefore, the LR model built with the old data might not work well for the new data, but the PLSR still perform well.

The results of this study proved FTIR microspectroscopy to be a useful technique in distinguishing normal from virus-infected cells or in discriminating between two kinds of viruses-infected cells at early stages of infection. Therefore, it seems certainly worthwhile to continue with the development of FTIR microscopy for the purpose of viruses' infection diagnosis.

Further studies will involve developing methods to achieve classification of three or more kinds of virus-infected cells. In addition, we would like to see if we can detect the difference between those cells in stages of infection which is earlier than 24 h p.i.

REFERENCES

W. J. Conover (1998). *Practical nonparametric statistics* (3rd ed).

Herve Abdi (2003). *Partial least squares (PLS) regression*. The University of Texas at Dallas.

Alan. Agresti (2002). Categorical data analysis. New York: Wiley-Interscience.

J.A. Swets (1995). Signal detection theory and ROC analysis in psychology and diagnostics: Collected papers. *Lawrence Erlbaum Associates.*

D. Hosmer, & S. Lemeshow (2000). *Applied logistic regression* (2nd ed). John Wiley & Sons, Inc.

SAS Institute Inc. 2002-2005 SAS OnlineDoc. Version 9.1.3.

M. Kathleen Alam, Jerilyn A. Timlim, Laura E. Martin, Darryl Williams, C. Rick Lyons, Kristin Carrison, & Brian Hjelle (2004). Spectroscopic evaluation of living murine macrophage cells before and after activation using attenuated total reflectance infrared spectroscopy. *Vibrational Spectroscopy* 34, 3-11.

Ahmad Salman, Vitaly Erukhimovitch, Marina Talyshinsky, Mahmoud Huleihil, & Mahmoud Huleihel (2002). FTIR spectroscopic method for detection of cells infected with herpes viruses, *Biopolymers (Biospectroscopy)*. Vol. 67, 406-412.

Vitaly Erukhimovitch, Igor Mukmanov, Marina Talyshinsky, Yelena Souprun, & Mahmoud Huleihel (2004). The use of FTIR microscopy for evaluation of herpes viruses infection development kinetics. *Spectrochimica Acta* Part A 60 2355-2361.

H.H Mantschm, & D. Chapman (1996). Assessment of tissue viability using near-infrared spectroscopy. *Infrared Spectroscopy of Biomolecules*, Wiley, New York.

E. Burattini, M. Cavagna, R. Dell'Anna, F. Malvezzi Campeggi, F. Monti, F. Rossi, & S. Torriani (2008). A FTIR microspectroscopy study of autolysis in cells of the wine yeast Saccharomyces cerevisiae. *Vibrational Spectroscopy* 47, 139-147.

Menashi A. Cohenford, Thomas A. Godwin, Frederick Cahn, Prashant Bhandare, Thomas A. Caputo, & Basil Rigas (1997). Infrared Spectroscopy of Normal and Abnormal Cervical Smears: Evaluation by Principal Component Analysis. *Gynecologic Oncology*, 66, 59-65.

P. T. Wong, R. K. Wong, T. A. Caputo, T. A. Godwin, & B. Rigas (1991). Infrared spectroscopy of exfoliated human cervical cells: evidence of extensive structural changes during carcinogenesis. *Proc. Natl. Acad. Sci. USA, 88,* 10988-10992.

Michael Jackson, Keith Kim, John Tetteh, James R. Mansfield, Brion Dolenko, Raymond L. Somorjai, F. W. Orr, Peter H. Watson, & Henry H. Mantsch (1998). Cancer diagnosis by infrared spectroscopy: methodological aspects. *Proc. SPIE*, 3257, 24-34.

H. Mantsch, D. Chapman (1996). Infrared spectroscopy of biomolecules. *Wiley: New York.*

H M Yazdi, M A Bertrand, P T Wong (1996). Detecting structural changes at the molecular level with Fourier transform infrared spectroscopy. A potential tool for prescreening preinvasive lesions of the cervix. *Acta Cytol,* 40, 664-668.

E. Benedetti, E.Bramanti, F. Papineschi, & I. Rossi (1997). Determination of the relative amount of nucleic acids and proteins in leukemic and normal lymphocytes by means of Fourier transform infrared microspectroscopy. *Applied Spectroscopy*. 51, 792-797.

L Chiriboga, P Xie, H Yee, D Zarou, D Zakim, & M Diem (1998). Infrared spectroscopy of human tissue. IV. Detection of dysplastic and neoplastic changes of human cervical tissue via infrared microscopy. *Cell Mol Biol*. 44, 219-29.

D. Yang, D. Castro, I. El-Sayed, M. El-Sayed, R. Saxton, & Y. Nancy(1995). Fourier-transform infrared spectroscopic comparison of human fibroblast and fibrosarcoma cells. *Proc. SPIE*. 2389, 543-550.

APPENDIX A: SAS Code for Creating and Standardizing Datasets

```
/*********************************************************** ******/
/*                      for the mock (old data)                  */
/*****************************************************************/
%macro inputdata;
    %do i=1 %to 69;
        proc import datafile="D:\tians new research_0503\data in CD_new\032408
data\mock\mock-24hpi-032408-2cm-1-1500-800(&i).csv" out=new_cd.mock&i
replace  ;
        run;
    %end;
%mend;

%inputdata

data whole;
       if 1=1 then delete;
       run;

%macro merge;
%do i=1 %to 69;
data whole;
    merge whole new_cd.mock&i(firstobs=5 rename=(wavenumber=mo&i));
    mock&i=mo&i+0;
    drop mo&i;
run;
%end;
%mend;

%merge

data new_cd.orginal_mock;
    set whole(drop=xlabel);
run;

/*  standardize          */

proc means data=new_cd.orginal_mock;
    var mock1-mock69;
    output out=mockmean mean(mock1-mock69)=mock1-mock69;
    output out=mockstd std(mock1-mock69)=mock1-mock69;
run;

data mockmean; /*mean*/
    set mockmean;
    drop _freq_ _type_;
run;
data mockstd; /*std*/
    set mockstd;
    drop _freq_ _type_;
run;
```

```
data std;
    set new_cd.orginal_mock mockmean mockstd;
run;

proc transpose data=std out=stdtr name=cell prefix=v;
    var mock1-mock69;
run;

data stdtr;
    set stdtr;
    rename v729=mean v730=std;
run;

%macro std;
%do i=1 %to 728;
data stdtr;
    set stdtr;
    v&i=(v&i-mean)/std;
run;
%end;
%mend;

%std

data whole_stdd;
    set odata.x stdtr;
    drop mean std;
run;

proc transpose data=whole_stdd out=whole_stdd1;/* for plot graph */
    var v1-v728;
    id cell;
run;

data new_cd.mock_analysis;  /* data analysis */
    set whole_stdd(firstobs=2);
run;

data new_cd.mock_graph; /*for plot graph */
    set whole_stdd1;
run;

/*************************************************************/
/*                     for the Hsv1 (old data)                */
/*************************************************************/
%macro inputdata;
    %do i=1 %to 79;
        proc import datafile="D:\tians new research_0503\data in CD_new\032408
data\hsv1\hsv1-24hpi-032508-2cm-1 1500-800(&i).csv" out=new_cd.hsv&i
replace  ;
        run;
    %end;
%mend;

%inputdata
```

```
data whole;
       if 1=1 then delete;
       run;

%macro merge;
%do i=1 %to 79;
data whole;
    merge whole new_cd.hsv&i(firstobs=5 rename=(wavenumber=hs&i));
    hsv&i=hs&i+0;
    drop hs&i;
run;
%end;
%mend;

%merge

data new_cd.orginal_hsv1;
    set whole(drop=xlabel);
run;

/*  standardize          */

proc means data=new_cd.orginal_hsv1;
    var hsv1-hsv79;
    output out=hsvmean mean(hsv1-hsv79)=hsv1-hsv79;
    output out=hsvstd std(hsv1-hsv79)=hsv1-hsv79;
run;

data hsvmean; /*mean*/
    set hsvmean;
    drop _freq_ _type_;
run;
data hsvstd; /*std*/
    set hsvstd;
    drop _freq_ _type_;
run;

data std;
    set new_cd.orginal_hsv1 hsvmean hsvstd;
run;

proc transpose data=std out=stdtr name=cell prefix=v;
    var hsv1-hsv79;
run;

data stdtr;
    set stdtr;
    rename v729=mean v730=std;
run;

%macro std;
%do i=1 %to 728;
data stdtr;
    set stdtr;
    v&i=(v&i-mean)/std;
run;
```

```
%end;
%mend;

%std

data whole_stdd;
    set odata.x stdtr;
    drop mean std;
run;

proc transpose data=whole_stdd out=whole_stdd1;/* for plot graph */
    var v1-v728;
    id cell;
run;

data new_cd.hsv1_analysis;  /* data analysis */
    set whole_stdd(firstobs=2);
run;

data new_cd.hsv1_graph; /*for plot graph */
    set whole_stdd1;
run;

/*****************************************************************/
/*                      for the adeno (old data)                 */
/*****************************************************************/
%macro inputdata;
    %do i=1 %to 94;
        proc import datafile="D:\tians new research_0503\data in CD_new\032408
data\adeno\had1-24hpi-032508-2cm-1 1500-800(&i).csv" out=new_cd.adeno&i
replace  ;
        run;
    %end;
%mend;

%inputdata

data whole;
       if 1=1 then delete;
       run;

%macro merge;
%do i=1 %to 94;
data whole;
    merge whole new_cd.adeno&i(firstobs=5 rename=(wavenumber=ad&i));
    adeno&i=ad&i+0;
    drop ad&i;
run;
%end;
%mend;

%merge

data new_cd.orginal_adeno;
    set whole(drop=xlabel);
run;
```

```
/*  standardize         */

proc means data=new_cd.orginal_adeno;
    var adeno1-adeno94;
    output out=adenomean mean(adeno1-adeno94)=adeno1-adeno94;
    output out=adenostd std(adeno1-adeno94)=adeno1-adeno94;
run;

data adenomean; /*mean*/
    set adenomean;
    drop _freq_ _type_;
run;
data adenostd; /*std*/
    set adenostd;
    drop _freq_ _type_;
run;

data std;
    set new_cd.orginal_adeno adenomean adenostd;
run;

proc transpose data=std out=stdtr name=cell prefix=v;
    var adeno1-adeno94;
run;

data stdtr;
    set stdtr;
    rename v729=mean v730=std;
run;

%macro std;
%do i=1 %to 728;
data stdtr;
    set stdtr;
    v&i=(v&i-mean)/std;
run;
%end;
%mend;

%std

data whole_stdd;
    set odata.x stdtr;
    drop mean std;
run;

proc transpose data=whole_stdd out=whole_stdd1;/* for plot graph */
    var v1-v728;
    id cell;
run;

data new_cd.adeno_analysis;  /* data analysis */
    set whole_stdd(firstobs=2);
run;

data new_cd.adeno_graph; /*for plot graph */
```

```
    set whole_stdd1;
run;

/*****************************************************************/
/*                    for the Mock data (validation)             */
/*****************************************************************/
%macro inputdata;
    %do i=1 %to 54;
        proc import datafile="D:\tians new research_0503\data in CD_new\041608
data\mock\mock-24hpi-041608-2cm-1-1500-800(&i).csv" out=new_cd.vali_mock&i
replace  ;
        run;
    %end;

    %do i=1 %to 26;
        proc import datafile="D:\tians new research_0503\data in CD_new\041608
data\mock\mock-24hpi-041708-2cm-1-1500-800(&i).csv" out=new_cd.vali_mock_&i
replace  ;
        run;
    %end;
%mend;

%inputdata

data whole;
       if 1=1 then delete;
       run;

%macro merge;
%do i=1 %to 54;
data whole;
    merge whole new_cd.vali_mock&i(firstobs=5 rename=(wavenumber=mo&i));
    mock&i=mo&i+0;
    drop mo&i;
run;
%end;
%do i=1 %to 26;
data whole;
    merge whole new_cd.vali_mock_&i(firstobs=5 rename=(wavenumber=mo&i));
    mock_&i=mo&i+0;
    drop mo&i;
run;
%end;
%mend;

%merge

data new_cd.orginal_vali_mock;
    set whole(drop=xlabel);
run;

/*  standardize         */

proc means data=new_cd.orginal_vali_mock;
    var mock1-mock54 mock_1-mock_26;
    output out=mockmean mean(mock1-mock54 mock_1-mock_26)=mock1-mock54
mock_1-mock_26;
```

```
    output out=mockstd std(mock1-mock54 mock_1-mock_26)=mock1-mock54
mock_1-mock_26;
run;

data mockmean; /*mean*/
    set mockmean;
    drop _freq_ _type_;
run;
data mockstd; /*std*/
    set mockstd;
    drop _freq_ _type_;
run;

data std;
    set new_cd.orginal_vali_mock mockmean mockstd;
run;

proc transpose data=std out=stdtr name=cell prefix=v;
    var mock1-mock54 mock_1-mock_26;
run;

data stdtr;
    set stdtr;
    rename v729=mean v730=std;
run;

%macro std;
%do i=1 %to 728;
data stdtr;
    set stdtr;
    v&i=(v&i-mean)/std;
run;
%end;
%mend;

%std

data whole_stdd;
    set odata.x stdtr;
    drop mean std;
run;

proc transpose data=whole_stdd out=whole_stdd1;/* for plot graph */
    var v1-v728;
    id cell;
run;

data new_cd.vali_mock_analysis;  /* data analysis */
    set whole_stdd(firstobs=2);
run;

data new_cd.vali_mock_graph; /*for plot graph */
    set whole_stdd1;
run;

/*************************************************************/
```

```
/*                      for the Hsv1 data (validation)             */
/*****************************************************************/
%macro inputdata;
    %do i=1 %to 79;
        proc import datafile="D:\tians new research_0503\data in CD_new\041608
data\hsv1\hsv1-24hpi-041808-2cm-1-1500-800(&i).csv" out=new_cd.vali_hsv&i
replace  ;
        run;
    %end;
%mend;

%inputdata

data whole;
       if 1=1 then delete;
       run;

%macro merge;
%do i=1 %to 79;
data whole;
    merge whole new_cd.vali_hsv&i(firstobs=5 rename=(wavenumber=hs&i));
    hsv&i=hs&i+0;
    drop hs&i;
run;
%end;
%mend;

%merge

data new_cd.orginal_vali_hsv1;
    set whole(drop=xlabel);
run;

/*  standardize         */

proc means data=new_cd.orginal_vali_hsv1;
    var hsv1-hsv79;
    output out=hsvmean mean(hsv1-hsv79)=hsv1-hsv79;
    output out=hsvstd std(hsv1-hsv79)=hsv1-hsv79;
run;

data hsvmean; /*mean*/
    set hsvmean;
    drop _freq_ _type_;
run;
data hsvstd; /*std*/
    set hsvstd;
    drop _freq_ _type_;
run;

data std;
    set new_cd.orginal_vali_hsv1 hsvmean hsvstd;
run;

proc transpose data=std out=stdtr name=cell prefix=v;
    var hsv1-hsv79;
run;
```

```
data stdtr;
    set stdtr;
    rename v729=mean v730=std;
run;

%macro std;
%do i=1 %to 728;
data stdtr;
    set stdtr;
    v&i=(v&i-mean)/std;
run;
%end;
%mend;

%std

data whole_stdd;
    set odata.x stdtr;
    drop mean std;
run;

proc transpose data=whole_stdd out=whole_stdd1;/* for plot graph */
    var v1-v728;
    id cell;
run;

data new_cd.vali_hsv1_analysis;  /* data analysis */
    set whole_stdd(firstobs=2);
run;

data new_cd.vali_hsv1_graph; /*for plot graph */
    set whole_stdd1;
run;

/*****************************************************************/
/*                      for the Adeno data (validation)          */
/*****************************************************************/
%macro inputdata;
    %do i=1 %to 50;
        proc import datafile="D:\tians new research_0503\data in CD_new\041608
data\adeno\had1-24hpi-041708-2cm-1-1500-800(&i).csv" out=new_cd.vali_adeno&i
replace  ;
        run;
    %end;

    %do i=1 %to 34;
        proc import datafile="D:\tians new research_0503\data in CD_new\041608
data\adeno\had1-24hpi-041808-2cm-1-1500-800(&i).csv" out=new_cd.vali_adeno_&i
replace  ;
        run;
    %end;
%mend;

%inputdata
```

```
data whole;
       if 1=1 then delete;
       run;

%macro merge;
%do i=1 %to 50;
data whole;
    merge whole new_cd.vali_adeno&i(firstobs=5 rename=(wavenumber=ad&i));
    adeno&i=ad&i+0;
    drop ad&i;
run;
%end;
%do i=1 %to 34;
data whole;
    merge whole new_cd.vali_adeno_&i(firstobs=5 rename=(wavenumber=ad&i));
    adeno_&i=ad&i+0;
    drop ad&i;
run;
%end;
%mend;

%merge

data new_cd.orginal_vali_adeno;
    set whole(drop=xlabel);
run;

/*  standardize         */

proc means data=new_cd.orginal_vali_adeno;
    var adeno1-adeno50 adeno_1-adeno_34;
    output out=adenomean mean(adeno1-adeno50 adeno_1-adeno_34)=adeno1-adeno50
adeno_1-adeno_34;
    output out=adenostd std(adeno1-adeno50 adeno_1-adeno_34)=adeno1-adeno50
adeno_1-adeno_34;
run;

data adenomean; /*mean*/
    set adenomean;
    drop _freq_ _type_;
run;
data adenostd; /*std*/
    set adenostd;
    drop _freq_ _type_;
run;

data std;
    set new_cd.orginal_vali_adeno adenomean adenostd;
run;

proc transpose data=std out=stdtr name=cell prefix=v;
    var adeno1-adeno50 adeno_1-adeno_34;
run;

data stdtr;
    set stdtr;
    rename v729=mean v730=std;
```

```
run;

%macro std;
%do i=1 %to 728;
data stdtr;
    set stdtr;
    v&i=(v&i-mean)/std;
run;
%end;
%mend;

%std

data whole_stdd;
    set odata.x stdtr;
    drop mean std;
run;

proc transpose data=whole_stdd out=whole_stdd1;/* for plot graph */
    var v1-v728;
    id cell;
run;

data new_cd.vali_adeno_analysis;  /* data analysis */
    set whole_stdd(firstobs=2);
run;

data new_cd.vali_adeno_graph; /*for plot graph */
    set whole_stdd1;
run;
```

APPENDIX B: SAS Code for Mock versus HSV1

```
/*****************************************************/
/*    mock(inf=0) vs hsv1(inf=1)   (old data)         */
/*****************************************************/

data new_cd.mock_hsv1_graph; /* for graph */
   merge new_cd.mock_graph new_cd.hsv1_graph ;
run;

data mock;
   set new_cd.mock_analysis;
   inf=0;
run;

data hsv1;
   set new_cd.hsv1_analysis;
   inf=1;
run;

data new_cd.mock_hsv1_analysis;
   set mock hsv1;
run;

/*************plot the standardized graph (overall graph)************/

goptions reset=global gunit=pct border
       ctext=black ftitle=swissb ftext=swiss htitle=4 htext=3;

symbol1 color=blue i=j line=1 w=1 h=2.5 repeat=69;
symbol2 color=red i=j line=1 w=1 h=3 repeat=79;

legend1 across=1 down=2 label=none
       mode=protect position=(top inside left)
       value=('Mock' 'HSV1') offset=(1cm, -1cm);
footnote1 h=3 j=l ' blue-->Mock,  red-->HSV1';

axis1 label=(h=3 c=black"Wavenumber" )order=(800 to 1500 by 100)
     major=(height=2) minor=(height=1)
     width=3;

axis2 label=(h=3 c=black"absorbance" )order=(-2.5 to 2.5 by 0.5)
     major=(height=2) minor=(height=1)
     width=3;

title 'Standardized Data of Mock vs. HSV1';
 proc gplot data=new_cd.mock_hsv1_graph; /* blue=Mock red=Hsv1 */
   plot (mock1-mock69 hsv1-hsv79)*x / overlay legend=legend1
                        haxis=axis1 hminor=4
                        vaxis=axis2 vminor=4;
run;
quit;
```

```
/*****************************************************/
/*       wilcoxon rank test (Z-score)                */
/*****************************************************/

data whole;
    set new_cd.mock_hsv1_analysis;
run;


%macro wilcoxon;
data w;
       if 1=1 then delete;
        run;

%do i=1 %to 728;

ods trace on;
ods listing close;
ods trace off;
ods output  Npar1way.WilcoxonTest=t1;
proc npar1way wilcoxon data=whole;
     class inf;
     var v&i;
     *exact;
   run;

data t2;
    set t1(firstobs=6 obs=6);
    keep nvalue1;
run;

data t3;
    set t1;
    if label1='Z';
    keep  nvalue1;
run;

data t4;
    merge t2(rename=(nvalue1=p_value)) t3(rename=(nvalue1=z_score));
run;

data w;
    set w t4;
run;

%end;

%mend;

%wilcoxon


data w1;
    merge odata.xt w;
run;
```

```
data new_cd.z_mock_hsv1;
    set w1;
run;


/*******z-score plot ****************/

goptions reset=global gunit=pct border
       ctext=black ftitle=swissb ftext=swiss htitle=4 htext=3;

symbol1 color=blue i=j line=1 w=1 h=2.5 repeat=1;

axis1 label=(h=4 c=black"Wavenumber" )order=(800 to 1500 by 100)
     major=(height=2) minor=(height=1)
     width=3;

axis2 label=(h=4 c=black"Z-score" )order=(-10 to 10 by 1)
     major=(height=2) minor=(height=1)
     width=3;

title 'Z Score for Mock vs. HSV1';
 proc gplot data=new_cd.z_mock_hsv1;
   plot z_score*x / overlay
                        haxis=axis1 hminor=4
                        vaxis=axis2 vminor=4
                         vref=4 -4 lvref=5;
run;
quit;

data z;
    set new_cd.z_mock_hsv1;
run;

/*************************************/
/*       Sumby5                      */
/*************************************/

data sumby5;
    set odata.x new_cd.mock_hsv1_analysis;
    keep cell inf v1-v90 v124-v223 v246-v350 v375-v424 v433-v507 v532-v611
v634-v728;
run;


%macro sumby5;
data sumby5;
    set sumby5;
    %do i=1 %to 18;
        c&i=0;
        %do j=0 %to 4;
            %let m=%sysevalf(1+5*(&i-1)+&j, integer);
            c&i=c&i+v&m;
       %end;
        c&i=c&i/5;
    %end;

    %do i=19 %to 38;
```

```
        c&i=0;
        %do j=0 %to 4;
            %let m=%sysevalf(124+5*(&i-18-1)+&j, integer);
            c&i=c&i+v&m;
        %end;
        c&i=c&i/5;
    %end;

    %do i=39 %to 59;
        c&i=0;
        %do j=0 %to 4;
            %let m=%sysevalf(246+5*(&i-38-1)+&j, integer);
            c&i=c&i+v&m;
        %end;
        c&i=c&i/5;
    %end;

    %do i=60 %to 69;
        c&i=0;
        %do j=0 %to 4;
            %let m=%sysevalf(375+5*(&i-59-1)+&j, integer);
            c&i=c&i+v&m;
        %end;
        c&i=c&i/5;
    %end;

    %do i=70 %to 84;
        c&i=0;
        %do j=0 %to 4;
            %let m=%sysevalf(433+5*(&i-69-1)+&j, integer);
            c&i=c&i+v&m;
        %end;
        c&i=c&i/5;
    %end;

    %do i=85 %to 100;
        c&i=0;
        %do j=0 %to 4;
            %let m=%sysevalf(532+5*(&i-84-1)+&j, integer);
            c&i=c&i+v&m;
        %end;
        c&i=c&i/5;
    %end;

    %do i=101 %to 119;
        c&i=0;
        %do j=0 %to 4;
            %let m=%sysevalf(634+5*(&i-100-1)+&j, integer);
            c&i=c&i+v&m;
        %end;
        c&i=c&i/5;
    %end;

keep cell inf c1-c119;
run;
%mend;
```

```
%sumby5

data new_cd.sumby5_mock_hsv1;
    set sumby5;
run;

/*********************************************************/
/*    for validation: mock(inf=0) vs hsv1(inf=1)         */
/*********************************************************/

data new_cd.vali_mock_hsv1_graph; /* for graph */
    merge new_cd.vali_mock_graph new_cd.vali_hsv1_graph ;
run;

data mock;
    set new_cd.vali_mock_analysis;
    inf=0;
run;

data hsv1;
    set new_cd.vali_hsv1_analysis;
    inf=1;
run;

data new_cd.vali_mock_hsv1_analysis;
    set mock hsv1;
run;

/*****************************************************/
/*       wilcoxon rank test (Z-score)                */
/*****************************************************/

data whole;
    set new_cd.vali_mock_hsv1_analysis;
run;

%wilcoxon

data w1;
    merge odata.xt w;
run;

data new_cd.z_vali_mock_hsv1;
    set w1;
run;

/*******************************/
/*       Sumby5                */
/*******************************/

data sumby5;
    set odata.x new_cd.vali_mock_hsv1_analysis;
    keep cell inf v1-v90 v124-v223 v246-v350 v375-v424 v433-v507 v532-v611
v634-v728;
run;

%sumby5
```

```
data new_cd.sumby5_vali_mock_hsv1;
   set sumby5;
run;

/*************************************************************/
/*    balanced 3-fold crossvalidation for logistic regression */
/*************************************************************/

data sumby5;
   set new_cd.sumby5_mock_hsv1(firstobs=2);
run;

ods listing;
proc logistic data=sumby5 DESCENDING ;
     model inf=c1-c119/selection=stepwise
                  sle=0.05
                  sls=0.05;
run;

/*selection the combination from c81 c109 c116 c67*/
/* choose c109 c116 c67*/

proc logistic data=sumby5 DESCENDING ;
     model inf=c109 c116 c67/outroc=table1;
run;

%spec(table1,10)

proc print data=spec10;
run;

/*************plot the standardized graph (partial graph)************/
data s;
   set new_cd.sumby5_mock_hsv1(obs=2);
run;

proc print data=s;
   var c109 c116 c67;
run;

goptions reset=global gunit=pct border
       ctext=black ftitle=swissb ftext=swiss htitle=4 htext=3;

symbol1 color=blue i=j line=1 w=1 h=2.5 repeat=69;
symbol2 color=red i=j line=1 w=1 h=3 repeat=79;

axis1 label=(h=3 c=black"Wavenumber" )order=(1150 to 1500 by 50)
     major=(height=2) minor=(height=1)
     width=3;

axis2 label=(h=3 c=black"absorbance" )order=(-2.5 to 2.5 by 0.5)
     major=(height=2) minor=(height=1)
     width=3;
footnote1 h=3 j=l ' blue-->Mock,  red-->HSV1';
footnote2 h=3 j=l ' Variable selection: c67, c109, c116 ';
footnote3 h=3 j=l '        c67=1195.86, c109=1450.47, c116=1484.22 ';
```

```
title1 'Location of the Chosen Variables ';
title2 'Standardized Data of Mock vs. HSV1';
 proc gplot data=new_cd.mock_hsv1_graph; /* blue=Mock red=Hsv1 */
   plot (mock1-mock69 hsv1-hsv79)*x / overlay legend=legend1
                        haxis=axis1 hminor=4
                        vaxis=axis2 vminor=4
                         href=1450.47    1484.22    1195.86 lhref=5;
run;
quit;

/*******z-score plot *****************/

goptions reset=global gunit=pct border
       ctext=black ftitle=swissb ftext=swiss htitle=4 htext=3;

symbol1 color=blue i=j line=1 w=1 h=2.5 repeat=1;
footnote2 h=3 j=l ' Variable selection: c67, c109, c116 ';
footnote3 h=3 j=l '        c67=1195.86, c109=1450.47, c116=1484.22 ';

axis1 label=(h=4 c=black"Wavenumber" )order=(800 to 1500 by 100)
     major=(height=2) minor=(height=1)
     width=3;

axis2 label=(h=4 c=black"Z-score" )order=(-10 to 10 by 1)
     major=(height=2) minor=(height=1)
     width=3;

title1 'Location of the Chosen Variables';
title2 'Z Score for Mock vs. HSV1';
 proc gplot data=new_cd.z_mock_hsv1;
   plot z_score*x / overlay
                        haxis=axis1 hminor=4
                        vaxis=axis2 vminor=4
                         vref=4 -4 lvref=5 href=1450.47    1484.22    1195.86;
run;
quit;

/*******blanced Cross-validation (3 folds 95% 85% 80% sencitivity) ******/

%macro spec(table,n);

data &table;
    set &table(keep=_sensit_ _1mspec_);
    spec=1-_1mspec_;
    drop _1mspec_;
run;

proc sort data=&table;
    by descending _sensit_ spec;
    run;

data aa;
    _sensit_=0;
    spec=1;
run;
```

```
data &table;
    set &table aa;
run;

proc iml;

    use &table;
    read all var _num_ into x;
    close &table;

    a=nrow(x);

    s=j(3,1,0);

    do i=1 to a-1;

        if x[i,1]=0.95 then do;
            m=i;
            do f=i+1 to a;
                if x[i,1]=x[f,1] then m=m+1;
            end;
            s[1]=x[m,2];
            end;
         else if x[i,1]>0.95 & x[i+1,1]<0.95 then do;
            n=i+1;
            do j=i+2 to a;
                if x[i+1,1]=x[j,1] then n=n+1;
            end;
                s[1]=(0.95-x[n,1])/(x[i,1]-x[n,1])*(x[i,2]-x[n,2])+x[n,2];
            end;
    end;

    do i=1 to a-1;

        if x[i,1]=0.90 then do;
            m=i;
            do f=i+1 to a;
                if x[i,1]=x[f,1] then m=m+1;
            end;
            s[2]=x[m,2];
            end;
         else if x[i,1]>0.90 & x[i+1,1]<0.90 then do;
            n=i+1;
            do j=i+2 to a;
                if x[i+1,1]=x[j,1] then n=n+1;
            end;
                s[2]=(0.90-x[n,1])/(x[i,1]-x[n,1])*(x[i,2]-x[n,2])+x[n,2];
            end;
    end;


    do i=1 to a-1;

        if x[i,1]=0.80 then do;
            m=i;
            do f=i+1 to a;
                if x[i,1]=x[f,1] then m=m+1;
```

```
            end;
            s[3]=x[m,2];
            end;
         else if x[i,1]>0.80 & x[i+1,1]<0.80 then do;
            n=i+1;
            do j=i+2 to a;
               if x[i+1,1]=x[j,1] then n=n+1;
            end;
               s[3]=(0.80-x[n,1])/(x[i,1]-x[n,1])*(x[i,2]-x[n,2])+x[n,2];
            end;
    end;


    st=t(s);

    cname={'s1' 's2' 's3'};

    create spec&n from st[colname=cname];
    append from st;
    close spec&n;

quit;
%mend;

%macro crossvalidation(datain1=, datain2=, datain3=, factor=, n=);

data subdata;
    set &datain3(drop=inf);
run;

data subdata2;
    set &datain3;
run;
data training;
            set &datain1 &datain2;
run;
data whole;
    set training(in=in1) subdata(in=in2);
    m1=in1;
    m2=in2;
run;

ods listing close;
proc logistic data=whole DESCENDING ;
     model inf=c109 c116 c67;
     output out=one PREDICTED=p;
run;

data logi1 logi2(drop=inf);
    set one(keep=inf p m1 m2 m);
    if m1=1 then output logi1;
    if m2=1 then output logi2;
run;

proc sort data=subdata2;
    by m;
run;
```

```
proc sort data=logi2;
   by m;
run;

data logi22;
   merge subdata2(keep=inf m) logi2(keep=m p);
   by m;
run;


ods listing close;
ods trace on;
ods trace off;
ods output Association=auc1;
PROC LOGISTIC data=logi1 descending;    /*training dataset */
   model inf=p/outroc=ctable1;
run;
ods listing;

ods listing close;
ods trace on;
ods trace off;
ods output Association=auc2;
PROC LOGISTIC data=logi22 descending;    /* subdata set   */
   model inf=p/outroc=ctable2;
run;
ods listing;

%spec(ctable1,1)
%spec(ctable2,2)

data bbb&n;
   merge spec1(rename=(s1=ts1 s2=ts2 s3=ts3)) spec2;
   spec1=ts1-s1;
   spec2=ts2-s2;
   spec3=ts3-s3;
run;

data aaa&n;
   merge auc1(keep=label2 nvalue2 rename=(nvalue2=c1)) auc2(keep=label2 nvalue2
rename=(nvalue2=c2));
   if label2^='c' then delete;
   drop label2;
   shi=c1-c2;
run;

data aaa&n;
   merge aaa&n bbb&n;
run;

%mend;

%macro compute(fac, nseed);

data p1 p2;
   set sumby5;
```

```
    if inf=1 then output p1;/* hsv1 79 */
    if inf=0 then output p2;/* mock 69 */
run;


data p1;
    set p1;
    retain n 0;
    n=n+1;
    index=ranuni(&nseed);
run;

data p2;
    set p2;
    retain n 0;
    n=n+1;
    index=ranuni(&nseed);
run;


proc sort data=p1;
    by index;
run;

proc sort data=p2;
    by index;
run;

data p1;
    set p1;
    retain m 0;
    m=m+1;
run;

data p2;
    set p2;
    retain m 0;
    m=m+1;
run;

data a11 a12 a13;
    set p1;
    if m>=1 & m<=26 then output a11;
    if m>=27 & m<=52 then output a12;
    if m>=53 & m<=79 then output a13;
run;

data a21 a22 a23;
    set p2;
    if m>=1 & m<=23 then output a21;
    if m>=24 & m<=46 then output a22;
    if m>=47 & m<=69 then output a23;
run;

data a1;
    set a11 a21;
run;
```

```
data a2;
    set a12 a22;
run;

data a3;
    set a13 a23;
run;

%crossvalidation(datain1=a1, datain2=a2, datain3=a3, factor=&fac, n=1)
%crossvalidation(datain1=a1, datain2=a3, datain3=a2, factor=&fac, n=2)
%crossvalidation(datain1=a2, datain2=a3, datain3=a1, factor=&fac, n=3)

data aaa;
    set aaa1 aaa2 aaa3;
run;

ods listing close;
ods trace on;
ods trace off;
ods output  Means.Summary=aa1;
proc means data=aaa;
    var shi spec1 spec2 spec3;
run;

data aa1;
    set aa1(keep=shi_mean spec1_mean spec2_mean spec3_mean rename=(shi_mean=shi_c
spec1_mean=shi_95 spec2_mean=shi_90 spec3_mean=shi_80));
run;

%mend;

%macro average(nfac, m);

data w;
       if 1=1 then delete;
       run;

%do j=1 %to &m;

%compute(&nfac, 0)

data w;
    set w aa1;
run;

%end;

ods listing;
proc means data=w;
    var  shi_c shi_95 shi_90 shi_80 ;
run;

%mend;

%average(2,100)
```

```
/*************************************************************/
/*   balanced 3-fold corssvalidation for pls regression      */
/*************************************************************/

proc pls data =sumby5 /*cv=split(10)cv=random*/ nfac=5;
    model inf=c1-c119;
output out=one PREDICTED=p;
run;

PROC LOGISTIC data=one descending;
    model inf=p/outroc=table1;
run;

%spec(table1,10)

proc print data=spec10;
run;

%macro crossvalidation(datain1=, datain2=, datain3=, factor=, n=);

data subdata;
    set &datain3(drop=inf);
run;

data subdata2;
    set &datain3;
run;
data training;
           set &datain1 &datain2;
run;
data whole;
    set training(in=in1) subdata(in=in2);
    m1=in1;
    m2=in2;
run;


ods listing close;
proc pls data = whole /*cv=split(10) cv=random */nfac=&factor;
    model inf=c1-c119;
output out=one PREDICTED=p;
run;


data logi1 logi2(drop=inf);
    set one(keep=inf p m1 m2 m);
    if m1=1 then output logi1;
    if m2=1 then output logi2;
run;

proc sort data=subdata2;
    by m;
run;

proc sort data=logi2;
    by m;
run;
```

```
data logi22;
    merge subdata2(keep=inf m) logi2(keep=m p);
    by m;
run;


ods listing close;
ods trace on;
ods trace off;
ods output Association=auc1;
PROC LOGISTIC data=logi1 descending;    /*training dataset */
    model inf=p/outroc=ctable1;
run;
ods listing;

ods listing close;
ods trace on;
ods trace off;
ods output Association=auc2;
PROC LOGISTIC data=logi22 descending;    /* subdata set   */
    model inf=p/outroc=ctable2;
run;
ods listing;

%spec(ctable1,1)
%spec(ctable2,2)

data bbb&n;
    merge spec1(rename=(s1=ts1 s2=ts2 s3=ts3)) spec2;
    spec1=ts1-s1;
    spec2=ts2-s2;
    spec3=ts3-s3;
run;

data aaa&n;
    merge auc1(keep=label2 nvalue2 rename=(nvalue2=c1)) auc2(keep=label2 nvalue2
rename=(nvalue2=c2));
    if label2^='c' then delete;
    drop label2;
    shi=c1-c2;
run;

data aaa&n;
    merge aaa&n bbb&n;
run;

%mend;

%macro compute(fac, nseed);

data p1 p2;
    set sumby5;
    if inf=1 then output p1;/* hsv1 79 */
    if inf=0 then output p2;/* mock 69 */
run;
```

```
data p1;
    set p1;
    retain n 0;
    n=n+1;
    index=ranuni(&nseed);
run;

data p2;
    set p2;
    retain n 0;
    n=n+1;
    index=ranuni(&nseed);
run;


proc sort data=p1;
    by index;
run;

proc sort data=p2;
    by index;
run;

data p1;
    set p1;
    retain m 0;
    m=m+1;
run;

data p2;
    set p2;
    retain m 0;
    m=m+1;
run;

data a11 a12 a13;
    set p1;
    if m>=1 & m<=26 then output a11;
    if m>=27 & m<=52 then output a12;
    if m>=53 & m<=79 then output a13;
run;

data a21 a22 a23;
    set p2;
    if m>=1 & m<=23 then output a21;
    if m>=24 & m<=46 then output a22;
    if m>=47 & m<=69 then output a23;
run;

data a1;
    set a11 a21;
run;

data a2;
    set a12 a22;
run;
```

```
data a3;
    set a13 a23;
run;

%crossvalidation(datain1=a1, datain2=a2, datain3=a3, factor=&fac, n=1)
%crossvalidation(datain1=a1, datain2=a3, datain3=a2, factor=&fac, n=2)
%crossvalidation(datain1=a2, datain2=a3, datain3=a1, factor=&fac, n=3)

data aaa;
    set aaa1 aaa2 aaa3;
run;

ods listing close;
ods trace on;
ods trace off;
ods output  Means.Summary=aa1;
proc means data=aaa;
    var shi spec1 spec2 spec3;
run;

data aa1;
    set aa1(keep=shi_mean spec1_mean spec2_mean spec3_mean rename=(shi_mean=shi_c
spec1_mean=shi_95 spec2_mean=shi_90 spec3_mean=shi_80));
run;

%mend;

%macro average(nfac, m);

data w;
       if 1=1 then delete;
        run;

%do j=1 %to &m;

%compute(&nfac, 0)

data w;
    set w aa1;
run;

%end;

ods listing;
proc means data=w;
    var  shi_c shi_95 shi_90 shi_80 ;
run;

%mend;

%average(5,100)

/*************************************************/
/*      Validate  for logistic regression          */
/*************************************************/

data sumby5_old;
```

```
    set new_cd.sumby5_mock_hsv1(firstobs=2);
run;

data sumby5_new;
    set new_cd.sumby5_vali_mock_hsv1(firstobs=2);
run;

/* check  old data*/
proc logistic data=sumby5_old DESCENDING ;
     model inf=c109 c116 c67/outroc=table1;
run;

data whole;
    set sumby5_new(drop=inf) sumby5_old;
run;

proc logistic data=whole DESCENDING ;
     model inf= c109 c116 c67;
     output out=one PREDICTED=p;
run;

data logi1(drop=inf) logi2;
    set one(keep=inf p);
    if _n_<=159 then output logi1; /* new data 159*/
       else output logi2; /* old data 148 */
run;

data logi1; /* new */
    set logi1;
    if _n_<=80 then inf=0;
       else inf=1;
run;

PROC LOGISTIC data=logi1 descending;    /*new data */
    model inf=p/outroc=ctable1;
run;

PROC LOGISTIC data=logi2 descending;    /* old data   */
    model inf=p/outroc=ctable2;
run;

%spec(ctable1,1) /* new data */
%spec(ctable2,2) /* old data */

data bbb;
    merge spec1(rename=(s1=ts1 s2=ts2 s3=ts3)) spec2;
    spec1=s1-ts1;
    spec2=s2-ts2;
    spec3=s3-ts3;
run;

proc print data=bbb;
run;

proc print data=ctable1;
run;
```

```
/*************************************************/
/*      Validate  for PLS regression              */
/*************************************************/

proc pls data = whole /*cv=split(10) cv=random */nfac=5;
    model inf=c1-c119;
output out=one PREDICTED=p;
run;

data logi1(drop=inf) logi2;
    set one(keep=inf p);
    if _n_<=159 then output logi1;
        else output logi2;
run;

data logi1;
    set logi1;
    if _n_<=80 then inf=0;
        else inf=1;
run;

PROC LOGISTIC data=logi1 descending;     /*new data */
    model inf=p/outroc=ctable1;
run;

PROC LOGISTIC data=logi2 descending;     /* old data   */
    model inf=p/outroc=ctable2;
run;

%spec(ctable1,1) /* new data */
%spec(ctable2,2) /* old data */

data bbb;
    merge spec1(rename=(s1=ts1 s2=ts2 s3=ts3)) spec2;
    spec1=s1-ts1;
    spec2=s2-ts2;
    spec3=s3-ts3;
run;

proc print data=bbb;
run;
```

APPENDIX C: SAS Code for Mock versus Adeno

```
/*****************************************/
/*   mock(inf=0) vs adeno(inf=1) (old data) */
/*****************************************/

data new_cd.mock_adeno_graph; /* for graph */
   merge new_cd.mock_graph new_cd.adeno_graph ;
run;

data mock;
   set new_cd.mock_analysis;
   inf=0;
run;

data adeno;
   set new_cd.adeno_analysis;
   inf=1;
run;

data new_cd.mock_adeno_analysis;
   set mock adeno;
run;

/*************plot the standardized graph (overall graph)************/

goptions reset=global gunit=pct border
       ctext=black ftitle=swissb ftext=swiss htitle=4 htext=3;

symbol1 color=blue i=j line=1 w=1 h=2.5 repeat=69;
symbol2 color=red i=j line=1 w=1 h=3 repeat=94;

legend1 across=1 down=2 label=none
       mode=protect position=(top inside left)
       value=('Mock' 'Adeno') offset=(1cm, -1cm);

axis1 label=(h=3 c=black"Wavenumber" )order=(800 to 1500 by 100)
     major=(height=2) minor=(height=1)
     width=3;

axis2 label=(h=3 c=black"absorbance" )order=(-2.5 to 2.5 by 0.5)
     major=(height=2) minor=(height=1)
     width=3;

title 'Standardized Data of Mock vs. Adeno';
 proc gplot data=new_cd.mock_adeno_graph; /* blue=Mock red=Adeno */
  plot (mock1-mock69 adeno1-adeno94)*x / overlay legend=legend1
                        haxis=axis1 hminor=4
                        vaxis=axis2 vminor=4;
run;
quit;
```

```
/*****************************************************/
/*       wilcoxon rank test (Z-score)                */
/*****************************************************/

data whole;
    set new_cd.mock_adeno_analysis;
run;

%wilcoxon

data w1;
    merge odata.xt w;
run;

data new_cd.z_mock_adeno;
    set w1;
run;

/*******z-score plot ****************/

goptions reset=global gunit=pct border
       ctext=black ftitle=swissb ftext=swiss htitle=4 htext=3;

symbol1 color=blue i=j line=1 w=1 h=2.5 repeat=1;

axis1 label=(h=4 c=black"Wavenumber" )order=(800 to 1500 by 100)
     major=(height=2) minor=(height=1)
     width=3;

axis2 label=(h=4 c=black"Z-score" )order=(-10 to 10 by 1)
     major=(height=2) minor=(height=1)
     width=3;

title 'Z Score for Mock vs. adeno';
 proc gplot data=new_cd.z_mock_adeno;
   plot z_score*x / overlay
                        haxis=axis1 hminor=4
                        vaxis=axis2 vminor=4
                         vref=4 -4 lvref=5;
run;
quit;

data z;
    set new_cd.z_mock_adeno;
run;

/*************************************/
/*       Sumby5                      */
/*************************************/

data sumby5;
    set odata.x new_cd.mock_adeno_analysis;
    keep cell inf v131-v160 v231-v350 v388-v422 v436-v490 v531-v615 v634-v728;
run;


%macro sumby5;
```

```
data sumby5;
    set sumby5;
    %do i=1 %to 6;
        c&i=0;
        %do j=0 %to 4;
            %let m=%sysevalf(131+5*(&i-1)+&j, integer);
            c&i=c&i+v&m;
        %end;
        c&i=c&i/5;
    %end;

    %do i=7 %to 30;
        c&i=0;
        %do j=0 %to 4;
            %let m=%sysevalf(231+5*(&i-6-1)+&j, integer);
            c&i=c&i+v&m;
        %end;
        c&i=c&i/5;
    %end;

    %do i=31 %to 37;
        c&i=0;
        %do j=0 %to 4;
            %let m=%sysevalf(388+5*(&i-30-1)+&j, integer);
            c&i=c&i+v&m;
        %end;
        c&i=c&i/5;
    %end;

    %do i=38 %to 48;
        c&i=0;
        %do j=0 %to 4;
            %let m=%sysevalf(436+5*(&i-37-1)+&j, integer);
            c&i=c&i+v&m;
        %end;
        c&i=c&i/5;
    %end;

    %do i=49 %to 65;
        c&i=0;
        %do j=0 %to 4;
            %let m=%sysevalf(531+5*(&i-48-1)+&j, integer);
            c&i=c&i+v&m;
        %end;
        c&i=c&i/5;
    %end;

    %do i=66 %to 84;
        c&i=0;
        %do j=0 %to 4;
            %let m=%sysevalf(634+5*(&i-65-1)+&j, integer);
            c&i=c&i+v&m;
        %end;
        c&i=c&i/5;
    %end;

keep cell inf c1-c84;
```

```
run;
%mend;

%sumby5

data new_cd.sumby5_mock_adeno;
    set sumby5;
run;

/*************************************************************/
/*    for validation: mock(inf=0) vs adeno(inf=1)        */
/*************************************************************/

data new_cd.vali_mock_adeno_graph; /* for graph */
    merge new_cd.vali_mock_graph new_cd.vali_adeno_graph ;
run;

data mock;
    set new_cd.vali_mock_analysis;
    inf=0;
run;

data adeno;
    set new_cd.vali_adeno_analysis;
    inf=1;
run;

data new_cd.vali_mock_adeno_analysis;
    set mock adeno;
run;

/*****************************************************/
/*       wilcoxon rank test (Z-score)             */
/*****************************************************/

data whole;
    set new_cd.vali_mock_adeno_analysis;
run;

%wilcoxon

data w1;
    merge odata.xt w;
run;

data new_cd.z_vali_mock_adeno;
    set w1;
run;

/*********************************/
/*       Sumby5                  */
/*********************************/

data sumby5;
    set odata.x new_cd.vali_mock_adeno_analysis;
    keep cell inf v131-v160 v231-v350 v388-v422 v436-v490 v531-v615 v634-v728;
run;
```

```
%sumby5

data new_cd.sumby5_vali_mock_adeno;
   set sumby5;
run;

/*************************************************************/
/*    balanced 3-fold crossvalidation for logistic regression */
/*************************************************************/

data sumby5;
   set new_cd.sumby5_mock_adeno(firstobs=2);
run;

ods listing;
proc logistic data=sumby5 DESCENDING ;
     model inf=c1-c84/selection=stepwise
                  sle=0.05
                  sls=0.05;
run;

/*selection the combination from c84 c77 c14 c30 c78 c71 c79 c59*/
/* choose C30 c59 c79 c84*/
proc logistic data=sumby5 DESCENDING ;
     model inf=C30 c59 c79 c84/outroc=table1;
run;

proc logistic data=sumby5 DESCENDING ;
     model inf=c84 c77 c14 c30/outroc=table1;
run;

/*choose c84 c78 c14 c30*/
proc logistic data=sumby5 DESCENDING ;
     model inf=c84 c78 c14 c30/outroc=table1;
run;

%spec(table1,10)

proc print data=spec10;
run;

/*************plot the standardized graph (partial graph)************/
data s;
   set new_cd.sumby5_mock_adeno(obs=2);
run;

proc print data=s;
   var c84 c78 c14 c30;
run;

goptions reset=global gunit=pct border
       ctext=black ftitle=swissb ftext=swiss htitle=4 htext=3;

symbol1 color=blue i=j line=1 w=1 h=2.5 repeat=69;
symbol2 color=red i=j line=1 w=1 h=3 repeat=94;
```

```
axis1 label=(h=3 c=black"Wavenumber" )order=(1050 to 1500 by 50)
     major=(height=2) minor=(height=1)
     width=3;

axis2 label=(h=3 c=black"absorbance" )order=(-2.5 to 2.5 by 0.5)
     major=(height=2) minor=(height=1)
     width=3;

title 'Location of chosen variables (Standardized Data of Mock vs. Adeno)';
 proc gplot data=new_cd.mock_adeno_graph; /* blue=Mock red=Adeno */
  plot (mock1-mock69 adeno1-adeno94)*x / overlay /*legend=legend1*/
                        haxis=axis1 hminor=4
                        vaxis=axis2 vminor=4
                         href= 1498.69    1469.75    1056.99    1134.14 lhref=5;
run;
quit;

goptions reset=global gunit=pct border
       ctext=black ftitle=swissb ftext=swiss htitle=4 htext=3;

symbol1 color=blue i=j line=1 w=1 h=2.5 repeat=69;
symbol2 color=red i=j line=1 w=1 h=3 repeat=94;

axis1 label=(h=3 c=black"Wavenumber" )order=(800 to 1500 by 100)
     major=(height=2) minor=(height=1)
     width=3;

axis2 label=(h=3 c=black"absorbance" )order=(-2.5 to 2.5 by 0.5)
     major=(height=2) minor=(height=1)
     width=3;

title 'Location of chosen variables (Standardized Data of Mock vs. Adeno)';
 proc gplot data=new_cd.mock_adeno_graph; /* blue=Mock red=Adeno */
  plot (mock1-mock69 adeno1-adeno94)*x / overlay
                        haxis=axis1 hminor=4
                        vaxis=axis2 vminor=4
                         href= 1498.69    1469.75    1056.99    1134.14 lhref=5;
run;
quit;

/*******z-score plot ****************/

goptions reset=global gunit=pct border
       ctext=black ftitle=swissb ftext=swiss htitle=4 htext=3;

symbol1 color=blue i=j line=1 w=1 h=2.5 repeat=1;

axis1 label=(h=4 c=black"Wavenumber" )order=(800 to 1500 by 100)
     major=(height=2) minor=(height=1)
     width=3;

axis2 label=(h=4 c=black"Z-score" )order=(-10 to 10 by 1)
     major=(height=2) minor=(height=1)
     width=3;

title 'Location of chosen variables (Z Score for Mock vs. Adeno)';
 proc gplot data=new_cd.z_mock_adeno;
```

```
   plot z_score*x / overlay
                        haxis=axis1 hminor=4
                        vaxis=axis2 vminor=4
                         vref=4 -4 lvref=5 href= 1498.69    1469.75    1056.99
1134.14
;
run;
quit;

%macro crossvalidation(datain1=, datain2=, datain3=, factor=, n=);

data subdata;
    set &datain3(drop=inf);
run;

data subdata2;
    set &datain3;
run;
data training;
           set &datain1 &datain2;
run;
data whole;
    set training(in=in1) subdata(in=in2);
    m1=in1;
    m2=in2;
run;

ods listing close;
proc logistic data=whole DESCENDING ;
     model inf=c84 c78 c14 c30;
     output out=one PREDICTED=p;
run;

data logi1 logi2(drop=inf);
    set one(keep=inf p m1 m2 m);
    if m1=1 then output logi1;
    if m2=1 then output logi2;
run;

proc sort data=subdata2;
    by m;
run;

proc sort data=logi2;
    by m;
run;

data logi22;
    merge subdata2(keep=inf m) logi2(keep=m p);
    by m;
run;


ods listing close;
ods trace on;
ods trace off;
ods output Association=auc1;
```

```
PROC LOGISTIC data=logi1 descending;    /*training dataset */
    model inf=p/outroc=ctable1;
run;
ods listing;

ods listing close;
ods trace on;
ods trace off;
ods output Association=auc2;
PROC LOGISTIC data=logi22 descending;    /* subdata set   */
    model inf=p/outroc=ctable2;
run;
ods listing;

%spec(ctable1,1)
%spec(ctable2,2)

data bbb&n;
    merge spec1(rename=(s1=ts1 s2=ts2 s3=ts3)) spec2;
    spec1=ts1-s1;
    spec2=ts2-s2;
    spec3=ts3-s3;
run;

data aaa&n;
    merge auc1(keep=label2 nvalue2 rename=(nvalue2=c1)) auc2(keep=label2 nvalue2
rename=(nvalue2=c2));
    if label2^='c' then delete;
    drop label2;
    shi=c1-c2;
run;

data aaa&n;
    merge aaa&n bbb&n;
run;

%mend;

%macro compute(fac, nseed);

data p1 p2;
    set sumby5;
    if inf=1 then output p1;/* adeno 94 */
    if inf=0 then output p2;/* mock 69 */
run;


data p1;
    set p1;
    retain n 0;
    n=n+1;
    index=ranuni(&nseed);
run;

data p2;
    set p2;
    retain n 0;
```

```
    n=n+1;
    index=ranuni(&nseed);
run;


proc sort data=p1;
    by index;
run;

proc sort data=p2;
    by index;
run;

data p1;
    set p1;
    retain m 0;
    m=m+1;
run;

data p2;
    set p2;
    retain m 0;
    m=m+1;
run;

data a11 a12 a13;
    set p1;
    if m>=1 & m<=31 then output a11;
    if m>=32 & m<=62 then output a12;
    if m>=63 & m<=94 then output a13;
run;

data a21 a22 a23;
    set p2;
    if m>=1 & m<=23 then output a21;
    if m>=24 & m<=46 then output a22;
    if m>=47 & m<=69 then output a23;
run;

data a1;
    set a11 a21;
run;

data a2;
    set a12 a22;
run;

data a3;
    set a13 a23;
run;



%crossvalidation(datain1=a1, datain2=a2, datain3=a3, factor=&fac, n=1)
%crossvalidation(datain1=a1, datain2=a3, datain3=a2, factor=&fac, n=2)
%crossvalidation(datain1=a2, datain2=a3, datain3=a1, factor=&fac, n=3)
```

```
data aaa;
    set aaa1 aaa2 aaa3;
run;

ods listing close;
ods trace on;
ods trace off;
ods output  Means.Summary=aa1;
proc means data=aaa;
    var shi spec1 spec2 spec3;
run;

data aa1;
    set aa1(keep=shi_mean spec1_mean spec2_mean spec3_mean rename=(shi_mean=shi_c
spec1_mean=shi_95 spec2_mean=shi_90 spec3_mean=shi_80));
run;

%mend;

%macro average(nfac, m);

data w;
       if 1=1 then delete;
        run;

%do j=1 %to &m;

%compute(&nfac, 0)

data w;
    set w aa1;
run;

%end;

ods listing;
proc means data=w;
    var  shi_c shi_95 shi_90 shi_80 ;
run;

%mend;

%average(2,100)

/*************************************************************/
/*   balanced 3-fold corssvalidation for pls regression     */
/*************************************************************/

proc pls data =sumby5 /*cv=split(10)cv=random*/ nfac=4;
    model inf=c1-c84;
output out=one PREDICTED=p;
run;

PROC LOGISTIC data=one descending;
    model inf=p/outroc=table1;
run;
```

```
%spec(table1,10)

proc print data=spec10;
run;

%macro crossvalidation(datain1=, datain2=, datain3=, factor=, n=);

data subdata;
    set &datain3(drop=inf);
run;

data subdata2;
    set &datain3;
run;
data training;
           set &datain1 &datain2;
run;
data whole;
    set training(in=in1) subdata(in=in2);
    m1=in1;
    m2=in2;
run;


ods listing close;
proc pls data = whole /*cv=split(10) cv=random */nfac=&factor;
    model inf=c1-c84;
output out=one PREDICTED=p;
run;


data logi1 logi2(drop=inf);
    set one(keep=inf p m1 m2 m);
    if m1=1 then output logi1;
    if m2=1 then output logi2;
run;

proc sort data=subdata2;
    by m;
run;

proc sort data=logi2;
    by m;
run;

data logi22;
    merge subdata2(keep=inf m) logi2(keep=m p);
    by m;
run;


ods listing close;
ods trace on;
ods trace off;
ods output Association=auc1;
PROC LOGISTIC data=logi1 descending;    /*training dataset */
    model inf=p/outroc=ctable1;
```

```
run;
ods listing;

ods listing close;
ods trace on;
ods trace off;
ods output Association=auc2;
PROC LOGISTIC data=logi22 descending;    /* subdata set   */
   model inf=p/outroc=ctable2;
run;
ods listing;

%spec(ctable1,1)
%spec(ctable2,2)

data bbb&n;
   merge spec1(rename=(s1=ts1 s2=ts2 s3=ts3)) spec2;
   spec1=ts1-s1;
   spec2=ts2-s2;
   spec3=ts3-s3;
run;

data aaa&n;
   merge auc1(keep=label2 nvalue2 rename=(nvalue2=c1)) auc2(keep=label2 nvalue2
rename=(nvalue2=c2));
   if label2^='c' then delete;
   drop label2;
   shi=c1-c2;
run;

data aaa&n;
   merge aaa&n bbb&n;
run;

%mend;

%macro compute(fac, nseed);

data p1 p2;
   set sumby5;
   if inf=1 then output p1;/* adeno 94 */
   if inf=0 then output p2;/* mock 69 */
run;

data p1;
   set p1;
   retain n 0;
   n=n+1;
   index=ranuni(&nseed);
run;

data p2;
   set p2;
   retain n 0;
   n=n+1;
   index=ranuni(&nseed);
run;
```

```
proc sort data=p1;
    by index;
run;

proc sort data=p2;
    by index;
run;

data p1;
    set p1;
    retain m 0;
    m=m+1;
run;

data p2;
    set p2;
    retain m 0;
    m=m+1;
run;

data a11 a12 a13;
    set p1;
    if m>=1 & m<=31 then output a11;
    if m>=32 & m<=62 then output a12;
    if m>=63 & m<=94 then output a13;
run;

data a21 a22 a23;
    set p2;
    if m>=1 & m<=23 then output a21;
    if m>=24 & m<=46 then output a22;
    if m>=47 & m<=69 then output a23;
run;

data a1;
    set a11 a21;
run;

data a2;
    set a12 a22;
run;

data a3;
    set a13 a23;
run;

%crossvalidation(datain1=a1, datain2=a2, datain3=a3, factor=&fac, n=1)
%crossvalidation(datain1=a1, datain2=a3, datain3=a2, factor=&fac, n=2)
%crossvalidation(datain1=a2, datain2=a3, datain3=a1, factor=&fac, n=3)

data aaa;
    set aaa1 aaa2 aaa3;
run;

ods listing close;
```

```
ods trace on;
ods trace off;
ods output  Means.Summary=aa1;
proc means data=aaa;
    var shi spec1 spec2 spec3;
run;

data aa1;
    set aa1(keep=shi_mean spec1_mean spec2_mean spec3_mean rename=(shi_mean=shi_c
spec1_mean=shi_95 spec2_mean=shi_90 spec3_mean=shi_80));
run;

%mend;

%macro average(nfac, m);

data w;
       if 1=1 then delete;
       run;

%do j=1 %to &m;

%compute(&nfac, 0)

data w;
    set w aa1;
run;

%end;

ods listing;
proc means data=w;
    var  shi_c shi_95 shi_90 shi_80 ;
run;

%mend;

%average(4,100)

/*************************************************/
/*      Validate  for logistic regression          */
/*************************************************/

data sumby5_old;
    set new_cd.sumby5_mock_adeno(firstobs=2);
run;

data sumby5_new;
    set new_cd.sumby5_vali_mock_adeno(firstobs=2);
run;

/* check  old data*/
proc logistic data=sumby5_old DESCENDING ;
     model inf=c84 c78 c14 c30/outroc=table1;
run;

data whole;
```

```
    set sumby5_new(drop=inf) sumby5_old;
run;

proc logistic data=whole DESCENDING ;
     model inf=c84 c78 c14 c30;
     output out=one PREDICTED=p;
run;

data logi1(drop=inf) logi2;
    set one(keep=inf p);
    if _n_<=164 then output logi1; /* new data 164*/
       else output logi2; /* old data 163 */
run;

data logi1; /* new */
    set logi1;
    if _n_<=80 then inf=0;
       else inf=1;
run;

PROC LOGISTIC data=logi1 descending;    /*new data */
    model inf=p/outroc=ctable1;
run;

PROC LOGISTIC data=logi2 descending;    /* old data   */
    model inf=p/outroc=ctable2;
run;


%spec(ctable1,1) /* new data */
%spec(ctable2,2) /* old data */

data bbb;
    merge spec1(rename=(s1=ts1 s2=ts2 s3=ts3)) spec2;
    spec1=s1-ts1;
    spec2=s2-ts2;
    spec3=s3-ts3;
run;

proc print data=bbb;
run;

proc print data=ctable1;
run;

/*************************************************/
/*      Validate  for PLS regression               */
/*************************************************/

proc pls data = whole /*cv=split(10) cv=random */nfac=4;
    model inf=c1-c84;
output out=one PREDICTED=p;
run;

data logi1(drop=inf) logi2;
    set one(keep=inf p);
    if _n_<=164 then output logi1;
```

```
       else output logi2;
run;

data logi1;
    set logi1;
    if _n_<=80 then inf=0;
       else inf=1;
run;

PROC LOGISTIC data=logi1 descending;     /*new data */
    model inf=p/outroc=ctable1;
run;

PROC LOGISTIC data=logi2 descending;     /* old data   */
    model inf=p/outroc=ctable2;
run;

%spec(ctable1,1) /* new data */
%spec(ctable2,2) /* old data */

data bbb;
    merge spec1(rename=(s1=ts1 s2=ts2 s3=ts3)) spec2;
    spec1=s1-ts1;
    spec2=s2-ts2;
    spec3=s3-ts3;
run;

proc print data=bbb;
run;
```

APPENDIX D: SAS Code for HSV1 versus Adeno

```
/*****************************************/
/*    hsv1(inf=0) vs adeno(inf=1) (old data)*/
/*****************************************/

data new_cd.hsv1_adeno_graph; /* for graph */
   merge new_cd.hsv1_graph new_cd.adeno_graph ;
run;

data hsv1;
   set new_cd.hsv1_analysis;
   inf=0;
run;

data adeno;
   set new_cd.adeno_analysis;
   inf=1;
run;

data new_cd.hsv1_adeno_analysis;
   set hsv1 adeno;
run;

/*************plot the standardized graph (overall graph)************/

goptions reset=global gunit=pct border
       ctext=black ftitle=swissb ftext=swiss htitle=4 htext=3;

symbol1 color=blue i=j line=1 w=1 h=2.5 repeat=79;
symbol2 color=red i=j line=1 w=1 h=3 repeat=94;

legend1 across=1 down=2 label=none
       mode=protect position=(top inside left)
       value=('HSV1' 'Adeno') offset=(1cm, -1cm);

axis1 label=(h=3 c=black"Wavenumber" )order=(800 to 1500 by 100)
     major=(height=2) minor=(height=1)
     width=3;

axis2 label=(h=3 c=black"absorbance" )order=(-2.5 to 2.5 by 0.5)
     major=(height=2) minor=(height=1)
     width=3;

title 'Standardized Data of HSV1 vs. Adeno';
 proc gplot data=new_cd.hsv1_adeno_graph; /* blue=HSV1 red=adeno */
  plot (hsv1-hsv79 adeno1-adeno94)*x / overlay legend=legend1
                        haxis=axis1 hminor=4
                        vaxis=axis2 vminor=4;
run;
quit;
```

```
/*****************************************************/
/*            wilcoxon rank test (Z-score)           */
/*****************************************************/

data whole;
    set new_cd.hsv1_adeno_analysis;
run;

%wilcoxon

data w1;
    merge odata.xt w;
run;

data new_cd.z_hsv1_adeno;
    set w1;
run;

/*******z-score plot ****************/

goptions reset=global gunit=pct border
       ctext=black ftitle=swissb ftext=swiss htitle=4 htext=3;

symbol1 color=blue i=j line=1 w=1 h=2.5 repeat=1;

axis1 label=(h=4 c=black"Wavenumber" )order=(800 to 1500 by 100)
     major=(height=2) minor=(height=1)
     width=3;

axis2 label=(h=4 c=black"Z-score" )order=(-10 to 10 by 1)
     major=(height=2) minor=(height=1)
     width=3;

title 'Z Score for HSV1 vs. Adeno';
 proc gplot data=new_cd.z_hsv1_adeno;
   plot z_score*x / overlay
                        haxis=axis1 hminor=4
                        vaxis=axis2 vminor=4
                         vref=4 -4 lvref=5;
run;
quit;

data z;
    set new_cd.z_hsv1_adeno;
run;

/*************************************/
/*       Sumby5                      */
/*************************************/

data sumby5;
    set odata.x new_cd.hsv1_adeno_analysis;
    keep cell inf v1-v85 v121-v145 v157-v236 v360-v384 v433-v517 v557-v601
v637-v681;
run;
```

```
%macro sumby5;
data sumby5;
    set sumby5;
    %do i=1 %to 17;
        c&i=0;
        %do j=0 %to 4;
            %let m=%sysevalf(1+5*(&i-1)+&j, integer);
            c&i=c&i+v&m;
        %end;
        c&i=c&i/5;
    %end;

    %do i=18 %to 22;
        c&i=0;
        %do j=0 %to 4;
            %let m=%sysevalf(121+5*(&i-17-1)+&j, integer);
            c&i=c&i+v&m;
        %end;
        c&i=c&i/5;
    %end;

    %do i=23 %to 38;
        c&i=0;
        %do j=0 %to 4;
            %let m=%sysevalf(157+5*(&i-22-1)+&j, integer);
            c&i=c&i+v&m;
        %end;
        c&i=c&i/5;
    %end;

    %do i=39 %to 43;
        c&i=0;
        %do j=0 %to 4;
            %let m=%sysevalf(360+5*(&i-38-1)+&j, integer);
            c&i=c&i+v&m;
        %end;
        c&i=c&i/5;
    %end;

    %do i=44 %to 60;
        c&i=0;
        %do j=0 %to 4;
            %let m=%sysevalf(433+5*(&i-43-1)+&j, integer);
            c&i=c&i+v&m;
        %end;
        c&i=c&i/5;
    %end;

    %do i=61 %to 69;
        c&i=0;
        %do j=0 %to 4;
            %let m=%sysevalf(557+5*(&i-60-1)+&j, integer);
            c&i=c&i+v&m;
        %end;
        c&i=c&i/5;
    %end;
```

```
    %do i=70 %to 78;
        c&i=0;
        %do j=0 %to 4;
            %let m=%sysevalf(637+5*(&i-69-1)+&j, integer);
            c&i=c&i+v&m;
        %end;
        c&i=c&i/5;
    %end;

keep cell inf c1-c78;
run;
%mend;

%sumby5

data new_cd.sumby5_hsv1_adeno;
    set sumby5;
run;

/*************************************************************/
/*    for validation: hsv1(inf=0) vs adeno(inf=1)         */
/*************************************************************/

data new_cd.vali_hsv1_adeno_graph; /* for graph */
    merge new_cd.vali_hsv1_graph new_cd.vali_adeno_graph ;
run;

data hsv1;
    set new_cd.vali_hsv1_analysis;
    inf=0;
run;

data adeno;
    set new_cd.vali_adeno_analysis;
    inf=1;
run;

data new_cd.vali_hsv1_adeno_analysis;
    set hsv1 adeno;
run;

/*****************************************************/
/*           wilcoxon rank test (Z-score)            */
/*****************************************************/

data whole;
    set new_cd.vali_hsv1_adeno_analysis;
run;

%wilcoxon

data w1;
    merge odata.xt w;
run;

data new_cd.z_vali_hsv1_adeno;
    set w1;
```

```
run;

/************************************/
/*            Sumby5                */
/************************************/

data sumby5;
    set odata.x new_cd.vali_hsv1_adeno_analysis;
    keep cell inf v1-v85 v121-v145 v157-v236 v360-v384 v433-v517 v557-v601
v637-v681;
run;

%sumby5

data new_cd.sumby5_vali_hsv1_adeno;
    set sumby5;
run;

/*************************************************************/
/*    balanced 3-fold crossvalidation for logistic regression */
/*************************************************************/
data sumby5;
    set new_cd.sumby5_hsv1_adeno(firstobs=2);
run;

ods listing;
proc logistic data=sumby5 DESCENDING ;
     model inf=c1-c78/selection=stepwise
                  sle=0.05
                  sls=0.05;
run;

/*selection the combination from c56 c23 c12 c38 c58 c19*/
/* choose c23 c12 c38 c58*/

proc logistic data=sumby5 DESCENDING ;
     model inf=c23 c12 c38 c58/outroc=table1;
run;

proc logistic data=sumby5 DESCENDING ;
     model inf=c56 c23 c12 c19/outroc=table1;
run;

proc logistic data=sumby5 DESCENDING ;
     model inf=c56 c23 c38 c12/outroc=table1;
run;

proc logistic data=sumby5 DESCENDING ;
     model inf=c23 c12 c58 c19/outroc=table1;
run;

proc logistic data=sumby5 DESCENDING ;
     model inf=c56 c23 c12 /outroc=table1;
run;

proc logistic data=sumby5 DESCENDING ;
     model inf=c58 c23 c12/outroc=table1;
```

```
run;


/* choose this one */
proc logistic data=sumby5 DESCENDING ;
     model inf=c56 c23 c12 c19/outroc=table1;
run;

%spec(table1,10)

proc print data=spec10;
run;

/*************plot the standardized graph (partial graph)************/
data s;
   set new_cd.sumby5_hsv1_adeno(obs=2);
run;

proc print data=s;
   var c56 c23 c12 c19;
run;

goptions reset=global gunit=pct border
       ctext=black ftitle=swissb ftext=swiss htitle=4 htext=3;

symbol1 color=blue i=j line=1 w=1 h=2.5 repeat=79;
symbol2 color=red i=j line=1 w=1 h=3 repeat=94;

axis1 label=(h=3 c=black"Wavenumber" )order=(850 to 1300 by 50)
     major=(height=2) minor=(height=1)
     width=3;

axis2 label=(h=3 c=black"absorbance" )order=(-2.5 to 2.5 by 0.5)
     major=(height=2) minor=(height=1)
     width=3;

title 'Location of chosen variables (Standardized Data of HSV1 vs. Adeno)';
 proc gplot data=new_cd.hsv1_adeno_graph; /* blue=hsv1 red=Adeno */
  plot (hsv1-hsv79 adeno1-adeno94)*x / overlay
                        haxis=axis1 hminor=4
                        vaxis=axis2 vminor=4
                         href= 1275.91    951.868    854.463    921.972
lhref=5;
run;
quit;

goptions reset=global gunit=pct border
       ctext=black ftitle=swissb ftext=swiss htitle=4 htext=3;

symbol1 color=blue i=j line=1 w=1 h=2.5 repeat=79;
symbol2 color=red i=j line=1 w=1 h=3 repeat=94;

axis1 label=(h=3 c=black"Wavenumber" )order=(800 to 1500 by 100)
     major=(height=2) minor=(height=1)
     width=3;

axis2 label=(h=3 c=black"absorbance" )order=(-2.5 to 2.5 by 0.5)
```

```
     major=(height=2) minor=(height=1)
     width=3;

title 'Location of chosen variables (Standardized Data of HSV1 vs. Adeno)';
 proc gplot data=new_cd.hsv1_adeno_graph; /* blue=hsv1 red=Adeno */
  plot (hsv1-hsv79 adeno1-adeno94)*x / overlay
                        haxis=axis1 hminor=4
                        vaxis=axis2 vminor=4
                         href=  1275.91    951.868    854.463    921.972
lhref=5;
run;
quit;


/*******z-score plot ****************/

goptions reset=global gunit=pct border
       ctext=black ftitle=swissb ftext=swiss htitle=4 htext=3;

symbol1 color=blue i=j line=1 w=1 h=2.5 repeat=1;

axis1 label=(h=4 c=black"Wavenumber" )order=(800 to 1500 by 100)
     major=(height=2) minor=(height=1)
     width=3;

axis2 label=(h=4 c=black"Z-score" )order=(-10 to 10 by 1)
     major=(height=2) minor=(height=1)
     width=3;

title 'Location of chosen variables (Z Score for HSV1 vs. Adeno)';
 proc gplot data=new_cd.z_hsv1_adeno;
  plot z_score*x / overlay
                        haxis=axis1 hminor=4
                        vaxis=axis2 vminor=4
                         vref=4 -4 lvref=5 href=  1275.91    951.868    854.463
921.972;
run;
quit;

%macro crossvalidation(datain1=, datain2=, datain3=, factor=, n=);

data subdata;
    set &datain3(drop=inf);
run;

data subdata2;
    set &datain3;
run;
data training;
          set &datain1 &datain2;
run;
data whole;
    set training(in=in1) subdata(in=in2);
    m1=in1;
    m2=in2;
run;
```

```
ods listing close;
proc logistic data=whole DESCENDING ;
     model inf=c56 c23 c12 c19;
     output out=one PREDICTED=p;
run;

data logi1 logi2(drop=inf);
    set one(keep=inf p m1 m2 m);
    if m1=1 then output logi1;
    if m2=1 then output logi2;
run;

proc sort data=subdata2;
    by m;
run;

proc sort data=logi2;
    by m;
run;

data logi22;
    merge subdata2(keep=inf m) logi2(keep=m p);
    by m;
run;


ods listing close;
ods trace on;
ods trace off;
ods output Association=auc1;
PROC LOGISTIC data=logi1 descending;    /*training dataset */
    model inf=p/outroc=ctable1;
run;
ods listing;

ods listing close;
ods trace on;
ods trace off;
ods output Association=auc2;
PROC LOGISTIC data=logi22 descending;    /* subdata set   */
    model inf=p/outroc=ctable2;
run;
ods listing;

%spec(ctable1,1)
%spec(ctable2,2)

data bbb&n;
    merge spec1(rename=(s1=ts1 s2=ts2 s3=ts3)) spec2;
    spec1=ts1-s1;
    spec2=ts2-s2;
    spec3=ts3-s3;
run;

data aaa&n;
    merge auc1(keep=label2 nvalue2 rename=(nvalue2=c1)) auc2(keep=label2 nvalue2
rename=(nvalue2=c2));
```

```
    if label2^='c' then delete;
    drop label2;
    shi=c1-c2;
run;

data aaa&n;
    merge aaa&n bbb&n;
run;

%mend;

%macro compute(fac, nseed);

data p1 p2;
    set sumby5;
    if inf=1 then output p1;/* adeno 94 */
    if inf=0 then output p2;/* hsv1 79 */
run;


data p1;
    set p1;
    retain n 0;
    n=n+1;
    index=ranuni(&nseed);
run;

data p2;
    set p2;
    retain n 0;
    n=n+1;
    index=ranuni(&nseed);
run;


proc sort data=p1;
    by index;
run;

proc sort data=p2;
    by index;
run;

data p1;
    set p1;
    retain m 0;
    m=m+1;
run;

data p2;
    set p2;
    retain m 0;
    m=m+1;
run;

data a11 a12 a13;
    set p1;
```

```
    if m>=1 & m<=31 then output a11;
    if m>=32 & m<=62 then output a12;
    if m>=63 & m<=94 then output a13;
run;

data a21 a22 a23;
    set p2;
    if m>=1 & m<=26 then output a21;
    if m>=27 & m<=52 then output a22;
    if m>=53 & m<=79 then output a23;
run;

data a1;
    set a11 a21;
run;

data a2;
    set a12 a22;
run;

data a3;
    set a13 a23;
run;

%crossvalidation(datain1=a1, datain2=a2, datain3=a3, factor=&fac, n=1)
%crossvalidation(datain1=a1, datain2=a3, datain3=a2, factor=&fac, n=2)
%crossvalidation(datain1=a2, datain2=a3, datain3=a1, factor=&fac, n=3)

data aaa;
    set aaa1 aaa2 aaa3;
run;

ods listing close;
ods trace on;
ods trace off;
ods output  Means.Summary=aa1;
proc means data=aaa;
    var shi spec1 spec2 spec3;
run;

data aa1;
    set aa1(keep=shi_mean spec1_mean spec2_mean spec3_mean rename=(shi_mean=shi_c
spec1_mean=shi_95 spec2_mean=shi_90 spec3_mean=shi_80));
run;

%mend;

%macro average(nfac, m);

data w;
       if 1=1 then delete;
        run;

%do j=1 %to &m;

%compute(&nfac, 0)
```

```
data w;
    set w aa1;
run;

%end;

ods listing;
proc means data=w;
    var  shi_c shi_95 shi_90 shi_80 ;
run;

%mend;

%average(2,100)

/*************************************************************/
/*    balanced 3-fold corssvalidation for pls regression     */
/*************************************************************/

proc pls data =sumby5 /*cv=split(10)cv=random*/ nfac=6;
    model inf=c1-c78;
output out=one PREDICTED=p;
run;

PROC LOGISTIC data=one descending;
    model inf=p/outroc=table1;
run;

%spec(table1,10)

proc print data=spec10;
run;

%macro crossvalidation(datain1=, datain2=, datain3=, factor=, n=);

data subdata;
    set &datain3(drop=inf);
run;

data subdata2;
    set &datain3;
run;
data training;
          set &datain1 &datain2;
run;
data whole;
    set training(in=in1) subdata(in=in2);
    m1=in1;
    m2=in2;
run;


ods listing close;
proc pls data = whole /*cv=split(10) cv=random */nfac=&factor;
    model inf=c1-c78;
output out=one PREDICTED=p;
run;
```

```
data logi1 logi2(drop=inf);
    set one(keep=inf p m1 m2 m);
    if m1=1 then output logi1;
    if m2=1 then output logi2;
run;

proc sort data=subdata2;
    by m;
run;

proc sort data=logi2;
    by m;
run;

data logi22;
    merge subdata2(keep=inf m) logi2(keep=m p);
    by m;
run;


ods listing close;
ods trace on;
ods trace off;
ods output Association=auc1;
PROC LOGISTIC data=logi1 descending;    /*training dataset */
    model inf=p/outroc=ctable1;
run;
ods listing;

ods listing close;
ods trace on;
ods trace off;
ods output Association=auc2;
PROC LOGISTIC data=logi22 descending;    /* subdata set   */
    model inf=p/outroc=ctable2;
run;
ods listing;

%spec(ctable1,1)
%spec(ctable2,2)

data bbb&n;
    merge spec1(rename=(s1=ts1 s2=ts2 s3=ts3)) spec2;
    spec1=ts1-s1;
    spec2=ts2-s2;
    spec3=ts3-s3;
run;

data aaa&n;
    merge auc1(keep=label2 nvalue2 rename=(nvalue2=c1)) auc2(keep=label2 nvalue2
rename=(nvalue2=c2));
    if label2^='c' then delete;
    drop label2;
    shi=c1-c2;
run;
```

```
data aaa&n;
    merge aaa&n bbb&n;
run;

%mend;

%macro compute(fac, nseed);

data p1 p2;
    set sumby5;
    if inf=1 then output p1;/* adeno 94 */
    if inf=0 then output p2;/* hsv1 79 */
run;

data p1;
    set p1;
    retain n 0;
    n=n+1;
    index=ranuni(&nseed);
run;

data p2;
    set p2;
    retain n 0;
    n=n+1;
    index=ranuni(&nseed);
run;


proc sort data=p1;
    by index;
run;

proc sort data=p2;
    by index;
run;

data p1;
    set p1;
    retain m 0;
    m=m+1;
run;

data p2;
    set p2;
    retain m 0;
    m=m+1;
run;

data a11 a12 a13;
    set p1;
    if m>=1 & m<=31 then output a11;
    if m>=32 & m<=62 then output a12;
    if m>=63 & m<=94 then output a13;
run;
```

```
data a21 a22 a23;
    set p2;
    if m>=1 & m<=26 then output a21;
    if m>=27 & m<=52 then output a22;
    if m>=53 & m<=79 then output a23;
run;

data a1;
    set a11 a21;
run;

data a2;
    set a12 a22;
run;

data a3;
    set a13 a23;
run;

%crossvalidation(datain1=a1, datain2=a2, datain3=a3, factor=&fac, n=1)
%crossvalidation(datain1=a1, datain2=a3, datain3=a2, factor=&fac, n=2)
%crossvalidation(datain1=a2, datain2=a3, datain3=a1, factor=&fac, n=3)

data aaa;
    set aaa1 aaa2 aaa3;
run;

ods listing close;
ods trace on;
ods trace off;
ods output  Means.Summary=aa1;
proc means data=aaa;
    var shi spec1 spec2 spec3;
run;

data aa1;
    set aa1(keep=shi_mean spec1_mean spec2_mean spec3_mean rename=(shi_mean=shi_c
spec1_mean=shi_95 spec2_mean=shi_90 spec3_mean=shi_80));
run;

%mend;

%macro average(nfac, m);

data w;
       if 1=1 then delete;
       run;

%do j=1 %to &m;

%compute(&nfac, 0)

data w;
    set w aa1;
run;

%end;
```

```
ods listing;
proc means data=w;
    var  shi_c shi_95 shi_90 shi_80 ;
run;

%mend;

%average(6,100)

/*************************************************/
/*        Validate  for logistic regression        */
/*************************************************/

data sumby5_old; /* old data 242*/
    set new_cd.sumby5_hsv1_adeno(firstobs=2);
run;

data sumby5_new;  /* new data 243*/
    set new_cd.sumby5_vali_hsv1_adeno(firstobs=2);
run;

/* check  old data*/
proc logistic data=sumby5_old DESCENDING ;
     model inf=c56 c23 c12 c19/outroc=table1;
run;

data whole;
    set sumby5_new(drop=inf) sumby5_old;
run;

proc logistic data=whole DESCENDING ;
     model inf=c56 c23 c12 c19;
     output out=one PREDICTED=p;
run;

data logi1(drop=inf) logi2;
    set one(keep=inf p);
    if _n_<=163 then output logi1; /* new data 163*/
       else output logi2; /* old data 173 */
run;

data logi1; /* new */
    set logi1;
    if _n_<=79 then inf=0;
       else inf=1;
run;

PROC LOGISTIC data=logi1 descending;    /*new data */
    model inf=p/outroc=ctable1;
run;

PROC LOGISTIC data=logi2 descending;    /* old data   */
    model inf=p/outroc=ctable2;
run;
```

```
%spec(ctable1,1) /* new data */
%spec(ctable2,2) /* old data */

data bbb;
    merge spec1(rename=(s1=ts1 s2=ts2 s3=ts3)) spec2;
    spec1=s1-ts1;
    spec2=s2-ts2;
    spec3=s3-ts3;
run;

proc print data=bbb;
run;


/*************************************************/
/*         Validate  for PLS regression          */
/*************************************************/

proc pls data = whole /*cv=split(10) cv=random */nfac=6;
    model inf=c1-c78;
output out=one PREDICTED=p;
run;

data logi1(drop=inf) logi2;
    set one(keep=inf p);
    if _n_<=163 then output logi1;
        else output logi2;
run;

data logi1;
    set logi1;
    if _n_<=79 then inf=0;
        else inf=1;
run;

PROC LOGISTIC data=logi1 descending;    /*new data */
    model inf=p/outroc=ctable1;
run;

PROC LOGISTIC data=logi2 descending;    /* old data   */
    model inf=p/outroc=ctable2;
run;

%spec(ctable1,1) /* new data */
%spec(ctable2,2) /* old data */

data bbb;
    merge spec1(rename=(s1=ts1 s2=ts2 s3=ts3)) spec2;
    spec1=s1-ts1;
    spec2=s2-ts2;
    spec3=s3-ts3;
run;

proc print data=bbb;
run;
```

APPENDIX E: SAS Code for Mock versus HSV1 and Adeno

```
/*****************************************************/
/*    mock(inf=0) vs hsv1 and adeno(inf=1) (old data) */
/*****************************************************/

data new_cd.mock_both_graph; /* for graph */
    merge new_cd.mock_graph new_cd.hsv1_graph new_cd.adeno_graph;
run;

data mock;
    set new_cd.mock_analysis;
    inf=0;
run;

data hsv1;
    set new_cd.hsv1_analysis;
    inf=1;
run;

data adeno;
    set new_cd.adeno_analysis;
    inf=1;
run;

data new_cd.mock_both_analysis;
    set mock hsv1 adeno;
run;

/*************plot the standardized graph (overall graph)************/

goptions reset=global gunit=pct border
       ctext=black ftitle=swissb ftext=swiss htitle=4 htext=3;

symbol1 color=blue i=j line=1 w=1 h=2.5 repeat=69;
symbol2 color=red i=j line=1 w=1 h=3 repeat=173;

legend1 across=1 down=2 label=none
       mode=protect position=(top inside left)
       value=('Mock' 'HSV1 and Adeno') offset=(1cm, -1cm);

axis1 label=(h=3 c=black"Wavenumber" )order=(800 to 1500 by 100)
     major=(height=2) minor=(height=1)
     width=3;

axis2 label=(h=3 c=black"absorbance" )order=(-2.5 to 2.5 by 0.5)
     major=(height=2) minor=(height=1)
     width=3;

title 'Standardized Data of Mock vs. HSV1 and Adeno';
 proc gplot data=new_cd.mock_both_graph; /* blue=Mock red=Hsv1 and adeno */
   plot (mock1-mock69 hsv1-hsv79 adeno1-adeno94)*x / overlay legend=legend1
```

```
                        haxis=axis1 hminor=4
                        vaxis=axis2 vminor=4;
run;
quit;

/*****************************************************/
/*       wilcoxon rank test (Z-score)                */
/*****************************************************/

data whole;
    set new_cd.mock_both_analysis;
run;

%wilcoxon

data w1;
   merge odata.xt w;
run;

data new_cd.z_mock_both;
    set w1;
run;

/*******z-score plot ****************/

goptions reset=global gunit=pct border
       ctext=black ftitle=swissb ftext=swiss htitle=4 htext=3;

symbol1 color=blue i=j line=1 w=1 h=2.5 repeat=1;

axis1 label=(h=4 c=black"Wavenumber" )order=(800 to 1500 by 100)
     major=(height=2) minor=(height=1)
     width=3;

axis2 label=(h=4 c=black"Z-score" )order=(-10 to 10 by 1)
     major=(height=2) minor=(height=1)
     width=3;

title 'Z Score for Mock vs. HSV1 and Adeno';
 proc gplot data=new_cd.z_mock_both;
  plot z_score*x / overlay
                        haxis=axis1 hminor=4
                        vaxis=axis2 vminor=4
                         vref=4 -4 lvref=5;
run;
quit;

data z;
    set new_cd.z_mock_both;
run;

/*************************************/
/*           Sumby5                  */
/*************************************/

data sumby5;
    set odata.x new_cd.mock_both_analysis;
```

```
    keep cell inf v1-v90 v127-v166 v181-v215 v237-v351 v380-v424 v434-v498
v530-v614 v634-v728;
run;


%macro sumby5;
data sumby5;
    set sumby5;
    %do i=1 %to 18;
        c&i=0;
        %do j=0 %to 4;
            %let m=%sysevalf(1+5*(&i-1)+&j, integer);
            c&i=c&i+v&m;
       %end;
        c&i=c&i/5;
    %end;

    %do i=19 %to 26;
        c&i=0;
        %do j=0 %to 4;
            %let m=%sysevalf(127+5*(&i-18-1)+&j, integer);
            c&i=c&i+v&m;
       %end;
        c&i=c&i/5;
    %end;

    %do i=27 %to 33;
        c&i=0;
        %do j=0 %to 4;
            %let m=%sysevalf(181+5*(&i-26-1)+&j, integer);
            c&i=c&i+v&m;
       %end;
        c&i=c&i/5;
    %end;

    %do i=34 %to 56;
        c&i=0;
        %do j=0 %to 4;
            %let m=%sysevalf(237+5*(&i-33-1)+&j, integer);
            c&i=c&i+v&m;
       %end;
        c&i=c&i/5;
    %end;

    %do i=57 %to 65;
        c&i=0;
        %do j=0 %to 4;
            %let m=%sysevalf(380+5*(&i-56-1)+&j, integer);
            c&i=c&i+v&m;
       %end;
        c&i=c&i/5;
    %end;

    %do i=66 %to 78;
        c&i=0;
        %do j=0 %to 4;
            %let m=%sysevalf(434+5*(&i-65-1)+&j, integer);
```

```
            c&i=c&i+v&m;
        %end;
        c&i=c&i/5;
    %end;

    %do i=79 %to 95;
        c&i=0;
        %do j=0 %to 4;
            %let m=%sysevalf(530+5*(&i-78-1)+&j, integer);
            c&i=c&i+v&m;
        %end;
        c&i=c&i/5;
    %end;

    %do i=96 %to 114;
        c&i=0;
        %do j=0 %to 4;
            %let m=%sysevalf(634+5*(&i-95-1)+&j, integer);
            c&i=c&i+v&m;
        %end;
        c&i=c&i/5;
    %end;

keep cell inf c1-c114;
run;
%mend;

%sumby5

data new_cd.sumby5_mock_both;
    set sumby5;
run;


/*************************************************************/
/*    for validation: mock(inf=0) vs hsv1 and adeno(inf=1) */
/*************************************************************/

data new_cd.vali_mock_both_graph; /* for graph */
    merge new_cd.vali_mock_graph new_cd.vali_hsv1_graph new_cd.vali_adeno_graph;
run;

data mock;
    set new_cd.vali_mock_analysis;
    inf=0;
run;

data hsv1;
    set new_cd.vali_hsv1_analysis;
    inf=1;
run;

data adeno;
    set new_cd.vali_adeno_analysis;
    inf=1;
run;
```

```
data new_cd.vali_mock_both_analysis;
    set mock hsv1 adeno;
run;

/*****************************************************/
/*           wilcoxon rank test (Z-score)            */
/*****************************************************/

data whole;
    set new_cd.vali_mock_both_analysis;
run;

%wilcoxon

data w1;
    merge odata.xt w;
run;

data new_cd.z_vali_mock_both;
    set w1;
run;


/************************************/
/*            Sumby5                */
/************************************/

data sumby5;
    set odata.x new_cd.vali_mock_both_analysis;
    keep cell inf v1-v90 v127-v166 v181-v215 v237-v351 v380-v424 v434-v498
v530-v614 v634-v728;
run;

%sumby5

data new_cd.sumby5_vali_mock_both;
    set sumby5;
run;

/*************************************************************/
/*    balanced 3-fold crossvalidation for logistic regression */
/*************************************************************/

data sumby5;
    set new_cd.sumby5_mock_both(firstobs=2);
run;

ods listing;
proc logistic data=sumby5 DESCENDING ;
     model inf=c1-c114/selection=stepwise
                  sle=0.05
                  sls=0.05;
run;

proc logistic data=sumby5 DESCENDING ;
     model inf=c103 c66 c106 c98 c39 /outroc=table1;
run;
```

```
%spec(table1,10)

proc print data=spec10;
run;

/*************plot the standardized graph (partial graph)************/
data s;
   set new_cd.sumby5_mock_both(obs=2);
run;

proc print data=s;
   var c103 c66 c106 c98 c39;
run;

goptions reset=global gunit=pct border
       ctext=black ftitle=swissb ftext=swiss htitle=4 htext=3;

symbol1 color=blue i=j line=1 w=1 h=2.5 repeat=69;
symbol2 color=red i=j line=1 w=1 h=3 repeat=173;

legend1 across=1 down=2 label=none
       mode=protect position=(top inside left)
       value=('Mock' 'HSV1 and Adeno') offset=(1cm, -1cm);

axis1 label=(h=3 c=black"Wavenumber" )order=(1000 to 1500 by 50)
     major=(height=2) minor=(height=1)
     width=3;

axis2 label=(h=3 c=black"absorbance" )order=(-2.5 to 2.5 by 0.5)
     major=(height=2) minor=(height=1)
     width=3;

title 'Location of chosen variables (Standardized Data of Mock vs. HSV1 and Adeno)';
 proc gplot data=new_cd.mock_both_graph; /* blue=Mock red=Hsv1 and adeno */
   plot (mock1-mock69 hsv1-hsv79 adeno1-adeno94)*x / overlay legend=legend1
                        haxis=axis1 hminor=4
                        vaxis=axis2 vminor=4
                         href= 1445.64   1219.01   1460.11   1421.53
1053.13 lhref=5;
run;
quit;

/*******z-score plot ****************/

goptions reset=global gunit=pct border
       ctext=black ftitle=swissb ftext=swiss htitle=4 htext=3;

symbol1 color=blue i=j line=1 w=1 h=2.5 repeat=1;

axis1 label=(h=4 c=black"Wavenumber" )order=(800 to 1500 by 100)
     major=(height=2) minor=(height=1)
     width=3;

axis2 label=(h=4 c=black"Z-score" )order=(-10 to 10 by 1)
     major=(height=2) minor=(height=1)
     width=3;
```

```
title 'Location of chosen variables (Z Score for Mock vs. HSV1 and Adeno)';
 proc gplot data=new_cd.z_mock_both;
   plot z_score*x / overlay
                        haxis=axis1 hminor=4
                        vaxis=axis2 vminor=4
                         vref=4 -4 lvref=5 href=1445.64    1219.01    1460.11
1421.53    1053.13;
run;
quit;

/*******blanced Cross-validation (3 folds 95% 85% 80% sencitivity) ******/

%macro crossvalidation(datain1=, datain2=, datain3=, factor=, n=);

data subdata;
    set &datain3(drop=inf);
run;

data subdata2;
    set &datain3;
run;
data training;
           set &datain1 &datain2;
run;
data whole;
    set training(in=in1) subdata(in=in2);
    m1=in1;
    m2=in2;
run;

ods listing close;
proc logistic data=whole DESCENDING ;
     model inf=c103 c66 c106 c98 c39;
     output out=one PREDICTED=p;
run;

data logi1 logi2(drop=inf);
    set one(keep=inf p m1 m2 m);
    if m1=1 then output logi1;
    if m2=1 then output logi2;
run;

proc sort data=subdata2;
    by m;
run;

proc sort data=logi2;
    by m;
run;

data logi22;
    merge subdata2(keep=inf m) logi2(keep=m p);
    by m;
run;
```

```
ods listing close;
ods trace on;
ods trace off;
ods output Association=auc1;
PROC LOGISTIC data=logi1 descending;    /*training dataset */
   model inf=p/outroc=ctable1;
run;
ods listing;

ods listing close;
ods trace on;
ods trace off;
ods output Association=auc2;
PROC LOGISTIC data=logi22 descending;    /* subdata set   */
   model inf=p/outroc=ctable2;
run;
ods listing;

%spec(ctable1,1)
%spec(ctable2,2)

data bbb&n;
   merge spec1(rename=(s1=ts1 s2=ts2 s3=ts3)) spec2;
   spec1=ts1-s1;
   spec2=ts2-s2;
   spec3=ts3-s3;
run;

data aaa&n;
   merge auc1(keep=label2 nvalue2 rename=(nvalue2=c1)) auc2(keep=label2 nvalue2
rename=(nvalue2=c2));
   if label2^='c' then delete;
   drop label2;
   shi=c1-c2;
run;

data aaa&n;
   merge aaa&n bbb&n;
run;

%mend;

%macro compute(fac, nseed);

data p1 p2;
   set sumby5;
   if inf=1 then output p1;/* hsv1 and adeno 173 */
   if inf=0 then output p2;/* mock 69 */
run;


data p1;
   set p1;
   retain n 0;
   n=n+1;
   index=ranuni(&nseed);
run;
```

```
data p2;
    set p2;
    retain n 0;
    n=n+1;
    index=ranuni(&nseed);
run;


proc sort data=p1;
    by index;
run;

proc sort data=p2;
    by index;
run;

data p1;
    set p1;
    retain m 0;
    m=m+1;
run;

data p2;
    set p2;
    retain m 0;
    m=m+1;
run;

data a11 a12 a13;
    set p1;
    if m>=1 & m<=57 then output a11;
    if m>=58 & m<=115 then output a12;
    if m>=116 & m<=173 then output a13;
run;

data a21 a22 a23;
    set p2;
    if m>=1 & m<=23 then output a21;
    if m>=24 & m<=46 then output a22;
    if m>=47 & m<=69 then output a23;
run;

data a1;
    set a11 a21;
run;

data a2;
    set a12 a22;
run;

data a3;
    set a13 a23;
run;

%crossvalidation(datain1=a1, datain2=a2, datain3=a3, factor=&fac, n=1)
%crossvalidation(datain1=a1, datain2=a3, datain3=a2, factor=&fac, n=2)
```

```
%crossvalidation(datain1=a2, datain2=a3, datain3=a1, factor=&fac, n=3)

data aaa;
    set aaa1 aaa2 aaa3;
run;

ods listing close;
ods trace on;
ods trace off;
ods output  Means.Summary=aa1;
proc means data=aaa;
    var shi spec1 spec2 spec3;
run;

data aa1;
    set aa1(keep=shi_mean spec1_mean spec2_mean spec3_mean rename=(shi_mean=shi_c
spec1_mean=shi_95 spec2_mean=shi_90 spec3_mean=shi_80));
run;

%mend;

%macro average(nfac, m);

data w;
       if 1=1 then delete;
        run;

%do j=1 %to &m;

%compute(&nfac, 0)

data w;
    set w aa1;
run;

%end;

ods listing;
proc means data=w;
    var  shi_c shi_95 shi_90 shi_80 ;
run;

%mend;

%average(2,100)

/*************************************************************/
/*   balanced 3-fold corssvalidation for pls regression     */
/*************************************************************/

proc pls data =sumby5 /*cv=split(10)cv=random*/ nfac=5;
    model inf=c1-c114;
output out=one PREDICTED=p;
run;

PROC LOGISTIC data=one descending;
    model inf=p/outroc=table1;
```

```
run;

%spec(table1,10)

proc print data=spec10;
run;

/*******balanced Cross-validation (3 folds 95% 85% 80% sencitivity) ******/

%macro crossvalidation(datain1=, datain2=, datain3=, factor=, n=);

data subdata;
    set &datain3(drop=inf);
run;

data subdata2;
    set &datain3;
run;
data training;
           set &datain1 &datain2;
run;
data whole;
    set training(in=in1) subdata(in=in2);
    m1=in1;
    m2=in2;
run;


ods listing close;
proc pls data = whole /*cv=split(10) cv=random */nfac=&factor;
    model inf=c1-c114;
output out=one PREDICTED=p;
run;


data logi1 logi2(drop=inf);
    set one(keep=inf p m1 m2 m);
    if m1=1 then output logi1;
    if m2=1 then output logi2;
run;

proc sort data=subdata2;
    by m;
run;

proc sort data=logi2;
    by m;
run;

data logi22;
    merge subdata2(keep=inf m) logi2(keep=m p);
    by m;
run;


ods listing close;
ods trace on;
```

```
ods trace off;
ods output Association=auc1;
PROC LOGISTIC data=logi1 descending;    /*training dataset */
   model inf=p/outroc=ctable1;
run;
ods listing;

ods listing close;
ods trace on;
ods trace off;
ods output Association=auc2;
PROC LOGISTIC data=logi22 descending;    /* subdata set   */
   model inf=p/outroc=ctable2;
run;
ods listing;

%spec(ctable1,1)
%spec(ctable2,2)

data bbb&n;
   merge spec1(rename=(s1=ts1 s2=ts2 s3=ts3)) spec2;
   spec1=ts1-s1;
   spec2=ts2-s2;
   spec3=ts3-s3;
run;

data aaa&n;
   merge auc1(keep=label2 nvalue2 rename=(nvalue2=c1)) auc2(keep=label2 nvalue2
rename=(nvalue2=c2));
   if label2^='c' then delete;
   drop label2;
   shi=c1-c2;
run;

data aaa&n;
   merge aaa&n bbb&n;
run;

%mend;

%macro compute(fac, nseed);

data p1 p2;
   set sumby5;
   if inf=1 then output p1;/* hsv1 and adeno 173 */
   if inf=0 then output p2;/* mock 69 */
run;

data p1;
   set p1;
   retain n 0;
   n=n+1;
   index=ranuni(&nseed);
run;

data p2;
   set p2;
```

```
    retain n 0;
    n=n+1;
    index=ranuni(&nseed);
run;


proc sort data=p1;
    by index;
run;

proc sort data=p2;
    by index;
run;

data p1;
    set p1;
    retain m 0;
    m=m+1;
run;

data p2;
    set p2;
    retain m 0;
    m=m+1;
run;

data a11 a12 a13;
    set p1;
    if m>=1 & m<=57 then output a11;
    if m>=58 & m<=115 then output a12;
    if m>=116 & m<=173 then output a13;
run;

data a21 a22 a23;
    set p2;
    if m>=1 & m<=23 then output a21;
    if m>=24 & m<=46 then output a22;
    if m>=47 & m<=69 then output a23;
run;

data a1;
    set a11 a21;
run;

data a2;
    set a12 a22;
run;

data a3;
    set a13 a23;
run;

%crossvalidation(datain1=a1, datain2=a2, datain3=a3, factor=&fac, n=1)
%crossvalidation(datain1=a1, datain2=a3, datain3=a2, factor=&fac, n=2)
%crossvalidation(datain1=a2, datain2=a3, datain3=a1, factor=&fac, n=3)

data aaa;
```

```
    set aaa1 aaa2 aaa3;
run;

ods listing close;
ods trace on;
ods trace off;
ods output  Means.Summary=aa1;
proc means data=aaa;
    var shi spec1 spec2 spec3;
run;

data aa1;
    set aa1(keep=shi_mean spec1_mean spec2_mean spec3_mean rename=(shi_mean=shi_c
spec1_mean=shi_95 spec2_mean=shi_90 spec3_mean=shi_80));
run;

%mend;

%macro average(nfac, m);

data w;
       if 1=1 then delete;
        run;

%do j=1 %to &m;

%compute(&nfac, 0)

data w;
    set w aa1;
run;

%end;

ods listing;
proc means data=w;
    var  shi_c shi_95 shi_90 shi_80 ;
run;

%mend;

%average(5,100)

proc print data=aaa;
run;

proc print data=w;
run;

/*****************************************************/
/*        Validate  for logistic regression          */
/*****************************************************/

data sumby5_old;
    set new_cd.sumby5_mock_both(firstobs=2);
run;
```

```
data sumby5_new;
    set new_cd.sumby5_vali_mock_both(firstobs=2);
run;

/* check  old data*/
proc logistic data=sumby5_old DESCENDING ;
     model inf=c103 c66 c106 c98 c39/outroc=table1;
run;

data whole;
    set sumby5_new(drop=inf) sumby5_old;
run;

proc logistic data=whole DESCENDING ;
     model inf= c103 c66 c106 c98 c39;
     output out=one PREDICTED=p;
run;

data logi1(drop=inf) logi2;
    set one(keep=inf p);
    if _n_<=243 then output logi1; /* new data 243*/
       else output logi2; /* old data 242 */
run;

data logi1; /* new */
    set logi1;
    if _n_<=80 then inf=0;
       else inf=1;
run;

PROC LOGISTIC data=logi1 descending;    /*new data */
    model inf=p/outroc=ctable1;
run;

PROC LOGISTIC data=logi2 descending;    /* old data   */
    model inf=p/outroc=ctable2;
run;


%spec(ctable1,1) /* new data */
%spec(ctable2,2) /* old data */

data bbb;
    merge spec1(rename=(s1=ts1 s2=ts2 s3=ts3)) spec2;
    spec1=s1-ts1;
    spec2=s2-ts2;
    spec3=s3-ts3;
run;

proc print data=bbb;
run;

/*************************************************/
/*      Validate  for PLS regression             */
/*************************************************/

proc pls data = whole /*cv=split(10) cv=random */nfac=5;
```

```
    model inf=c1-c114;
output out=one PREDICTED=p;
run;

data logi1(drop=inf) logi2;
    set one(keep=inf p);
    if _n_<=243 then output logi1;
        else output logi2;
run;

data logi1;
    set logi1;
    if _n_<=80 then inf=0;
        else inf=1;
run;

PROC LOGISTIC data=logi1 descending;     /*new data */
    model inf=p/outroc=ctable1;
run;

PROC LOGISTIC data=logi2 descending;     /* old data   */
    model inf=p/outroc=ctable2;
run;

%spec(ctable1,1) /* new data */
%spec(ctable2,2) /* old data */

data bbb;
    merge spec1(rename=(s1=ts1 s2=ts2 s3=ts3)) spec2;
    spec1=s1-ts1;
    spec2=s2-ts2;
    spec3=s3-ts3;
run;

proc print data=bbb;
run;
```