7-13-2009

Advanced Statistical Methodologies in Determining the Observation Time to Discriminate Viruses Using FTIR

Shan Luo
Georgia State University

ADVANCED STATISTICAL METHODOLOGIES IN DETERMINING THE OBSERVATION TIME TO DISCRIMINATE VIRUSES USING FTIR

by

SHAN LUO

Under the Direction of Yu-Sheng Hsu

ABSTRACT

Fourier transform infrared (FTIR) spectroscopy, one method of electromagnetic radiation for detecting specific cellular molecular structure, can be used to discriminate different types of cells. The objective is to find the minimum time (choice among 2 hour, 4 hour and 6 hour) to record FTIR readings such that different viruses can be discriminated. A new method is adopted for the datasets. Briefly, inner differences are created as the control group, and Wilcoxon Signed Rank Test is used as the first selecting variable procedure in order to prepare the next stage of discrimination. In the second stage we propose either partial least squares (PLS) method or simply taking significant differences as the discriminator. Finally, k-fold cross-validation method is used to estimate the shrinkages of the goodness measures, such as sensitivity, specificity and area under the ROC curve (AUC). There is no doubt in our mind 6 hour is enough for discriminating mock from Hsv1, and Coxsackie viruses. Adeno virus is an exception.

INDEX WORDS: Inner-difference, Intra-difference, Wilcoxon Signed-Rank Test, Partial Least Square Regression, Area Under the ROC Curve, Specificity, Sensitivity, Shrinkage, K-fold Cross-Validation, Bootstrap method

ADVANCED STATISTICAL METHODOLOGIES IN DETERMINING THE OBSERVATION TIME TO DISCRIMINATE VIRUSES USING FTIR

by

SHAN LUO

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

2009

ADVANCED STATISTICAL METHODOLOGIES IN DETERMINING THE OBSERVATION TIME TO DISCRIMINATE VIRUSES USING FTIR

by

SHAN LUO

Committee Chair: Dr. Yu-Sheng Hsu

Committee: Dr. Xu Zhang
Dr. Yuanhui Xiao

Electronic Version Approved:

Office of Graduate Studies
College of Arts and Sciences
Georgia State University
August 2009

DEDICATION

This Thesis is dedicated to Yu-sheng Hsu,

My parents,

and all my friends

ACKNOWLEDGEMENTS

I would like to appreciate my deep and sincere gratitude to every people who have given me help in the process of completing this thesis.

Under the direct of my supervisor, Dr. Yu-Sheng Hsu, finally I can finish this study step by step. From discussion on detailed SAS codes, through specific theories on research and simulation, to the final results, it cannot be accomplished without his help. I have consolidated my knowledge in statistics and learned a lot of new statistical methods from Dr. Yu-Sheng Hsu, which cannot be measured only by language. The research path to the final results is long and tough. From Dr. Yu-Sheng Hsu, I find that not only knowledge and skills are necessary in research, enthusiasm, patience and perseverance also cannot be missed.

I would like to express many thanks to Dr. Hasting, Jing Guo and Ruili Wang who provide me the original data and their source, also for their help in assisting me to understand the biological and physical background of the research.

I would like to express many thanks to Dr. Xu Zhang. I become more and more proficient in SAS programming starting from her classes. My thesis is improved a lot based on her help.

I would like to express many thanks to Dr. Yixin Fang, Tian Tang, Xin Huang and Dongmei Wang. I get more familiar with either biostatistical background or SAS programming under their help.

I would also like to express many thanks to Dr. Yuanhui Xiao, who takes time to participate as proofreaders and reviewer in my committee member, providing comments and suggestions in improving my thesis.

Finally, I want to give my thanks to my family and friends for their love, encouragement and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Microscopic Fourier Transform Infrared (FTIR) is a measurement technique of the electromagnetic radiation by penetrating to cell structure and reflecting the absorbance of cell tissues. FTIR, which has been approved to be an accurate method in detecting diagnosis, is used all over this study and provides the whole dataset to our research.

There are four kinds of monkey kidney cells, including Mock, Hsv1, Adeno and Coxsackie. The purpose of this study to find a method that can discriminate these four cells.

In the original dataset, it takes 24 hours to detect absorbance by Microscopic Fourier Transform Infrared (FTIR), which is time-consuming. Thus, we changed the time measurement to 2 hours, 4 hours and 6 hours. Through advanced statistical methods mentioned in the abstract, we found that the 6 hour measurement is more reasonable than the 2 hour and 4 hour measurement. Using this method, we improved the efficiency of FTIR's measurement and saved huge amount of time and resources. The absorbance data are detected by FTIR machine on a spectra range from wavenumber of 799-1500 cm^{-1}.728 measurements are taken respectively.

In this study, we do statistical analysis for 2 hour, 4 hour and 6 hour dataset. The final results have shown that 6 hour dataset is sufficient to distinguish among these four types of cells except Mock vs. Adeno paired comparison.

The thesis is organized as follows: In Chapter 2, the whole process and all the statistical methodologies used are introduced. The main methods include Wilcoxon Signed Rank Test, Model built with positive terms minus negative terms, Partial Least Square Regression (PLSR), Area Under the ROC Curve (AUC), bootstrap simulation to build confidence interval, Cholesky Decomposition to generate multivariate normal distribution, k-fold cross-validation. Meanwhile, the comparison between the result of model with positive terms minus negative terms and the result of PLSR are described. In Chapter 3, the paired comparison of 6 hour Mock vs. Hsv1 are used as the main example in the whole study. Certainly, the integrated paired comparisons include Mock vs. Hsv1, Mock vs. Adeno, Mock vs. Coxsackie, Hsv1 vs. Coxsackie, Hsv1 vs. Adeno, Adeno vs. Coxsackie. Chapter 4 gives a discussion on further studies. Parts of SAS codes involved in this thesis are attached as Appendix D.

Chapter 2

Methodology

2.1 Data Manipulation

In this study, four kinds of monkey kidney cells, namely Mock, Hsv1, Adeno and Coxsackie, are available for statistical data analysis.

Totally, there are 21 paired comparison datasets for Mock vs. Hsv1, 20 paired comparison datasets for Mock vs. Adeno, 18 paired comparison datasets for Mock vs. Coxsackie, 20 paired comparison datasets for Hsv1 vs. Adeno, 17 paired comparison datasets for Adeno vs. Coxsackie and 18 paired comparison datasets for Hsv1 vs. Coxsackie. Please refer to Appendix A for details of these date group.

Before starting our statistical data analysis, we polished out data first which included dropping useless character information from our data set, dealing with missing values, standardization and so on. The process for standardization is as follows

$$\text{Standardized data } y_i = \frac{x_i x + {}^{-}}{s_x}$$

$$\text{Mean } \bar{x}x = \frac{1}{n}\overset{n}{\underset{i=1}{\longleftrightarrow}}{}_i$$

$$\text{Standard deviation } s_{xi}xx = +\sqrt{\frac{1}{n+1}\overset{n}{\underset{i=1}{\longleftrightarrow}}x) \qquad {}^{-\ 2}}$$

Where $x_i, i = 1,2,\cdots,728$ are the 728 absorbencies at each point

The methodology will be described by the comparison of 6 hour Mock vs. Hsv1's data. All the other comparisons will follow the same pattern.

First, Mock vs. Hsv1 are hard to compare due to large variations during date of observations. Therefore, we adopted a pair wise comparison method, i.e. we only compare Mock & Hsv1 at the same date.

Since we did a pair wise comparison, we do not have the control groups. To compensate for the lack of control we constructed a control group by analyzing the differences between two Mocks and between two Hsv1s. Therefore, we randomly split the set of observations of each date into two equal-size parts, i.e. two Mock sets and two Hsv1 sets for each date. Then the differences between two Mock sets and between two Hsv1 sets can serve as control groups.

Specifically, for 03/26/08 data, there are 57 Mocks and 70 Hsv1s. We denote two subgroups of Hsv1 by Hsv_{1i} and Hsv_{2j}, where i, j=1,2, … ,35. Similarly, we denote two subgroups of Mocks by $Mock_{1i}$ and $Mock_{2j}$, i=1, … , 29 and j=1, … , 28. Their averages will be denoted by Hsv1, Hsv2, Mock1 and Mock2, respectively.

We define the inner difference as

INN1=Mock1-Mock2

INN2=Hsv1-Hsv2

We define the intra difference as

INT1=Mock1-Hsv1

INT2=Mock2-Hsv2

In this case, each pair wise date group should have two inner-differences and two intra-differences.

From Central Limit Theorem, INN and INT are both normally distributed. Notice that there are 728 INN1, INN2, INT1, INT2, respectively at 728 frequencies/wavenumbers. These inner-differences and intra-differences are assumed to be independent.

Because there are 728 wavenumbers, a sum by n method is used to smooth the lines on the plot. No significant difference is detected between the two situations after comparing the plots of sumby 2 and sumby 4. Thus, Sumby 4 is chosen for inner-difference and intra-difference.

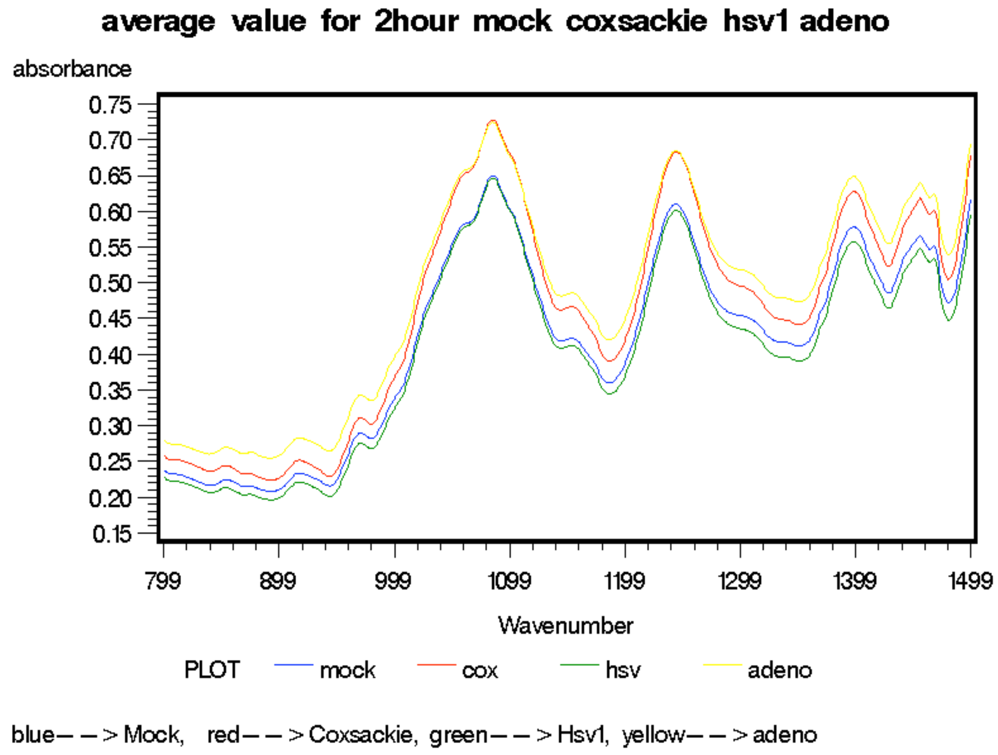


Figure 1. Average line for 2 hour Mock, Coxsackie Hsv1 and Adeno

2.2 Wilcoxon Signed-Rank Test

Wilcoxon Signed-Rank Test, also called One-Tailed Wilcoxon Rank Test, is a nonparametric method used to test whether the location of the measurement is equal to a prespecified value. Moreover, Wilcoxon Signed-Rank Test can also be used as an alternative way to the paired student's t-test in a case when the population is not normally distributed. Even if normal distribution is not satisfied, we can still use Wilcoxon Signed-Rank Test.

Let Z_i denotes intra-differences, for i=1… n. There are two assumptions about Wilcoxon signed-rank test. One is that Z_i are assumed to be independent; the other is that Z_i are drawn from a continuous population and is symmetric about a specified value θ, given that the null hypothesis test of Wilcoxon signed-rank test is $H_0 : \theta = 0$.

Excluding intra-difference with a zero value, after ranking the absolute values of the intra-differences as $|Z_i|$, we attach the signs of the differences to the ranks. The ranking of each ordered $|Z_i|$ is given a rank of R_i, which are called signed ranks. Let us denote φ_i for the positive Z_i values, where $\varphi_i = I(\text{indicator function})\,(Z_i > 0)$. Now that we can set up the Wilcoxon signed-rank statistic value W_+ by

$$W_+ = \sum_{i=1}^{n} \varphi_i R_i$$

We call the number of signed ranks as N, N may be less than or equal to the number of intra-differences.

From the graph for Wilcoxon signed-rank test showing below, we can select significant regions of wavenumbers. We already use sumby 4, so there should be $\frac{782}{4} = 182$ wavenumbers. According to Bonferroni correction, the criterion of P-value is equal to 5% divided by 182, which is nearly 0.0002. So we only consider all the inner-difference and intra-difference with selected wavenumbers regions whose P-value<0.0002

One thing should be noticed is that the selected significant wavenumbers are different for different datasets.

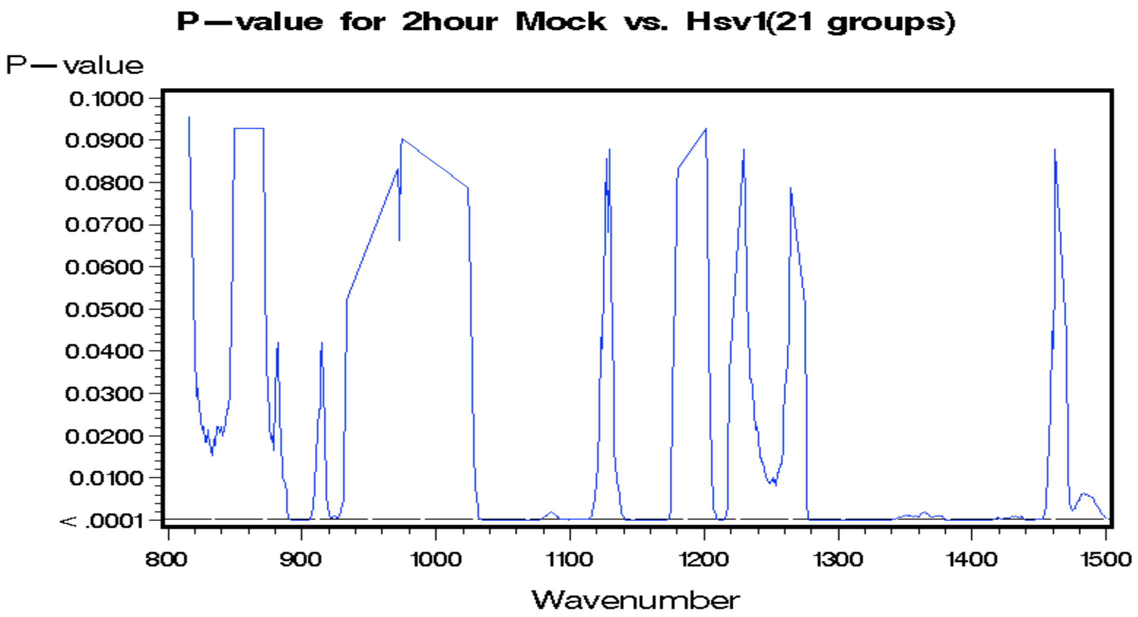


Figure 2. P-value for 2 hour Mock vs. Hsv1

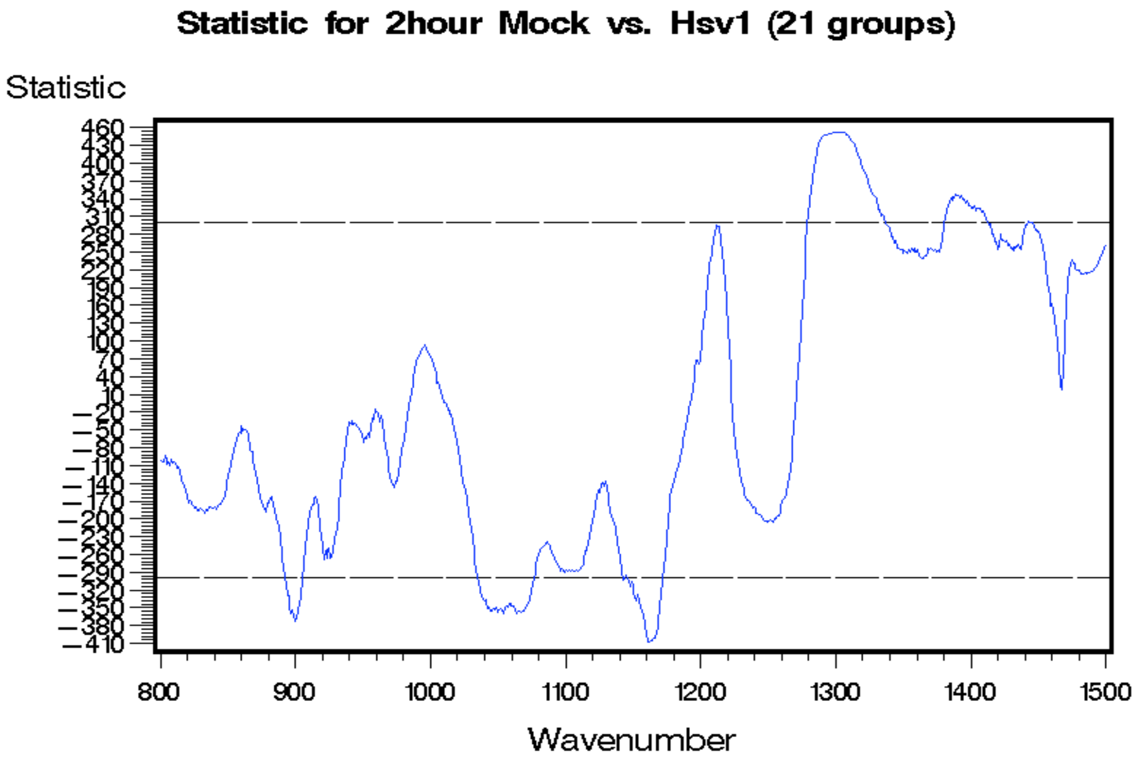


Figure 3. Statistic value for 2 hour Mock vs. Hsv1

2.3 Model with Positive Terms Minus Negative Terms

We denote

$$Y=\sum_{Positive\ range}(INT)-\sum_{Negative\ range}(INT)$$

$$X=\sum_{Positive\ range}(INN)-\sum_{Negative\ range}(INN)$$

as our discriminating statistic. In practice, we do not know if it is X or Y. A pre-assigned cutoff point will determine if it is Hsv1 or Mock. In other words, we constructed a linear-combination model with all the coefficients equal to 1 and -1.

Before moving to the next step, we need to make sure if we can combine Mock1-Mock2 and Hsv1-Hsv2 as inner differences. We check the equal variances between Mock1-Mock2 and Hsv1-Hsv2 by F-test, and find no evidence of unequal variances.

The relationship between Inner-difference of Mock and Inner difference of Hsv is verified by checking their variance first via F-test. The null hypothesis is constructed that the variance of the two groups (Mock & Hsv) is equal. If the result of F-test is significant, it may be needed to find out some other methods; if it is not significant, the null hypothesis can be accepted.

2.4 Partial Least Square Regression

Before PLS-regression, we would like to briefly talk about the Principle Component Regression (PCR), which explains the variance-covariance matrix by a set of fewer linear combinations of variables that take more weights. PCR depends solely on the covariance matrix $\sum$ (or the correlation matrix ρ) of $X_1, X_2, ..., X_p$.

At the first step,

The first principal component p1 = linear combination with maximum variance subject to $a_1^{'}a_1$.

At the second step,

The second principal component p2 = linear combination with maximum variance subject to $a_2^{'}a_2$.

……

At the ith step,

The ith principal component pi = linear combination with maximum variance subject to $a_i^{'}a_i$

Partial Least Square Regression is an extension use of the multiple linear regression model. Multiple Linear Regression may suffer over-fitting problems---when the number of factors get too large, the model can fit the sample data well but with high prediction errors. In this case, PLS could avoid this problem by extracting latent factors, which account for most of the variations in the response value.

Principal components regression and partial least squares regression differ in the methods used in extracting factor. PCR only generates matrix that will reflect the covariance character among the predictor variables, while PLS generates matrix reflecting the covariance character between the predictor and response variables. Actually, PCR is a special case of PLSR. This is the reason why we choose PLS, instead of PCR for our study.

PLS model can be defined as

$$Y = \sum_{i=1}^{n} a_i x_i \ ,$$

where $x_i, i = 1, \cdots, n$ are factors in the PLS model while $a_i, i = 1, \cdots n$ are coefficients of independent variables.

Unlike another linear-combination model with coefficients all equal to 1 or -1, we build PLS model with coefficient not all equal to 1 or -1. It is obvious that PLS model will give us more accurate coefficients in the linear model. The reason we still consider linear-combination model with coefficients all equal to 1 or -1 is that it may provide a better shrinkage, which will be discussed later.

2.5 Generating Multivariate Normal Distribution

An easy way to generate multivariate normal distribution is Cholesky Decomposition. Basically, the Cholesky Decomposition is to decompose a symmetric positive-definite matrix into the product of a lower triangular matrix and its conjugate transpose. Since **M** is a symmetric positive definite matrix, it can be decomposed as

$$M = D^{'}D$$

where **D** is a lower triangular matrix with positive diagonal entries, and $D^{'}$ denotes the conjugate transpose of **D**. This factorization of M is called the Cholesky decomposition. Another point we should pay attention to is that Cholesky decomposition is unique: given a positive-definite matrix **M**, there will be only one triangular matrix **D** corresponding to M such that $M = D^{'}D$.

2.6 Compute Specificity and AUC of Two Normal Distributions

According to Central Limit Theorem, we have assumed that both inner-differences and intra-differences have normal distributions. In the next step, we want to use the area under the Receiver Operating Characteristic (AUC), specificity and sensitivity to evaluate the model we have built.

The following graph shows two normal distributions, without disease-Mock and with disease-Hsv1. We use AUC, specificity and sensitivity to discriminate those two types of cells.

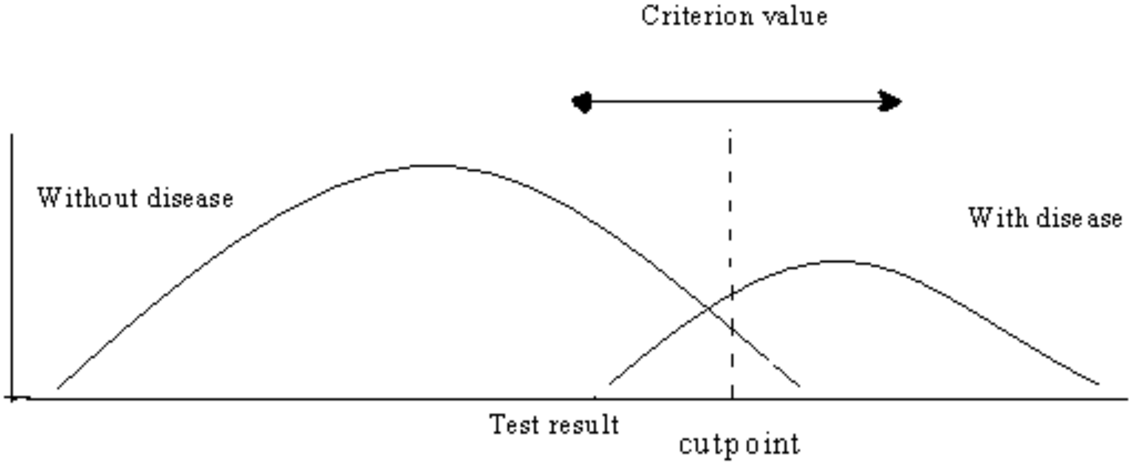


Figure 4. Two normal distributions

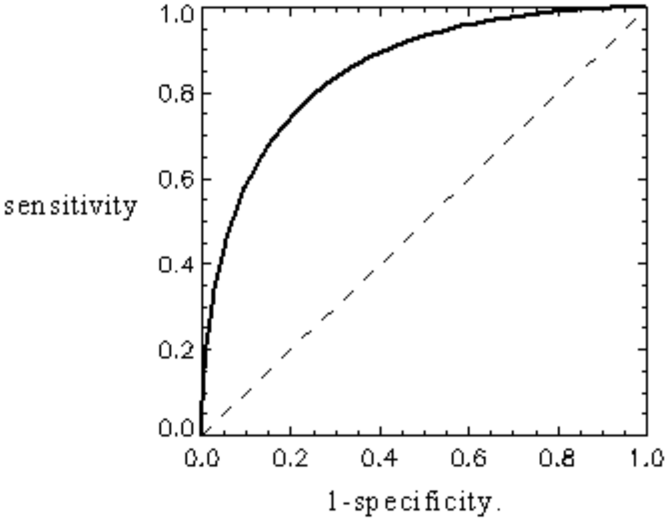


Figure 5. ROC Curve

From the graph above, we know that Receiver Operating Characteristic (ROC), is a plot of sensitivity against 1-specificity for different possible cut-off points in a specified model, where

Sensitivity = P (correct diagnosis among all positives)

Specificity = P (correct diagnosis among all negatives)

Both the range for specificity and sensitivity is from 0 to 1. Generally speaking, for a specified model, the larger the sensitivity is, the smaller the specificity will be. Because AUC is fixed, we usually improve sensitivity by sacrificing specificity.

A rough criterion to evaluate AUC for discrimination is:

(1) Excellent discrimination: $0.9 < AUC < 1$

(2) Good discrimination: $0.8 < AUC < 0.9$

(3) Fair discrimination: $0.7< AUC < 0.8$

(4) Poor discrimination: $0.6 < AUC < 0.7$

The sensitivity and specificity in our study are defined as:

Sensitivity = the probability of correct diagnosis for the Hsv1 population,

and

Specificity = the probability of correct diagnosis for the Mock population.

We only considered the specificity with sensitivity equal to 95%, 90% and 80%, respectively.

Recall part 2.3, we already know that

$$Y=\sum_{Positive\ range}(INT) - \sum_{Negative\ range}(INT), \qquad (1)$$

and

$$X=\sum_{Positive\ range}(INN) - \sum_{Negative\ range}(INN) \qquad (2)$$

Since both X and Y are normally distributed (Central Limit Theorem), AUC can be computed as

$$AUC = P\ (Y>X) = P(Y\text{-}X>0) \qquad (3)$$

$$E\ (Y\text{-}X) = \mu_Y\text{-}\mu_X$$

$$Var\ (Y\text{-}X) = \sigma_Y^2 + \sigma_X^2 \qquad (4)$$

So we have:

$$\frac{(Y-X)-(\mu_Y-\mu_X)}{\sqrt{\sigma_Y^2+\sigma_X^2}} \sim \text{N }(0,1) \text{ (standard normal distribution).}$$

Hence,

$$\text{AUC}=1 - \Phi(\frac{-(\mu_Y-\mu_X)}{\sqrt{\sigma_Y^2+\sigma_X^2}}), \qquad (5)$$

Sensitivity with cutoff point c is $P(Y>c) = P(\frac{Y-\mu_Y}{\sigma_Y}>\frac{C-\mu_Y}{\sigma_Y})=1- \Phi(\frac{C-\mu_Y}{\sigma_Y})$, (6)

Specificity with cutoff point c is $P(X<c)= P(\frac{X-\mu_X}{\sigma_X}<\frac{C-\mu_X}{\sigma_X})=\Phi(\frac{C-\mu_X}{\sigma_X})$, (7)

where Φ is the distribution function of the standard normal distribution.

The estimated AUC, sensitivity and specificity can be obtained by replacing μ_Y, μ_X, σ_Y, σ_X with the estimated ones $\bar{X}$,$\bar{Y}$, S_x, $S_{y.}$

2.7 Construct Confidence Interval by Parametric Bootstrap Method

The reason for building a confidence interval is that we wanted to see the range of specificity and AUC, although we already had their value. We will discuss parametric bootstrap method.

Nonparametric bootstrap simulates bootstrap sample that are independent and identically distributed from empirical distribution while parametric bootstrap simulates bootstrap sample from estimated parametric model.

Instead of drawing and random sampling with replacement from the original population dataset, bootstrap method uses the existing sample having an approximating distribution from the original dataset as a population, and draw random samples from this population. We can estimate the difference between the sample characters and the population characters through bootstrap samples. Any bootstrap sample can be represented by

$$\{(x_{i1}^*, x_{i2}^*, y_{i1}^*, y_{i2}^*)|\ \text{i}=1, \ldots,17\},$$

where $(x_{i1}^*, x_{i2}^*, y_{i1}^*, y_{i2}^*)$ are from a multivariate normal distribution with mean vector and variance-covariance matrix we already computed.

Using bootstrap method, we simulated 1000 sample dataset whose distributions are similar to the existing sample which we treated as the population. Then, we continue the following two steps to get the confidence interval for AUC and specificity with sensitivity equal to 95%, 90% and 80%, respectively, maybe obtained as follows.

(1) Compute $\bar{X}_i^*$, $(S_{x_i}^*)^2$, $\bar{Y}_i^*$, $(S_{y_i}^*)^2$ from this bootstrap sample.

(2) Compute $l_i^* = \frac{\bar{X}_i^* - \bar{Y}_i^*}{\sqrt{(S_{x_i}^*)^2 + (S_Y^*)^2}}$

$q_{i(\alpha)}^* = \frac{\bar{X}_i^* - \bar{Y}_i^* - Z_\alpha * S_{x_i}^*}{S_{y_i}^*}$, for α=0.5, 0.1, and 0.2

We first find the cutoff point c_1, c_2, c_3 for three specified sensitivities 95%, 90% and 80%, respectively. Then we calculated the three corresponding specificities by substituting these three cutoff point values c_1, c_2, c_3.

After repeating N times (we select N to be 1000), we obtain 2.5th and 97.5th quartile of

$\begin{cases} l_i^* \\ q_{i(\alpha)}^* \end{cases}$, say $\begin{cases} l_{2.5},\ l_{97.5} \\ q_{2.5(\alpha)},\ q_{97.5(\alpha)} \end{cases}$

The 95% C.I. for AUC is:

[P (Z>-$l_{2.5}$), P (Z>-$l_{97.5}$)]

The 95% C.I. for specificity at sensitivity=1-α is:

[P (Z<$q_{2.5(\alpha)}$), P (Z<$q_{97.5(\alpha)}$)]

Detailed computations are as follows:

Step I : Generate the multivariate normal distribution 21 times

Since the discriminator has a normal distribution for the INT, we can use these 42 INTs to estimate the mean and the standard deviation of the normal distribution. Similarly, we can use 42 INNs to estimate the mean and standard deviation of its normal distribution.

To generate a bootstrap sample, we generate 4-variate normal vectors using Cholesky decomposition method. The detailed generating part will be skipped here.

Step II: Compute mean and standard deviation & specificity and AUC

From the two normal date set we simulate, one is inner group (x), the other one is intra group (y), we compute mean of x ($\hat{\mu}_1$), std of x ($\hat{\sigma}_1$), mean of y ($\hat{\mu}_2$), std of y ($\hat{\sigma}_2$).

Using $\hat{\mu}_1$, $\hat{\mu}_2$, $\hat{\sigma}_1$, $\hat{\sigma}_2$, we can compute specificities corresponding to 3 specified sensitivities and the AUC.

We denote the specificity corresponding to sensitivity 95% as sp1. Similarly, sp2 is for 90% and sp3 is for 80%.

AUC is also computed.

Step III: Repeat step I and step II 1000 times

After repeating step I and step II 1000 times, we obtain 1000 of sp1, 1000 of sp2, 1000 of sp3 and 1000 of AUC.

We rank the 3 groups of 1000 specificities and 1000 AUCs from the smallest to the largest, respectively, i.e. we will have four ordered arrays with 1000 each. The bootstrap confidence intervals can be read through these arrays. For instance, denote 1000 ordered bootstrap AUCs as $\{A_1, A_2 \ldots A_{1000}\}$. Then a 95% confidence interval for AUC is

$((A_{25} + A_{26})/2, (A_{974} + A_{975})/2)$. Other confidence intervals can be read in a similar fashion.

2.8 K-Fold Cross-Validation

In the former parts, we have discussed linear-combination with coefficients all equal to 1 and -1. Comparing with this linear-combination regression, PLS have better specificity, AUC and their confidence interval. We need to estimate the shrinkage of all methods. The final estimates of AUC and Specificities at various sensitivity levels can be obtained by the original estimates subtract the estimated shrinkages. We are using k-fold cross-validation to estimate the shrinkages.

K-fold cross-validation can be explained as follows with k=3. We first randomly split the original dataset into 3 equal parts. Here, inner-differences and intra-differences constitute the original dataset. We still use Mock vs. Hsv1 as an example, there are a total of 21 date groups. After randomly split them into 3 equal subsets, each subset should have 7 date groups. Given the fact that each date group contains 2 intra-differences and 2 inner-differences, there are 42 intra-differences and 42 inner-differences. So each subset should have 14 intra-differences and 14 inner-differences. We also need to point out that intra-differences and inner-differences at the same date are assigned in the same group out of 3.

We use subset 1, subset 2 and subset 3 to represent these three subsets. Within these three subsets, we randomly select two of them as training dataset, e.g., subset 1 and subset 2; the other one is the validation dataset, e.g. subset 3. The estimates from training datasets subtract the validated estimates from validation datasets will be used as the estimates of the shrinkage.

The procedure to build the model from training datasets will be exactly the same as how we build the original model, which went through Wilcoxon Signed-rank test and Partial Least Square or simply using sum of positives subtract sum of negatives.

In this 3-fold cross-validation for our study, we repeat the split 100 times. Each time, there should be 3 shrinkages. So the final results should contain 300 shrinkages of AUC and others. The average value of the shrinkages will be used as the estimates of the shrinkages.

Notice that the k-fold cross-validation estimates of the shrinkages are conservative. This is because our training data size is only (k-1)/k of the original sample size, and shrinkage usually decrease as the sample size increase. Other estimates, such as bootstrap method can also be sued, which may under estimate the shrinkages. Therefore, we used k-fold cross-validation method. The details of these resampling methods will be discussed in Chapter III.

The shrinkage of specificity and AUC are computed by the k-fold cross-validation. In order to get the right specificity and AUC, we should use the original specificity and AUC of the whole original dataset after subtracting the shrinkage. The result is the final step we want.

Chapter 3

Calculation and Results

3.1 Overview

From the data description, we know that there are totally 21 paired comparisons for Mock vs. Hsv1, 20 paired comparisons for Mock and Adeno, 20 paired comparisons for Hsv1 and Adeno, 17 paired comparisons for Adeno and Coxsackie, 18 paired comparisons for Hsv1 and Adeno, 18 paired comparisons for Mock and Coxsackie corresponding to date.

The original data for 2 hour Mock, Hsv1, Adeno and Coxsackie are shown in Graph. The vertical coordinate is the absorbance while the horizontal coordinate is the wavenumber. In this plot, Mock is in blue color, Coxsackie in red, Hsv1 in green and Adeno in yellow. It seems that no wavenumbers with their absorbance can discriminate among those four cells.

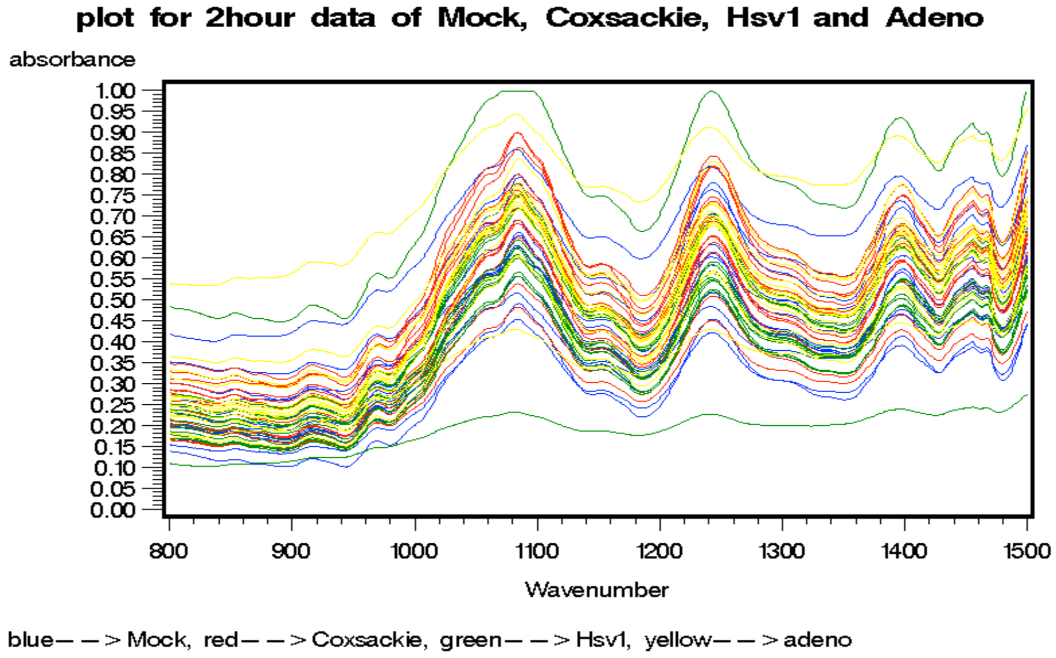


Figure 6. Absorbance of original data

After taking average, the graph of average value is as follows, the significant wavenumber regions are still unclear.

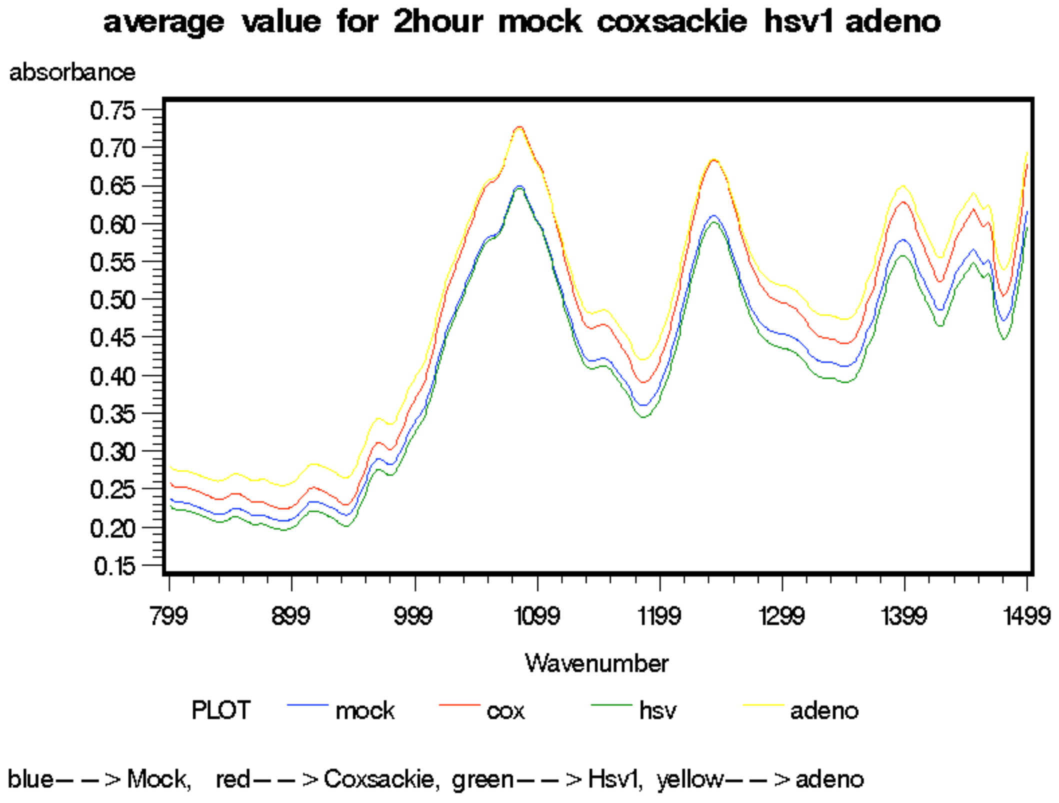


Figure 7. Average line of absorbance

3.2 F-Test

Since we combine Mock minus Mock vs. Hsv1 minus Hsv1 as the inner group, we need to check if they have the same normal distribution. Both means are zero. Therefore, all we need to check if two have the same variance. Two equal sample variances test is performed. The ANOVA table is shown in Table 1.

Because P-value = 0.9051, not significant, we can accept the Null hypothesis that the variances between inner differences of Mock vs. Hsv1 groups are the same.

Table 1. F-test to check consistence

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	0.03666017	0.03666017	0.01	0.9051
Error	22	55.44635685	2.52028895		
Corrected Total	23	55.48301702			

3.3 Model with Coefficients Equal to 1 or -1

3.3.1 Plot of Wilcoxon Signed-Rank Test for Mock vs. Hsv1

Let us still keep Mock vs. Hsv1 as an example to describe our methods, and similar statistical methods and data analysis are applied to other paired comparisons, including Mock and Coxsackie, Mock and Adeno, Coxsackie and Adeno, Coxsackie and Hsv1, Adeno and Hsv1.

Figure 8 is the standardized data for the first date group of 6 hour Mock vs. Hsv1 pair comparison. The blues lines represent Mock while red lines represent Hsv1.

After we standardized the original data, we obtained the inner-difference and intra-difference after each date group being randomly split into two subgroups. For the intra-difference, the Wilcoxon signed rank test is employed to select the significant regions, in which Mock vs. Hsv1 can be distinguished. Figures 9 and 10 are drawn from Wilcoxon signed rank test. We select regions with p-value smaller than 0.0001. Then, we set coefficients equal to 1to the regions with Signed rank test value larger than 200 and -1 to the regions with Signed rank test value smaller than -200.

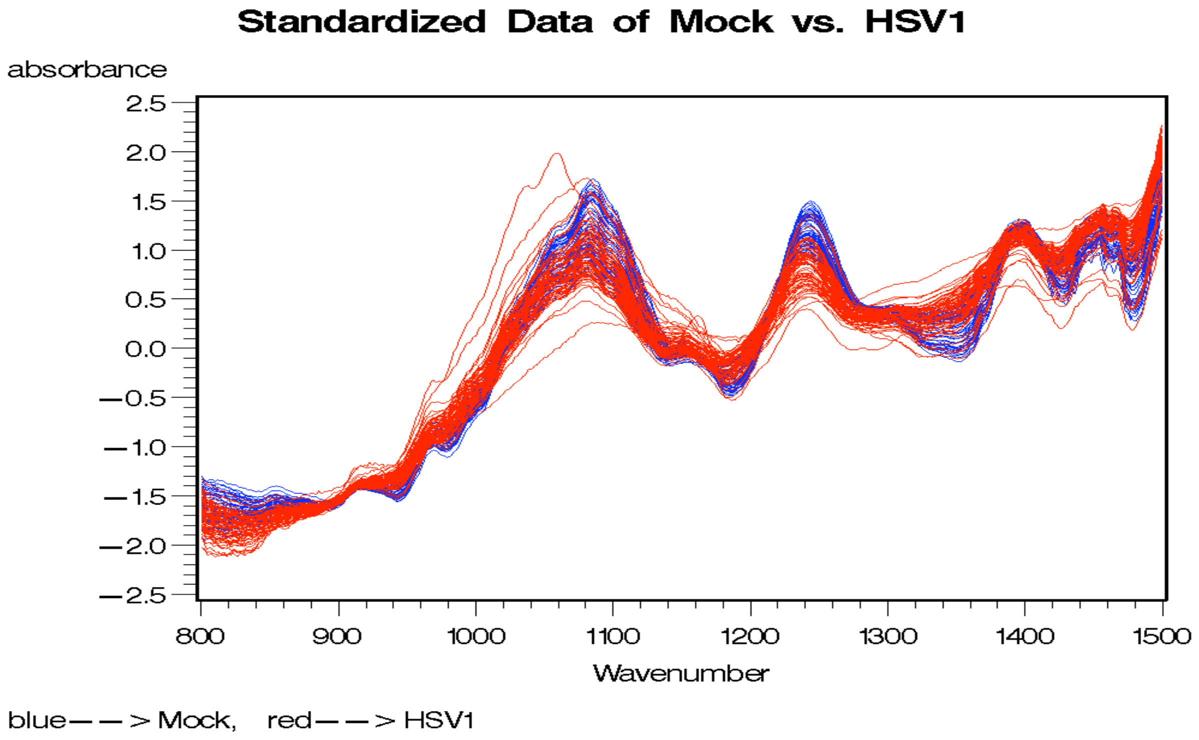


Figure 8. Standardized data of Mock vs. Hsv1 in first date group

The p-value and signed rank test value plot for 2 hour Mock vs. Hsv1 are as follows:

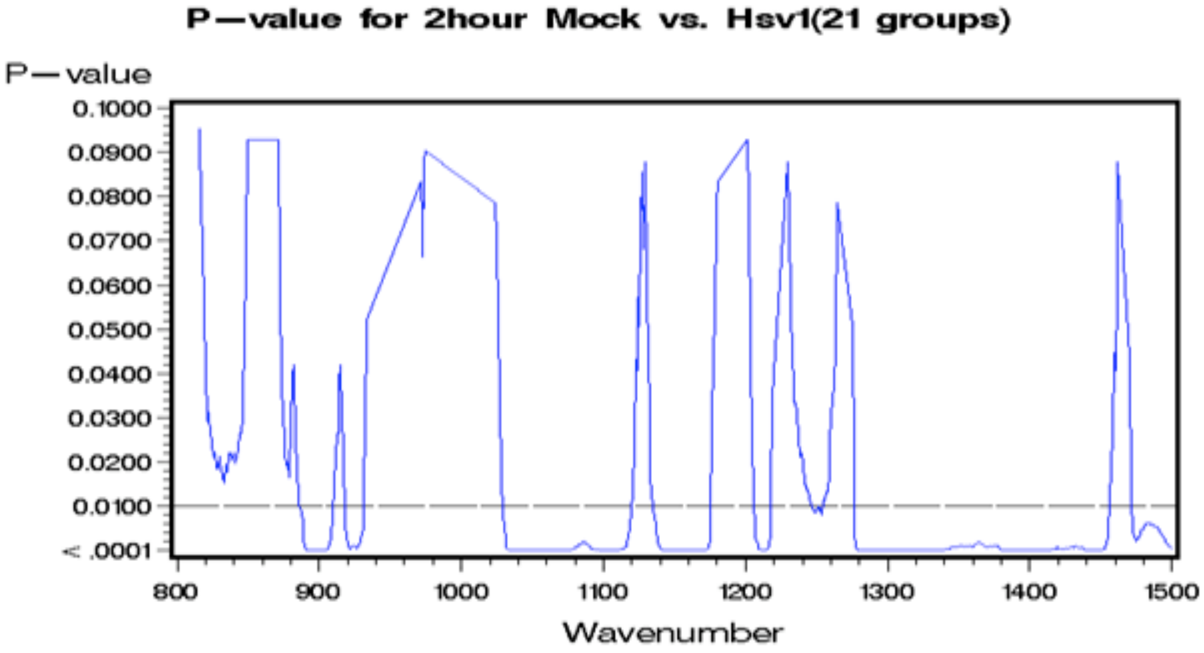


Figure 9. P-value for 2 hour Mock vs. Hsv1

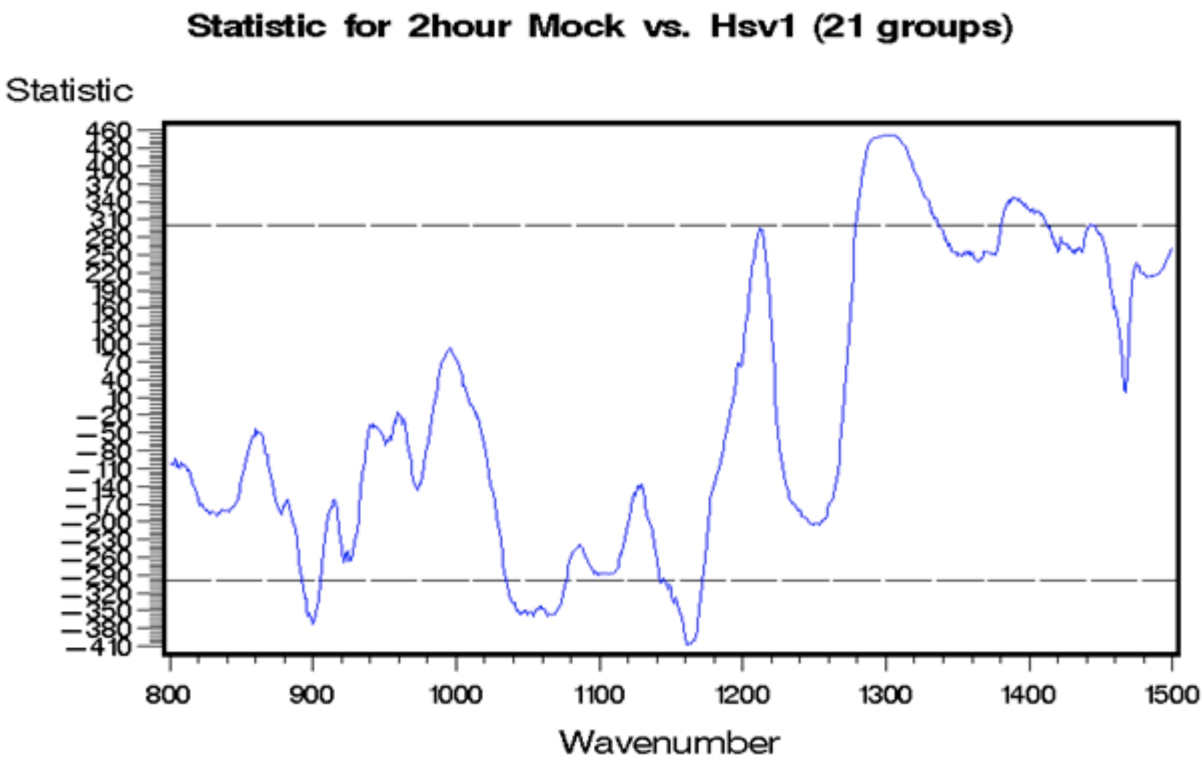


Figure 10. Statistic value for 2 hour Mock vs. Hsv1

From figures 9 and 10, the significant positive regions for 2 hour Mock vs. Hsv1 are 1279-1336 cm^{-1} and 1381-1411 cm^{-1} while the significant negative regions for 2 hour Mock vs. Hsv1 are 893-905 cm^{-1}, 1034-1077 cm^{-1} and 1145-1171 cm^{-1}.

(2)

The p-value and signed rank test value plot for 4 hour Mock vs. Hsv1 are shown in figures 11 and 12.

From figures 11 and 12, the significant positive regions for 4 hour Mock vs. Hsv1 are 1208-1218 cm^{-1}, 1270-1330 cm^{-1}, 1417-1451cm^{-1} while the significant negative regions for 4 hour Mock vs. Hsv1are 1032-1114 cm^{-1}.

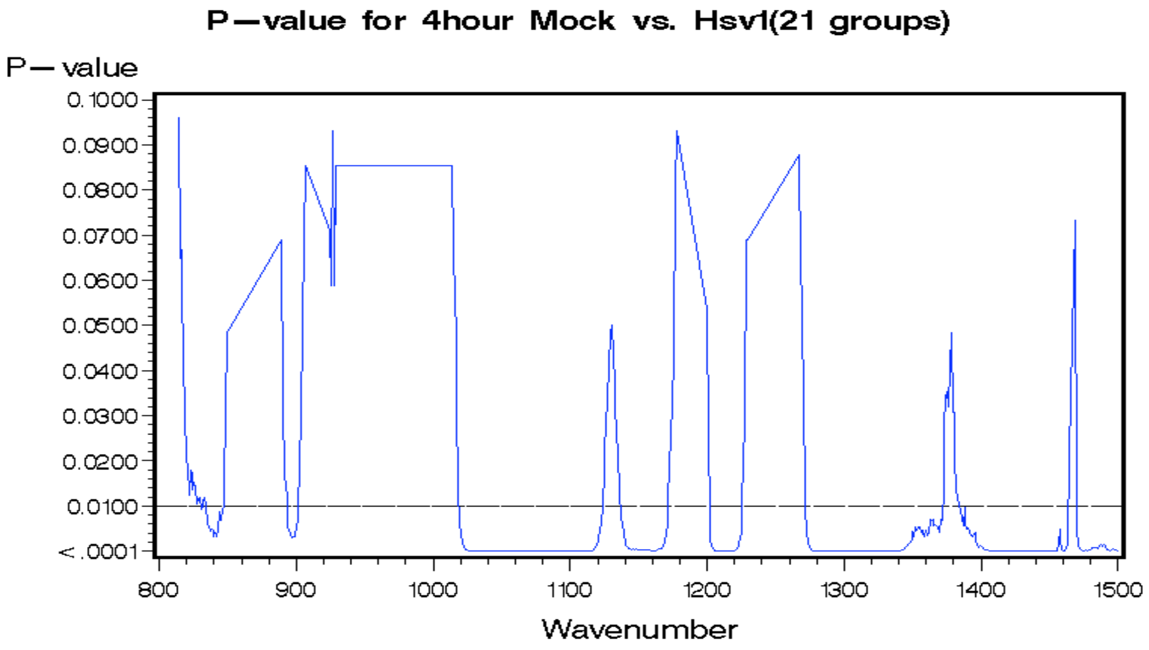


Figure 11. P-value for 4 hour Mock vs. Hsv1

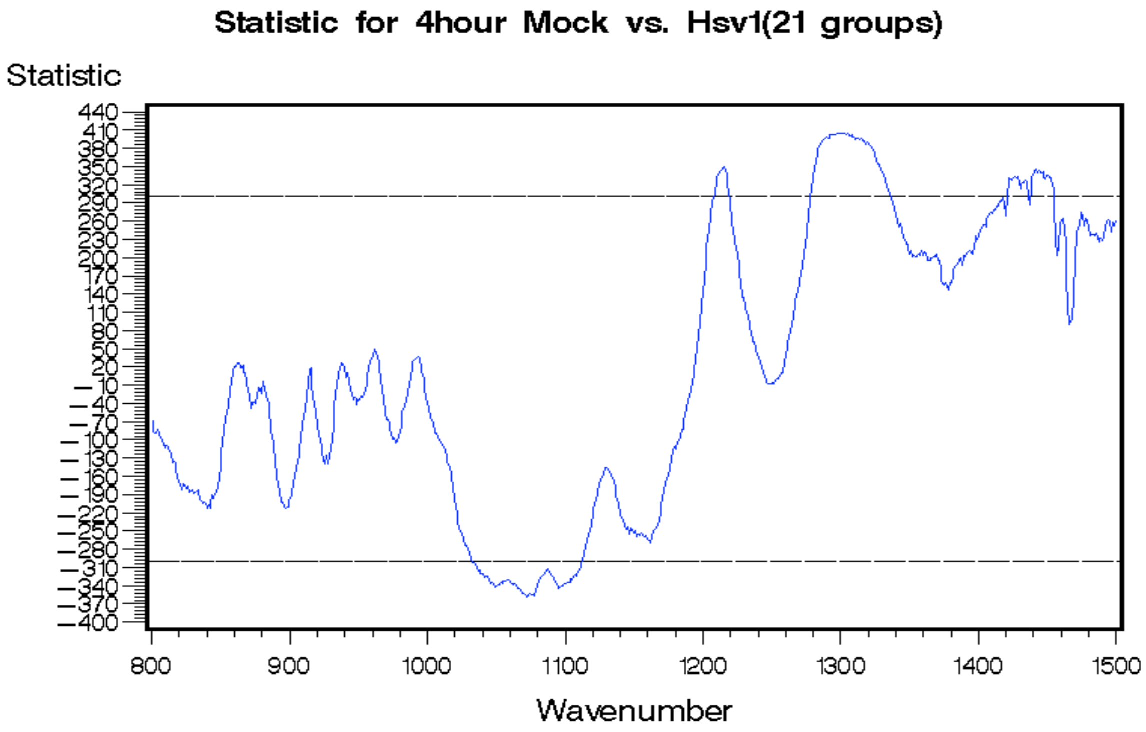


Figure 12. Statistic value for 4 hour Mock vs. Hsv1

(3)

For figures 13 and 14, the significant positive regions for 6 hour Mock vs. Hsv1 are 1205-1231 cm^{-1}, 1260-1327 cm^{-1} while the significant negative regions for 6 hour Mock vs. Hsv1 are 1045-1078 cm^{-1}, 1096-1105cm^{-1}, 1126-1167cm^{-1}.

The p-value and signed rank test value plot for 6 hour Mock vs. Hsv1 are as follows:

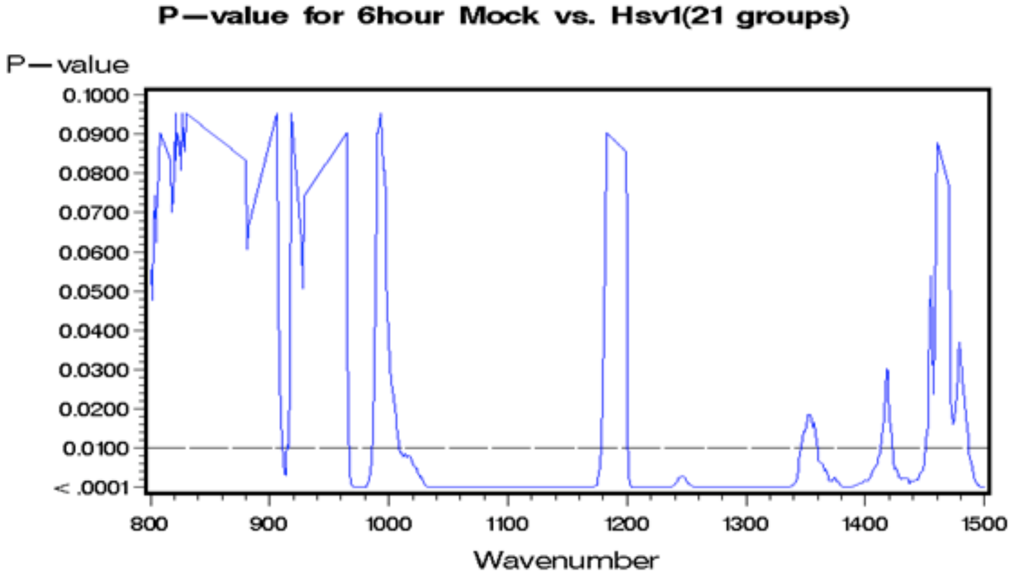


Figure 13. P-value for 6 hour Mock vs. Hsv1

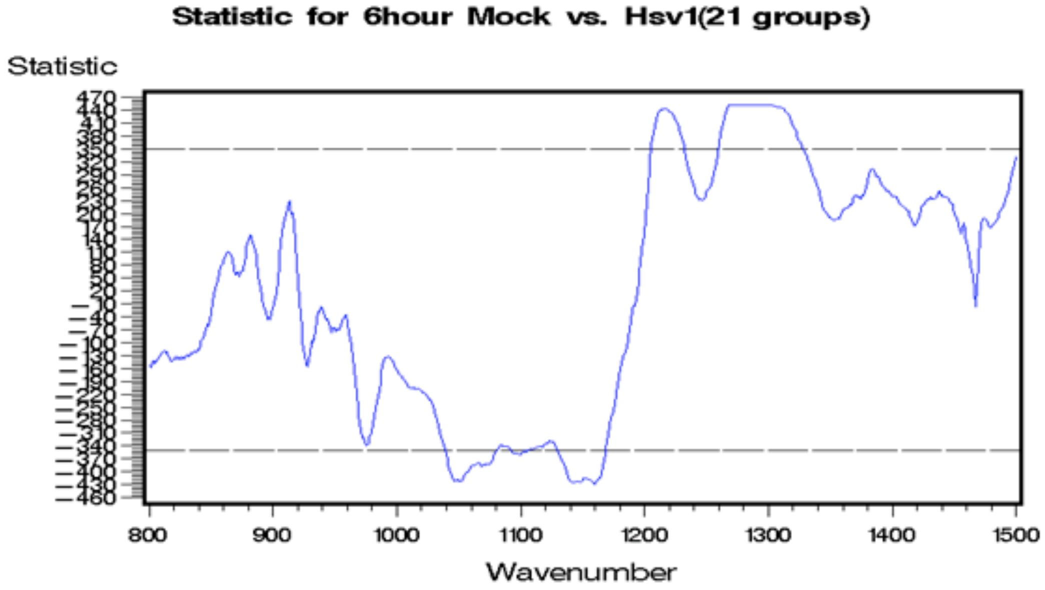


Figure 14. Statistic value for 6 hour Mock vs. Hsv1

3.3.2 Selected Significant Regions from Wilcoxon Signed-Rank Test

Using Wilcoxon Signed Rank Test, the selected variables represent positive regions and negative regions for 6 hour Mock vs. Hsv1 are collected in the following tables, respectively

Table 2. Selected variables for 6 hour Mock vs. Hsv1 in positive regions

Obs	VarName	Test	Testlab	Stat	pType	pValue
1	t106	Signed Rank	S	364.5	Pr >= |S|	<.0001
2	t107	Signed Rank	S	423.5	Pr >= |S|	<.0001
3	t108	Signed Rank	S	440.5	Pr >= |S|	<.0001
4	t109	Signed Rank	S	443.5	Pr >= |S|	<.0001
5	t110	Signed Rank	S	435.5	Pr >= |S|	<.0001
6	t111	Signed Rank	S	417.5	Pr >= |S|	<.0001
7	t112	Signed Rank	S	378.5	Pr >= |S|	<.0001
8	t113	Signed Rank	S	340.5	Pr >= |S|	<.0001
9	t114	Signed Rank	S	287.5	Pr >= |S|	0.0001
10	t119	Signed Rank	S	297.5	Pr >= |S|	<.0001
11	t120	Signed Rank	S	356.5	Pr >= |S|	<.0001
12	t121	Signed Rank	S	418.5	Pr >= |S|	<.0001
13	t122	Signed Rank	S	447.5	Pr >= |S|	<.0001
14	t123	Signed Rank	S	451.5	Pr >= |S|	<.0001
15	t124	Signed Rank	S	451.5	Pr >= |S|	<.0001
16	t125	Signed Rank	S	451.5	Pr >= |S|	<.0001
17	t126	Signed Rank	S	451.5	Pr >= |S|	<.0001
18	t127	Signed Rank	S	451.5	Pr >= |S|	<.0001
19	t128	Signed Rank	S	451.5	Pr >= |S|	<.0001
20	t129	Signed Rank	S	451.5	Pr >= |S|	<.0001
21	t130	Signed Rank	S	451.5	Pr >= |S|	<.0001
22	t131	Signed Rank	S	451.5	Pr >= |S|	<.0001
23	t132	Signed Rank	S	447.5	Pr >= |S|	<.0001
24	t133	Signed Rank	S	446.5	Pr >= |S|	<.0001
25	t134	Signed Rank	S	436.5	Pr >= |S|	<.0001
26	t135	Signed Rank	S	420.5	Pr >= |S|	<.0001
27	t136	Signed Rank	S	390.5	Pr >= |S|	<.0001
28	t137	Signed Rank	S	363.5	Pr >= |S|	<.0001
29	t138	Signed Rank	S	338.5	Pr >= |S|	<.0001
30	t139	Signed Rank	S	315.5	Pr >= |S|	<.0001
31	t152	Signed Rank	S	303.5	Pr >= |S|	<.0001
32	t153	Signed Rank	S	287.5	Pr >= |S|	0.0001
33	t182	Signed Rank	S	319.5	Pr >= |S|	<.0001

Table 3. Selected variables for 6 hour Mock vs. Hsv1 in negative regions

Obs	VarName	Test	Testlab	Stat	pType	pValue
1	t45	Signed Rank	S	-296.5	Pr >= |S|	<.0001
2	t46	Signed Rank	S	-332.5	Pr >= |S|	<.0001
3	t47	Signed Rank	S	-319.5	Pr >= |S|	<.0001
4	t61	Signed Rank	S	-279.5	Pr >= |S|	0.0002
5	t62	Signed Rank	S	-323.5	Pr >= |S|	<.0001
6	t63	Signed Rank	S	-365.5	Pr >= |S|	<.0001
7	t64	Signed Rank	S	-406.5	Pr >= |S|	<.0001
8	t65	Signed Rank	S	-420.5	Pr >= |S|	<.0001
9	t66	Signed Rank	S	-421.5	Pr >= |S|	<.0001
10	t67	Signed Rank	S	-404.5	Pr >= |S|	<.0001
11	t68	Signed Rank	S	-390.5	Pr >= |S|	<.0001
12	t69	Signed Rank	S	-383.5	Pr >= |S|	<.0001
13	t70	Signed Rank	S	-385.5	Pr >= |S|	<.0001
14	t71	Signed Rank	S	-383.5	Pr >= |S|	<.0001
15	t72	Signed Rank	S	-381.5	Pr >= |S|	<.0001
16	t73	Signed Rank	S	-362.5	Pr >= |S|	<.0001
17	t74	Signed Rank	S	-344.5	Pr >= |S|	<.0001
18	t75	Signed Rank	S	-340.5	Pr >= |S|	<.0001
19	t76	Signed Rank	S	-342.5	Pr >= |S|	<.0001
20	t77	Signed Rank	S	-357.5	Pr >= |S|	<.0001
21	t78	Signed Rank	S	-358.5	Pr >= |S|	<.0001
22	t79	Signed Rank	S	-353.5	Pr >= |S|	<.0001
23	t80	Signed Rank	S	-349.5	Pr >= |S|	<.0001
24	t81	Signed Rank	S	-346.5	Pr >= |S|	<.0001
25	t82	Signed Rank	S	-346.5	Pr >= |S|	<.0001
26	t83	Signed Rank	S	-343.5	Pr >= |S|	<.0001
27	t84	Signed Rank	S	-337.5	Pr >= |S|	<.0001
28	t85	Signed Rank	S	-330.5	Pr >= |S|	<.0001
29	t86	Signed Rank	S	-350.5	Pr >= |S|	<.0001
30	t87	Signed Rank	S	-374.5	Pr >= |S|	<.0001
31	t88	Signed Rank	S	-394.5	Pr >= |S|	<.0001
32	t89	Signed Rank	S	-422.5	Pr >= |S|	<.0001
33	t90	Signed Rank	S	-423.5	Pr >= |S|	<.0001
34	t91	Signed Rank	S	-423.5	Pr >= |S|	<.0001
35	t92	Signed Rank	S	-415.5	Pr >= |S|	<.0001
36	t93	Signed Rank	S	-418.5	Pr >= |S|	<.0001
37	t94	Signed Rank	S	-427.5	Pr >= |S|	<.0001
38	t95	Signed Rank	S	-415.5	Pr >= |S|	<.0001
39	t96	Signed Rank	S	-366.5	Pr >= |S|	<.0001
40	t97	Signed Rank	S	-300.5	Pr >= |S|	<.0001

The discriminator will be built by the summarization of variables from Table minus the summarization of variables from Tables, which is

(t106+…+t114+t119+…+t139+t152+t153+t182) - (t45+…+t47+t61+…+t97)

Similarly, the selected variables represent positive regions and negative regions for 6 hour Mock vs. Coxsackie are collected in the following tables, respectively.

Table 4. Selected variables for 6 hour Mock vs. Coxsackie in positive regions

Obs	VarName	Test	Testlab	Stat	pType	pValue
1	t57	Signed Rank	S	230	Pr >= |S|	<.0001
2	t58	Signed Rank	S	260	Pr >= |S|	<.0001
3	t59	Signed Rank	S	260	Pr >= |S|	<.0001
4	t60	Signed Rank	S	242	Pr >= |S|	<.0001
5	t126	Signed Rank	S	264	Pr >= |S|	<.0001
6	t127	Signed Rank	S	272	Pr >= |S|	<.0001
7	t128	Signed Rank	S	269	Pr >= |S|	<.0001
8	t129	Signed Rank	S	266	Pr >= |S|	<.0001
9	t130	Signed Rank	S	254	Pr >= |S|	<.0001
10	t131	Signed Rank	S	248	Pr >= |S|	<.0001
11	t132	Signed Rank	S	232	Pr >= |S|	<.0001
12	t154	Signed Rank	S	225	Pr >= |S|	0.0001
13	t155	Signed Rank	S	255	Pr >= |S|	<.0001
14	t156	Signed Rank	S	264	Pr >= |S|	<.0001
15	t157	Signed Rank	S	255	Pr >= |S|	<.0001

Table 5. Selected variables for 6 hour Mock vs. Coxsackie in negative regions

Obs	VarName	Test	Testlab	Stat	pType	pValue
1	t26	Signed Rank	S	-257	Pr >= |S|	<.0001
2	t27	Signed Rank	S	-258	Pr >= |S|	<.0001
3	t28	Signed Rank	S	-263	Pr >= |S|	<.0001
4	t29	Signed Rank	S	-234	Pr >= |S|	<.0001
5	t30	Signed Rank	S	-232	Pr >= |S|	<.0001
6	t31	Signed Rank	S	-270	Pr >= |S|	<.0001
7	t32	Signed Rank	S	-305	Pr >= |S|	<.0001
8	t33	Signed Rank	S	-325	Pr >= |S|	<.0001
9	t34	Signed Rank	S	-330	Pr >= |S|	<.0001
10	t35	Signed Rank	S	-333	Pr >= |S|	<.0001
11	t36	Signed Rank	S	-333	Pr >= |S|	<.0001
12	t37	Signed Rank	S	-307	Pr >= |S|	<.0001
13	t38	Signed Rank	S	-264	Pr >= |S|	<.0001
14	t39	Signed Rank	S	-243	Pr >= |S|	<.0001
15	t40	Signed Rank	S	-239	Pr >= |S|	<.0001
16	t41	Signed Rank	S	-222	Pr >= |S|	0.0002
17	t43	Signed Rank	S	-229	Pr >= |S|	<.0001
18	t44	Signed Rank	S	-260	Pr >= |S|	<.0001
19	t45	Signed Rank	S	-272	Pr >= |S|	<.0001
20	t46	Signed Rank	S	-239	Pr >= |S|	<.0001

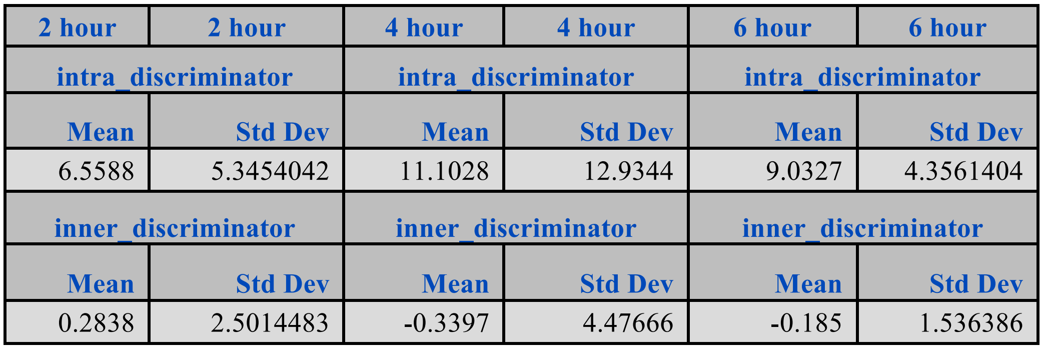
The discriminator will be built by the summarization of variables from Table minus the summarization of variables from Tables, which is

(t57+… +t60+t126+…+t132+t154+…t157) - (t26+…+t41+t43+…t46)

3.3.3 AUC and Specificities at Various Sensitivity Levels

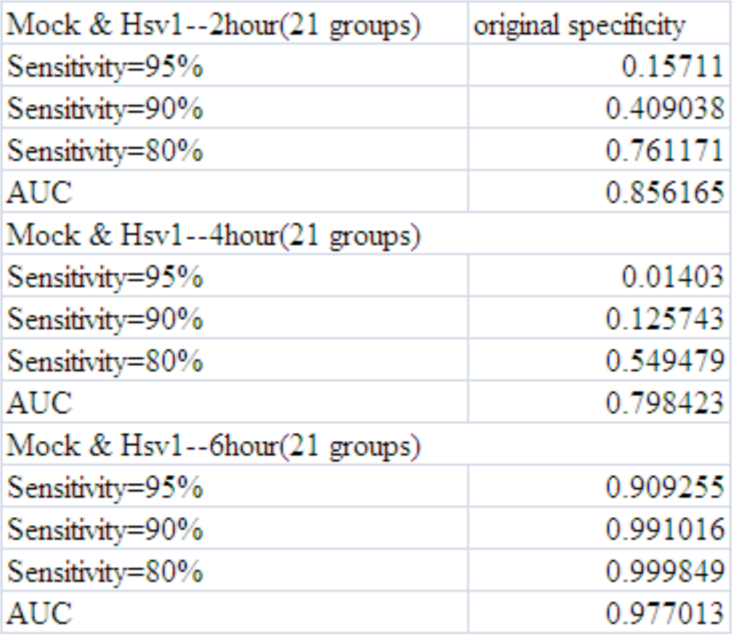
The mean and standardization of the discriminator for both inner and intra cases are as follows, we can use them to calculate specificity and AUC.

Table 6. Mean and standard deviation for intra-discriminator and inner-discriminator



2 hour	2 hour	4 hour	4 hour	6 hour	6 hour
intra_discriminator		intra_discriminator		intra_discriminator	
Mean	Std Dev	Mean	Std Dev	Mean	Std Dev
6.5588	5.3454042	11.1028	12.9344	9.0327	4.3561404
inner_discriminator		inner_discriminator		inner_discriminator	
Mean	Std Dev	Mean	Std Dev	Mean	Std Dev
0.2838	2.5014483	-0.3397	4.47666	-0.185	1.536386

Table 7. Specificity and AUC computed from intra-discriminator and inner-discriminator



Mock & Hsv1--2hour(21 groups)	original specificity
Sensitivity=95%	0.15711
Sensitivity=90%	0.409038
Sensitivity=80%	0.761171
AUC	0.856165
Mock & Hsv1--4hour(21 groups)	
Sensitivity=95%	0.01403
Sensitivity=90%	0.125743
Sensitivity=80%	0.549479
AUC	0.798423
Mock & Hsv1--6hour(21 groups)	
Sensitivity=95%	0.909255
Sensitivity=90%	0.991016
Sensitivity=80%	0.999849
AUC	0.977013

It is clear that the comparison between Mock and Hsv in 6 hour is the best because it has the largest AUC and specificity with sensitivity corresponding to 95%, 90% and 80%, respectively. Neither two hours nor four hours data discriminate well for Mock vs. Hsv1.

3.3.4 Parametric Bootstrap to Build Confidence Intervals

We adopt parametric bootstrap method to find confidence intervals for AUC and Specificities at various sensitivity levels. The procedure is illustrated by Mock vs. Hsv1 paired comparisons in 6 hour.

First, we use our sample to estimate the mean vectors and variance-covariance matrix of four variables, 2 inners and 2 intras. Let x1, x2, y1, y2 denote these variables defined as follow.

x1 = M1-M2

x2 = H1-H2

y1 = M1-H1

y2 = M2-H2

For Mock vs. Hsv1, we obtained mean [-0.3049 -0.0659 8.9132 9.1522], and variance-covariance matrix

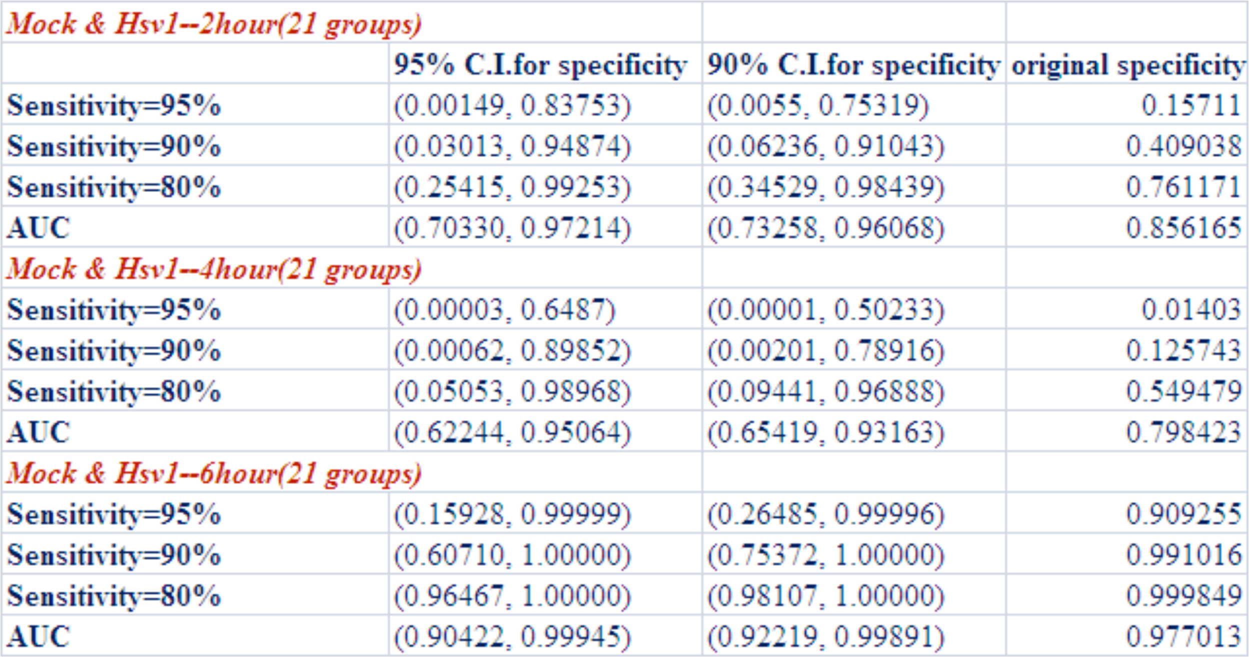
$$\begin{bmatrix} 1.2419 & -0.3459 & 2.2908 & 0.7030 \\ -0.3459 & 3.5671 & -1.1854 & 2.7276 \\ 2.2908 & -1.1854 & 20.1612 & 16.6850 \\ 0.7030 & 2.7276 & 16.6850 & 18.7096 \end{bmatrix}$$

Using Cholesky decomposition, we generated 1000 sets of vectors of (x1, x2, y1, y2) with each set containing exact 21 (our sample size for Mock vs. Hsv1) derived from a multivariate distribution with above mean vector and variance-covariance matrix.

From each simulated set, means and variances of inners and intra can be computed. Then 4 quantities (AUC, Specificities are 95%, 90% and 80% sensitivities) are obtained using formula

(1) – (7). The 2.5 and 97.5 percentiles form the desired confidence intervals. The results are shows in the following table.

Table 8. Confidence interval for specificity and AUC of Mock vs. Hsv1



Mock & Hsv1--2hour(21 groups)			
	95% C.I.for specificity	**90% C.I.for specificity**	**original specificity**
Sensitivity=95%	(0.00149, 0.83753)	(0.0055, 0.75319)	0.15711
Sensitivity=90%	(0.03013, 0.94874)	(0.06236, 0.91043)	0.409038
Sensitivity=80%	(0.25415, 0.99253)	(0.34529, 0.98439)	0.761171
AUC	(0.70330, 0.97214)	(0.73258, 0.96068)	0.856165
Mock & Hsv1--4hour(21 groups)			
Sensitivity=95%	(0.00003, 0.6487)	(0.00001, 0.50233)	0.01403
Sensitivity=90%	(0.00062, 0.89852)	(0.00201, 0.78916)	0.125743
Sensitivity=80%	(0.05053, 0.98968)	(0.09441, 0.96888)	0.549479
AUC	(0.62244, 0.95064)	(0.65419, 0.93163)	0.798423
Mock & Hsv1--6hour(21 groups)			
Sensitivity=95%	(0.15928, 0.99999)	(0.26485, 0.99996)	0.909255
Sensitivity=90%	(0.60710, 1.00000)	(0.75372, 1.00000)	0.991016
Sensitivity=80%	(0.96467, 1.00000)	(0.98107, 1.00000)	0.999849
AUC	(0.90422, 0.99945)	(0.92219, 0.99891)	0.977013

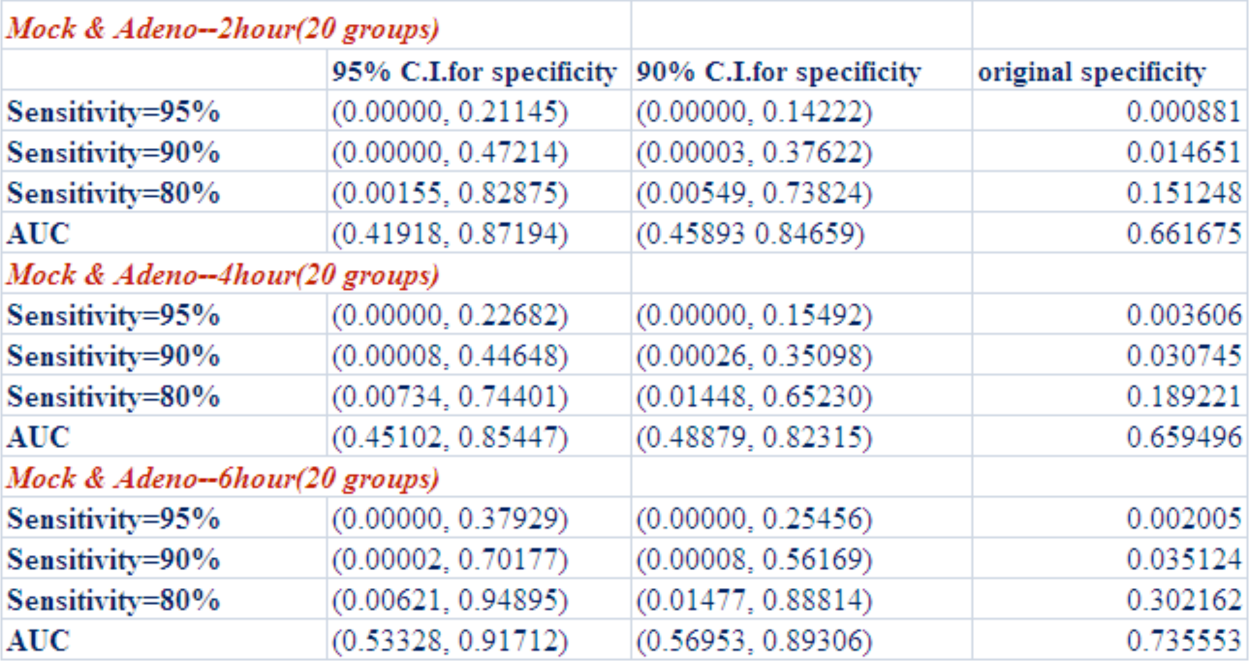
The confidence interval also stands for the result of Mock vs. Hsv1 in 6 hour is the best.

3.3.5 Specificity, AUC and Their Confidence Intervals for All Others Paired Comparisons

From Appendix, we obtain confidence intervals for others comparisons as following tables.

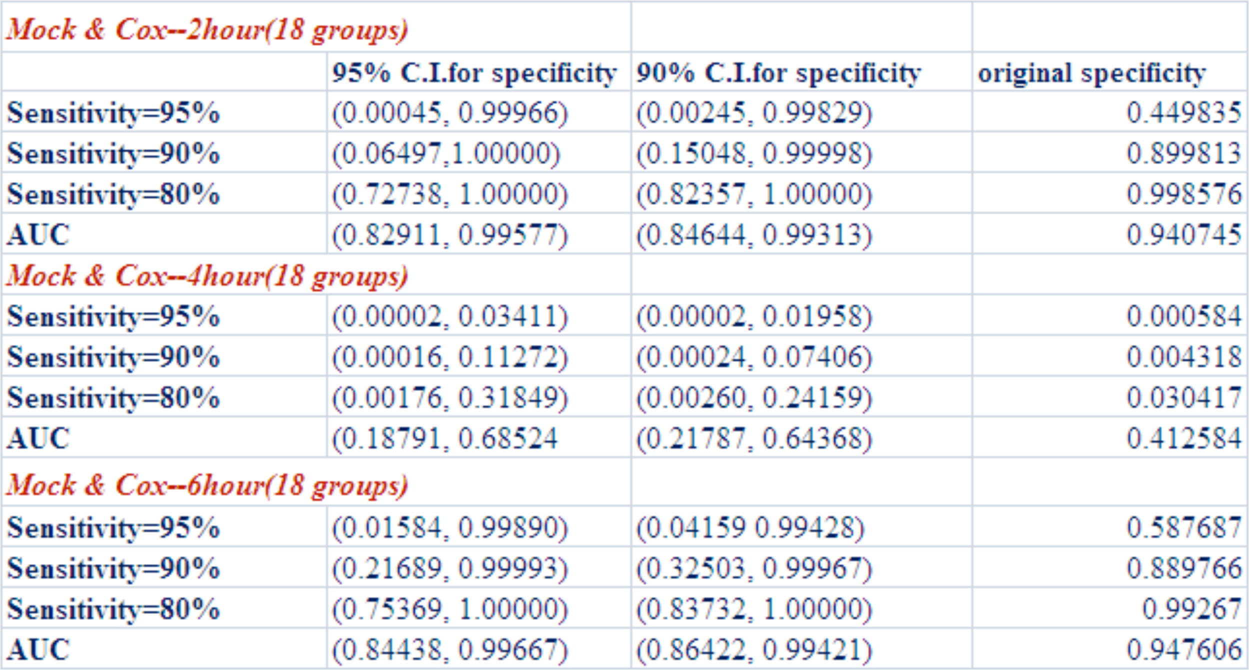
We can tell that 6 hour paired comparisons is the best in all 2 hour, 4 hour and 6 hour paired comparisons, among which, the result for Mock vs. Hsv1 has a clear discrimination while the result for Mock and Adeno is not clear.

Table 9. Specificity, AUC and 95%, 90% confidence interval for Mock and Adeno



Mock & Adeno--2hour(20 groups)			
	95% C.I.for specificity	**90% C.I.for specificity**	**original specificity**
Sensitivity=95%	(0.00000, 0.21145)	(0.00000, 0.14222)	0.000881
Sensitivity=90%	(0.00000, 0.47214)	(0.00003, 0.37622)	0.014651
Sensitivity=80%	(0.00155, 0.82875)	(0.00549, 0.73824)	0.151248
AUC	(0.41918, 0.87194)	(0.45893 0.84659)	0.661675
Mock & Adeno--4hour(20 groups)			
Sensitivity=95%	(0.00000, 0.22682)	(0.00000, 0.15492)	0.003606
Sensitivity=90%	(0.00008, 0.44648)	(0.00026, 0.35098)	0.030745
Sensitivity=80%	(0.00734, 0.74401)	(0.01448, 0.65230)	0.189221
AUC	(0.45102, 0.85447)	(0.48879, 0.82315)	0.659496
Mock & Adeno--6hour(20 groups)			
Sensitivity=95%	(0.00000, 0.37929)	(0.00000, 0.25456)	0.002005
Sensitivity=90%	(0.00002, 0.70177)	(0.00008, 0.56169)	0.035124
Sensitivity=80%	(0.00621, 0.94895)	(0.01477, 0.88814)	0.302162
AUC	(0.53328, 0.91712)	(0.56953, 0.89306)	0.735553

Table 10. Specificity, AUC and 95%, 90% confidence interval for Mock and Coxsackie



Mock & Cox--2hour(18 groups)			
	95% C.I.for specificity	**90% C.I.for specificity**	**original specificity**
Sensitivity=95%	(0.00045, 0.99966)	(0.00245, 0.99829)	0.449835
Sensitivity=90%	(0.06497,1.00000)	(0.15048, 0.99998)	0.899813
Sensitivity=80%	(0.72738, 1.00000)	(0.82357, 1.00000)	0.998576
AUC	(0.82911, 0.99577)	(0.84644, 0.99313)	0.940745
Mock & Cox--4hour(18 groups)			
Sensitivity=95%	(0.00002, 0.03411)	(0.00002, 0.01958)	0.000584
Sensitivity=90%	(0.00016, 0.11272)	(0.00024, 0.07406)	0.004318
Sensitivity=80%	(0.00176, 0.31849)	(0.00260, 0.24159)	0.030417
AUC	(0.18791, 0.68524	(0.21787, 0.64368)	0.412584
Mock & Cox--6hour(18 groups)			
Sensitivity=95%	(0.01584, 0.99890)	(0.04159 0.99428)	0.587687
Sensitivity=90%	(0.21689, 0.99993)	(0.32503, 0.99967)	0.889766
Sensitivity=80%	(0.75369, 1.00000)	(0.83732, 1.00000)	0.99267
AUC	(0.84438, 0.99667)	(0.86422, 0.99421)	0.947606

3.4 Partial Least Square Regression

Since Partial Least Squares (PLS) method is widely used in the discrimination analysis, we would like to try PLS for our data analysis. We are not directly using PLS. The procedure is explained in two steps. Selecting regions with Wilcoxon Signed Rank Test is still used as the first step. In the second step, we use PLS on the regions selected in the first step.

The following three tables show the means and the standard deviations of the intra and inner values for 2, 4, and 6 hours data. Using formula (1) – (7), the specificities at 3 sensitivity levels and AUCs are all equal to 1. We found except Mock vs. Adeno, all other comparisons yield 1.

Table11. Mean and standard deviation 6 hour Mock vs. Hsv1

Intra-values		**Inner- values**	
Mean	**Std Dev**	**Mean**	**Std Dev**
0.9874	0.0845084	0.0126	0.0734869

Table 12. Mean and standard deviation 2 hour Mock vs. Hsv1

Intra- values		**Inner- values**	
Mean	**Std Dev**	**Mean**	**Std Dev**
0.9697582	0.1217444	0.0302418	0.1195155

Table13. Mean and standard deviation 4 hour Mock vs. Hsv1

Intra- values		**Inner- values**	
Mean	**Std Dev**	**Mean**	**Std Dev**
0.9821457	0.0987069	0.0178543	0.0888459

3.5 K-Fold Cross-Validation

3.5.1 3-Fold Cross-Validation on Results Derived from PLSR

As we discussed before, the different coefficients in the PLSR model mainly account for the shrinkage in k-fold cross-validation. Different from model with positive terms minus negative terms, all coefficients of which are equal to 1 or -1, the coefficients in PLSR model will vary in wide range.

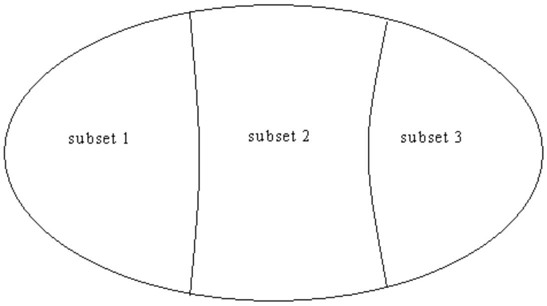


Figure 15. 3-fold cross-validation

The following table is the results for 6 hour 3-fold cross-validation of PLSR.

Table14. Shrinkage for 6 hour 3-fold cross-validation of PLSR

Obs	Mock_Hsv	Mock_Cox	Adeno_Hsv	Adeno_Cox	Hsv_Cox
shrinkage(95% sensitivity)	0.36322	0.36847			
shrinkage(90% sensitivity)	0.32179	0.32676			
shrinkage(80% sensitivity)	0.2748	0.27974			
shrinkage of AUC	0.2082	0.2135	0.295749	0.199987	0.565121

3.5.2 3-Fold Cross-Validation on Results Derived from Model with Positive Terms Minus Negative Terms

The results for 6 hour 3-fold cross-validation of model with all coefficients equal to 1 and -1 are

Table15. Shrinkage for 6 hour 3-fold cross-validation of model with positive terms minus negative terms

Obs	Mock_Hsv	Mock_Cox	Adeno_Hsv	Adeno_Cox	Hsv_Cox
shrinkage(95% sensitivity)	0.22599	0.20422			
shrinkage(90% sensitivity)	0.18567	0.28839			
shrinkage(80% sensitivity)	0.06842	0.24917			
shrinkage of AUC	0.03557	0.09911	0.08701	0.1077	0.05662

3.5.3 2-Fold Cross-Validation on Results Derived from PLSR

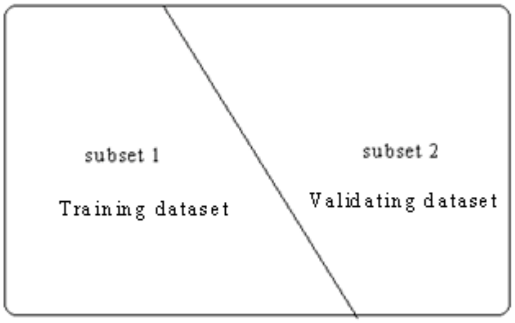


Figure 16. 2-fold cross-validation

We repeated this cross-validation process for 100 times. When calculating 2-fold cross-validation, the summarizations of all the shrinkage are divided by 200 instead of 300 in 3-fold cross-validation. The results for 6 hour 2-fold cross-validation of PLSR are shown in table 16.

Table16. Shrinkage for 6 hour 2-fold cross-validation of PLSR

Obs	Mock_Hsv	Mock_Cox	Adeno_Hsv	Adeno_Cox	Hsv_Cox
shrinkage(95% sensitivity)	0.34706	0.48378			
shrinkage(90% sensitivity)	0.30174	0.44939			
shrinkage(80% sensitivity)	0.25054	0.40294			
shrinkage of AUC	0.18050	0.32094	0.21137	0.23024	0.53793

3.5.4 2-Fold Cross-Validation on Results Derived from Model with Positive Terms Minus Negative Terms

The results for 6 hour 2-fold cross-validation of model with all coefficients equal to 1 and -1 are shown in table 17.

Table17. Shrinkage for 6 hour 2-fold cross-validation of model with positive terms minus negative terms

Obs	Mock_Hsv	Mock_Cox	Adeno_Cox	Hsv_Adeno	Hsv_Cox
shrinkage(95% sensitivity)	0.29248	0.24511			
shrinkage(90% sensitivity)	0.19191	0.36376			
shrinkage(80% sensitivity)	0.06945	0.37856			
shrinkage of AUC	0.04383	0.13935	0．13122	0．08463	0．08031

P-value of Wilcoxon Signed Rank Test nearly all larger than 0.5, this brings up a problem of no shrinkage for Mock and Adeno. There are totally 5 paired comparisons.

Chapter 4

Conclusion

Based on the final results for specificity and AUC, the 6 hour measurement is better than 2 hour measurement and 4 hour measurement. This is the reason why the 6 hour results are mainly used to explain the whole study.

However, there is one exception, the results for Mock and Adeno paired comparison is not significant, regardless of whether 2 hour, 4 hour or 6 hour data is used. As far as we know, the difference between Mock and Adeno are not easy to distinguish. It is need to do research on other new methods.

Two different regression models are used here. One is simply use sum of positive significance terms subtract the sum of negative significance terms.

The other model is Partial Least Square Regression. All the processes are the same with first method by selecting significant wavenumber regions in the first step. PLSR also uses the wavenumber selected by Wilcoxon Signed-rank Test.

In consolidate sample size, only 2-fold cross-validation is discussed here. After comparing the shrinkage of PLSR and the first method, it is clear that shrinkages of PLSR are inferior to shrinkages from the first method. As mentioned before, the coefficients will explain the shrinkage of PLSR while different variables from selected significant wavenumber regions account for the shrinkage of positive minus negative method.

Paired comparisons are employed here. Further studies will deal with longer time measurements such as 8hour, 10hour or 12 hour, tridimensional or even multidimensional comparisons. New detecting machines other than FTIR microspectroscopy maybe used in future measurements, with new measuring methods, which may give us an advanced expectation.

REFERENCES

[1] George H.Dunteman(1984). Introduction to multivariate analysis. Thousand Oaks, CA:Sage Publications.

[2] Tabachnick , Barbara G.and Linda S. Fidell (2001). Using Multivariate Statistics, 4th ed.

[3] Jardine, N. & Sibson, R.(1968). The contruction of hierarchin and non-hierarchic classifications. The computer Journal 11:177.

[4] James J.Higgins. Introduction To Modern Nonparametric Statistics.

[5] Bradley Efron. The Jackknife, the Bootstrap and Other Resampling Plans. Society for INDUSTRIAL and APPLIED MATHEMATICS (1982)

[6] W. J. Conover (1998). Practical nonparametric statistics (3rd ed).

[7] Hartigan, J. (1975). Clustering Algorithms. Wiley, New York.

[8] SAS Institute Inc. 2002-2005 SAS OnlineDoc. Version 9.1.3.

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

[10] Randall D. Tobias. An Introduction to Partial Least Squares Regression. SAS Institute Inc., Cary, NC.

[11] Shafagh Fallah, David Tritchler and Joseph Beyene, Estimating Number of Clusters Based on a General Similarity Matrix with Application to Microarray Data, Statistical Applications in Genetics and Molecular Biology, Volume 7, Issue 1, Article 24.

[12] Calinski, R.B. and Harabasz, J.(1974). A dendrite method for cluster analysis, Communications in Statistics, 3: 1-27.

[13] Dudoit, S. and Fridlyand, S. (2002). A prediction-based resampling method for estimating the number of clusters in a dataset. Genome Biology, 3(7): research 0036.1-0036.21.

[14] Juhasz, F.(1989), On the theoretical backgrounds of cluster analysis based on the eigenvalue problem of the association matrix. Statistics, 20: 572-581.

[15] Thorsten Joachims, Support Vector Machine, University of Dortmund, Informatic, AI-Unit Collaborative Research Center on 'Complexity Reduction in Multivariate Data'(SFB475).

[16] J.B.MacQueen (1967): 'Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability', Berkeley, University of California Press, 1:281-297.

[17] M. Bianco and V.J.Yohai(1996). Robust estimation in the logistic regression model. In H. Rieder, Ed. Robust Statistics, Data Analysis and Computer Intensive Methods, pp 17034.

[18] Tian Tang, Infrared Spectroscopy In Combination With Advanced Statistical Methods For Distinguishing Viral Infected Biological Cells

[19] C.Croux and K.Joossens(2005). Influence of Observations on the Misclassification Probability in Quadratic Discriminant Analysisi', Journal of Multivariate Analysis.

[20] http://en.wikipedia.org/wiki

APPENDIX A:

Summarization of Date Groups

Mock	1	2	3	4	5	6	7	8	9	10	11	12	13(09)	14	15	16	17	18	19	20	21
	0326	0506	0613	0909	0911	1009	1020	1023	1028	1106	1113	1120	0115	0128	0129	0204	0205	0212	0213	0305	0309
2	53	63	56	27	25	33	32	35	28	28	27	29	27	29	29	58	58	46	67	49	42
4	78	91	55	20	25	26	28	27	28	28	27	26	31	30	34	61	56	48	36	44	47
6	57	69	39	26	21	26	29	33	26	29	27	25	24	37	43	23	72	26	46	64	71
Hsv1																					
2	70	63	35	25	25	31	29	31	27	29	27	29	24	31	26	52	37	50	59	49	24
4	43	72	43	33	28	26	27	27	29	26	29	28	29	23	31	61	51	39	57	46	43
6	70	41	59	26	27	27	30	30	28	29	32	26	25	43	48	35	71	29	68	69	84
Adeno																					
2	74	51	42	25	22	29	29	28	27	29	25		28	35	26	60	64	30	72	55	19
4	48	44	40	26	29	27	28	27	28	30	28		32	32	29	56	62	38	49	57	48
6	44	59	38	34	26	23	29	35	30	29	30		27	44	55	20	52	34	59	66	86
Cox																					
2				26	25	29	31	27	27	29	28	27	24	31	28	56	66	57	39	40	36
4				26	25	24	27	27	27	31	28	27	27	28	29	60	53	76	78	47	52
6				29	26	25	36	40	27	29	39	28	27	67	55	26	41	21	49	69	44
compute				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

From(12)- 112008, we get data from coxsackie.

APPENDIX B:

Plots of Wilcoxon Signed-Rank Test for Other Paired Comparisons

Appendix B.1. Mock and Adeno

(1) The p-value and signed rank test value plot for 2 hour Mock and Adeno are as follows:

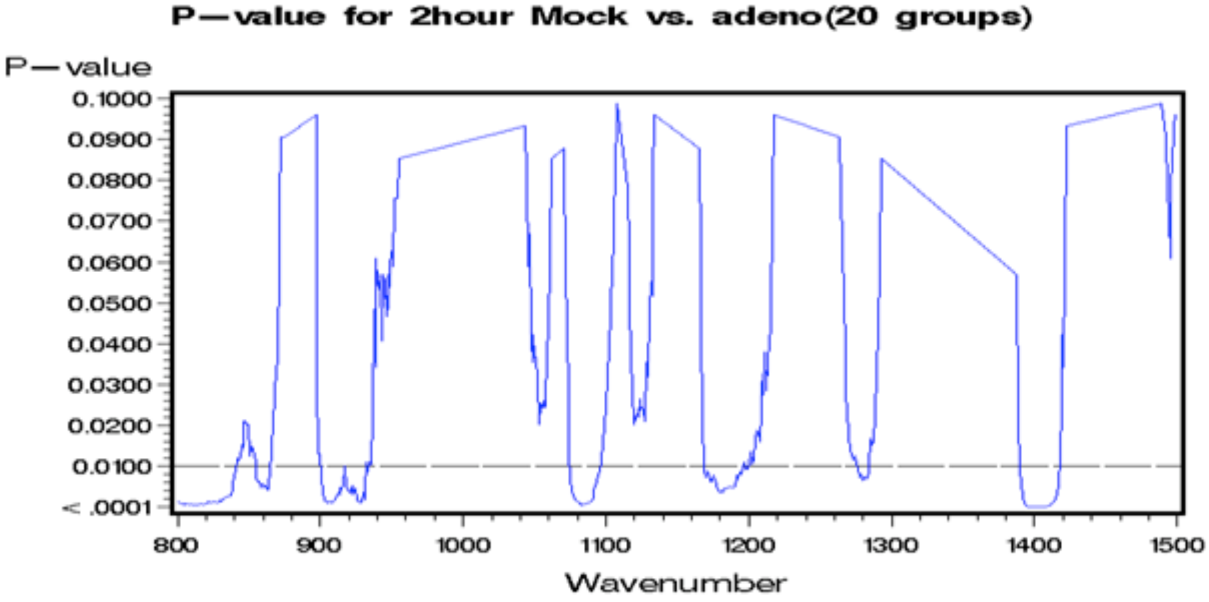


Figure B.1.1. P-value for 2 hour Mock vs. Adeno

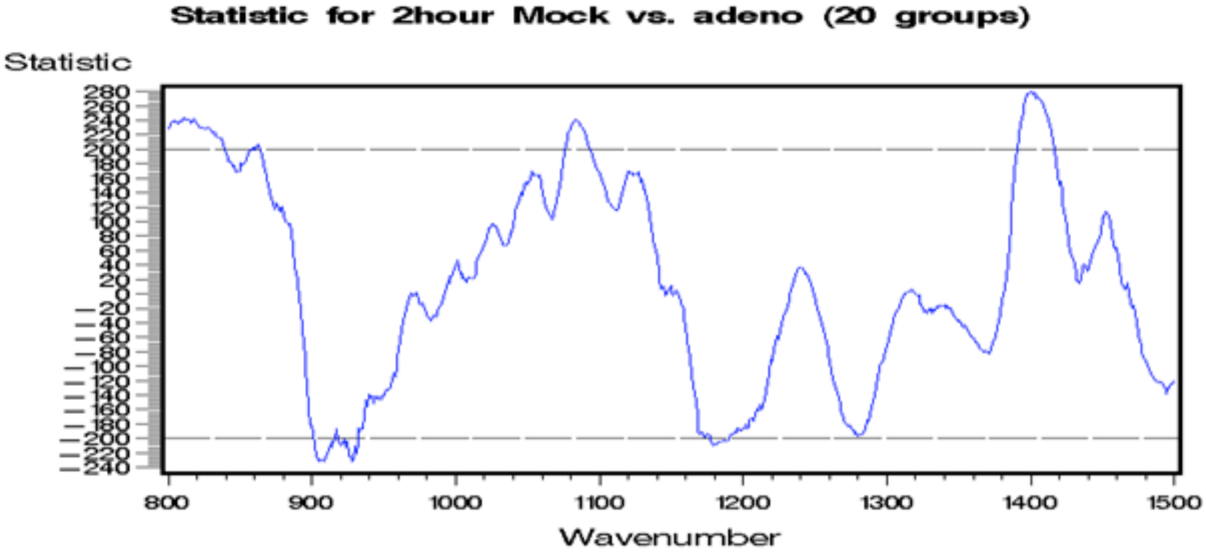


Figure B.1.2. Statistic value for 2 hour Mock vs. Adeno

From the above two plots, the significant positive regions for 2 hour Mock and Adeno are 799-838 cm^{-1}, 860-863 cm^{-1} ,1076-1092cm^{-1} and 1391-1416cm^{-1} while the significant negative regions for 2 hour Mock and Adeno are 901-916 cm^{-1}, 918-931cm^{-1} and 1177-1190cm^{-1}.

(2)The p-value and signed rank test value plot for 4 hour Mock and Adeno are as follows:

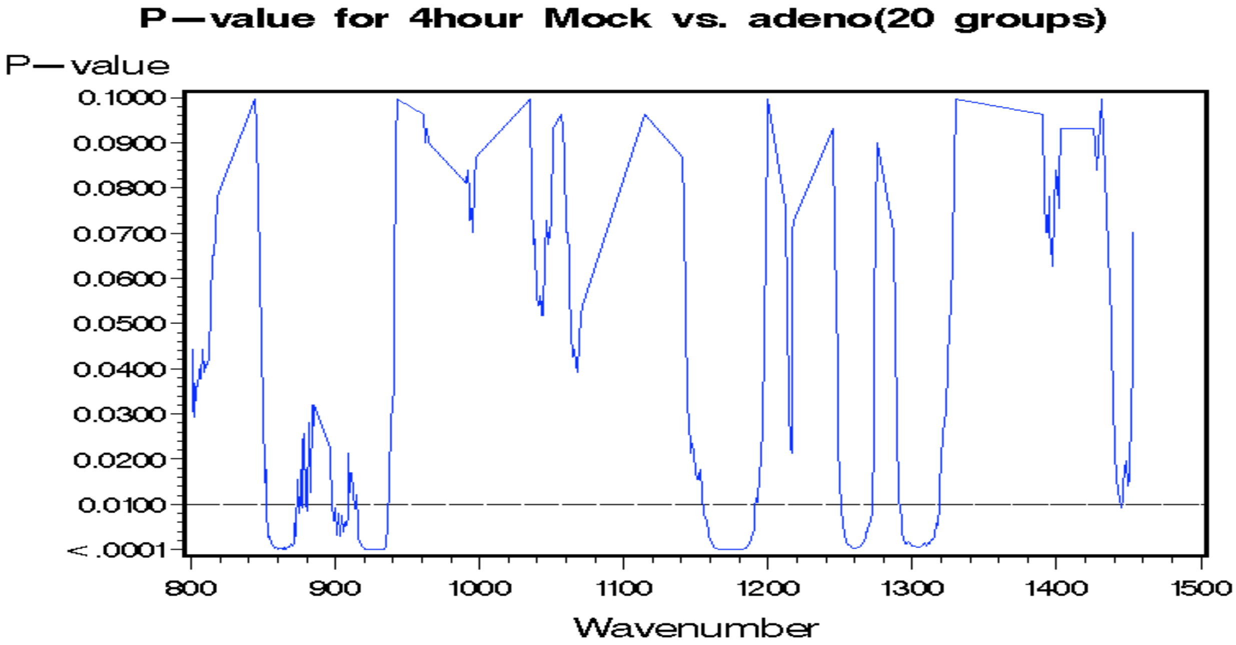


Figure B.1.3. P-value for 4 hour Mock vs. Adeno

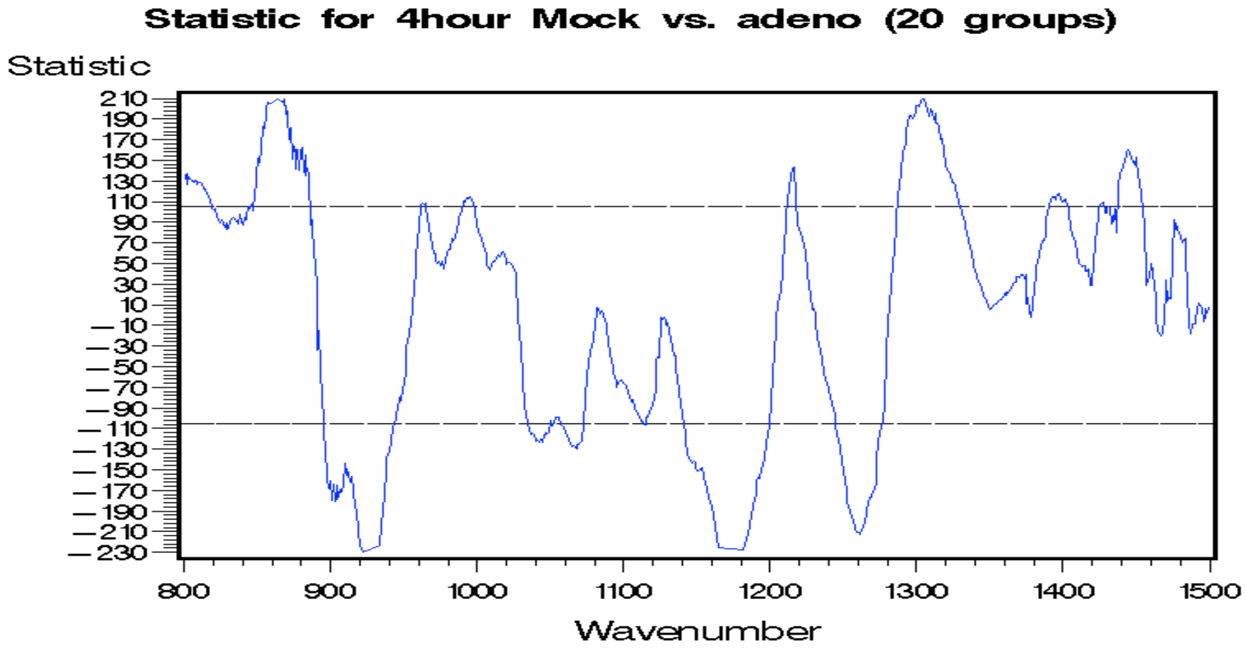


Figure B.1.4. Statistic value for 4 hour Mock vs. Adeno

From the above two plots, the significant positive regions for 4 hour Mock and Adeno are 856-868 cm^{-1}, 1300-1307 cm^{-1} while the significant negative regions for 4 hour Mock and Adeno are 919-934cm^{-1}, 1162-1187cm^{-1} and 1257-1264cm^{-1}.

(3)The p-value and signed rank test value plot for 6 hour Mock and Adeno are as follows:

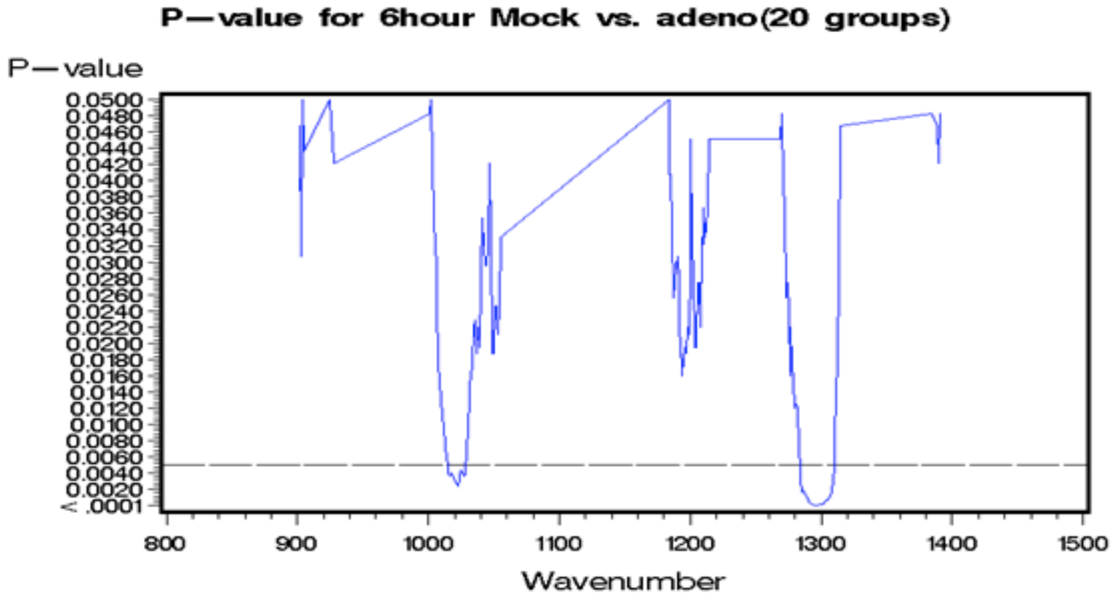


Figure B.1.5. P-value for 6 hour Mock vs. Adeno

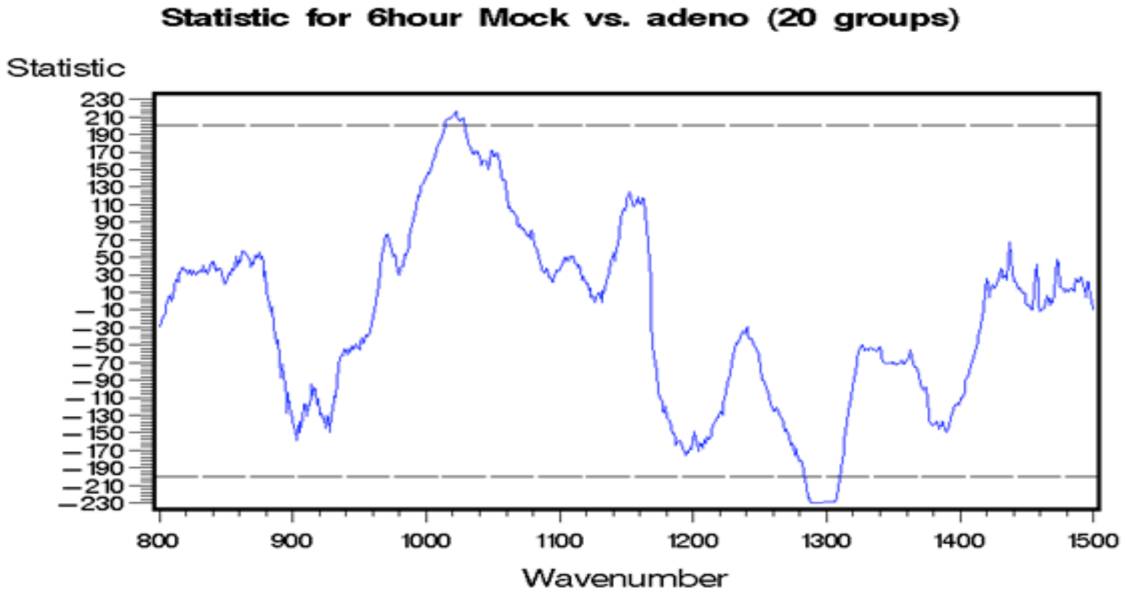


Figure B.1.6. Statistic value for 6 hour Mock vs. Adeno

From the above two plots, the significant positive regions for 6 hour Mock and Adeno is 1012-1029 cm^{-1} while the significant negative regions for 6 hour Mock and Adeno is1282-1311cm^{-1}.

Appendix B. 2. Mock and Coxsackie

(1)The p-value and signed rank test value plot for 2 hour Mock and Coxsackie are as follows:

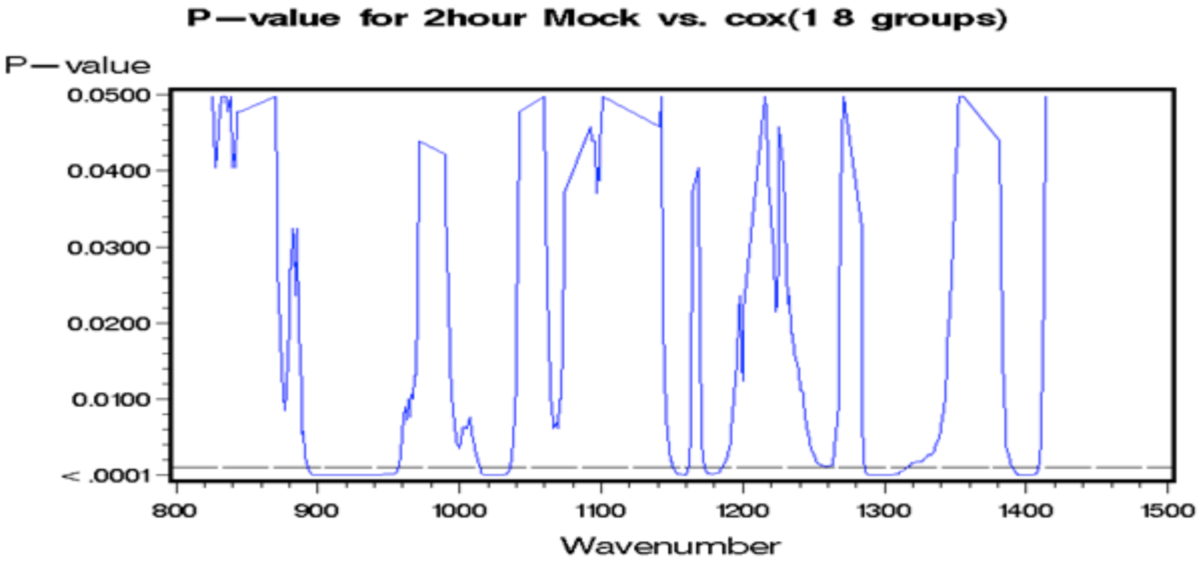


Figure B.2.1. P-value for 2 hour Mock vs. Coxsackie

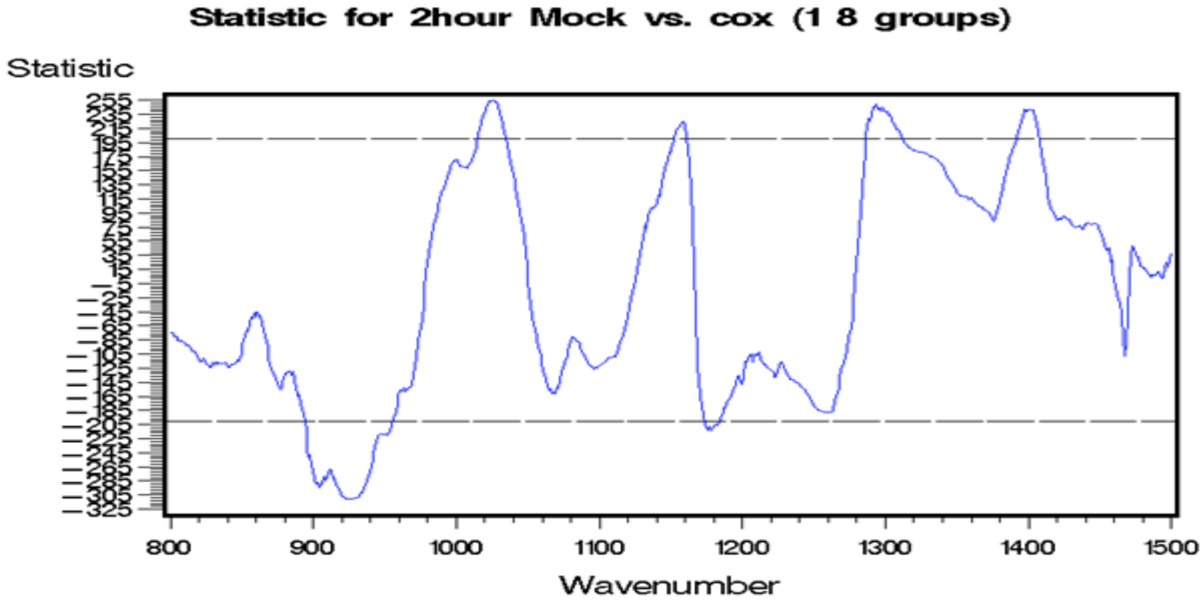


Figure B.2.2. Statistic value for 2 hour Mock vs. Coxsackie

From the above two plots, the significant positive regions for 2 hour Mock and Coxsackie are 1018-1030 cm^{-1}, 1290-1303cm^{-1}, 1397-1405 cm^{-1} while the significant negative regions for 2 hour Mock and Coxsackie is 895-943cm^{-1}.

(2)The p-value and signed rank test value plot for 4 hour Mock and Coxsackie are as follows:

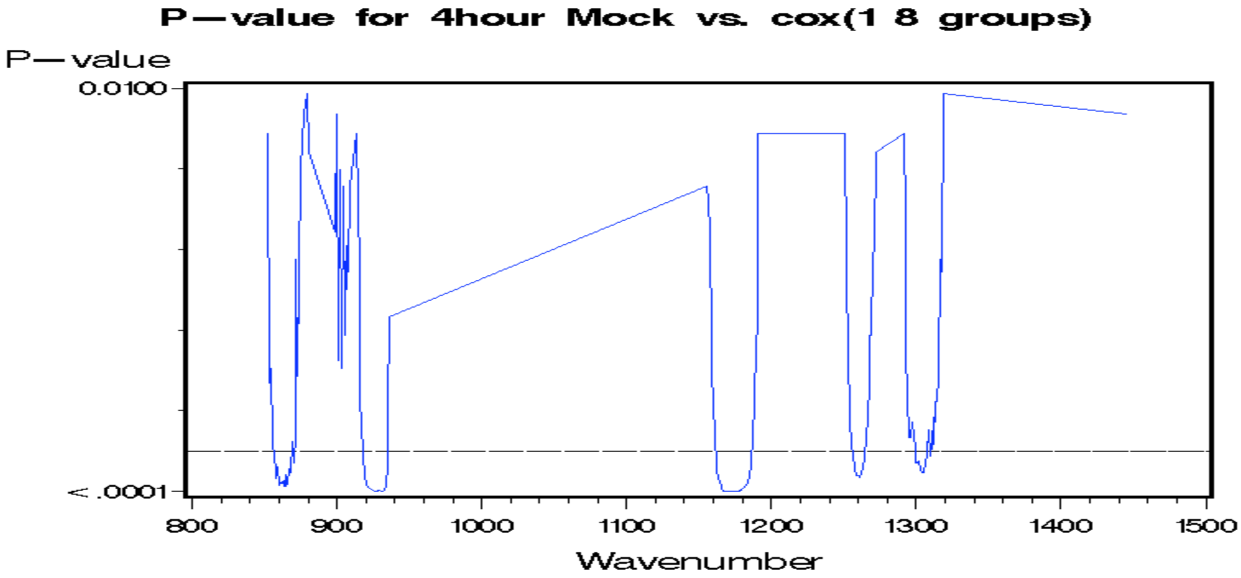


Figure B.2.3. P-value for 4 hour Mock vs. Coxsackie

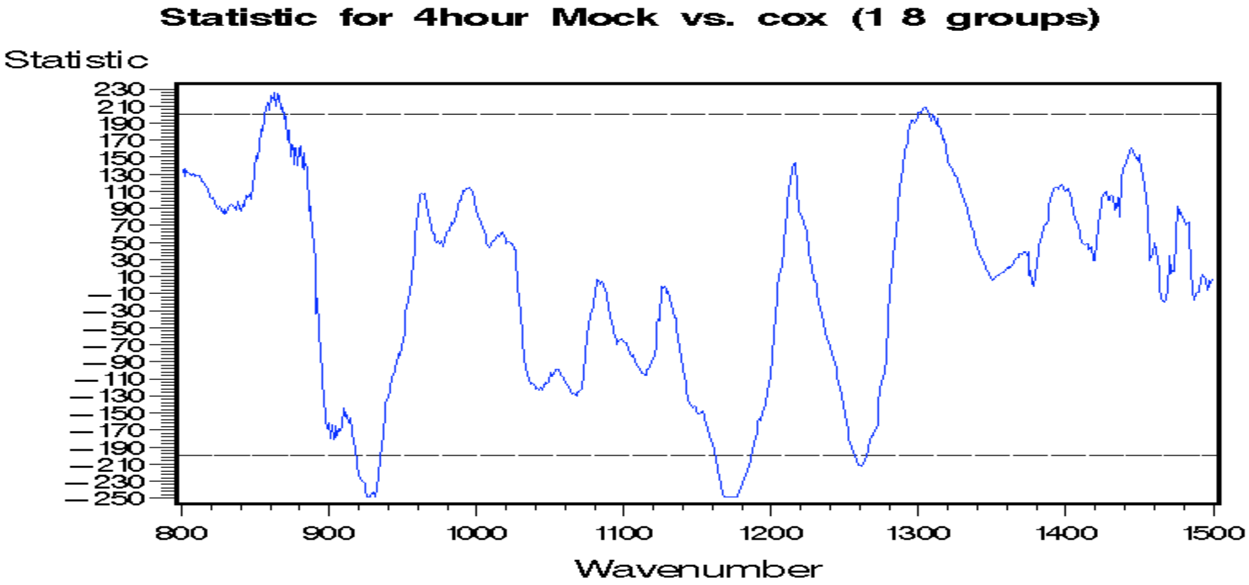


Figure B.2.4. Statistic value for 4 hour Mock vs. Coxsackie

From the above two plots, the significant positive regions for 4 hour Mock and Coxsackie are 856-868 cm^{-1}, 1300-1307cm^{-1} while the significant negative regions for 4 hour Mock and Coxsackie are 919-935cm^{-1}, 1162-1187cm^{-1}, 1257-1264cm^{-1}.

(3)The p-value and signed rank test value plot for 6 hour Mock and Coxsackie are as follows:

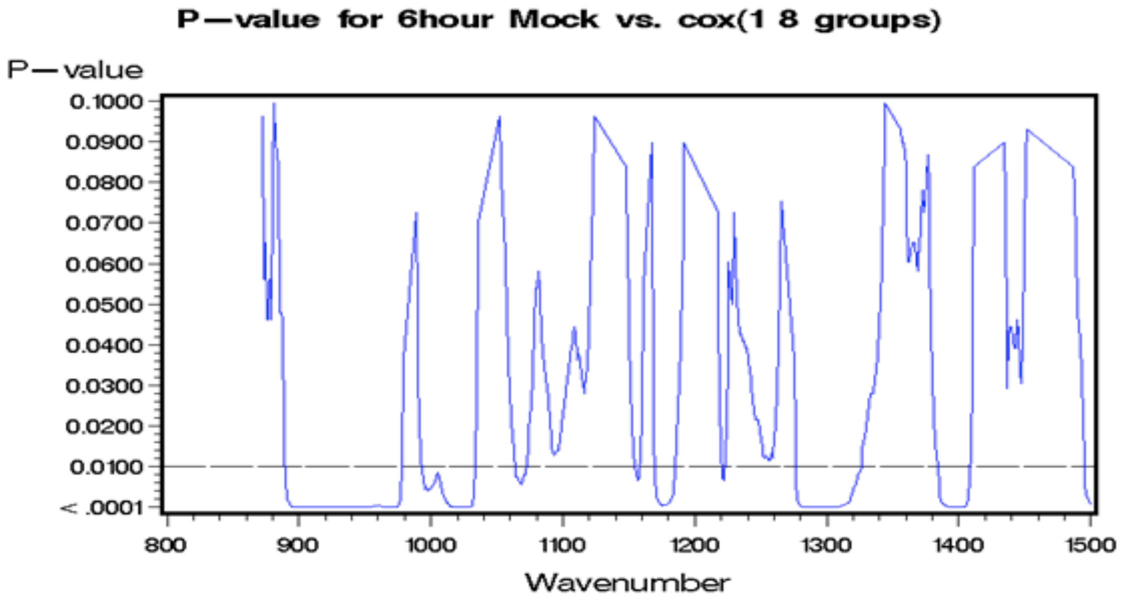


Figure B.2.5. P-value for 6 hour Mock vs. Coxsackie

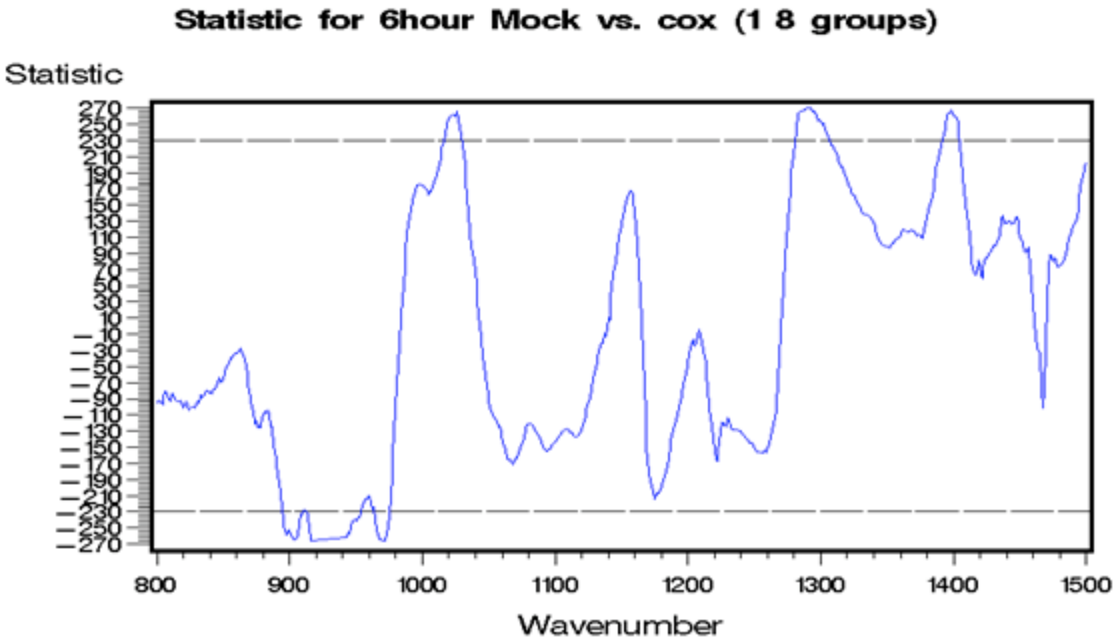


Figure B.2.6. Statistic value for 6 hour Mock vs. Coxsackie

From the above two plots, the significant positive regions for 6 hour Mock and Coxsackie are 1017-1029cm^{-1}, 1281-1307cm^{-1}, 1391-1405cm^{-1} while the significant negative regions for 6 hour Mock and Coxsackie are 894-954cm^{-1}, 964-975cm^{-1}.

Appendix B. 3. Hsv1 and Coxsackie

(1)The p-value and signed rank test value plot for 2 hour Hsv1 and Coxsackie are as follows:

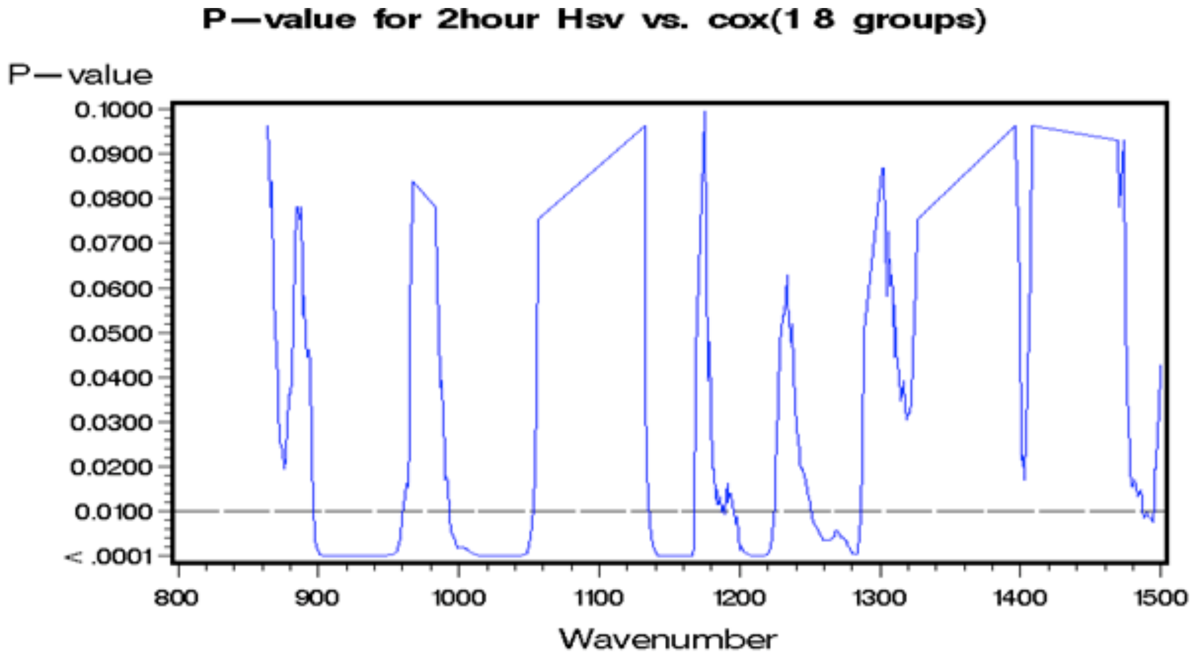


Figure B.3.1. P-value for 2 hour Hsv1 vs. Coxsackie

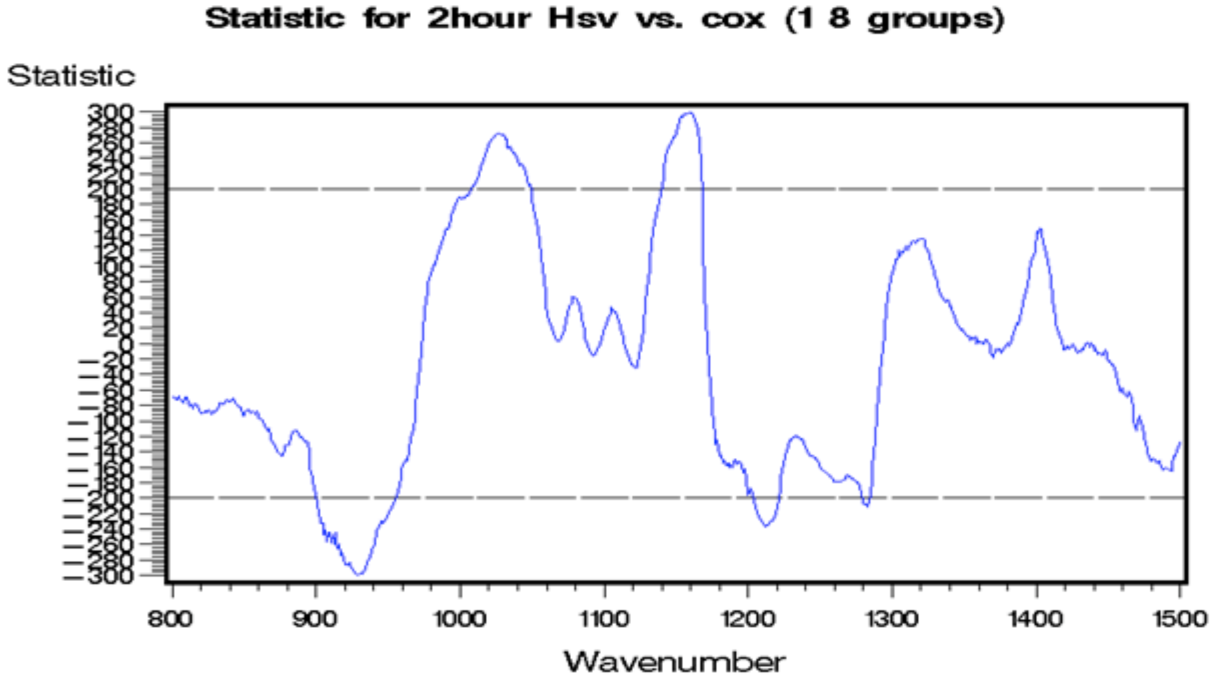


Figure B.3.2. Statistic value for 2 hour Hsv1 vs. Coxsackie

From the above two plots, the significant positive regions for 2 hour Hsv1 and Coxsackie are 1014-1044 cm^{-1}, 1142-1166cm^{-1} while the significant negative regions for 2 hour Hsv1 and Coxsackie are 902-948cm^{-1}, 1209-1218cm^{-1}.

(2)The p-value and signed rank test value plot for 4 hour Hsv1 and Coxsackie are as follows:

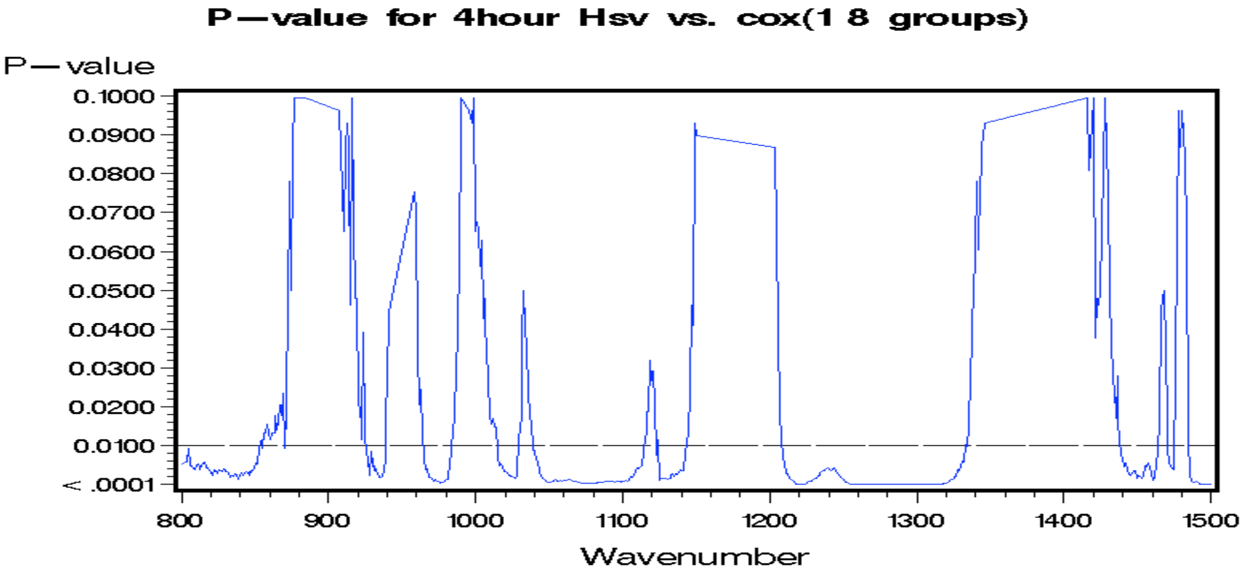


Figure B.3.3. P-value for 4 hour Hsv1 vs. Coxsackie

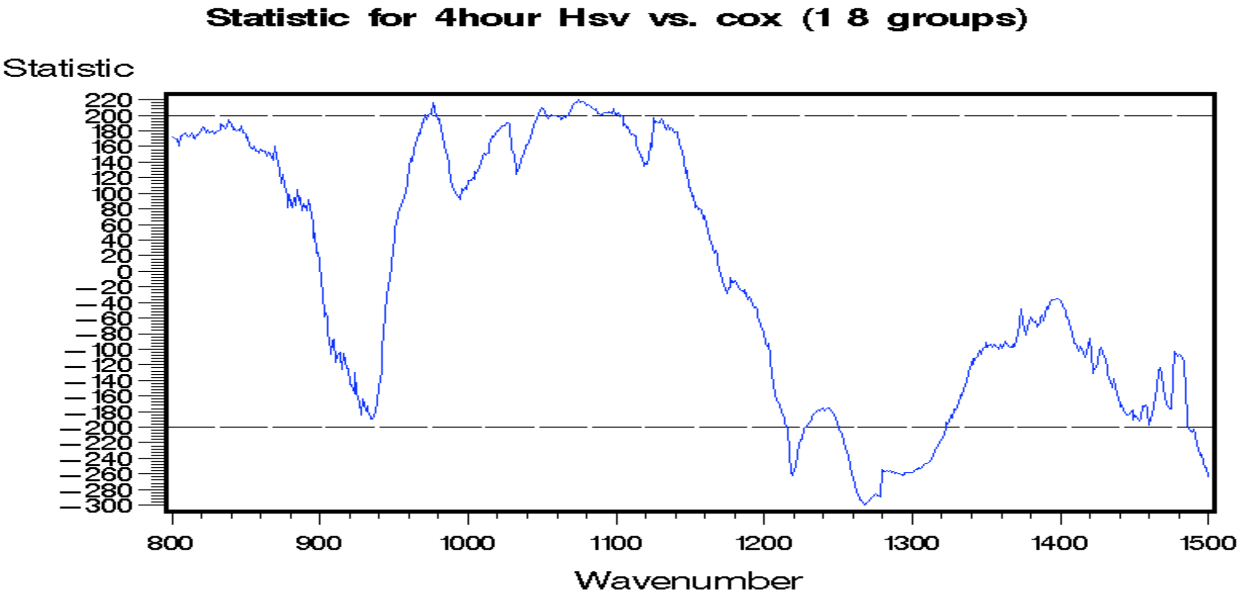
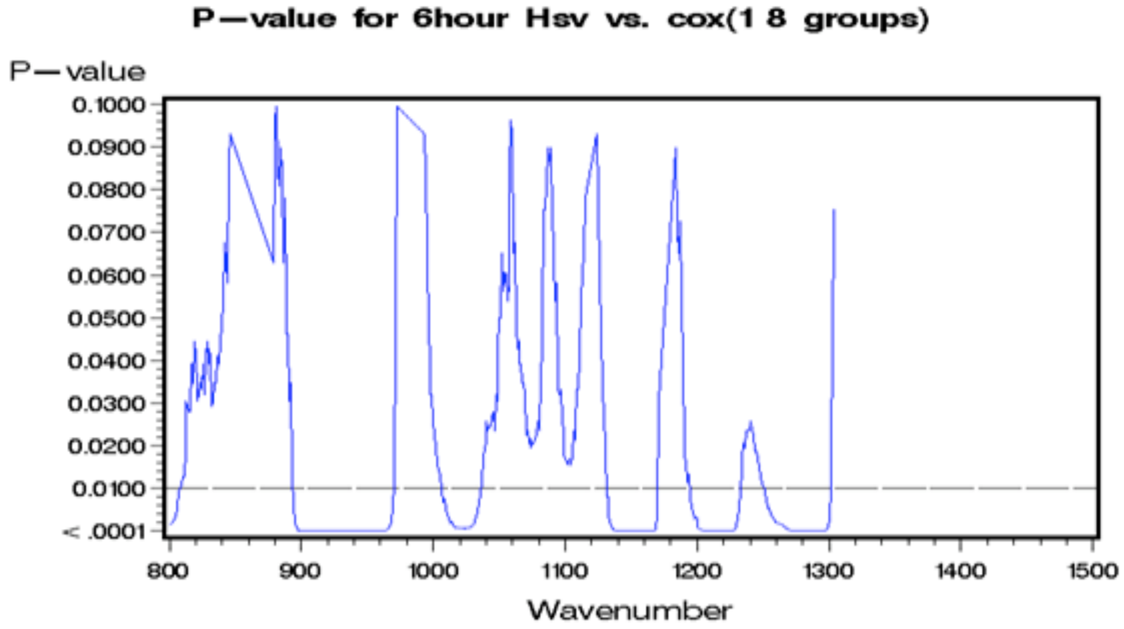


Figure B.3.4. Statistic value for 4 hour Hsv1 vs. Coxsackie

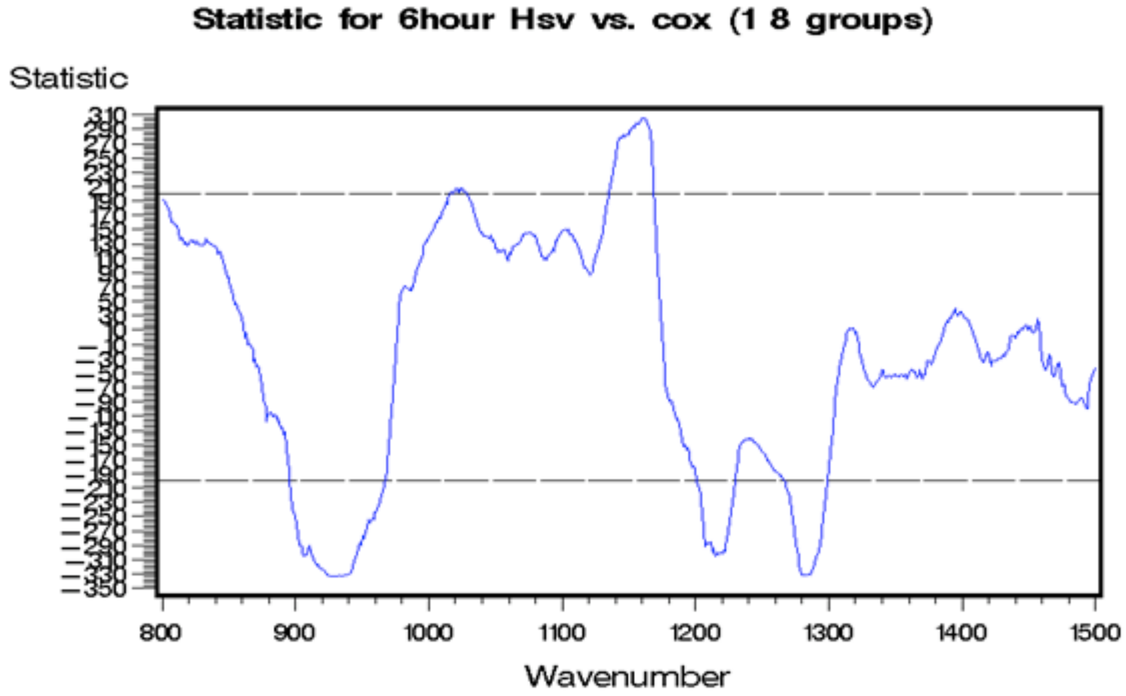
From the above two plots, the significant positive regions for 4 hour Hsv1 and Coxsackie are 974-979 cm^{-1}, 1047-1053cm^{-1}, 1068-1102cm^{-1} while the significant negative regions for 4 hour Hsv1 and Coxsackie are 1216-1227cm^{-1}, 1250-1323cm^{-1}, 1487-1500cm^{-1}.

(3)The p-value and signed rank test value plot for 6 hour Hsv1 and Coxsackie are as follows:



P—value for 6hour Hsv vs. cox(1 8 groups)

Figure B.3.5. P-value for 6 hour Hsv1 vs. Coxsackie



Statistic for 6hour Hsv vs. cox (1 8 groups)

Figure B.3.6. Statistic value for 6 hour Hsv1 vs. Coxsackie

From the above two plots, the significant positive regions for 6 hour Hsv1 and Coxsackie is 1137-1167cm^{-1} while the significant negative regions for 6 hour Hsv1 and Coxsackie are 897-963cm^{-1}, 1204-1228cm^{-1}, 1272-1297cm^{-1}.

Appendix B. 4. Hsv1 and Adeno

(1) The p-value and signed rank test value plot for 2 hour Hsv1 and Adeno are as follows:

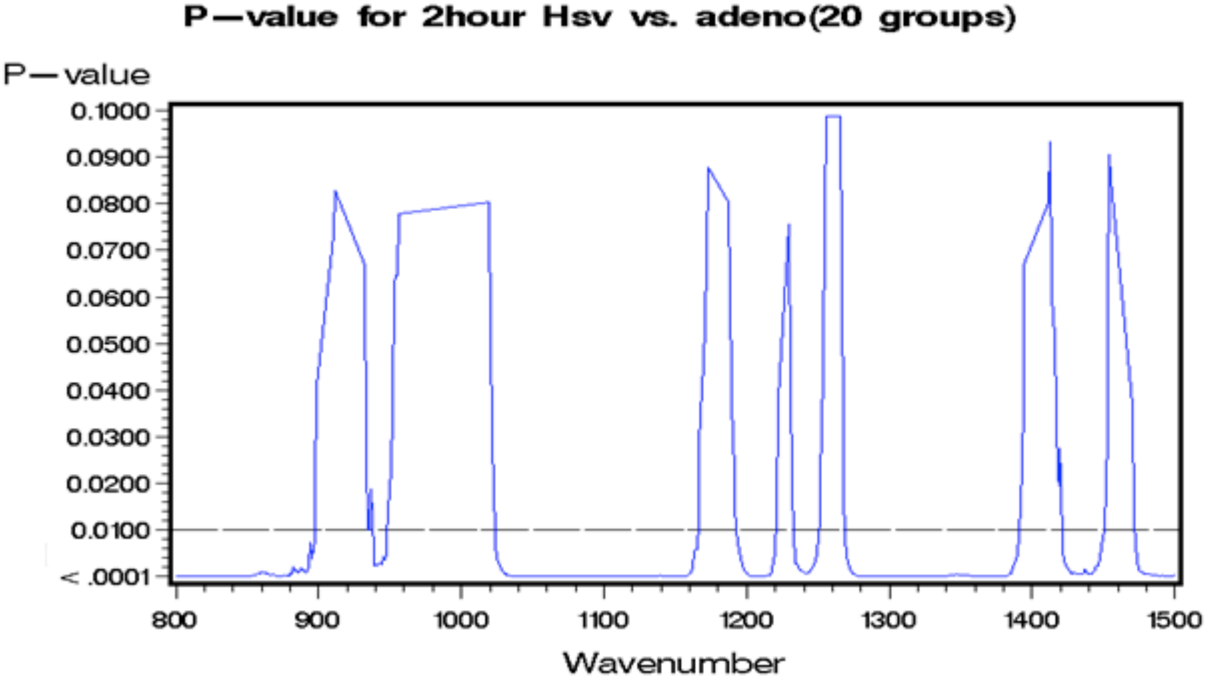


Figure B.4.1. P-value for 2 hour Hsv1 vs. Adeno

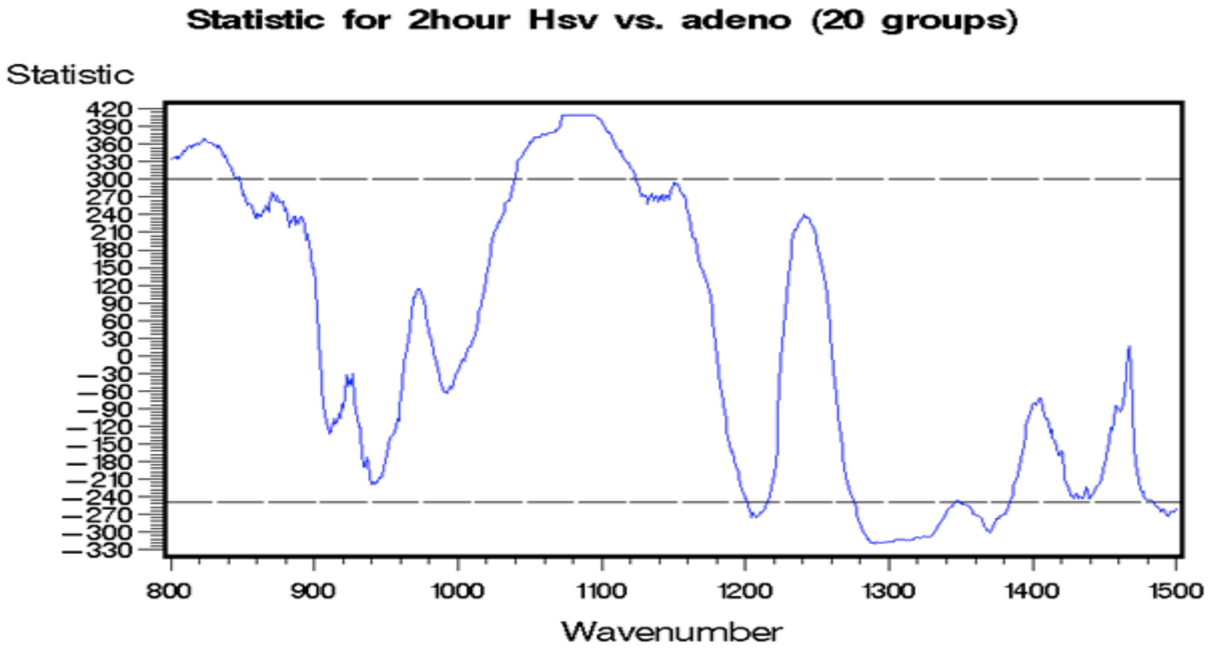


Figure B.4.2. Statistic value for 2 hour Hsv1 vs. Adeno

From the above two plots, the significant positive regions for 2 hour Hsv1 and Adeno are 799-847 cm^{-1}, 1039-1123cm^{-1} while the significant negative regions for 2 hour Hsv1 and Adeno are 1201-1215cm^{-1}, 1275-1344cm^{-1}, 1351-1382cm^{-1}, 1484-1500cm^{-1}.

(2) The p-value and signed rank test value plot for 4 hour Hsv1 and Adeno are as follows:

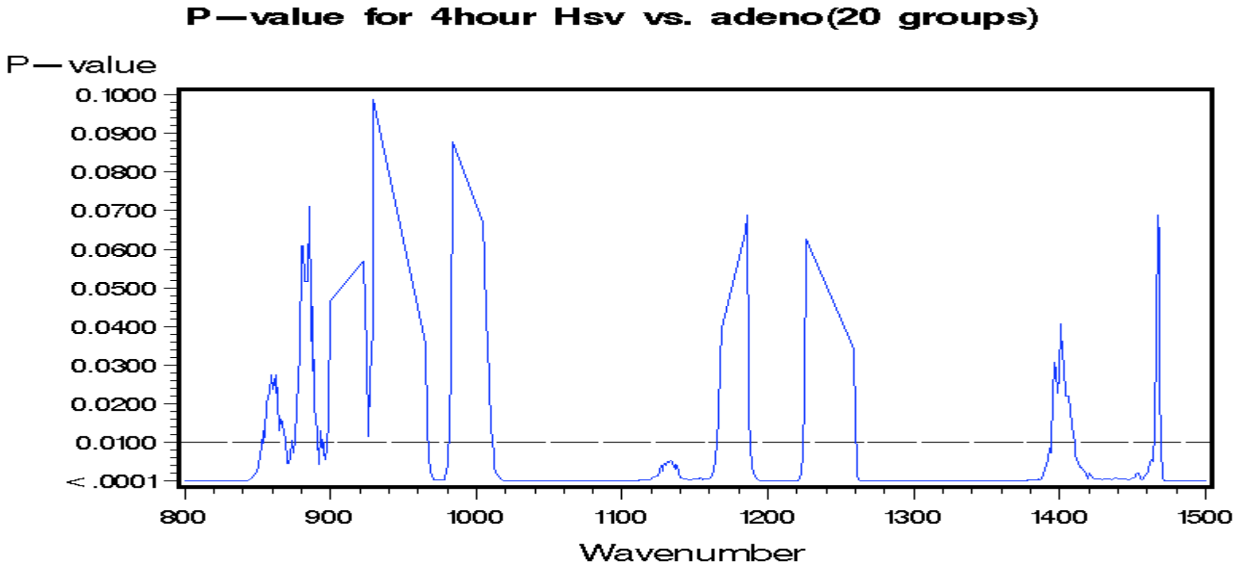


Figure B.4.3. P-value for 4 hour Hsv1 vs. Adeno

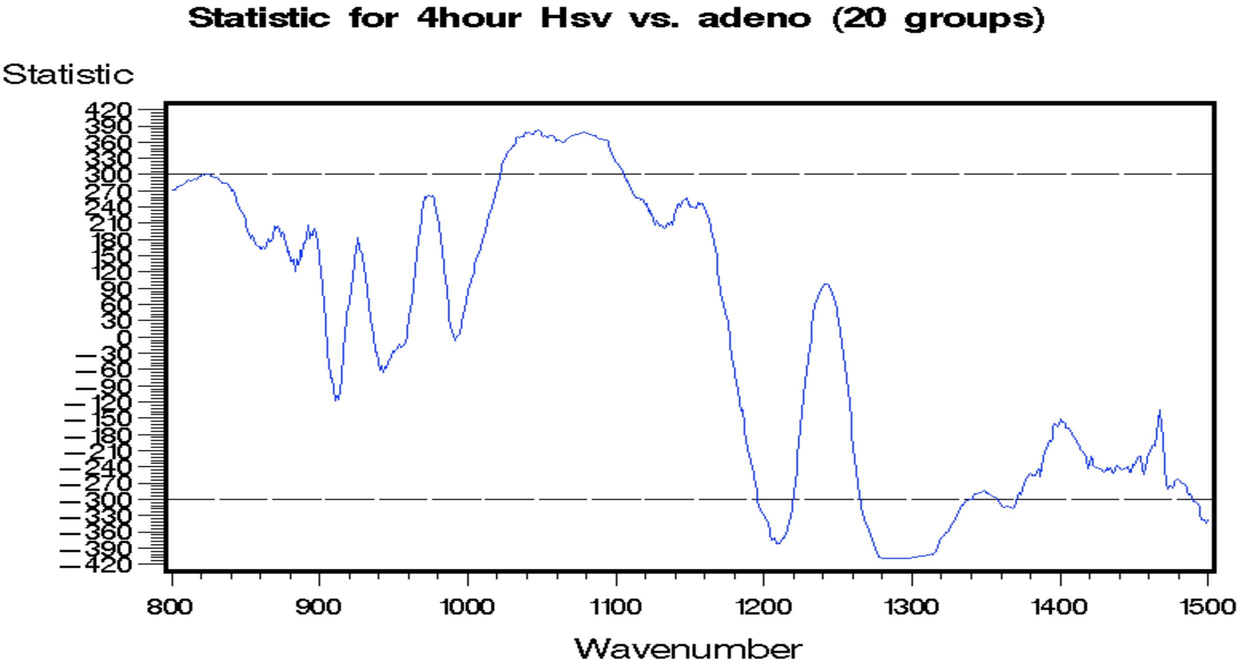


Figure B.4.4. Statistic value for 4 hour Hsv1 vs. Adeno

From the above two plots, the significant positive regions for 4 hour Hsv1 and Adeno is 1022-1219 cm^{-1} while the significant negative regions for 4 hour Hsv1 and Adeno are 1195-1219cm^{-1}, 1265-1339cm^{-1}, 1358-1370cm^{-1}, 1490-1500cm^{-1}.

(3) The p-value and signed rank test value plot for 6 hour Hsv1 and Adeno are as follows:

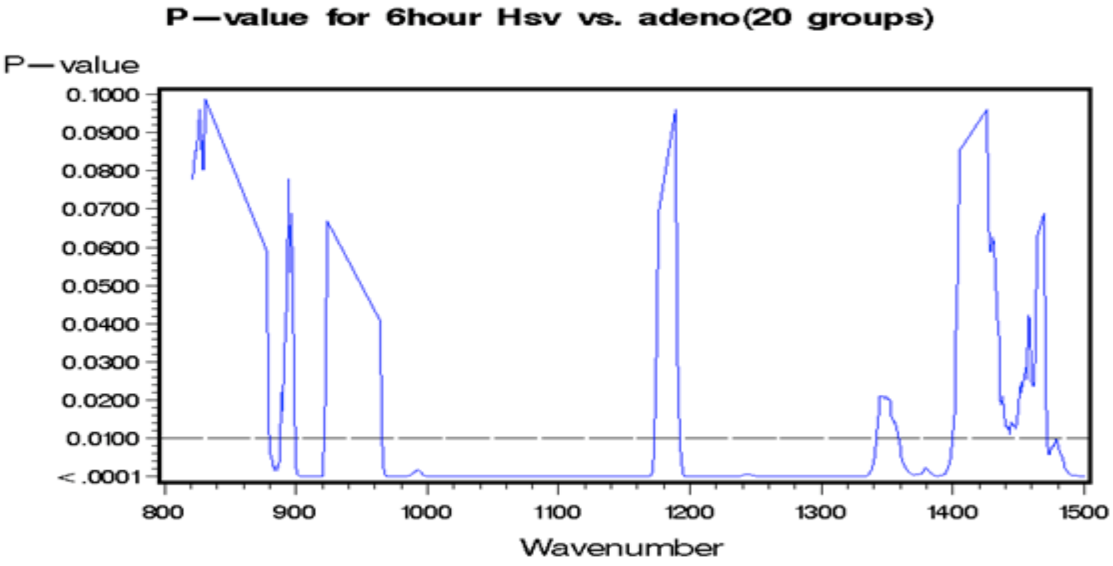


Figure B.4.5. P-value for 6 hour Hsv1 vs. Adeno

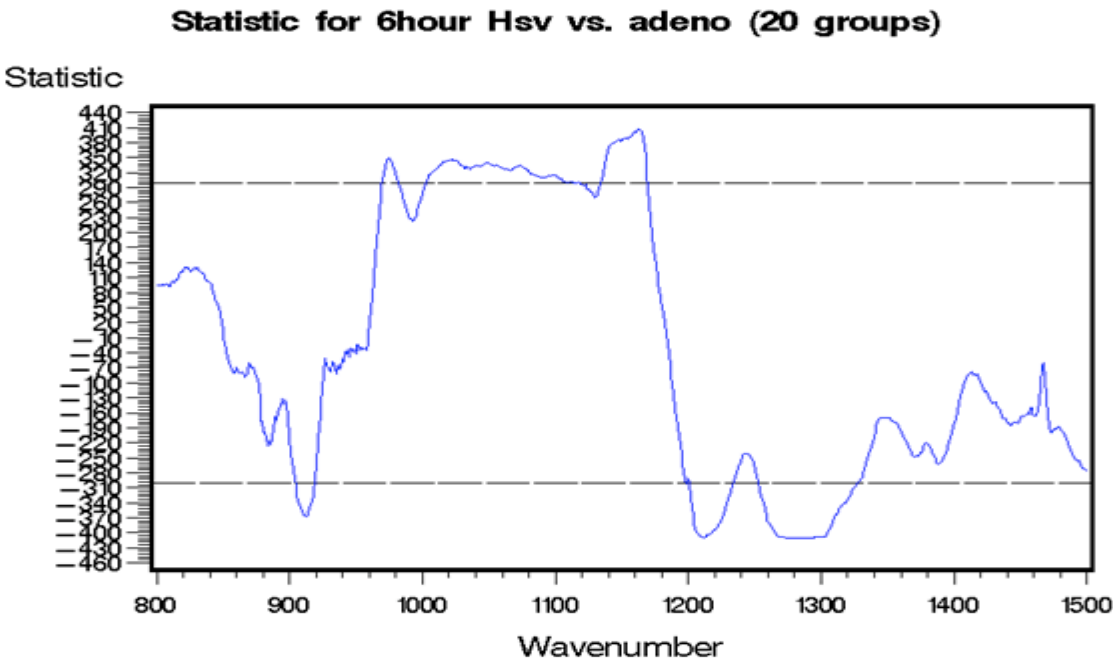


Figure B.4.6. Statistic value for 6 hour Hsv1 vs. Adeno

From the above two plots, the significant positive regions for 6 hour Hsv1 and Adeno are 970-980cm^{-1}, 1002-1112cm^{-1} and 1135-1168cm^{-1} while the significant negative regions for 6 hour Hsv1 and Adeno are 904-918cm^{-1}, 1201-1233cm^{-1}, 1253-1327cm^{-1}.

Appendix B. 5. Adeno and Coxsackie

(1) The p-value and signed rank test value plot for 2 hour Adeno and Coxsackie are as follows:

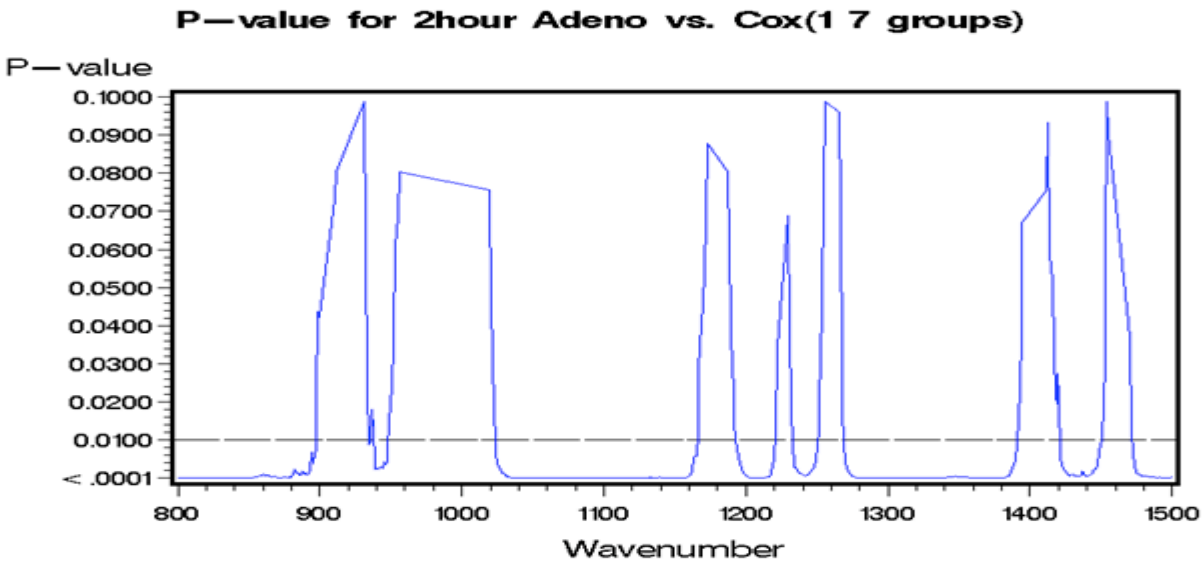


Figure B.5.1. P-value for 2 hour Adeno vs. Coxsackie

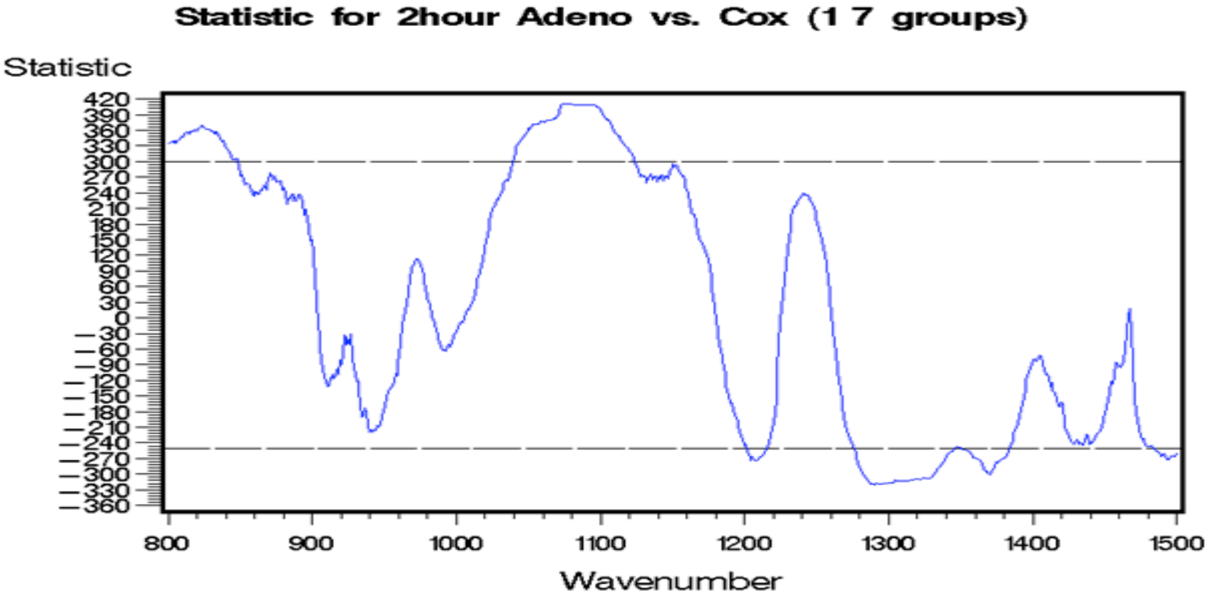


Figure B.5.2. Statistic value for 2 hour Adeno vs. Coxsackie

From the above two plots, the significant positive regions for 2 hour Adeno and Coxsackie are 808-836cm^{-1} and 1045-1113cm^{-1} while the significant negative regions for 2 hour Adeno and Coxsackie is 1282-1331cm^{-1}.

(2) The p-value and signed rank test value plot for 4 hour Adeno and Coxsackie are as follows:

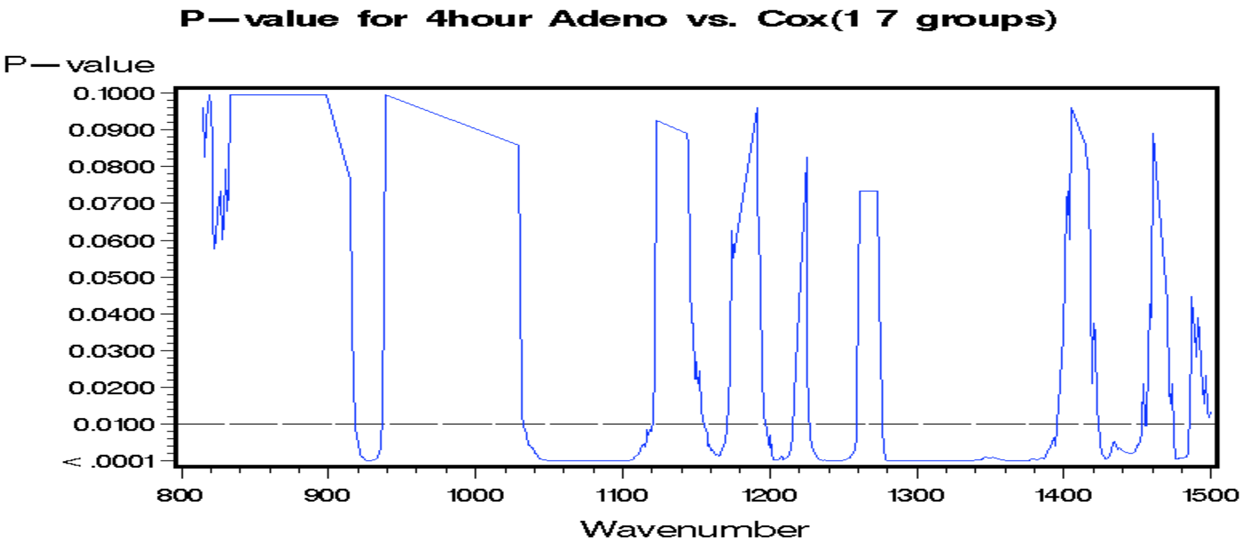


Figure B.5.3. P-value for 4 hour Adeno vs. Coxsackie

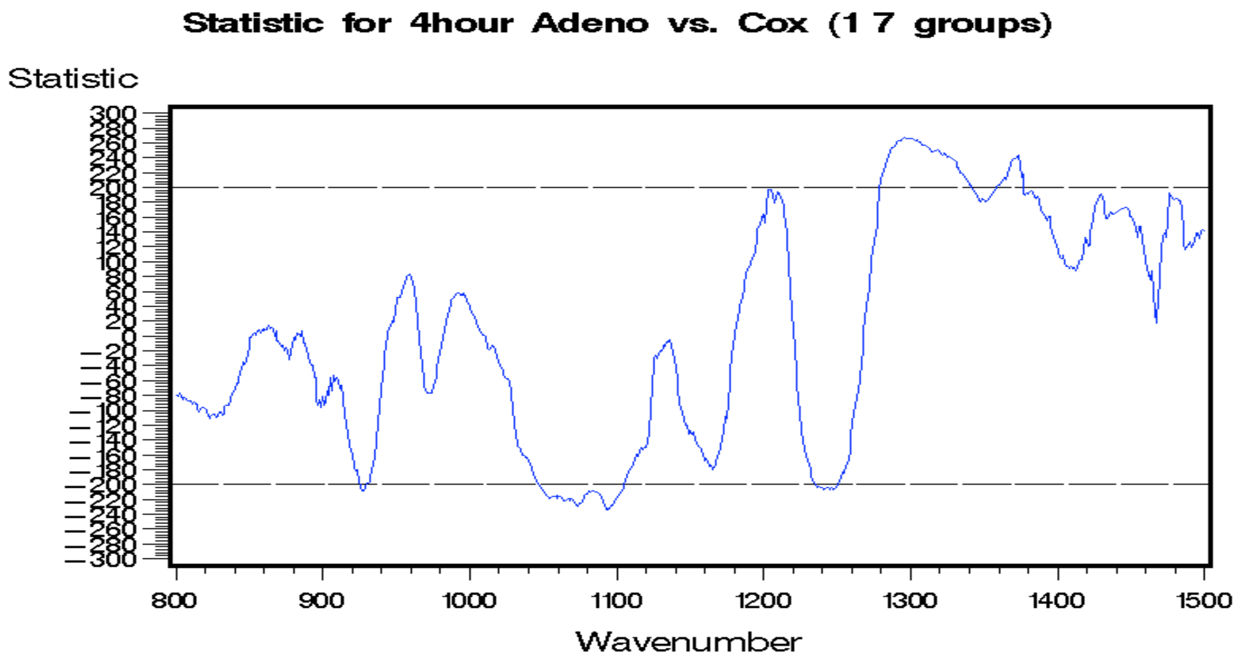


Figure B.5.4. Statistic value for 4 hour Adeno vs. Coxsackie

From the above two plots, the significant positive regions for 4 hour Adeno and Coxsackie are 1279-1338 cm^{-1} and 1362-1376 cm^{-1}while the significant negative regions for 4 hour Adeno and Coxsackie is 1049-1102cm^{-1}.

(3) The p-value and signed rank test value plot for 6 hour Adeno and Coxsackie are as follows:

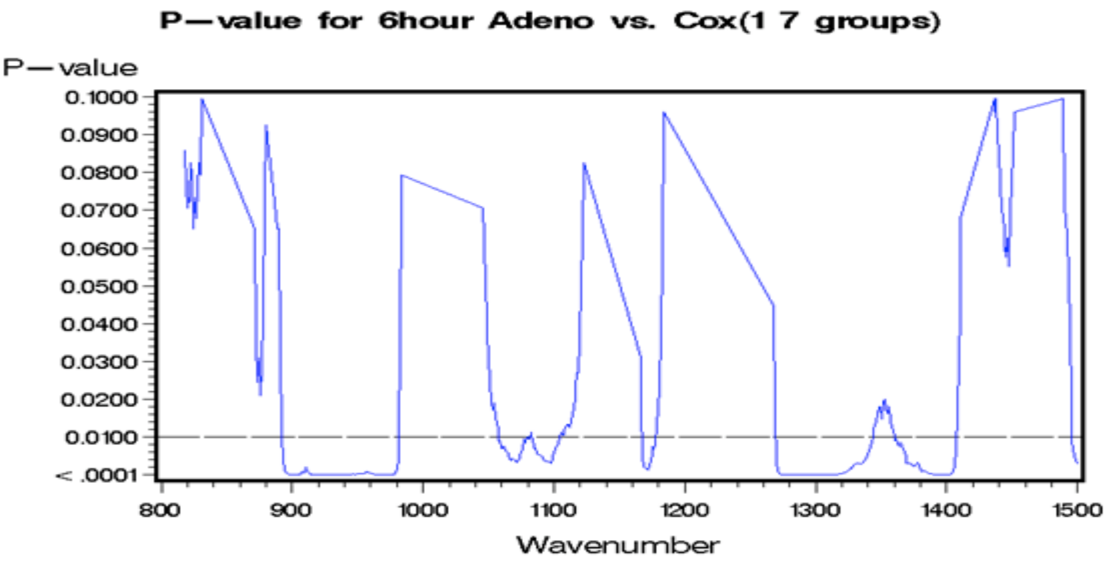


Figure B.5.5. P-value for 6 hour Adeno vs. Coxsackie

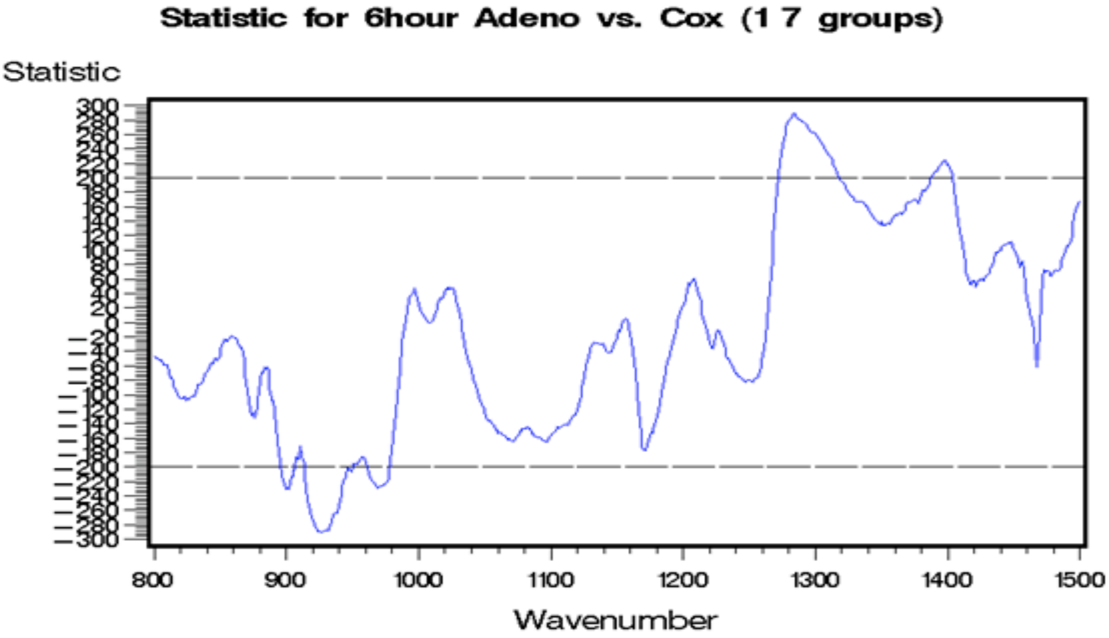


Figure B.5.6 Statistic value for 6 hour Adeno vs. Coxsackie

From the above two figures, the significant positive regions for 6 hour Adeno and Coxsackie is 1273-1317cm^{-1} while the significant negative regions for 6 hour Adeno and Coxsackie is 913-942cm^{-1}.

APPENDIX C: Significant Regions by Wilcoxon Signed-Rank Test

Appendix C.1 Selected variables for 6 hour Mock vs. Hsv1

Obs	VarName	Test	Testlab	Stat	pType	P-Value	Positive Num	Negative Num
1	t1	Signed Rank	S	-154.5	Pr >= |S|	0.0522	0	0
2	t2	Signed Rank	S	-144.5	Pr >= |S|	0.0702	0	0
3	t3	Signed Rank	S	-133.5	Pr >= |S|	0.0954	0	0
4	t4	Signed Rank	S	-120.5	Pr >= |S|	0.1336	0	0
5	t5	Signed Rank	S	-134.5	Pr >= |S|	0.0929	0	0
6	t6	Signed Rank	S	-134.5	Pr >= |S|	0.0929	0	0
7	t7	Signed Rank	S	-133.5	Pr >= |S|	0.0954	0	0
8	t8	Signed Rank	S	-132.5	Pr >= |S|	0.098	0	0
9	t9	Signed Rank	S	-124.5	Pr >= |S|	0.1208	0	0
10	t10	Signed Rank	S	-120.5	Pr >= |S|	0.1336	0	0
11	t11	Signed Rank	S	-114.5	Pr >= |S|	0.1546	0	0
12	t12	Signed Rank	S	-91.5	Pr >= |S|	0.2575	0	0
13	t13	Signed Rank	S	-57.5	Pr >= |S|	0.4788	0	0
14	t14	Signed Rank	S	-12.5	Pr >= |S|	0.878	0	0
15	t15	Signed Rank	S	42.5	Pr >= |S|	0.6012	0	0
16	t16	Signed Rank	S	82.5	Pr >= |S|	0.308	0	0
17	t17	Signed Rank	S	106.5	Pr >= |S|	0.1862	0	0
18	t18	Signed Rank	S	96.5	Pr >= |S|	0.2319	0	0
19	t19	Signed Rank	S	61.5	Pr >= |S|	0.4486	0	0
20	t20	Signed Rank	S	67.5	Pr >= |S|	0.4052	0	0
21	t21	Signed Rank	S	107.5	Pr >= |S|	0.182	0	0
22	t22	Signed Rank	S	141.5	Pr >= |S|	0.0765	0	0
23	t23	Signed Rank	S	106.5	Pr >= |S|	0.1862	0	0
24	t24	Signed Rank	S	29.5	Pr >= |S|	0.717	0	0
25	t25	Signed Rank	S	-30.5	Pr >= |S|	0.7078	0	0
26	t26	Signed Rank	S	-43.5	Pr >= |S|	0.5926	0	0
27	t27	Signed Rank	S	-0.5	Pr >= |S|	0.9951	0	0
28	t28	Signed Rank	S	85.5	Pr >= |S|	0.2905	0	0
29	t29	Signed Rank	S	176.5	Pr >= |S|	0.0255	0	0
30	t30	Signed Rank	S	220.5	Pr >= |S|	0.0045	0	0
31	t31	Signed Rank	S	173.5	Pr >= |S|	0.0282	0	0
32	t32	Signed Rank	S	16.5	Pr >= |S|	0.8394	0	0
33	t33	Signed Rank	S	-98.5	Pr >= |S|	0.2222	0	0
34	t34	Signed Rank	S	-135.5	Pr >= |S|	0.0904	0	0
35	t35	Signed Rank	S	-99.5	Pr >= |S|	0.2175	0	0
36	t36	Signed Rank	S	-46.5	Pr >= |S|	0.5672	0	0
37	t37	Signed Rank	S	-29.5	Pr >= |S|	0.717	0	0
38	t38	Signed Rank	S	-44.5	Pr >= |S|	0.5841	0	0
39	t39	Signed Rank	S	-69.5	Pr >= |S|	0.3913	0	0
40	t40	Signed Rank	S	-69.5	Pr >= |S|	0.3913	0	0
41	t41	Signed Rank	S	-54.5	Pr >= |S|	0.5022	0	0
42	t42	Signed Rank	S	-38.5	Pr >= |S|	0.6359	0	0
43	t43	Signed Rank	S	-111.5	Pr >= |S|	0.1659	0	0
44	t44	Signed Rank	S	-212.5	Pr >= |S|	0.0063	0	0
45	t45	Signed Rank	S	-296.5	Pr >= |S|	<.0001	0	1
46	t46	Signed Rank	S	-332.5	Pr >= |S|	<.0001	0	2
47	t47	Signed Rank	S	-319.5	Pr >= |S|	<.0001	0	3

48	t48	Signed Rank	S	-264.5	Pr >= |S|	0.0005	0	3
49	t49	Signed Rank	S	-199.5	Pr >= |S|	0.0108	0	3
50	t50	Signed Rank	S	-142.5	Pr >= |S|	0.0744	0	3
51	t51	Signed Rank	S	-137.5	Pr >= |S|	0.0856	0	3
52	t52	Signed Rank	S	-147.5	Pr >= |S|	0.0644	0	3
53	t53	Signed Rank	S	-173.5	Pr >= |S|	0.0282	0	3
54	t54	Signed Rank	S	-184.5	Pr >= |S|	0.0191	0	3
55	t55	Signed Rank	S	-201.5	Pr >= |S|	0.01	0	3
56	t56	Signed Rank	S	-205.5	Pr >= |S|	0.0085	0	3
57	t57	Signed Rank	S	-208.5	Pr >= |S|	0.0075	0	3
58	t58	Signed Rank	S	-216.5	Pr >= |S|	0.0053	0	3
59	t59	Signed Rank	S	-224.5	Pr >= |S|	0.0037	0	3
60	t60	Signed Rank	S	-245.5	Pr >= |S|	0.0013	0	3
61	t61	Signed Rank	S	-279.5	Pr >= |S|	0.0002	0	4
62	t62	Signed Rank	S	-323.5	Pr >= |S|	<.0001	0	5
63	t63	Signed Rank	S	-365.5	Pr >= |S|	<.0001	0	6
64	t64	Signed Rank	S	-406.5	Pr >= |S|	<.0001	0	7
65	t65	Signed Rank	S	-420.5	Pr >= |S|	<.0001	0	8
66	t66	Signed Rank	S	-421.5	Pr >= |S|	<.0001	0	9
67	t67	Signed Rank	S	-404.5	Pr >= |S|	<.0001	0	10
68	t68	Signed Rank	S	-390.5	Pr >= |S|	<.0001	0	11
69	t69	Signed Rank	S	-383.5	Pr >= |S|	<.0001	0	12
70	t70	Signed Rank	S	-385.5	Pr >= |S|	<.0001	0	13
71	t71	Signed Rank	S	-383.5	Pr >= |S|	<.0001	0	14
72	t72	Signed Rank	S	-381.5	Pr >= |S|	<.0001	0	15
73	t73	Signed Rank	S	-362.5	Pr >= |S|	<.0001	0	16
74	t74	Signed Rank	S	-344.5	Pr >= |S|	<.0001	0	17
75	t75	Signed Rank	S	-340.5	Pr >= |S|	<.0001	0	18
76	t76	Signed Rank	S	-342.5	Pr >= |S|	<.0001	0	19
77	t77	Signed Rank	S	-357.5	Pr >= |S|	<.0001	0	20
78	t78	Signed Rank	S	-358.5	Pr >= |S|	<.0001	0	21
79	t79	Signed Rank	S	-353.5	Pr >= |S|	<.0001	0	22
80	t80	Signed Rank	S	-349.5	Pr >= |S|	<.0001	0	23
81	t81	Signed Rank	S	-346.5	Pr >= |S|	<.0001	0	24
82	t82	Signed Rank	S	-346.5	Pr >= |S|	<.0001	0	25
83	t83	Signed Rank	S	-343.5	Pr >= |S|	<.0001	0	26
84	t84	Signed Rank	S	-337.5	Pr >= |S|	<.0001	0	27
85	t85	Signed Rank	S	-330.5	Pr >= |S|	<.0001	0	28
86	t86	Signed Rank	S	-350.5	Pr >= |S|	<.0001	0	29
87	t87	Signed Rank	S	-374.5	Pr >= |S|	<.0001	0	30
88	t88	Signed Rank	S	-394.5	Pr >= |S|	<.0001	0	31
89	t89	Signed Rank	S	-422.5	Pr >= |S|	<.0001	0	32
90	t90	Signed Rank	S	-423.5	Pr >= |S|	<.0001	0	33
91	t91	Signed Rank	S	-423.5	Pr >= |S|	<.0001	0	34
92	t92	Signed Rank	S	-415.5	Pr >= |S|	<.0001	0	35
93	t93	Signed Rank	S	-418.5	Pr >= |S|	<.0001	0	36
94	t94	Signed Rank	S	-427.5	Pr >= |S|	<.0001	0	37
95	t95	Signed Rank	S	-415.5	Pr >= |S|	<.0001	0	38
96	t96	Signed Rank	S	-366.5	Pr >= |S|	<.0001	0	39
97	t97	Signed Rank	S	-300.5	Pr >= |S|	<.0001	0	40
98	t98	Signed Rank	S	-253.5	Pr >= |S|	0.0009	0	40
99	t99	Signed Rank	S	-191.5	Pr >= |S|	0.0148	0	40
100	t100	Signed Rank	S	-131.5	Pr >= |S|	0.1007	0	40
101	t101	Signed Rank	S	-91.5	Pr >= |S|	0.2575	0	40

102	t102	Signed Rank	S	-21.5	Pr >= |S|	0.7917	0	40
103	t103	Signed Rank	S	31.5	Pr >= |S|	0.6987	0	40
104	t104	Signed Rank	S	124.5	Pr >= |S|	0.1208	0	40
105	t105	Signed Rank	S	248.5	Pr >= |S|	0.0011	0	40
106	t106	Signed Rank	S	364.5	Pr >= |S|	<.0001	1	40
107	t107	Signed Rank	S	423.5	Pr >= |S|	<.0001	2	40
108	t108	Signed Rank	S	440.5	Pr >= |S|	<.0001	3	40
109	t109	Signed Rank	S	443.5	Pr >= |S|	<.0001	4	40
110	t110	Signed Rank	S	435.5	Pr >= |S|	<.0001	5	40
111	t111	Signed Rank	S	417.5	Pr >= |S|	<.0001	6	40
112	t112	Signed Rank	S	378.5	Pr >= |S|	<.0001	7	40
113	t113	Signed Rank	S	340.5	Pr >= |S|	<.0001	8	40
114	t114	Signed Rank	S	287.5	Pr >= |S|	0.0001	9	40
115	t115	Signed Rank	S	252.5	Pr >= |S|	0.0009	9	40
116	t116	Signed Rank	S	234.5	Pr >= |S|	0.0023	9	40
117	t117	Signed Rank	S	237.5	Pr >= |S|	0.002	9	40
118	t118	Signed Rank	S	259.5	Pr >= |S|	0.0006	9	40
119	t119	Signed Rank	S	297.5	Pr >= |S|	<.0001	10	40
120	t120	Signed Rank	S	356.5	Pr >= |S|	<.0001	11	40
121	t121	Signed Rank	S	418.5	Pr >= |S|	<.0001	12	40
122	t122	Signed Rank	S	447.5	Pr >= |S|	<.0001	13	40
123	t123	Signed Rank	S	451.5	Pr >= |S|	<.0001	14	40
124	t124	Signed Rank	S	451.5	Pr >= |S|	<.0001	15	40
125	t125	Signed Rank	S	451.5	Pr >= |S|	<.0001	16	40
126	t126	Signed Rank	S	451.5	Pr >= |S|	<.0001	17	40
127	t127	Signed Rank	S	451.5	Pr >= |S|	<.0001	18	40
128	t128	Signed Rank	S	451.5	Pr >= |S|	<.0001	19	40
129	t129	Signed Rank	S	451.5	Pr >= |S|	<.0001	20	40
130	t130	Signed Rank	S	451.5	Pr >= |S|	<.0001	21	40
131	t131	Signed Rank	S	451.5	Pr >= |S|	<.0001	22	40
132	t132	Signed Rank	S	447.5	Pr >= |S|	<.0001	23	40
133	t133	Signed Rank	S	446.5	Pr >= |S|	<.0001	24	40
134	t134	Signed Rank	S	436.5	Pr >= |S|	<.0001	25	40
135	t135	Signed Rank	S	420.5	Pr >= |S|	<.0001	26	40
136	t136	Signed Rank	S	390.5	Pr >= |S|	<.0001	27	40
137	t137	Signed Rank	S	363.5	Pr >= |S|	<.0001	28	40
138	t138	Signed Rank	S	338.5	Pr >= |S|	<.0001	29	40
139	t139	Signed Rank	S	315.5	Pr >= |S|	<.0001	30	40
140	t140	Signed Rank	S	278.5	Pr >= |S|	0.0002	30	40
141	t141	Signed Rank	S	248.5	Pr >= |S|	0.0011	30	40
142	t142	Signed Rank	S	212.5	Pr >= |S|	0.0063	30	40
143	t143	Signed Rank	S	193.5	Pr >= |S|	0.0137	30	40
144	t144	Signed Rank	S	183.5	Pr >= |S|	0.0199	30	40
145	t145	Signed Rank	S	190.5	Pr >= |S|	0.0153	30	40
146	t146	Signed Rank	S	208.5	Pr >= |S|	0.0075	30	40
147	t147	Signed Rank	S	215.5	Pr >= |S|	0.0056	30	40
148	t148	Signed Rank	S	227.5	Pr >= |S|	0.0032	30	40
149	t149	Signed Rank	S	241.5	Pr >= |S|	0.0016	30	40
150	t150	Signed Rank	S	244.5	Pr >= |S|	0.0014	30	40
151	t151	Signed Rank	S	270.5	Pr >= |S|	0.0003	30	40
152	t152	Signed Rank	S	303.5	Pr >= |S|	<.0001	31	40
153	t153	Signed Rank	S	287.5	Pr >= |S|	0.0001	32	40
154	t154	Signed Rank	S	267.5	Pr >= |S|	0.0004	32	40
155	t155	Signed Rank	S	254.5	Pr >= |S|	0.0008	32	40

156	t156	Signed Rank	S	244.5	Pr >= |S|	0.0014	32	40
157	t157	Signed Rank	S	239.5	Pr >= |S|	0.0018	32	40
158	t158	Signed Rank	S	221.5	Pr >= |S|	0.0043	32	40
159	t159	Signed Rank	S	212.5	Pr >= |S|	0.0063	32	40
160	t160	Signed Rank	S	193.5	Pr >= |S|	0.0137	32	40
161	t161	Signed Rank	S	174.5	Pr >= |S|	0.0273	32	40
162	t162	Signed Rank	S	199.5	Pr >= |S|	0.0108	32	40
163	t163	Signed Rank	S	219.5	Pr >= |S|	0.0047	32	40
164	t164	Signed Rank	S	231.5	Pr >= |S|	0.0027	32	40
165	t165	Signed Rank	S	234.5	Pr >= |S|	0.0023	32	40
166	t166	Signed Rank	S	250.5	Pr >= |S|	0.001	32	40
167	t167	Signed Rank	S	240.5	Pr >= |S|	0.0017	32	40
168	t168	Signed Rank	S	235.5	Pr >= |S|	0.0022	32	40
169	t169	Signed Rank	S	215.5	Pr >= |S|	0.0056	32	40
170	t170	Signed Rank	S	179.5	Pr >= |S|	0.0229	32	40
171	t171	Signed Rank	S	167.5	Pr >= |S|	0.0345	32	40
172	t172	Signed Rank	S	128.5	Pr >= |S|	0.1089	32	40
173	t173	Signed Rank	S	53.5	Pr >= |S|	0.5101	32	40
174	t174	Signed Rank	S	28.5	Pr >= |S|	0.7262	32	40
175	t175	Signed Rank	S	179.5	Pr >= |S|	0.0229	32	40
176	t176	Signed Rank	S	179.5	Pr >= |S|	0.0229	32	40
177	t177	Signed Rank	S	167.5	Pr >= |S|	0.0345	32	40
178	t178	Signed Rank	S	183.5	Pr >= |S|	0.0199	32	40
179	t179	Signed Rank	S	206.5	Pr >= |S|	0.0081	32	40
180	t180	Signed Rank	S	234.5	Pr >= |S|	0.0023	32	40
181	t181	Signed Rank	S	271.5	Pr >= |S|	0.0003	32	40
182	t182	Signed Rank	S	319.5	Pr >= |S|	<.0001	33	40

Appendix B.2 Selected variables for 6 hour Mock vs. Coxsackie

Obs	VarName	Test	Testlab	Stat	pType	P-Value	Positive Num	Negative Num
1	t1	Signed Rank	S	-96	Pr >= |S|	0.1334	0	0
2	t2	Signed Rank	S	-89	Pr >= |S|	0.1651	0	0
3	t3	Signed Rank	S	-86	Pr >= |S|	0.1802	0	0
4	t4	Signed Rank	S	-88	Pr >= |S|	0.1701	0	0
5	t5	Signed Rank	S	-92	Pr >= |S|	0.1509	0	0
6	t6	Signed Rank	S	-97	Pr >= |S|	0.1293	0	0
7	t7	Signed Rank	S	-102	Pr >= |S|	0.11	0	0
8	t8	Signed Rank	S	-100	Pr >= |S|	0.1175	0	0
9	t9	Signed Rank	S	-90	Pr >= |S|	0.1603	0	0
10	t10	Signed Rank	S	-85	Pr >= |S|	0.1855	0	0
11	t11	Signed Rank	S	-81	Pr >= |S|	0.2076	0	0
12	t12	Signed Rank	S	-80	Pr >= |S|	0.2134	0	0
13	t13	Signed Rank	S	-67	Pr >= |S|	0.299	0	0
14	t14	Signed Rank	S	-60	Pr >= |S|	0.3531	0	0
15	t15	Signed Rank	S	-43	Pr >= |S|	0.507	0	0
16	t16	Signed Rank	S	-34	Pr >= |S|	0.6003	0	0
17	t17	Signed Rank	S	-30	Pr >= |S|	0.644	0	0
18	t18	Signed Rank	S	-45	Pr >= |S|	0.4873	0	0
19	t19	Signed Rank	S	-88	Pr >= |S|	0.1701	0	0
20	t20	Signed Rank	S	-121	Pr >= |S|	0.0561	0	0

21	t21	Signed Rank	S	-125	Pr >= |S|	0.048	0	0
22	t22	Signed Rank	S	-106	Pr >= |S|	0.0963	0	0
23	t23	Signed Rank	S	-128	Pr >= |S|	0.0426	0	0
24	t24	Signed Rank	S	-165	Pr >= |S|	0.0076	0	0
25	t25	Signed Rank	S	-216	Pr >= |S|	0.0003	0	0
26	t26	Signed Rank	S	-257	Pr >= |S|	<.0001	0	1
27	t27	Signed Rank	S	-258	Pr >= |S|	<.0001	0	2
28	t28	Signed Rank	S	-263	Pr >= |S|	<.0001	0	3
29	t29	Signed Rank	S	-234	Pr >= |S|	<.0001	0	4
30	t30	Signed Rank	S	-232	Pr >= |S|	<.0001	0	5
31	t31	Signed Rank	S	-270	Pr >= |S|	<.0001	0	6
32	t32	Signed Rank	S	-305	Pr >= |S|	<.0001	0	7
33	t33	Signed Rank	S	-325	Pr >= |S|	<.0001	0	8
34	t34	Signed Rank	S	-330	Pr >= |S|	<.0001	0	9
35	t35	Signed Rank	S	-333	Pr >= |S|	<.0001	0	10
36	t36	Signed Rank	S	-333	Pr >= |S|	<.0001	0	11
37	t37	Signed Rank	S	-307	Pr >= |S|	<.0001	0	12
38	t38	Signed Rank	S	-264	Pr >= |S|	<.0001	0	13
39	t39	Signed Rank	S	-243	Pr >= |S|	<.0001	0	14
40	t40	Signed Rank	S	-239	Pr >= |S|	<.0001	0	15
41	t41	Signed Rank	S	-222	Pr >= |S|	0.0002	0	16
42	t42	Signed Rank	S	-211	Pr >= |S|	0.0004	0	16
43	t43	Signed Rank	S	-229	Pr >= |S|	<.0001	0	17
44	t44	Signed Rank	S	-260	Pr >= |S|	<.0001	0	18
45	t45	Signed Rank	S	-272	Pr >= |S|	<.0001	0	19
46	t46	Signed Rank	S	-239	Pr >= |S|	<.0001	0	20
47	t47	Signed Rank	S	-144	Pr >= |S|	0.0214	0	20
48	t48	Signed Rank	S	-36	Pr >= |S|	0.5789	0	20
49	t49	Signed Rank	S	66	Pr >= |S|	0.3064	0	20
50	t50	Signed Rank	S	133	Pr >= |S|	0.0347	0	20
51	t51	Signed Rank	S	164	Pr >= |S|	0.008	0	20
52	t52	Signed Rank	S	174	Pr >= |S|	0.0046	0	20
53	t53	Signed Rank	S	172	Pr >= |S|	0.0052	0	20
54	t54	Signed Rank	S	166	Pr >= |S|	0.0072	0	20
55	t55	Signed Rank	S	181	Pr >= |S|	0.003	0	20
56	t56	Signed Rank	S	209	Pr >= |S|	0.0005	0	20
57	t57	Signed Rank	S	230	Pr >= |S|	<.0001	1	20
58	t58	Signed Rank	S	260	Pr >= |S|	<.0001	2	20
59	t59	Signed Rank	S	260	Pr >= |S|	<.0001	3	20
60	t60	Signed Rank	S	242	Pr >= |S|	<.0001	4	20
61	t61	Signed Rank	S	193	Pr >= |S|	0.0014	4	20
62	t62	Signed Rank	S	106	Pr >= |S|	0.0963	4	20
63	t63	Signed Rank	S	55	Pr >= |S|	0.3951	4	20
64	t64	Signed Rank	S	-8	Pr >= |S|	0.9021	4	20
65	t65	Signed Rank	S	-71	Pr >= |S|	0.2706	4	20
66	t66	Signed Rank	S	-105	Pr >= |S|	0.0996	4	20
67	t67	Signed Rank	S	-119	Pr >= |S|	0.0605	4	20
68	t68	Signed Rank	S	-132	Pr >= |S|	0.0361	4	20
69	t69	Signed Rank	S	-160	Pr >= |S|	0.0099	4	20
70	t70	Signed Rank	S	-166	Pr >= |S|	0.0072	4	20
71	t71	Signed Rank	S	-164	Pr >= |S|	0.008	4	20

72	t72	Signed Rank	S	-146	Pr >= |S|	0.0195	4	20
73	t73	Signed Rank	S	-126	Pr >= |S|	0.0461	4	20
74	t74	Signed Rank	S	-123	Pr >= |S|	0.0519	4	20
75	t75	Signed Rank	S	-135	Pr >= |S|	0.0319	4	20
76	t76	Signed Rank	S	-147	Pr >= |S|	0.0187	4	20
77	t77	Signed Rank	S	-154	Pr >= |S|	0.0133	4	20
78	t78	Signed Rank	S	-147	Pr >= |S|	0.0187	4	20
79	t79	Signed Rank	S	-139	Pr >= |S|	0.0268	4	20
80	t80	Signed Rank	S	-131	Pr >= |S|	0.0377	4	20
81	t81	Signed Rank	S	-127	Pr >= |S|	0.0443	4	20
82	t82	Signed Rank	S	-133	Pr >= |S|	0.0347	4	20
83	t83	Signed Rank	S	-138	Pr >= |S|	0.028	4	20
84	t84	Signed Rank	S	-120	Pr >= |S|	0.0582	4	20
85	t85	Signed Rank	S	-95	Pr >= |S|	0.1376	4	20
86	t86	Signed Rank	S	-62	Pr >= |S|	0.3371	4	20
87	t87	Signed Rank	S	-37	Pr >= |S|	0.5684	4	20
88	t88	Signed Rank	S	-15	Pr >= |S|	0.8175	4	20
89	t89	Signed Rank	S	14	Pr >= |S|	0.8295	4	20
90	t90	Signed Rank	S	71	Pr >= |S|	0.2706	4	20
91	t91	Signed Rank	S	114	Pr >= |S|	0.0727	4	20
92	t92	Signed Rank	S	139	Pr >= |S|	0.0268	4	20
93	t93	Signed Rank	S	160	Pr >= |S|	0.0099	4	20
94	t94	Signed Rank	S	149	Pr >= |S|	0.017	4	20
95	t95	Signed Rank	S	50	Pr >= |S|	0.4399	4	20
96	t96	Signed Rank	S	-90	Pr >= |S|	0.1603	4	20
97	t97	Signed Rank	S	-184	Pr >= |S|	0.0025	4	20
98	t98	Signed Rank	S	-211	Pr >= |S|	0.0004	4	20
99	t99	Signed Rank	S	-203	Pr >= |S|	0.0007	4	20
100	t100	Signed Rank	S	-182	Pr >= |S|	0.0029	4	20
101	t101	Signed Rank	S	-147	Pr >= |S|	0.0187	4	20
102	t102	Signed Rank	S	-119	Pr >= |S|	0.0605	4	20
103	t103	Signed Rank	S	-92	Pr >= |S|	0.1509	4	20
104	t104	Signed Rank	S	-61	Pr >= |S|	0.345	4	20
105	t105	Signed Rank	S	-28	Pr >= |S|	0.6663	4	20
106	t106	Signed Rank	S	-21	Pr >= |S|	0.7465	4	20
107	t107	Signed Rank	S	-17	Pr >= |S|	0.7936	4	20
108	t108	Signed Rank	S	-55	Pr >= |S|	0.3951	4	20
109	t109	Signed Rank	S	-119	Pr >= |S|	0.0605	4	20
110	t110	Signed Rank	S	-166	Pr >= |S|	0.0072	4	20
111	t111	Signed Rank	S	-125	Pr >= |S|	0.048	4	20
112	t112	Signed Rank	S	-124	Pr >= |S|	0.0499	4	20
113	t113	Signed Rank	S	-128	Pr >= |S|	0.0426	4	20
114	t114	Signed Rank	S	-128	Pr >= |S|	0.0426	4	20
115	t115	Signed Rank	S	-133	Pr >= |S|	0.0347	4	20
116	t116	Signed Rank	S	-141	Pr >= |S|	0.0245	4	20
117	t117	Signed Rank	S	-146	Pr >= |S|	0.0195	4	20
118	t118	Signed Rank	S	-156	Pr >= |S|	0.0121	4	20
119	t119	Signed Rank	S	-157	Pr >= |S|	0.0115	4	20
120	t120	Signed Rank	S	-152	Pr >= |S|	0.0147	4	20
121	t121	Signed Rank	S	-126	Pr >= |S|	0.0461	4	20
122	t122	Signed Rank	S	-75	Pr >= |S|	0.244	4	20

123	t123	Signed Rank	S	32	Pr >= |S|	0.622	4	20
124	t124	Signed Rank	S	140	Pr >= |S|	0.0256	4	20
125	t125	Signed Rank	S	204	Pr >= |S|	0.0007	4	20
126	t126	Signed Rank	S	264	Pr >= |S|	<.0001	5	20
127	t127	Signed Rank	S	272	Pr >= |S|	<.0001	6	20
128	t128	Signed Rank	S	269	Pr >= |S|	<.0001	7	20
129	t129	Signed Rank	S	266	Pr >= |S|	<.0001	8	20
130	t130	Signed Rank	S	254	Pr >= |S|	<.0001	9	20
131	t131	Signed Rank	S	248	Pr >= |S|	<.0001	10	20
132	t132	Signed Rank	S	232	Pr >= |S|	<.0001	11	20
133	t133	Signed Rank	S	217	Pr >= |S|	0.0002	11	20
134	t134	Signed Rank	S	205	Pr >= |S|	0.0006	11	20
135	t135	Signed Rank	S	189	Pr >= |S|	0.0018	11	20
136	t136	Signed Rank	S	171	Pr >= |S|	0.0055	11	20
137	t137	Signed Rank	S	160	Pr >= |S|	0.0099	11	20
138	t138	Signed Rank	S	147	Pr >= |S|	0.0187	11	20
139	t139	Signed Rank	S	138	Pr >= |S|	0.028	11	20
140	t140	Signed Rank	S	135	Pr >= |S|	0.0319	11	20
141	t141	Signed Rank	S	122	Pr >= |S|	0.0539	11	20
142	t142	Signed Rank	S	103	Pr >= |S|	0.1065	11	20
143	t143	Signed Rank	S	98	Pr >= |S|	0.1253	11	20
144	t144	Signed Rank	S	100	Pr >= |S|	0.1175	11	20
145	t145	Signed Rank	S	107	Pr >= |S|	0.093	11	20
146	t146	Signed Rank	S	112	Pr >= |S|	0.0781	11	20
147	t147	Signed Rank	S	118	Pr >= |S|	0.0628	11	20
148	t148	Signed Rank	S	119	Pr >= |S|	0.0605	11	20
149	t149	Signed Rank	S	114	Pr >= |S|	0.0727	11	20
150	t150	Signed Rank	S	111	Pr >= |S|	0.081	11	20
151	t151	Signed Rank	S	133	Pr >= |S|	0.0347	11	20
152	t152	Signed Rank	S	157	Pr >= |S|	0.0115	11	20
153	t153	Signed Rank	S	196	Pr >= |S|	0.0012	11	20
154	t154	Signed Rank	S	225	Pr >= |S|	0.0001	12	20
155	t155	Signed Rank	S	255	Pr >= |S|	<.0001	13	20
156	t156	Signed Rank	S	264	Pr >= |S|	<.0001	14	20
157	t157	Signed Rank	S	255	Pr >= |S|	<.0001	15	20
158	t158	Signed Rank	S	195	Pr >= |S|	0.0012	15	20
159	t159	Signed Rank	S	132	Pr >= |S|	0.0361	15	20
160	t160	Signed Rank	S	79	Pr >= |S|	0.2193	15	20
161	t161	Signed Rank	S	74	Pr >= |S|	0.2505	15	20
162	t162	Signed Rank	S	67	Pr >= |S|	0.299	15	20
163	t163	Signed Rank	S	88	Pr >= |S|	0.1701	15	20
164	t164	Signed Rank	S	96	Pr >= |S|	0.1334	15	20
165	t165	Signed Rank	S	108	Pr >= |S|	0.0899	15	20
166	t166	Signed Rank	S	134	Pr >= |S|	0.0332	15	20
167	t167	Signed Rank	S	130	Pr >= |S|	0.0393	15	20
168	t168	Signed Rank	S	128	Pr >= |S|	0.0426	15	20
169	t169	Signed Rank	S	124	Pr >= |S|	0.0499	15	20
170	t170	Signed Rank	S	99	Pr >= |S|	0.1213	15	20
171	t171	Signed Rank	S	81	Pr >= |S|	0.2076	15	20
172	t172	Signed Rank	S	24	Pr >= |S|	0.7118	15	20
173	t173	Signed Rank	S	-43	Pr >= |S|	0.507	15	20

174	t174	Signed Rank	S	-82	Pr >= |S|	0.2019	15	20
175	t175	Signed Rank	S	74	Pr >= |S|	0.2505	15	20
176	t176	Signed Rank	S	80	Pr >= |S|	0.2134	15	20
177	t177	Signed Rank	S	75	Pr >= |S|	0.244	15	20
178	t178	Signed Rank	S	88	Pr >= |S|	0.1701	15	20
179	t179	Signed Rank	S	111	Pr >= |S|	0.081	15	20
180	t180	Signed Rank	S	127	Pr >= |S|	0.0443	15	20
181	t181	Signed Rank	S	162	Pr >= |S|	0.0089	15	20
182	t182	Signed Rank	S	195	Pr >= |S|	0.0012	15	20

APPENDIX D: SAS Code

Appendix D.1 SAS code for data clean (data import, standardized, randomly split into two groups and construct inner-difference and intra-difference)

D.1.1 First Date Group for 6 hour Mock vs. Hsv1 paired comparison

```
/***********************************************************************/
/*          import data Mock & Hsv1 data(6 hour)-032608
*/
/***********************************************************************/

proc import datafile="G:\code running in lab\new data\032608\v-mock-human-t-
nofix-6hpi-032808-2cm-1_2000-700 filter.csv"
out=sixh.mock1 replace;
run;
proc import datafile="G:\code running in lab\new data\032608\v-hsv1-human-t-
nofix-6hpi-033108-2cm-1_2000-700 filter.csv"
out=sixh.hsv1 replace;
run;

data mock6h;
        if 1=1 then delete;
       run;
data hsv6h;
        if 1=1 then delete;
       run;

data mock6h;
    merge mock6h sixh.mock1(firstobs=5 rename=(wavenumber=VAR2));
run;
data hsv6h;
    merge hsv6h sixh.hsv1(firstobs=5 rename=(wavenumber=VAR2));
run;

data sixh.mock6h;
    set mock6h(drop=xlabel);
    mock1=var2*1;
    rename var3-var58=mock2-mock57;
    drop var2;
run;
data sixh.hsv6h;
    set hsv6h(drop=xlabel);
    hsv1=var2*1;
    rename var3-var71=hsv2-hsv70;
     drop var2;
run;

/**************************/
/*  standardize           */
/**************************/
proc means data=sixh.mock6h;
    var mock1-mock57;
    output out=mockmean mean(mock1-mock57)=mock1-mock57;
    output out=mockstd std(mock1-mock57)=mock1-mock57;
run;
```

```
proc means data=sixh.hsv6h;
    var hsv1-hsv70;
    output out=hsvmean mean(hsv1-hsv70)=hsv1-hsv70;
    output out=hsvstd std(hsv1-hsv70)=hsv1-hsv70;
run;

data mockmean; /*mean*/
    set mockmean;
    drop _freq_ _type_;
run;
data hsvmean; /*mean*/
    set hsvmean;
    drop _freq_ _type_;
run;

data mockstd; /*std*/
    set mockstd;
    drop _freq_ _type_;
run;
data hsvstd; /*std*/
    set hsvstd;
    drop _freq_ _type_;
run;

data mock6h;
    set sixh.mock6h mockmean mockstd;
run;
data hsv6h;
    set sixh.hsv6h hsvmean hsvstd;
run;

proc transpose data=mock6h out=mock6htr name=cell prefix=v;
    var mock1-mock57;
run;
proc transpose data=hsv6h out=hsv6htr name=cell prefix=v;
    var hsv1-hsv70;
run;

data mock6htr;
    set mock6htr;
    rename v729=mean v730=std;
run;
data hsv6htr;
    set hsv6htr;
    rename v729=mean v730=std;
run;

%macro mockstd;
%do i=1 %to 728;
data mock6htr;
    set mock6htr;
    v&i=(v&i-mean)/std;
run;
%end;
%mend;
%mockstd
```

```
%macro hsvstd;
%do i=1 %to 728;
data hsv6htr;
    set hsv6htr;
    v&i=(v&i-mean)/std;
run;
%end;
%mend;
%hsvstd

/************************************************************************/
/*       randomly separate into 2 groups(mock 57)(hsv 70)               */
/************************************************************************/
data mocksplit;
  set mock6htr;
  drop mean std;
run;

data mocksp1;
set mocksplit;
retain n 0;
n=n+1;
index=ranuni(370548);
run;

proc sort data=mocksp1;
by index;
run;

data mocksp1;
set mocksp1;
retain m 0;
m=m+1;
run;

data mock6h1sp1 mock6h1sp2;
set mocksp1;
    if m>=1 & m<=28 then output mock6h1sp1;
    if m>=29 & m<=57 then output mock6h1sp2;
run;

data hsvsplit;
  set hsv6htr;
  drop mean std;
run;

data hsvsp1;
set hsvsplit;
retain n 0;
n=n+1;
index=ranuni(674647);
run;

proc sort data=hsvsp1;
by index;
run;
```

```
data hsvsp1;
set hsvsp1;
retain m 0;
m=m+1;
run;

data hsv6h1sp1 hsv6h1sp2;
set hsvsp1;
    if m>=1 & m<=35 then output hsv6h1sp1;
    if m>=36 & m<=70 then output hsv6h1sp2;
run;

/*****************************************************************/
/*          take average & difference dt1 dt2 dn1 dn2           */
/*****************************************************************/
data avm1;
set mock6h1sp1;
drop cell index n m;
run;
data avm2;
set mock6h1sp2;
drop cell index n m;
run;
data avh1;
set hsv6h1sp1;
drop cell index n m;
run;
data avh2;
set hsv6h1sp2;
drop cell index n m;
run;

proc transpose data=avm1 out=am1 prefix=v;
run;
data am1;
set am1;
tm1=sum(of v1-v28);
am1=tm1/28;
run;

proc transpose data=avm2 out=am2 prefix=v;
run;
data am2;
set am2;
tm2=sum(of v1-v29);
am2=tm2/29;
run;

proc transpose data=avh1 out=ah1 prefix=v;
run;
data ah1;
set ah1;
th1=sum(of v1-v35);
ah1=th1/35;
run;

proc transpose data=avh2 out=ah2 prefix=v;
```

```
run;
data ah2;
set ah2;
th2=sum(of v1-v35);
ah2=th2/35;
run;

data sixh.a1m1;
set am1;
keep am1;
run;
data sixh.a1m2;
set am2;
keep am2;
run;
data sixh.a1h1;
set ah1;
keep ah1;
run;
data sixh.a1h2;
set ah2;
keep ah2;
run;

data dt1;
merge sixh.a1m1 sixh.a1h1;
dt1=am1-ah1;
run;
data sixh.dt1;
set dt1;
keep dt1;
run;

data dt2;
merge sixh.a1m2 sixh.a1h2;
dt2=am2-ah2;
run;
data sixh.dt2;
set dt2;
keep dt2;
run;

data dn1;
merge sixh.a1m1 sixh.a1m2;
dn1=am1-am2;
run;
data sixh.dn1;
set dn1;
keep dn1;
run;

data dn2;
merge sixh.a1h1 sixh.a1h2;
dn2=ah1-ah2;
run;
data sixh.dn2;
set dn2;
```

```
keep dn2;
run;
```

D.1.2 Adeno of all dates groups

```
/**************************/
/*  Group 1 (032608)      */
/**************************/
libname sixh 'G:\sixh';
proc import datafile="G:\code running in lab\new data\032608\v-had1-human-t-
nofix-6hpi-040108-2cm-1_2000-700 filter.csv"
out=sixh.adeno1 replace;
run;

data adeno6h;
        if 1=1 then delete;
       run;

data adeno6h;
   merge adeno6h sixh.adeno1(firstobs=5 rename=(wavenumber=VAR2));
run;

data sixh.adeno6h;
   set adeno6h(drop=xlabel);
   adeno1=var2*1;
   rename var3-var45=adeno2-adeno44;
   drop var2;
run;

/**************************/
/*  standardize           */
/**************************/
proc means data=sixh.adeno6h;
   var adeno1-adeno44;
   output out=adenomean mean(adeno1-adeno44)=adeno1-adeno44;
   output out=adenostd std(adeno1-adeno44)=adeno1-adeno44;
run;

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

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

data adeno6h;set sixh.adeno6h adenomean adenostd;run;

proc transpose data=adeno6h out=adeno6htr name=cell prefix=v;
   var adeno1-adeno44;
run;

data adeno6htr;set adeno6htr;rename v729=mean v730=std;run;

%macro adenostd;
%do i=1 %to 728;
data adeno6htr;
   set adeno6htr;
```

```
    v&i=(v&i-mean)/std;
run;
%end;
%mend;
%adenostd

/************************************************************************/
/*      randomly separate into 2 groups(adeno 44)                       */
/************************************************************************/
data adenosplit;set adeno6htr;drop mean std;run;

data adenosp1;
set adenosplit;
retain n 0;
n=n+1;
index=ranuni(752226);
run;

proc sort data=adenosp1;
by index;
run;

data adenosp1;
set adenosp1;
retain m 0;
m=m+1;
run;

data adeno6h1sp1 adeno6h1sp2;
set adenosp1;
    if m>=1 & m<=22 then output adeno6h1sp1;
   if m>=23 & m<=44 then output adeno6h1sp2;
run;

/*****************************************************************/
/*          take average & difference dt1 dt2 dn1 dn2            */
/*****************************************************************/
data ava1;set adeno6h1sp1;drop cell index n m;run;
data ava2;set adeno6h1sp2;drop cell index n m;run;

proc transpose data=ava1 out=aa1 prefix=v;run;
data aa1;set aa1;ta1=sum(of v1-v22);aa1=ta1/22;run;
proc transpose data=ava2 out=aa2 prefix=v;run;
data aa2;set aa2;ta2=sum(of v1-v22);aa2=ta2/22;run;

data sixh.a1a1;set aa1;keep aa1;run;/*a1a1 & a1a2 is adeno; a1m1 & a1m2 is
mock;a1h1 & a1h2 is hsv*/
data sixh.a1a2;set aa2;keep aa2;run;

/*dt,dn for mock_hsv;dt_m_a is for mock & adeno; dt_h_a is for hsv & adeno*/
data dt_ma1;merge sixh.a1m1 sixh.a1a1;dt_ma1=am1-aa1;run;
data sixh.dt_ma1;set dt_ma1;keep dt_ma1;run;

data dt_ma2;merge sixh.a1m2 sixh.a1a2;dt_ma2=am2-aa2;run;
data sixh.dt_ma2;set dt_ma2;keep dt_ma2;run;

data dn_a;merge sixh.a1a1 sixh.a1a2;dn_a=aa1-aa2;run;
data sixh.dn_ma2;set dn_a;keep dn_a;run;
```

Appendix D.2 F-test

```
data inner;
set sixh.inner_discrmin_mock_hsv1(keep=inner_discriminator);
run;

data sixh.inner;
set inner;
proc print;run;

/***************************/
/*F-test for Mock vs. Hsv1 */
/***************************/
data mock;
set inner;
if(mod(_N_,2)=1); *mod(_N_,2)=1: odds, mod(_N_,2)=0: even;
*_N_=1;
run;

data hsv;
set inner;
if (mod(_N_,2)=0); *mod(_N_,2)=1: odds, mod(_N_,2)=0: even;
*_N_=2;
run;


data f_test;
input type innerdif;
datalines;
1   -3.10433
2   0.76149
1   -1.82758
2   -0.41556
1   4.81687
2   1.24504
1   -1.78082
2   3.06975
1   3.60794
2   2.23788
1   -0.51593
2   -0.93661
1   1.60792
2   3.396
1   -1.12962
2   -0.86353
1   0.23016
2   0.88635
1   0.11559
2   3.93707
1   -0.65214
2   4.08347
1   -1.07944
2   -0.51139
;
```

```
proc glm data=f_test;
class type;
model innerdif = type;
means type;
run;

proc print;run;
```

Appendix D.3 Compute Specificity and AUC of model with positive terms minus negative terms for 6 hour Mock vs. Hsv1 paired comparisons, generating multivariate normal distribution and bootstrap for confidence interval

```
libname sixh 'E:\code running in lab\new data\sixh';
libname twoh 'E:\code running in lab\new data\library';

data sixh.intdif;
merge sixh.dt1 sixh.dt2 sixh.dt3 sixh.dt4 sixh.dt3 sixh.dt5 sixh.dt6 sixh.dt7
sixh.dt8 sixh.dt9 sixh.dt10 sixh.dt11 sixh.dt12
sixh.dt13 sixh.dt14 sixh.dt15 sixh.dt16 sixh.dt17 sixh.dt18 sixh.dt19
sixh.dt20 sixh.dt21 sixh.dt22 sixh.dt23 sixh.dt24
sixh.dt25 sixh.dt26 sixh.dt27 sixh.dt28 sixh.dt29 sixh.dt30 sixh.dt31
sixh.dt32 sixh.dt33 sixh.dt34 sixh.dt35 sixh.dt36
sixh.dt37 sixh.dt38 sixh.dt39 sixh.dt40 sixh.dt41 sixh.dt42;
run;

data sixh.inndif;
merge sixh.dn1 sixh.dn2 sixh.dn3 sixh.dn4 sixh.dn3 sixh.dn5 sixh.dn6 sixh.dn7
sixh.dn8 sixh.dn9 sixh.dn10 sixh.dn11 sixh.dn12
sixh.dn13 sixh.dn14 sixh.dn15 sixh.dn16 sixh.dn17 sixh.dn18 sixh.dn19
sixh.dn20 sixh.dn21 sixh.dn22 sixh.dn23 sixh.dn24
sixh.dn25 sixh.dn26 sixh.dn27 sixh.dn28 sixh.dn29 sixh.dn30 sixh.dn31
sixh.dn32 sixh.dn33 sixh.dn34 sixh.dn35 sixh.dn36
sixh.dn37 sixh.dn38 sixh.dn39 sixh.dn40 sixh.dn41 sixh.dn42;
run;

proc transpose data=sixh.intdif out=sixh.intdiftr prefix=v;
run;

/******************************************************/
/*         one-tailed wilcoxon rank test (P-value)        */
/******************************************************/
/*data try;
set sixh.intdiftr(keep=v1 v2);
run;
*/

ods trace on;
ods listing close;

proc univariate data=sixh.intdiftr;
ods trace off;
ods output  TestsForLocation=t1; run;

data t2;
set t1;
```

```
if Testlab="S" then output;
run;

data sixh.mock_hsv1_unip;
set t2(keep=Stat pValue);
run;

data sixh.mock_hsv1_unip;
merge twoh.xt sixh.mock_hsv1_unip;
run;

/*******p-value 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"P-value" )order=(0 to 0.1 by 0.01)
      major=(height=1) minor=(height=0.5)
      width=3;

title 'P-value for 6 hour Mock vs. Hsv1(21 groups)';
 proc gplot data=sixh.mock_hsv1_unip;
   plot pValue*XLabe2 / overlay
                             haxis=axis1 hminor=4
                             vaxis=axis2 vminor=4
                          vref=0.01   lvref=5;
run;
quit;

/*******Statistic 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"Statistic" )order=(-460 to 490 by 30)
      major=(height=2) minor=(height=1)
      width=3;

title 'Statistic for 6 hour Mock vs. Hsv1(21 groups)';
 proc gplot data=sixh.mock_hsv1_unip;
   plot Stat*XLabe2 / overlay
                             haxis=axis1 hminor=4
                             vaxis=axis2 vminor=4
                          vref=350 -350 lvref=5;
run;
```

```
quit;

data try;
set sixh.mock_hsv1_unip;
*if Stat<-350 & pValue<0.0002 then inf=1;
if Stat>350 & pValue<0.0002 then inf=2;
else inf=0;
run;

proc print data=try;
run;


data t;
merge twoh.xt sixh.intdif;
run;
data n;
merge twoh.xt sixh.inndif;
run;

/*
proc print data=sixh.xt;
run;
*/

/************************************************************************/
/*         find 20 discriminators for intra-difference                 */
/************************************************************************/
data dtminus;
set t;
if XLabe2>=1045 & XLabe2<=1078 or XLabe2>=1096 & XLabe2<=1105 or XLabe2>=1129
& XLabe2<=1167;
drop XLabe2;
run;

data dtadd;
set t;
if XLabe2>=1205 & XLabe2<=1231 or XLabe2>=1260 & XLabe2<=1327;
drop XLabe2;
run;

proc transpose data=dtadd out=dtatr prefix=v;
run;

data dtatr;
set dtatr;
t1=sum(of v1-v96);
run;

proc transpose data=dtminus out=dtmtr prefix=v;
run;

data dtmtr;
set dtmtr;
t2=sum(of v1-v83);
run;
```

```
data dtotal;
merge dtatr(keep=t1) dtmtr(keep=t2);
run;

data sixh.intra_discrmin_mock_hsv1;
set dtotal;
intra_discriminator=t1-t2;
run;

proc print data=sixh.intra_discrmin_mock_hsv1;
run;
/*************************************************************************/
/*         find 12 discriminators for inner-difference                   */
/*************************************************************************/
data dtminus;
set n;
if XLabe2>=1045 & XLabe2<=1078 or XLabe2>=1096 & XLabe2<=1105 or XLabe2>=1129
& XLabe2<=1167;
drop XLabe2;
run;

data dtadd;
set n;
if XLabe2>=1205 & XLabe2<=1231 or XLabe2>=1260 & XLabe2<=1327;
drop XLabe2;
run;

proc transpose data=dtadd out=dtatr prefix=v;
run;

data dtatr;
set dtatr;
t1=sum(of v1-v96);
run;

proc transpose data=dtminus out=dtmtr prefix=v;
run;

data dtmtr;
set dtmtr;
t2=sum(of v1-v83);
run;

data dtotal;
merge dtatr(keep=t1) dtmtr(keep=t2);
run;

data sixh.inner_discrmin_mock_hsv1;
set dtotal;
inner_discriminator=t1-t2;
run;

proc print data=sixh.inner_discrmin_mock_hsv1;
run;

/*************************************************************************/
/*        two normal distribution,find mean and stadardization           */
```

```
/*************************************************************************/
/*************************/
/*        intra          */
/*************************/

data intra;
set sixh.intra_discrmin_mock_hsv1(keep=intra_discriminator);
run;
/*
proc print data=intra;
run;
 */
proc means data=intra mean std;
 var intra_discriminator;
 output out=meansd mean=meanintra std=sdintra;
 run;

 data _null_;
   set meansd;
   call symput('intramean',trim(left(meanintra)));
   call symput('intrasd',trim(left(sdintra)));
 run;

data intra;
 set intra;
 standardized=(intra_discriminator-&intramean)/&intrasd;
run;

proc print;run;

/*************************/
/*        inner              */
/*************************/
data inner;
set sixh.inner_discrmin_mock_hsv1(keep=inner_discriminator);
run;

proc means data=inner mean std;
 var inner_discriminator;
 output out=meansd1 mean=meaninner std=sdinner;
 run;

 data _null_;
   set meansd1;
   call symput('innermean',trim(left(meaninner)));
   call symput('innersd',trim(left(sdinner)));
 run;

data inner;
 set inner;
 standardized=(inner_discriminator-&innermean)/&innersd;
run;

proc print;run;


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

```
/*  step 1: Compute sample mean & covriance-variance matrix           */
/**********************************************************************/
/*************************/
/*      M1-M2  & H1-H2    */
/*************************/
data inner;set sixh.inner_discrmin_mock_hsv1(keep=inner_discriminator);run;
data mock;set inner;if(mod(_N_,2)=1); *mod(_N_,2)=1: odds, mod(_N_,2)=0:
even;*_N_=1;run;
data mock;set mock (rename=(inner_discriminator=x1));run;
data hsv;set inner;if (mod(_N_,2)=0); *mod(_N_,2)=1: odds, mod(_N_,2)=0:
even;*_N_=2;run;
data hsv;set hsv (rename=(inner_discriminator=x2));run;
/*************************/
/*   M1-H1  & M2-H2       */
/*************************/
data intra;set sixh.intra_discrmin_mock_hsv1(keep=intra_discriminator);run;
data intraone;set intra;if(mod(_N_,2)=1); *mod(_N_,2)=1: odds, mod(_N_,2)=0:
even;run;
data intraone;set intraone (rename=(intra_discriminator=x3));run;
data intratwo;set intra;if (mod(_N_,2)=0); *mod(_N_,2)=1: odds, mod(_N_,2)=0:
even;run;
data intratwo;set intratwo (rename=(intra_discriminator=x4));run;
/**************************************************************************/
/*M1-M2(mock-x1) & H1-H2(hsv-x2) & M1-H1(intraone-x3) & M2-H2(intratwo-x4)*/
/**************************************************************************/
data sample;merge mock hsv intraone intratwo;run;

 proc corr data=sample cov outp=outcov(type=cov) nocorr;
      var x1 x2 x3 x4;
     * by _Imputation_;
   run;

   proc print data=outcov; title 'Sample Means and Covariance Matrices';run;


/**********************************************************************/
/*  step 2: Generate the multivariate normal data in Macro(12 times) */
/**********************************************************************/
data sixh.outcov;set outcov;run;
/* Cholesky Decomposition *//*please see reference*/
%macro multivariate(varcov=, means=, n=, mul=, seed=);
    /* arguments for the macro:
    1. covcov: data set for variance-covariance matrix
    2. means: data set for mean vector
    3. n: sample size
    4. mul: output data set name */
    proc iml;
    use &varcov; /* read in data for variance-covariance matrix */
    read all into sigma;
    use &means; /* read in data for means */
    read all into mu;
    p = nrow(sigma); /* calculate number of variables */
    n = &n;
    l = t(half(sigma)); /* calculate cholesky root of cov matrix */
    z = normal(j(p,&n,&seed)); /* generate nvars*samplesize normals */
    y = l*z; /* premultiply by cholesky root */
    yall = t(repeat(mu,1,&n)+y); /* add in the means */
    varnames = { x1 x2 y1 y2 };
    create &mul from yall (|colname = varnames|);
    append from yall;
    quit;
%mend multivariate;
data mean;
input x @@;
cards;
 -0.3049 -0.0659 8.9132 9.1522
;
run;
data varcov;
input x1-x4;
cards;
1.2419 -0.3459 2.2908 0.7030
-0.3459 3.5671 -1.1854 2.7276
2.2908 -1.1854 20.1612 16.6850
0.7030 2.7276 16.6850 18.7096
;
run;

%macro average(iter);
%do i=0 %to &iter;
    %multivariate(varcov=varcov, means=mean, n=21, mul=mvnormal, seed=&i)

    data x1;set mvnormal(keep=x1 rename=(x1=x));run;
    data x2;set mvnormal(keep=x2 rename=(x2=x));run;
    data x;set x1 x2;run;
    data y1;set mvnormal(keep=y1 rename=(y1=y));run;
    data y2;set mvnormal(keep=y2 rename=(y2=y));run;
    data y;set y1 y2;run;

    /* Step 3: compute mean,std for x and y */
     proc means noprint data=x mean std; var x; output out=xnormal mean=meanx
std=stdx;run;
     proc means noprint data=y mean std; var y; output out=ynormal mean=meany
std=stdy;run;

    data spec;merge xnormal ynormal;run;
    data spec;set spec(drop=_type_ _freq_);run;

    /* Step 4: compute spec1, spec2, spec3, AUC*/
    data norm;/*PROBIT() and PROBNORM()*/
    set spec;
    cutpt1=meany+(-1.6448536)*stdy;tr1=(cutpt1-
meanx)/stdx;spec1=probnorm(tr1);/*95% sensitivity*/
    cutpt2=meany+(-1.2815516)*stdy;tr2=(cutpt2-
meanx)/stdx;spec2=probnorm(tr2);/*90% sensitivity*/
    cutpt3=meany+(-0.8416212)*stdy;tr3=(cutpt3-
meanx)/stdx;spec3=probnorm(tr3);/*80% sensitivity*/
    se=sqrt(stdy*stdy+stdx*stdx);meandiff=meany-meanx;tile=meandiff/se;
    AUC=probnorm(tile);
    run;
    data normal;set norm (keep=spec1 spec2 spec3 AUC);run;
    data result;set result normal ;run;
%end;
%mend average;
```

```
/* Step 5: Repeat step1-step4 1000 times */
%average(1000);

/*proc print data=result;run;*/

data spec1;set result(keep=spec1);run;proc sort data=spec1 out=specficity1;by
spec1;run;
data spec2;set result(keep=spec2);run;proc sort data=spec2 out=specficity2;by
spec2;run;
data spec3;set result(keep=spec3);run;proc sort data=spec3 out=specficity3;by
spec3;run;
data AUC;set result(keep=AUC);run;proc sort data=AUC out=AUCnew;by AUC;run;

data toresult;merge specficity1 specficity2 specficity3 AUCnew;run;
proc print data=toresult;run;

data sixh.bootstrapdata;set toresult;run;


data s;set sixh.bootstrapdata;run;
proc print data=s;run;
```

Appendix D.4 Compute Specificity and AUC of PLSR for 6 hour Mock vs. Hsv1 paired comparisons, generating multivariate normal distribution and bootstrap for confidence interval

```
libname sixh 'G:\code running in lab\new data\sixh';
libname twoh 'G:\code running in lab\new data\library';

data t;
merge twoh.xt sixh.intdif;
run;
data n;
merge twoh.xt sixh.inndif;
run;

data intra;
set t;
if XLabe2>=1045 & XLabe2<=1079 or XLabe2>=1098 & XLabe2<=1103 or XLabe2>=1129
& XLabe2<=1167
or XLabe2>=1206 & XLabe2<=1230 or XLabe2>=1260 & XLabe2<=1327.5;
run;

data inner;
set n;
if XLabe2>=1045 & XLabe2<=1079 or XLabe2>=1098 & XLabe2<=1103 or XLabe2>=1129
& XLabe2<=1167
or XLabe2>=1206 & XLabe2<=1230 or XLabe2>=1260 & XLabe2<=1327.5;
run;

data total;merge intra(drop=Xlabe2) inner(drop=Xlabe2);run;
proc transpose data=total out=totaltr prefix=v; run;
```

```
%macro sumby5;
data sumby5;
    set totaltr;
    %do i=1 %to 7;
        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=8 %to 8;
        c&i=0;
        %do j=0 %to 4;
            %let m=%sysevalf(36+5*(&i-7-1)+&j, integer);
            c&i=c&i+v&m;
         %end;
        c&i=c&i/5;
    %end;

    %do i=9 %to 16;
        c&i=0;
        %do j=0 %to 4;
            %let m=%sysevalf(41+5*(&i-8-1)+&j, integer);
            c&i=c&i+v&m;
         %end;
        c&i=c&i/5;
    %end;

    %do i=17 %to 21;
        c&i=0;
        %do j=0 %to 4;
            %let m=%sysevalf(81+5*(&i-16-1)+&j, integer);
            c&i=c&i+v&m;
         %end;
        c&i=c&i/5;
    %end;

    %do i=22 %to 35;
        c&i=0;
        %do j=0 %to 4;
            %let m=%sysevalf(106+5*(&i-21-1)+&j, integer);
            c&i=c&i+v&m;
         %end;
        c&i=c&i/5;
    %end;
keep c1-c35;
run;
%mend;

%sumby5
```

```
data sixh.pls_mock_hsv;set sumby5;inf=1;if _n_ >=43 then inf=0;run;

proc print data=one;run;

proc pls data =sixh.pls_mock_hsv /*cv=split(10)cv=random*/  details;
   model inf=c1-c35/solution;
output out=one P=PRED;
run;

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

/*        intra          */

data intra;set one (keep=PRED inf rename=(PRED=intra_discriminator));if
inf=1;drop inf;run;

proc means data=intra mean std;
 var intra_discriminator;
 output out=meansd mean=meanintra std=sdintra;
 run;

/*        inner          */

data inner;set one (keep=PRED inf rename=(PRED=inner_discriminator));if
inf=0;drop inf;run;

proc means data=inner mean std;
 var inner_discriminator;
 output out=meansd1 mean=meaninner std=sdinner;
 run;


/*************************/
/*      M1-M2  & H1-H2    */
/*************************/
data mock;set inner;if(mod(_N_,2)=1); *mod(_N_,2)=1: odds, mod(_N_,2)=0:
even;*_N_=1;run;
data mock;set mock (rename=(inner_discriminator=x1));run;
data hsv;set inner;if (mod(_N_,2)=0); *mod(_N_,2)=1: odds, mod(_N_,2)=0:
even;*_N_=2;run;
data hsv;set hsv (rename=(inner_discriminator=x2));run;
/*************************/
/*   M1-H1  & M2-H2       */
/*************************/
data intraone;set intra;if(mod(_N_,2)=1); *mod(_N_,2)=1: odds, mod(_N_,2)=0:
even;run;
data intraone;set intraone (rename=(intra_discriminator=x3));run;
data intratwo;set intra;if (mod(_N_,2)=0); *mod(_N_,2)=1: odds, mod(_N_,2)=0:
even;run;
data intratwo;set intratwo (rename=(intra_discriminator=x4));run;
/****************************************************************************
***/
/*   M1-M2(mock-x1) & H1-H2(hsv-x2) & M1-H1(intraone-x3) & M2-H2(intratwo-x4)
*/
/****************************************************************************
```

```
***/
data sample;merge mock hsv intraone intratwo;run;

 proc corr data=sample cov outp=outcov(type=cov) nocorr;
      var x1 x2 x3 x4;
     * by _Imputation_;
   run;

   proc print data=outcov; title 'Sample Means and Covariance Matrices';run;

/*********************************************************************/
/*  step 2: Generate the multivariate normal data in Macro(21 times) */
/*********************************************************************/

/* Cholesky Decomposition *//*please see reference*/
%macro multivariate(varcov=, means=, n=, mul=, seed=);
   /* arguments for the macro:
   1. covcov: data set for variance-covariance matrix
   2. means: data set for mean vector
   3. n: sample size
   4. mul: output data set name */
   proc iml;
   use &varcov; /* read in data for variance-covariance matrix */
   read all into sigma;
   use &means; /* read in data for means */
   read all into mu;
   p = nrow(sigma); /* calculate number of variables */
   n = &n;
   l = t(half(sigma)); /* calculate cholesky root of cov matrix */
   z = normal(j(p,&n,&seed)); /* generate nvars*samplesize normals */
   y = l*z; /* premultiply by cholesky root */
   yall = t(repeat(mu,1,&n)+y); /* add in the means */
   varnames = { x1 x2 y1 y2 };
   create &mul from yall (|colname = varnames|);
   append from yall;
   quit;
%mend multivariate;
data mean;
input x @@;
cards;
0.0068 0.0183 0.9817 0.9932
;
run;
data varcov;
input x1-x4;
cards;
0.0085 0.0004 0.0020 -0.0041
0.0004 0.0046 -0.0027 0.0016
0.0020 -0.0027 0.0068 0.0022
-0.0041 0.0016 0.0022 0.0078
;
run;

%macro average(iter);
%do i=0 %to &iter;
   %multivariate(varcov=varcov, means=mean, n=21, mul=mvnormal, seed=&i)
```

```
    data x1;set mvnormal(keep=x1 rename=(x1=x));run;
    data x2;set mvnormal(keep=x2 rename=(x2=x));run;
    data x;set x1 x2;run;
    data y1;set mvnormal(keep=y1 rename=(y1=y));run;
    data y2;set mvnormal(keep=y2 rename=(y2=y));run;
    data y;set y1 y2;run;

    /* Step 3: compute mean,std for x and y */
     proc means noprint data=x mean std; var x; output out=xnormal mean=meanx
std=stdx;run;
     proc means noprint data=y mean std; var y; output out=ynormal mean=meany
std=stdy;run;

    data spec;merge xnormal ynormal;run;
    data spec;set spec(drop=_type_ _freq_);run;

    /* Step 4: compute spec1, spec2, spec3, AUC*/
    data norm;/*PROBIT() and PROBNORM()*/
    set spec;
    cutpt1=meany+(-1.6448536)*stdy;tr1=(cutpt1-
meanx)/stdx;spec1=probnorm(tr1);/*95% sensitivity*/
    cutpt2=meany+(-1.2815516)*stdy;tr2=(cutpt2-
meanx)/stdx;spec2=probnorm(tr2);/*90% sensitivity*/
    cutpt3=meany+(-0.8416212)*stdy;tr3=(cutpt3-
meanx)/stdx;spec3=probnorm(tr3);/*80% sensitivity*/
    se=sqrt(stdy*stdy+stdx*stdx);meandiff=meany-meanx;tile=meandiff/se;
    AUC=probnorm(tile);
    run;
    data normal;set norm (keep=spec1 spec2 spec3 AUC);run;
    data result;set result normal ;run;
%end;
%mend average;

/* Step 5: Repeat step1-step4 1000 times */
%average(1000);


data spec1;set result(keep=spec1);run;proc sort data=spec1 out=specficity1;by
spec1;run;
data spec2;set result(keep=spec2);run;proc sort data=spec2 out=specficity2;by
spec2;run;
data spec3;set result(keep=spec3);run;proc sort data=spec3 out=specficity3;by
spec3;run;
data AUC;set result(keep=AUC);run;proc sort data=AUC out=AUCnew;by AUC;run;

data toresult;merge specficity1 specficity2 specficity3 AUCnew;run;

proc print;run;
```

Appendix D.5 Comparing sumby2 and sumby4 by plot

```
libname twoh 'G:\code running in lab\new data\library';
libname sixh 'G:\code running in lab\new data\sixh';

data intdif;set sixh.intdif;run;
data inndif;set sixh.inndif;run;
proc transpose data=intdif out=intr prefix=v; run;
data intr;set intr (drop=_NAME_);run;

%macro sumby2 (dataset);
data sumby2;set &dataset;
ARRAY old (728) v1 - v728;
ARRAY new (364) t1 - t364;
Do i = 1 To 728;
    IF (mod(i,2)=1) THEN DO;
        new((i+1)/2) = old(i)+old(i+1);
    END;
END;
keep t1-t364;
run;
%mend;
%sumby2(intr);

%macro sumby4 (dataset);
data sumby4;set &dataset;
ARRAY old (728) v1 - v728;
ARRAY new (182) t1 - t182;
Do i = 1 To 728;
    IF (mod(i,4)=1) THEN DO;
        new((i+3)/4) = old(i)+old(i+1)+old(i+2)+old(i+3);
    END;
END;
keep t1-t182;
run;
%mend;
%sumby4(intr);


ods trace on;
ods listing close;

proc univariate data=sumby4;
run;
ods trace off;
ods output  TestsForLocation=t3;
data t4;
set t3;
if Testlab="S" then output;
run;

data sumby4wilcoxon;
set t4(keep=Stat pValue);
run;

data sumby4xt;set twoh.xt;if(mod(_N_,4)=1);run;

data one;
merge sumby2xt sumby2wilcoxon;
run;

data two;
merge sumby4xt sumby4wilcoxon;
```

```
run;

/*******p-value 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"P-value" )order=(0 to 0.002 by 0.0001)
      major=(height=1) minor=(height=0.5)
      width=3;

title "P-value for 6 hour Mock vs. Hsv1 sumby&n (21 groups)";
 proc gplot data=two;
   plot pValue*XLabe2 / overlay
                             haxis=axis1 hminor=4
                             vaxis=axis2 vminor=4
                         vref=0.0002   lvref=5;
run;
quit;


%macro wilcoxon (dataset/*sumby2 or sumby4*/
                 ,sumbydata/*sumby2xt or sumby4xt*/
               ,n/*2 or 4*/
                 ,hourdata/*sixh.mock_hsv_sumby2_plot*/);

ods trace on;
ods listing close;

proc univariate data=&dataset;
run;
ods trace off;
ods output  TestsForLocation=t&n;

data newt&n;
set t&n;
if Testlab="S" then output;
run;

data sumbywilcoxon;
set newt&n(keep=Stat pValue);
run;

data &sumbydata;set twoh.xt;if(mod(_N_,&n)=1);run;
/*proc print data=twoh.xt;run;*/

data &hourdata;
merge &sumbydata sumbywilcoxon;
run;

/*******p-value 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"P-value" )order=(0 to 0.002 by 0.0001)
      major=(height=1) minor=(height=0.5)
      width=3;

title "P-value for 6 hour Mock vs. Hsv1 sumby&n (21 groups)";
 proc gplot data=&hourdata;
   plot pValue*XLabe2 / overlay
                             haxis=axis1 hminor=4
                             vaxis=axis2 vminor=4
                          vref=0.0002   lvref=5;
run;
quit;

%mend;

%wilcoxon(sumby2,sumby2xt,2,sixh.mock_hsv_sumby2_plot);
%wilcoxon(sumby4,sumby4xt,4,sixh.mock_hsv_sumby4_plot);


/*******Statistic 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"Statistic" )order=(-460 to 490 by 30)
      major=(height=2) minor=(height=1)
      width=3;

title 'Statistic for 6 hour Mock vs. Hsv1 sumby&n (21 groups)';
 proc gplot data=sixh.mock_hsv_sumby4_plot;
   plot Stat*XLabe2 / overlay
                             haxis=axis1 hminor=4
                             vaxis=axis2 vminor=4
                          vref=350 -350 lvref=5;
run;
quit;


data try;
set sixh.mock_hsv1_unip;
*if Stat<-350 & pValue<0.0002 then inf=1;
```

```
if Stat>350 & pValue<0.0002 then inf=2;
else inf=0;
run;
```

Appendix D.6 Results of PLSR by Sumby4_3-fold cross-validation for 6 hour mock and hsv paired comparison

```
libname twoh 'E:\code running in lab\new data\library';
libname sixh 'E:\code running in lab\new data\sixh';

ods select none;


/*****************/
/*  6h_mock_hsv  */
/*****************/

data intdif;set sixh.intdif;run;
data inndif;set sixh.inndif;run;
proc transpose data=intdif out=intr prefix=v; run;
proc transpose data=inndif out=innr prefix=v; run;
data intr;set intr ;run;
data innr;set innr ;run;

%macro sumby4 (sumby4dataset,dataset);
data &sumby4dataset;set &dataset;
ARRAY old (728) v1 - v728;
ARRAY new (182) t1 - t182;
Do i = 1 To 728;
    IF (mod(i,4)=1) THEN DO;
        new((i+3)/4) = old(i)+old(i+1)+old(i+2)+old(i+3);
    END;
END;
keep t1-t182;
run;
%mend;
%sumby4(intrsumby4,intr);
%sumby4(innrsumby4,innr);

data p1; set intrsumby4;inf=1;run;
data p2;set innrsumby4;inf=0;run;

data p1; set p1;
retain cc 0;
if (mod(_N_,2)=1) then cc=(_N_+1)/2;
if (mod(_N_,2)=0) then cc=_N_/2;
run;

data p2; set p2;
retain cc 0;
if (mod(_N_,2)=1) then cc=(_N_+1)/2;
if (mod(_N_,2)=0) then cc=_N_/2;
run;

data t1;set p1;if(mod(_N_,2)=1);run; *mod(_N_,2)=1: odds, mod(_N_,2)=0: even;
data t2;set p1;if(mod(_N_,2)=0);run;
data n1;set p2;if(mod(_N_,2)=1);run;
data n2;set p2;if(mod(_N_,2)=0);run;/*t1,t2 is intra; n1,n2in inner*/

%macro split(n);/*n=100*/
%do i=0 %to &n;
data t1;set t1;index=ranuni(&i);run;
data t2;merge t2 t1(keep=index);run;
data n1;merge n1 t1(keep=index);run;
data n2;merge n2 t1(keep=index);run;

proc sort data=t1;by index;run;
proc sort data=t2;by index;run;
proc sort data=n1;by index;run;
proc sort data=n2;by index;run;

data a11 a12 a13;/*intraone*/
    set t1;
    if _N_>=1 & _N_<=7 then output a11;
    if _N_>=8 & _N_<=14 then output a12;
    if _N_>=15 & _N_<=21 then output a13;
run;

data b11 b12 b13;/*intratwo*/
    set t2;
    if _N_>=1 & _N_<=7 then output b11;
    if _N_>=8 & _N_<=14 then output b12;
    if _N_>=15 & _N_<=21 then output b13;
run;

data a21 a22 a23;/*innerone*/
    set n1;
    if _N_>=1 & _N_<=7 then output a21;
    if _N_>=8 & _N_<=14 then output a22;
    if _N_>=15 & _N_<=21 then output a23;
run;

data b21 b22 b23;/*innertwo*/
    set n2;
    if _N_>=1 & _N_<=7 then output b21;
    if _N_>=8 & _N_<=14 then output b22;
    if _N_>=15 & _N_<=21 then output b23;
run;

/*intra----a11,a12,a13,is used in wilicoxon rank test*/
data a11;set a11 b11;run;data a21;set a21 b21;run;
data a12;set a12 b12;run;data a22;set a22 b22;run;
data a13;set a13 b13;run;data a23;set a23 b23;run;

proc sort data=a11;by cc;run;
proc sort data=a21;by cc;run;
proc sort data=a12;by cc;run;
proc sort data=a22;by cc;run;
proc sort data=a13;by cc;run;
proc sort data=a23;by cc;run;

data a1;set a11 a21;run;
data a1;set a1 (drop=index rename=(cc=m));run;

data a2;set a12 a22;run;
data a2;set a2 (drop=index rename=(cc=m));run;

data a3;set a13 a23;run;
data a3;set a3 (drop=index rename=(cc=m));run;


%crossvalidation(wix1=a11,wix2=a12,wix3=a13,pls1=a1, pls2=a2, pls3=a3);
%crossvalidation(wix1=a11,wix2=a13,wix3=a12,pls1=a1, pls2=a3, pls3=a2);
%crossvalidation(wix1=a13,wix2=a12,wix3=a11,pls1=a3, pls2=a2, pls3=a1);


%end;
%mend;

/*x-inner y-intra*/
%macro spec(datainner,dataintra,c1,c2,normal,result);
proc means noprint data=&datainner mean std; var &c1; output out=xnormal
mean=meanx std=stdx;run;
proc means noprint data=&dataintra mean std; var &c2; output out=ynormal
mean=meany std=stdy;run;

data spec;merge xnormal ynormal;run;
data spec;set spec(drop=_type_ _freq_);run;

    /*(3) compute spec1, spec2, spec3, AUC*/
    data norm;/*PROBIT() and PROBNORM()*/
    set spec;
    cutpt1=meany+(-1.6448536)*stdy;tr1=(cutpt1-
meanx)/stdx;spec1=probnorm(tr1);/*95% sensitivity*/
    cutpt2=meany+(-1.2815516)*stdy;tr2=(cutpt2-
meanx)/stdx;spec2=probnorm(tr2);/*90% sensitivity*/
    cutpt3=meany+(-0.8416212)*stdy;tr3=(cutpt3-
meanx)/stdx;spec3=probnorm(tr3);/*80% sensitivity*/
    se=sqrt(stdy*stdy+stdx*stdx);meandiff=meany-meanx;tile=meandiff/se;
    AUC=probnorm(tile);
    run;
  data &normal;set norm (keep=spec1 spec2 spec3 AUC);run;

data &result;set &result &normal;run;

%mend;


%macro crossvalidation(wix1,wix2,wix3,pls1,pls2,pls3);/*n=100*/

data trainning;
           set &wix1(drop=inf cc) &wix2(drop=inf cc);
run;

*ods trace on;
*ods listing close;

proc univariate data=trainning;
*ods trace off;
```

```
ods output  TestsForLocation=t3; run;

data t4;
set t3;
if Testlab="S" then output;
run;

data depend;
set t4(keep=pValue);
retain z 0;
if pValue<=0.0002 then z=z+1;
if _n_=182 then call symput("counterx",z);
run;

%let counterxzero=%sysevalf(&counterx+0);
%let counterxone=%sysevalf(&counterx+1);
%let counterxtwo=%sysevalf(&counterx+2);

proc transpose data=&pls1 out=a1tr prefix=v; run;
proc transpose data=&pls2 out=a2tr prefix=v; run;
proc transpose data=&pls3 out=a3tr prefix=v; run;

data splsa1;merge a1tr t4(keep=pValue); if pValue <=0.0002;run;
data splsa2;merge a2tr t4(keep=pValue); if pValue <=0.0002;run;
data splsa3;merge a3tr t4(keep=pValue); if pValue <=0.0002;run;

proc transpose data=splsa1(drop=pValue) out=plsa1(drop=_NAME_
rename=(v&counterxone=inf v&counterxtwo=m)) prefix=v; run;
proc transpose data=splsa2(drop=pValue) out=plsa2(drop=_NAME_
rename=(v&counterxone=inf v&counterxtwo=m)) prefix=v; run;
proc transpose data=splsa3(drop=pValue) out=plsa3(drop=_NAME_
rename=(v&counterxone=inf v&counterxtwo=m)) prefix=v; run;

/*data ji; z=symget('counterxtwo');proc print data=ji;run;*/

data plstrainning;
          set plsa1 plsa2;
run;


ods output ParameterEstimates=coefficient;
proc pls data = plstrainning details;
   model inf=v1-v&counterxzero /SOLUTION;
output out=one PREDICTED=p;
run;

data coefficient;set coefficient(firstobs=2 drop=RowName
rename=(inf=coeff));run;


/*****************************************************************/
/*   compute two dataset's(1st & 2nd) specificiet in pls model  */
/*****************************************************************/

data oneintra oneinner;set one;
if inf=1 then output oneintra;
if inf=0 then output oneinner;
```

```
run;

data p;set oneintra(keep=p);run;
data q;set oneinner(keep=p rename=(p=q));run;


/*****************************************************************/
/*   (1)compute 3rd dataset's specificiet in validating pls model  */
/*****************************************************************/

data intra inner;set plsa3;
if inf=1 then output intra;
if inf=0 then output inner;
run;

/*(2)specificity & sensitivity*/
proc transpose data=intra(drop=inf m) out=intra prefix=v; run;
proc transpose data=inner(drop=inf m) out=inner prefix=v; run;

data validateintra;merge intra(drop=_NAME_) coefficient;
array old v1-v14;
array new t1-t14;
do i=1 to 14;
  new(i)=old(i)*14;
end;
run;

data validateinner;merge inner(drop=_NAME_) coefficient;
array old v1-v14;
array new n1-n14;
do i=1 to 14;
  new(i)=old(i)*14;
end;
run;

proc transpose data=validateintra(keep=t1-t14) out=intra prefix=v; run;
data intra;set intra;intrastar=sum(v1-v&counterxzero);run;

proc transpose data=validateinner(keep=n1-n14) out=inner prefix=v; run;
data inner;set inner;innerstar=sum(v1-v&counterxzero);run;

data x;set intra(keep=intrastar rename=(intrastar=x));run;
data y;set inner(keep=innerstar rename=(innerstar=y));run;


/*first-inner second-intra*/
%spec(x,y,x,y,normalvalidate,resultvalidate);
%spec(q,p,q,p,normalorig,resultorig);

%mend;

%split(100);

data sixh.mock_hsv_3fold_auc;set resultvalidate;run;
data shirinkage;merge resultorig(rename=(spec1=speco1 spec2=speco2
spec3=speco3 auc=auco)) resultvalidate;run;
data shirinkage;set shirinkage;
```

```
shi1=speco1-spec1;
shi2=speco2-spec2;
shi3=speco3-spec3;
shiauc=auco-auc;
run;


proc transpose data=shirinkage(keep=shi1 shi2 shi3 shiauc)
out=totalshirinkage prefix=v; run;

data totalshi;set totalshirinkage;
avg=Mean(of v1-v300);
run;

data sixh.mock_hsv_3fold_shi;set totalshi(keep=avg);run;
```

Appendix D.7 Results of model with positive terms minus negative terms by Sumby4_2-fold cross-validation for 6 hour mock and hsv paired comparison

```
libname twoh 'E:\code running in lab\new data\library';
libname sixh 'E:\code running in lab\new data\sixh';

ods select none;

/*****************/
/*  6h_mock_hsv  */
/*****************/

data intdif;set sixh.intdif;run;
data inndif;set sixh.inndif;run;
proc transpose data=intdif out=intr prefix=v; run;
proc transpose data=inndif out=innr prefix=v; run;
data intr;set intr ;run;
data innr;set innr ;run;

%macro sumby4 (sumby4dataset,dataset);
data &sumby4dataset;set &dataset;
ARRAY old (728) v1 - v728;
ARRAY new (182) t1 - t182;
Do i = 1 To 728;
    IF (mod(i,4)=1) THEN DO;
       new((i+3)/4) = old(i)+old(i+1)+old(i+2)+old(i+3);
    END;
END;
keep t1-t182;
run;
%mend;
%sumby4(intrsumby4,intr);
%sumby4(innrsumby4,innr);

data p1; set intrsumby4;inf=1;run;
data p2;set innrsumby4;inf=0;run;

data p1; set p1;
retain cc 0;
```

```
if (mod(_N_,2)=1) then cc=(_N_+1)/2;
if (mod(_N_,2)=0) then cc=_N_/2;
run;

data p2; set p2;
retain cc 0;
if (mod(_N_,2)=1) then cc=(_N_+1)/2;
if (mod(_N_,2)=0) then cc=_N_/2;
run;

data t1;set p1;if(mod(_N_,2)=1);run; *mod(_N_,2)=1: odds, mod(_N_,2)=0: even;
data t2;set p1;if(mod(_N_,2)=0);run;
data n1;set p2;if(mod(_N_,2)=1);run;
data n2;set p2;if(mod(_N_,2)=0);run;/*t1,t2 is intra; n1,n2in inner*/

%macro split(n);/*n=100*/
%do i=0 %to &n;
data t1;set t1;index=ranuni(&i);run;
data t2;merge t2 t1(keep=index);run;
data n1;merge n1 t1(keep=index);run;
data n2;merge n2 t1(keep=index);run;

proc sort data=t1;by index;run;
proc sort data=t2;by index;run;
proc sort data=n1;by index;run;
proc sort data=n2;by index;run;

data a11 a12 ;/*intraone*/
    set t1;
    if _N_>=1 & _N_<=10 then output a11;
    if _N_>=11 & _N_<=21 then output a12;

run;

data b11 b12;/*intratwo*/
    set t2;
    if _N_>=1 & _N_<=10 then output b11;
    if _N_>=11 & _N_<=21 then output b12;
run;

data a21 a22;/*innerone*/
    set n1;
    if _N_>=1 & _N_<=10 then output a21;
    if _N_>=11& _N_<=21 then output a22;
run;

data b21 b22;/*innertwo*/
    set n2;
    if _N_>=1 & _N_<=10 then output b21;
    if _N_>=11 & _N_<=21 then output b22;
run;

/*intra----a11,a12,is used in wilicoxon rank test*/
data a11;set a11 b11;run;/*intraone=10*/
data a21;set a21 b21;run;/*innerone=10*/
data a12;set a12 b12;run;/*intratwo=11*/
data a22;set a22 b22;run;/*innertwo=11*/
```

```
proc sort data=a11;by cc;run;
proc sort data=a21;by cc;run;
proc sort data=a12;by cc;run;
proc sort data=a22;by cc;run;

/*data a1;set a11 a21;run;
data a1;set a1 (drop=index rename=(cc=m));run;

data a2;set a12 a22;run;
data a2;set a2 (drop=index rename=(cc=m));run;
*/
%crossvalidation(a11,a21,a12,a22);
%crossvalidation(a12,a22,a11,a21);

%end;
%mend;

%macro spec(datainner,dataintra,c1,c2,normal,result);
proc means noprint data=&datainner mean std; var &c1; output out=xnormal
mean=meanx std=stdx;run;
proc means noprint data=&dataintra mean std; var &c2; output out=ynormal
mean=meany std=stdy;run;

data spec;merge xnormal ynormal;run;
data spec;set spec(drop=_type_ _freq_);run;

    /*(3) compute spec1, spec2, spec3, AUC*/
    data norm;/*PROBIT() and PROBNORM()*/
    set spec;
    cutpt1=meany+(-1.6448536)*stdy;tr1=(cutpt1-
meanx)/stdx;spec1=probnorm(tr1);/*95% sensitivity*/
    cutpt2=meany+(-1.2815516)*stdy;tr2=(cutpt2-
meanx)/stdx;spec2=probnorm(tr2);/*90% sensitivity*/
    cutpt3=meany+(-0.8416212)*stdy;tr3=(cutpt3-
meanx)/stdx;spec3=probnorm(tr3);/*80% sensitivity*/
    se=sqrt(stdy*stdy+stdx*stdx);meandiff=meany-meanx;tile=meandiff/se;
    AUC=probnorm(tile);
    run;
  data &normal;set norm (keep=spec1 spec2 spec3 AUC);run;

data &result;set &result &normal;run;

%mend;


%macro crossvalidation(trainintra,traininner,valiintra,valiinner);/*n=100*/

data trainning;
           set &trainintra(drop=index inf cc);
run;

*ods trace on;
*ods listing close;

proc univariate data=trainning;
*ods trace off;
```

```
ods output  TestsForLocation=t3; run;

data t4;
set t3;
if Testlab="S" then output;
run;

data depend;
set t4;
retain z 0;
retain n 0;
if pValue<=0.0002 & Stat>0 then z=z+1;
if pValue<=0.0002 & Stat<0 then n=n+1;
if _n_=182 then call symput("counterx",z);/*z is used to computing the number
of positive value of Stat*/
if _n_=182 then call symput("countern",n);/*n is used to computing the number
of negative value of Stat*/
run;

%let counterxposi=%sysevalf(&counterx+0);
%let counterxnega=%sysevalf(&countern+0);

/**********************/
/*   training subset  */
/**********************/
proc transpose data=&trainintra(drop=inf cc index) out=a1tra prefix=v;
run;/*intra*/
proc transpose data=&traininner(drop=inf cc index) out=a1ner prefix=v;
run;/*inner*/

data a1intra;merge a1tra t4(keep=Stat pValue); if pValue <=0.0002;run;
data a1inner;merge a1ner t4(keep=Stat pValue); if pValue <=0.0002;run;

/*
inner1p are positive of inner-difference of training subset
inner1n are negative of inner-difference of training subset
intra1p are positive of inner-difference of training subset
intra1n are negative of inner-difference of training subset*/
data inner1p inner1n;set a1inner;
if Stat>0 then output inner1p;
if Stat<0 then output inner1n;run;

data intra1p intra1n;set a1intra;
if Stat>0 then output intra1p;
if Stat<0 then output intra1n;run;

proc transpose data=inner1p(drop=_NAME_ pValue Stat) out=ner1p prefix=v; run;
proc transpose data=inner1n(drop=_NAME_ pValue Stat) out=ner1n prefix=v; run;
proc transpose data=intra1p(drop=_NAME_ pValue Stat) out=tra1p prefix=v; run;
proc transpose data=intra1n(drop=_NAME_ pValue Stat) out=tra1n prefix=v; run;

/*inner*/
data ner1p;set ner1p;t1=sum(of v1-v&counterxposi);run;
data ner1n;set ner1n;t2=sum(of v1-v&counterxnega);run;
data inner1;merge ner1p(keep=t1) ner1n(keep=t2);inner1_discriminator=t1-
t2;run;
```

```
data innertraining;set inner1(keep=inner1_discriminator
rename=(inner1_discriminator=x));run;
/*intra*/
data tra1p;set tra1p;t1=sum(of v1-v&counterxposi);run;
data tra1n;set tra1n;t2=sum(of v1-v&counterxnega);run;
data intra1;merge tra1p(keep=t1) tra1n(keep=t2);intra1_discriminator=t1-
t2;run;

data intratraining;set intra1(keep=intra1_discriminator
rename=(intra1_discriminator=y));run;
/**********************/
/*  validation subset */
/**********************/

proc transpose data=&valiintra(drop=inf cc index) out=a2tra prefix=v;
run;/*intra*/
proc transpose data=&valiinner(drop=inf cc index) out=a2ner prefix=v;
run;/*inner*/
data a2intra;merge a2tra t4(keep=Stat pValue); if pValue <=0.0002;run;
data a2inner;merge a2ner t4(keep=Stat pValue); if pValue <=0.0002;run;
/*
inner2p are positive of inner-difference of validation subset
inner2n are negative of inner-difference of validation subset
intra2p are positive of inner-difference of validation subset
intra2n are negative of inner-difference of validation subset*/
data inner2p inner2n;set a2inner;
if Stat>0 then output inner2p;
if Stat<0 then output inner2n;run;

data intra2p intra2n;set a2intra;
if Stat>0 then output intra2p;
if Stat<0 then output intra2n;run;

proc transpose data=inner2p(drop=_NAME_ pValue Stat) out=ner2p prefix=v; run;
proc transpose data=inner2n(drop=_NAME_ pValue Stat) out=ner2n prefix=v; run;
proc transpose data=intra2p(drop=_NAME_ pValue Stat) out=tra2p prefix=v; run;
proc transpose data=intra2n(drop=_NAME_ pValue Stat) out=tra2n prefix=v; run;

/*inner*/
data ner2p;set ner2p;t1=sum(of v1-v&counterxposi);run;
data ner2n;set ner2n;t2=sum(of v1-v&counterxnega);run;
data inner2;merge ner2p(keep=t1) ner2n(keep=t2);inner2_discriminator=t1-
t2;run;

data innervali;set inner2(keep=inner2_discriminator
rename=(inner2_discriminator=p));run;

/*intra*/
data tra2p;set tra2p;t1=sum(of v1-v&counterxposi);run;
data tra2n;set tra2n;t2=sum(of v1-v&counterxnega);run;
data intra2;merge tra2p(keep=t1) tra2n(keep=t2);intra2_discriminator=t1-
t2;run;

data intravali;set intra2(keep=intra2_discriminator
rename=(intra2_discriminator=q));run;

%spec(innertraining, intratraining,x,y, normalorig,resultorig);
```

```
%spec(innervali,intravali,p,q, normalvalidate,resultvalidate);

%mend;

%split(100);

data shirinkage;merge resultorig(rename=(spec1=speco1 spec2=speco2
spec3=speco3 auc=auco)) resultvalidate;run;
data shirinkage;set shirinkage;
shi1=speco1-spec1;
shi2=speco2-spec2;
shi3=speco3-spec3;
shiauc=auco-auc;
run;

data sixh.linear_mock_hsv_2fold_auc;set shirinkage;run;

proc transpose data=shirinkage(keep=shi1 shi2 shi3 shiauc)
out=totalshirinkage prefix=v; run;

data totalshi;set totalshirinkage;
avg=Mean(of v1-v200);
run;

data sixh.linear_mock_hsv_2fold_shi;set totalshi(keep=avg);run;

data z1;set shirinkage(keep=shi1 shi2 shi3 shiauc);
if (mod(_N_,2)=1);
run;

data z2;set shirinkage(keep=shi1 shi2 shi3 shiauc);
if (mod(_N_,2)=0);
run;

proc transpose data=z1 out=b1 prefix=v; run;
proc transpose data=z2 out=b2 prefix=v; run;

data b1;set b1;avg1=Mean(of v1-v100);run;
data b2;set b2;avg2=Mean(of v1-v100);run;

data b;merge b1(keep=_NAME_ avg1) b2(keep=avg2);run;

proc print data=b;run;

data sixh.linear_2fold_odd_mock_hsv;set b;run;

proc print data=sixh.linear_mock_hsv_2fold_auc;run;
```