TESIS SS09-2304

PRACTICAL METHODS VALIDATION FOR VARIABLES SELECTION IN THE HIGH DIMENSION DATA: APPLICATION FOR THREE METABOLOMICS DATASETS

ACHMAD CHOIRUDDIN
NRP 1312 201 905

PEMBIMBING
Jean-Charles MARTIN
Matthieu MAILLOT
Florent VIEUX
Cécile CAPPONI

PROGRAM MAGISTER
JURUSAN STATISTIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2015

THESIS SS09-2304

PRACTICAL METHODS VALIDATION FOR VARIABLES SELECTION IN THE HIGH DIMENSION DATA: APPLICATION FOR THREE METABOLOMICS DATASETS

ACHMAD CHOIRUDDIN
NRP 1312 201 905

SUPERVISORS
Jean-Charles MARTIN
Matthieu MAILLOT
Florent VIEUX
Cécile CAPPONI

MAGISTER PROGRAM
DEPARTMENT OF STATISTICS
FACULTY OF MATHEMATICS AND NATURAL SCIENCES
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2015

PRACTICAL METHODS VALIDATION FOR VARIABLES
SELECTION IN THE HIGH DIMENSION DATA:
APPLICATION FOR THREE METABOLOMICS DATASETS

This thesis is composed to fulfill the requirement to get Master degree

Magister Sains (M.Si)

in

Institut Teknologi Sepuluh Nopember

By:

ACHMAD CHOIRUDDIN

NRP. 1312 201 905

Examination date : September, 19th 2014

Graduation period : March 2015

Approved by :

1.   Dr. Jean-charles MARTIN   (Supervisor)

2.   Dr. Matthieu MAILLOT   (Supervisor)

3.   Dr. Florent VIEUX   (Supervisor)

4.   Dr. Cécile CAPPONI   (Reviewer)

Director of Master Program ITS,

Prof. Dr. Ir. Adi Soeprijanto, M.T.
NIP. 19640405 199002 1 001

ABSTRACT

Background: Variable selection on high throughput metabolomics data are becoming inevitable to select relevant information since they often imply a high degree of multicolinearity, and, as a result, lead to severely ill conditioned problems. Both in supervised classification framework and machine learning algorithms, one solution is to reduce their data dimensionality either by performing features selection, or by introducing artificial variables in order to enhance the generalization performance of a given algorithm as well as to gain some insight about the concept to learned.

Objective: The main objective of this study is to select a set of features from thousands of variables in dataset. We divide this objective into two sides: (1) To identify small sets of features (fewer than 15 features) that could be used for diagnostic purpose in clinical practice, called low-level analysis and (2) We do the identification to a larger set of features (around 50-100 features), called middle-level analysis; this involves obtaining a set of variables that are related to the outcome of interest. Besides that, we would like to compare the performances of several proposed techniques in feature selection procedure for Metabolomics study.

Method: This study is facilitated by four proposed techniques, which are two machine learning techniques (i.e., RSVM and RFFS) and two supervised classification techniques (i.e., PLS-DA VIP and sPLS-DA), to classify our three datasets, i.e., human urines, rat's urines, and rat's plasma datasets, which contains two classes sample each dataset.

Results: RSVM-LOO always leads the accuracy performance compare to the other two cross-validation methods, i.e., bootstrap and N-fold. However, this RSVM results is not much better since RFFS could achieve the higher accuracy performance. Another side, PLS-DA and sPLS-DA could reach a good performance either for variability explanation or predictive ability. In biological sense, RFFS and PLS-DA VIP show their performance by finding the more common selected features than RSVM and sPLS-DA compare to previous metabolomics study. This is also confirmed in the statistical comparison that RFFS and PLS-DA could lead the similarity percentage of selected features. Furthermore, RFFS and PLS-DA VIP have their better performance since they could select three metabolites of five confirmed metabolites from previous metabolomics study which couldn't be achieved by RSVM and sPLS-DA.

Conclusion: RFFS seems to become the most appropriate techniques in features selection study, particularly in low-level analysis when having small sets features is often desirable. Both PLS-DA VIP and sPLS-DA lead to a good performance either for variability explanation or predictive ability, but PLS-DA VIP is slightly better in term of biological insight. Besides it is only limited for two class problem, RSVM unfortunately couldn't achieve a quite good performance both in statistical and biological interpretation.

Keywords: High dimension data, Features selection, Classification analysis, Metabolomics.

ACKNOWLEDGMENTS

Conducting a research with some experts in mostly metabolomics is really my first experience. Although I did take big efforts, I really enjoy doing this project in the sense that I could learn many new things. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

My first thanks go to my supervisors for their continuous support and their precious feedback in conducting this project. I am highly indebted to **Jean-Charles MARTIN** for his kind to allow me carrying out a six-month internship in UMR NORT, getting involved in his research project team could deepen my analysis skill and broaden my knowledge. In addition, his experience as an expert taught me many things, not only on the project implementation, but also being a professional researcher. I would like to express my deep gratitude to **Matthieu MAILLOT** and **Florent VIEUX** for their guidance and constant supervision as well as for providing necessary information regarding the project. I also particularly appreciated for their support and motivation both in completing the project and being my new family. Advice given by my university advisor, **Cécile CAPPONI**, has been a great help in understanding several concepts I learned.

My thanks and appreciations also go to my colleagues in the laboratory of UMR NORT in developing the project and people who have willingly helped me out with their abilities.

I would further like to thank all the persons who helped me during my stay in the Department of Applied Mathematics in Social Sciences: a very kind people, **Nicolas PECH**, head of department, who gave a lot of time, energy, and motivation to keep me stand up until the end of this Master program, my great lecturers: **Marie-christine ROUBAUD, Thomas WILLER, Laurence REBOUL, Sébastian OLIVEAU, Christine CAMPIONI, Rebecca McKenna,** and **Richard LALOU** who inspired me a lot, **Fabienne PICOLET** for her useful help in administration purpose, and for all my classmates : **Dian, Julien, Lylia, Mahmoud, Armand, Alpha, Sofiane,** and **Youssoupha**, we are truly the rainbow troops.

I would also like to acknowledge the Ministry of Education and Culture Republic of Indonesia, ITS Postgraduate program, and Ministry of Foreign Affair of France for the financial support.

Finally, I wish to thank my parents, my brothers and sisters in PPI Marseille-PACA and Indonesia, and all my big family for their support and encouragement throughout my study.

TABLE OF CONTENT

CHAPTER 1
PRESENTATION OF UMR NORT

UMR NORT (Nutrition, Obésité et Risque Thrombotique) had many complementary expertise in the field of nutrition and metabolic diseases at the Faculty of Medicine in Timone, University of Aix-Marseille.

Nutrition Obésité Risque Thrombotique

UMR INSERM 1062 / INRA 1260 / AMU
Directrice : Pr Marie-Christine ALESSI marie-christine.alessi@univ-amu.fr
Directrice Adjointe : Dr Marie-Josèphe AMIOT-CARLIN marie-jo.amiot-carlin@univ-amu.fr

EQUIPE 1	EQUIPE 2	EQUIPE 3	EQUIPE 4 Eq Emergente
Biodisponibilité, lipides et micronutriments	Micronutriments, tissus adipeux et résistance à l'insuline	Hémostase, thrombose, et biomarqueurs vasculaires	Modelisations et Interventions durables pour la sécurité nutritionnelle et la santé des populations
Responsable d'Equipe	**Responsable d'Equipe**	**Responsable d'Equipe**	**Responsable d'Equipe**
P. Borel (DR1 INRA 100%)	F. Peiretti (CR1 INSERM 100%)	M. C. Alessi (PU-PH)	J Amiot-Carlin (DR1 INRA 100%)
			N. Darmon (DR2 INRA 100%)
Chercheurs	**Chercheurs**	**Chercheurs**	**Chercheurs**
E. Reboul (CR1 INRA 100%)	R. Govers (CR1 INSERM 100%)	M. Canault (CR2 INSERM 100%)	
	J.F. Landrier (CR1 INRA 100%)	M. Grino (CR1 INSERM 100%)	
		J.C. Martin (CR1 INRA 100%)	
Enseignants Chercheurs	**Enseignants Chercheurs**	**Enseignants Chercheurs**	**Enseignants Chercheurs**
M. Gastaldi (MCU-PH AMU 30%)	P. Darmon (PU-PH AP-HM(3...)	V. Baccini (MCU-PH 5%)	
M. Maraninchi (MCU-PH AM...)	M. Fontes (PU-PH AMU 50%)	T. Cuisset (MCU-PH 40%)	
A. Nicolay (MCU AMU 30%)	E. Sérée (MCU AMU 50%)	C. Défoort (MCU AMU 25%)	
H. Portugal (PU-PH AMU 5%)		A. Dutour (PU-PH 40%)	
R. Valéro (PU-PH AMU 40%)		P. Morange (MU-PH 40%)	
E. Wolff (MCU AMU 30%)		M. Poggi (MCU AMU 100%)	
S. Béliard (MCU PH AMU 35%)		B. Gaborit (MCU-PH 10%)	
		J.L. Bonnet (PU-PH 5%)	
BIATSS	**BIATSS**	**BIATSS**	**BIATSS**
J. Dupont Roussel (T.AMU 10...)	J. Astier (Tech INRA 100%)	P. Ancel (CDD AI AMU 100%)	
C. Halimi (Tech INRA 80%)	B. Bonardo (Tech AMU 100%)	C. Antona (ADT INRA 90%)	
M. Nowicki (Tech INRA 80%)	T. Gonzalez (Tech AMU 80%)	D. Bastelica (AI AMU 100%)	
	C. Couturier (CDD AMU 100%)	C. Ginies (AI INRA 90% Mobilité)	
		M. Verdier (IE AMU 100%)	
Contractuels/ Post-Doc	**Contractuels/Post-Doc**	**Contractuels/Post-Doc**	**Contractuels/Post-Doc**
C. Desmarchelier (CDD 100%)	F. Tourniaire (CDD 100%)	G. Favé (CDD 100%)	H. Gaigi (CDD 50%)
M. Margier (CDD AMU 100%)		A. Mezzapesa (CDD 100%)	C. Dubois (CDD 50%)
		L. Svilar (IE CDD 100%)	A. Lesturgeon (CDD)
			A. Maidon (CDD)
			M. Gobard (CDD)
			M. Pérignon (CDD 100%)
			I. Denes (CDD 30%)
			T. Barré (CDD 100%)
			L. Francès (CDD étr 100%)
Thésards	**Thésards**	**Thésards**	**Thésards**
D. Préveraud	L. Bonnet	N. Aidoud	G. Ferrari
	I. Kara	I. Abdesselam	C. Morrissey
	E. Karkeni	D. Ghaloussi	J. Peyrol
	S. Fenni	F. Al Frouh	T. Barré
	N. Sreng	C. Grosdidier	
		M. Favier	
Services Communs Administratifs			
C. Bellavia (CDD ADT AMU 1...)	M. H. Goletto (AI INRA 100%)	V. Kogalama (Tech INSERM 80%)	
L. Patron Chassende (CDD A...)	B. Sauvan (Tech INSERM 80%)		
Animalerie			
	P. Guichard (ADT AMU 100%)		
Services Communs Laverie Nettoyage			
E. Pelloux (ADT INSERM 80%)	I. Khelifi (ADT AMU 100%)	C. Pétrolési (ATRF AMU 100%)	M. Ayello (ATRF AMU 100%)

TOTAL DES EFFECTIFS : 80 agents : 6 chercheurs INRA, 5 BIATSS INRA, 4 chercheurs INSERM, 3 BIATSS INSERM, 14 BIATSS AMU, 6 MCU-PH, 5 MCU AMU, 7 PU-PH, 1 MU-PH, 14 Contractuels, 15 Doctorants

Figure 1.1. Structure Team of UMR NORT

Two key complementary themes developed are: (1) digestion, bioavailability of lipophilic micro constituent and postprandial lipids metabolism, and (2) nutrition and vascular and thrombotic diseases. They combined both descriptive and mechanistic approaches using various and complementary methodologies ranging from molecular and cell biology to clinical studies.

The activities are divided into 4 teams (presented in Figure 1.1). Under this draft unit, they associate the tools of physical chemistry and biochemistry, molecular biology, culture and cell biology, analytical chemistry, genetic and nutritional animal models. Metabolic and nutritional studies in human volunteers or patients are also carried out. Various local hospital collaborations (IFR Site Timone CRNH) and international or industry partnership (food and pharmaceutical sectors) are being set. Concerned human pathologies are cardiovascular disease, obesity and metabolic syndrome, diabetes Type II and its complications, lipid malabsorption, malnutrition. The research must have direct benefits for optimizing nutritional recommendations and to strengthen the role of nutrition in public health policy (cf NFHP).

CHAPTER 2
INTRODUCTION

Metabolomics is an emerging field providing insight into physiological processes. There have been a lot of metabolomics studies, and it is becoming more and more developed. In this part, we would like to describe the background overview of our metabolomics study, including background of our study and our study objectives.

2.1   Background

Metabolomics can be defined as the field of science that deals with the measurement of metabolites in an organism for the study of the physiological processes and their reaction to various stimuli such as infection, disease, or drug use (Nicholson, Lindon, & Holmes, 1999). It is an effective tool to investigate disease diagnosis in metabolite concentration in various biofluids.

Metabolomics allows analyzing hundreds of metabolites in a given biological sample. When applied to urine or plasma samples, it allows differentiating individual phenotypes better than with conventional clinical endpoints or with small sets of metabolites. It also allows exploring the metabolic effects of a nutrient in a more global way. In the field of nutrition, metabolomics has been used to characterize the effects of both a deficiency or a supplementation of different nutrients, and to compare the metabolic effects of closely related foods such as whole-grain or refined wheat flours (Scalbert, et al., 2009).

There have been several metabolomics studies which have been carried out, such as Kind, Tolstikov, Fiehn, & Weiss (2007) who did a research of urinary metabolomics approach for identifying kidney cancer, Gu, et al., (2007) who tested the effect of diet on metabolites using rat urine samples, and many others (Scalbert, et al., 2009; Suhre, et al., 2010; Dai, et al., 2010; and Grison, et al., 2013).

Like in many other biological studies, a key difficulty in the Metabolomic study is the noisy nature of the data (Rakotomamonjy, 2003 and Zhang, et al., 2006), which can be caused by the intrinsic complexity of the biological problem, as well as experimental and technical biases. Another difficulty arises from the high dimensionality of the data while the training samples are very scarce. Similar to the situation in microarray studies, typically one metabolomics investigation only involves several samples (usually less than 100 samples) but the measured points on the mass spectrum can be in thousands or more. Even after pre-processing steps such as peak and/or biomarker detection, the dimensionality is usually much larger than the sample size.

As these high throughput data are characterized by thousands of variables and small number of samples, they often imply a high degree of multicolinearity, and, as a result, lead to severely ill conditioned problems (Lê Cao, Boitard, & Besse, 2011). If directly working in this high dimensional space with limited samples, most conventional pattern recognition algorithms may not work well (Zhang & Wong, 2001). Some algorithms may not be able to achieve a solution when the number of sample is less than the dimensionality. For others that can achieve a solution, it may not be able to work well on samples other than that used for training.

Both in supervised classification framework and machine learning algorithms, one solution is to reduce the dimensionality of the data either by performing features selection, or by introducing artificial variables (i.e., latent variables) that summarize most of information. The purpose of the features or variables selection is to eliminate irrelevant variables to enhance the generalization performance of a given algorithm (Rakotomamonjy, 2003) as well as to gain some insight about the concept to be learned (Diaz-Uriarte & Andres, 2006). The technique of introducing artificial variables that summarize most of information, such as Partial Least Squares (PLS), has an objective to overcome the problem of high multicolinearity (Pérez-Enciso & Tenenhaus, 2003). Other advantages of feature selection and introducing artificial variables include cost reduction of data gathering and storage, and also on computational speedup.

Several features selection studies have been carried out. Golub, et al. (1999) defined a metric to evaluate the correlation of a feature with a classification scheme, thus determining whether the feature is relevant or not. Obviously, this kind of strategy does not take possible unless it can be proven that the features are statistically independent each other. Zhang & Wong (2001) proposed features selection algorithms named Recursive Support Vector Machines (R-SVM) based on the features contribution built by their weights and class means difference, while Guyon, Weston, Barnhill, & Vapnik (2002) proposed Support Vector Machine – Recursive Features Elimination (SVM-RFE) built by their weights in the SVM classifiers. In 2006, Zhang, et al., compared R-SVM and SVM-RFE and they concluded that R-SVM and SVM-RFE cross-validation prediction performances were nearly the same, but R-SVM was more robust to noise and outliers in discovering informative features and therefore had better accuracy on independent test data.

Based on Random Forest defined by Breiman (2001), Diaz-Uriarte and Andés (2006) developed features selection algorithm based on the decrease of classification accuracy when values of a variable in a node of tree are permuted randomly, called Random Forest for Feature Selection (RFFS). Another side, in supervised classification framework, Pérez-Enciso and Tenenhaus (2003) have developed Partial Least Square Discriminant Analysis – Variable Importance in the Projection (PLSDA-VIP) to select the most important variables based on PLS rules. Besides that, Lê Cao, Boitard, and Besse (2011) introduced a sparse version of PLS for discrimination purpose, called

sparse PLS-DA (or sPLS-DA), which was a natural extension to the sPLS proposed by Lê Cao, Rossouw, Robert-Granié, & Besse (2008).

In this study, we will focus on the four methods proposed above (R-SVM, RFFS, PLSDA-VIP, and sPLS-DA) to analyze metabolomics data. It is important to identify the discriminating features that cause the categorization to enable an in-depth understanding of the system that generated the data. This can be achieved through feature selection, which involves identifying the optimum subset of the variables in data set that gives the best separation (Mahadevan, Shah, Marrie, & Slupsky, 2008).

2.2   Objective

Selection of relevant variables for sample classification is a common task in most features expression studies, including this study. When there are much larger features than the number of sample(s), this problem may undermine the success of classification techniques that is strongly affected by data quality: redundant, noisy, and unreliable information as well as a confusing selection of relevant variables. Because of that, our interest objectives in this study are as follows.

1. To identify small sets of features that could be used for diagnostic purpose in clinical practice; this involves obtaining the smallest possible set of variables that can still achieve good predictive performance. In this point, our purpose is to select the most relevant variables that contribute maximally in the classification (we define to choose under 15 features).

2. Beside the stringency depicted above to focus on the least number of variables enabling the best classification, our other purpose is to select a much wider set of variables (around 50-100 features) that detailed the outcome to be explained. This could provide a mechanistic view of the biological outcome to be described.

3. There were several features selection studies, especially in metabolomics studies, which have been carried out. In this study, we would like to compare the performance of four feature selection techniques proposed in our three datasets, either in statistical or biological insight. Besides that, we would like also to compare the performance of different cross-validation used in RSVM. Finally, we would like to find the most appropriate feature selection techniques, particularly for metabolomics data, for future study.

CHAPTER 3
LITERATURE REVIEW

In this part, we explain the statistical techniques used in this study. As mentioned in Chapter 2, headline of this study is to select a set of features from thousands of variables in dataset. Besides that, we would like to compare the performances of several proposed techniques in feature selection procedure for Metabolomics study. We consider three technique rules, which are Support Vector Machine (SVM), Random Forest (RF), and Partial Least Square Discriminant Analysis (PLS-DA). For feature selection, we will compare four techniques based on three rules mentioned. They are Recursive Support Vector Machine, Random Forest for Feature Selection, PLS-DA Feature Selection based on Variable Importance in the Projection, and Sparse PLS-DA.

3.1   Support Vector Machine

The basic principle of support vector machine classifier is a binary classifier algorithm that looks for an optimal hyper plane as a decision function in a high-dimensional space. The foundations of SVM have been developed by Cortes & Vapnik (1995) and are gaining popularity due to many attractive features, and promising empirical performance. In this problem, the goal is to separate the two classes by a function which is induced from available examples. The goal is to produce a classifier that will work well on *unseen* examples, i.e., it generalizes well. Consider the example in Figure 3.1. Here there are many possible linear classifiers that can separate the data, but there is only one that maximizes the margin (maximizes the distance between it and the nearest data point of each class). This linear classifier is termed the optimal separating hyper plane. Intuitively, we would expect this boundary to generalize well as opposed to the other possible boundaries.
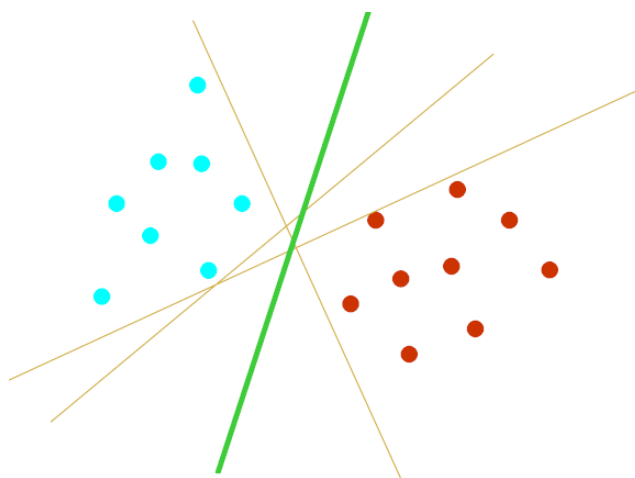

Figure 3.1. Optimal Separating Hyperplane

SVM has been developed for multiclass purpose and even for regression problem. SVM techniques both in classification and regression purpose have been explained by Gunn (1998). In this study, we will focus on SVM for binary classification. The key idea of SVM is on generalization; where a classifier needs not only to work well on the training samples, but also work equally well on previously unseen samples.

Consider one has a training data set $\{x_k, y_k\} \in \mathbb{R}^n \times \{-1, 1\}$ where x_k are the training examples and y_k are the class labels. The method consists in first mapping $\mathbf{x}$ into a high dimensional space via a function $\mathbf{\Phi}$, then computing a decision function of the form:

$$f(x) = \langle w, \mathbf{\Phi}(x) \rangle + b$$

by maximizing the distance between the set of points $\mathbf{\Phi}(x)$ to the hyperplane parameterized by (w, b) while being consistent on the training set. The set of vectors is said to be *optimally separated* by the hyper plane if it is separated without error and the distance among the closest vectors to the hyper plane is maximal. The class label of $\mathbf{x}$ is obtained by considering the sign of $f(x)$. For the SVM classifier with misclassified examples being quadratically penalized, this optimization problem can be written as:

$$\min_{w, \xi} \frac{1}{2} \|w\|^2 + C \sum_{k=1}^{m} \xi_k^2$$

under the constraint $\forall k, y_k f(x) \geq 1 - \xi_k$. Here, C is the regularization parameter, which is trade off between the training accuracy and prediction term. The solution of this problem is obtained using the Lagrangian theory and one can prove that vector $\mathbf{w}$ is of the form:

$$w = \sum_{k=1}^{m} \alpha_k^* y_k \, \mathbf{\Phi}(x)$$

where α_k^* is the solution of the following quadratic optimization problem:

$$\max_{\alpha} W(\alpha) = \sum_{k=1}^{m} \alpha_k - \frac{1}{2} \sum_{k,l=1}^{m} \alpha_k \, \alpha_l y_k y_l \left(K(x_k, x_l) + \frac{1}{C} \delta_{k,l} \right)$$

Subject to $\sum_{k=1}^{m} y_k \alpha_k = 0$ and $\forall k, \alpha_k \geq 0$, where $\delta_{k,l}$ is Kronecker symbol and $K(x_k, x_l) = \langle \mathbf{\Phi}(x_k), \mathbf{\Phi}(x_l) \rangle$ is the Gram matrix of the training examples.

3.1.1 Recursive SVM

Recursive SVM (R-SVM) is an algorithm to recursively classifies the sample using SVM rules and selects the variables according to their weight in the SVM classifiers. R-SVM has been developed by Zhang & Wong (2001) and Zhang, et al., (2006). The main objective of R-SVM is to select a subset of features with maximum discriminatory power between two classes. Since the feature dimension is large and the sample size is

small, there are usually many combinations of features that can give zero error on the training data. Therefore, the "minimal error" cannot work. Intuitively, it is desirable to find a set of features that give the maximum separation between two classes of samples.

3.1.2 Ranking the Features according to Their Contribution

For linear SVM, the final decision function $f(x)$ is a linear one, which is the weighted sum of all the features plus a constant term as a threshold. If $f(x) > 0$, then the sample is class 1, otherwise class 2. To achieve our objective, the simplest way is to select a subset of features that contributes the most in the classification based on the decision function; the idea is to rank all the features according to their relative contribution in classification function. When calculating the contribution, Zhang & Wong (2001) and Zhang, et al., (2006) consider the use of the mean values of samples in the same class. The expression of feature i of the two class means is:

$$m_j^+ = \sum_{x^+ \in class1} x_j^+ \quad and \quad m_j^- = \sum_{x^- \in class2} x_j^-$$

The difference of two class means in decision function is:

$$S = \sum_{j=1}^{d} w_j\, m_j^+ - \sum_{j=1}^{d} w_j\, m_j^- = \sum_{j=1}^{d} w_j \left(m_j^+ - m_j^-\right)$$

Where d is the total of features, and w_j is the jth component of the weight vector **w** in SVM. Then we define the contribution of feature j in S as:

$$s_j = w_j\left(m_j^+ - m_j^-\right)$$

The contribution of feature j is not only decided by the weight w_j in the classifier function, but also decided by the data (the class-means). According to the idea of large-margin in statistical learning theory, a larger S corresponds better generalization ability. Therefore, if we want to select a subset of features from all the d features, the proper way is to keep those features that give largest positive contribution in S.

3.1.3 Assessing the Performance of Feature Selection

When the sample size is small so that we cannot afford to use an independent test set, cross validation is the usual choice for assessing the performance of the classifier. In this technique, we use three types of cross validations, which are Bootstrap, Leave-one-out (LOO), and N-fold.

Efron (1979) described clearly the bootstrap prosedures. Bootstrap prosedure is now getting more and more useful in the classification method, such as in Ambroise & McLachlan (2002) and Zhang, et al., (2006). The principle of bootstrap method in

classification is: (1) constructing the sample n size and resampling randomly these n sample size with replacement in many times (usually more than 100 times), (2) calculating the prediction error of each iteration, and (3) calculating the mean prediction error.

The procedures and the using of N-Fold CV method in SVM has been explained by (Ambroise & McLachlan, 2002), (Bhardwaj, Langlois, Zhao, & Lu, 2005), and also (Mahadevan, Shah, Marrie, & Slupsky, 2008). The dataset is divided into N non overlapping subsets of roughly equal size. The rule is trained on N-1 of these subsets combined together and then applied to the remaining subset to obtain an estimate of the prediction error. This process is repeated in turn for each of N subsets, and the CV error is given by the average of the N estimates of the prediction error thus obtained. If we take $N = n - 1$, where n is the number of observation, so our N-Fold CV method is equal to leave-one-out CV method.

It should be emphasized that when sample size is small, the feature selection depends heavily on the specific samples used for the selection, no matter what method is used. The feature selection procedure is a part of the whole classification system. In some literature, feature selection steps were external to the cross validation procedures, i.e., the feature selection was done with all the samples and the cross-validation was only done for the classification procedure. We call this kind of cross validation CV1. As pointed out by Ambroise & McLachlan (2002), CV1 may severely bias the evaluation in favor of the studied method due to "information leak" in the feature selection step. A more proper approach is to include the feature selection procedure in the cross validation, i.e., to leave the test sample(s) out from the training set before undergoing any feature selection. In this way, not only the classification algorithm, but also the feature selection method is validated. We call this scheme CV2 and use it in all of our investigations throughout. Thus, for the cross validation, the sample to be left out as test sample should be removed from the data set at the very beginning, before any feature selection procedure.

3.1.4 Recursive Classification and Feature Selection

The selection of an optimal subset of features from a feature set is a combinatorial problem, which cannot be solved when the dimension is high without the involvement of certain assumptions or compromise, which results in only suboptimal solutions. Here we use a recursive procedure to approach the problem. To select a subset of features that contribute the most in the classification, we rank all the features according to s_j defined in sub 3.1.2 and choose the top ones from the list. We use this strategy recursively in the following procedures:

Step 0. Define a decreasing series of feature numbers $d_0 > d_1 > d_2 > \cdots > d_k$ to be selected in the series of selection steps. Set $i = 0$ and $d_0 = d$ (i.e., start with all features).

Step 1. At step i, build the SVM decision function with current d_i features.

Step 2. Rank the features according to their contribution factors s_j in the trained SVM and select the top d_{i+1} features (eliminate the bottom $d_i - d_{i+1}$ features).

Step 3. Set $i = i + 1$. Repeat from Step 1 until $i = k$.

This is an implementation of backward feature elimination scheme described in pattern recognition textbooks with criteria defined on SVM models at each feature-selection level. It should be noted that this scheme is suboptimal as it does not exhaustively search in the space of all possible combinations. Our choices of the number of iterations and the number of features to be selected in each iteration are very *ad hoc*. Although different settings of these parameters may affect the results, we have observed that, for most cases when the two classes can be reasonably separated with the expression data, the classification performances achieved with different settings were very close to each other, and the majority of features ranked at the top positions were also very stable.

Zhang, et al., (2006) follow the CV2 scheme to estimate the error rate a teach level. In cross-validation experiments, different training subsets generate different lists of features (although many or most of them overlap in usual experiments). A frequency-based selection method is adopted to decide the lists of features to be reported. That is, after the recursive feature selection steps on each subset, we count at each of the d_i levels the frequency of the features being selected among all rounds of cross-validation experiments. The top d_i most frequently selected features are reported as the final d_i features (called the top features).

In most situations, CV2 errors usually follow a U-shaped curve along the selection steps (feature numbers). Finding the minimal number of features that can give the minimal CV2 error rate is often desirable for

Figure 3.2. Workflow of R-SVM algorithm

real applications. Another realistic consideration is the limited ability of follow-up biological investigations on the selected features. As a compromise, we decide the final number of features to be reported in an experiment by considering both the error rates and the limitation of follow-up biological investigations. The entire workflow is depicted in Figure 3.2; we call this whole scheme R-SVM (recursive SVM).

3.2   Random Forest

Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. Random forest is an algorithm for classification developed by Breiman (2001) that uses an ensemble of classification trees. Each of the classification trees is built using a bootstrap sample of data, and each split the candidate set of variables is a random subset of the variables. Thus, random forest uses both bagging and random variable selection for tree building. The algorithm yields an ensemble that can achieve both low bias and low variance.

The random forests algorithm is: (1) draw *ntree* bootstrap samples from the original data, (2) for each of the bootstrap samples, grow an unpruned classification or regression tree, with the following modification: at each node, rather than choosing the best split among all predictors, randomly sample *mtry* of the predictors and choose the best split from among those variables (Bagging can be thought of as the special case of random forests obtained when $mtry = p$, the number of predictors), and (3) predict new data by aggregating the predictions of the *ntree* trees (i.e., majority votes for classification, average for regression).

Random forest has excellent performance in classification tasks. Random forest has also several characteristics that make it deal for metabolomics data, such as: (a) can be used when there are many more variables than observations, (b) can be used both for two-class and multi-class problem, (c) has good predictive performance even when most predictive variable are noise, and therefore it does not require a pre-selection of features, (d) does not over fit, (e) can handle a mixture of categorical and continuous predictors, (f) incorporates interactions among predictor variables, (g) the output is invariant to monotone transformations of the predictors, (h) there are high quality and free implementations: the original Fortran code from Breiman and Cutler, and an R package (Liaw & Wiener, 2002), (i) returns measures of variable importance, and (j) there is a little need to fine-tune parameters to achieve excellent performance. The most important parameter to choose is *mtry* (the number of input variables tried at each split), but it has been reported that the default value of *mtry* in R package is a good choice (Liaw & Wiener, 2002). In addition, the user needs to decide how many trees to grow for each forest (*ntree*) as well as the minimum size of the terminal nodes (*nodesizes*). All these three parameters were examined by Diaz-Uriarte & Andres (2006) in the purpose of feature selection using random forest.

3.2.1 Feature Selection Using Random Forest

Random forest returns several measures of variable importance, which were used by a few authors. Diaz-Uriarte & Andres (2006) found in (Dudoit & Fridlyand, 2003) and (Wu, et al., 2003) that they use filtering approaches and, thus, do not take advantage of the measures of variable importance returned by random forest as part of the algorithm. Diaz-Uriarte & Andres (2006) also found a similarity of variable importance measures in (Svetnik, Liaw, Tong, & Wang, 2004). However, their strategy is to achieve the accurate predictors, which might not be the most appropriate for a purpose as it shifts the emphasis away from selection the specific variables.

This feature selection strategy was proposed by Diaz-Uriarte & Andres (2006). The most reliable measure is based on the decrease of classification accuracy when values of a variable in a node of a tree are permuted randomly. To select features they iteratively fit random forests, at each iteration building a new forest after discarding those variables (features) with the smallest variable importance; the selected set of features is the one that yields the smallest out-of-bag (OOB) error rate.

It is proposed using out-of-bag estimates as an ingredient in estimates of generalization error. Assume a method for constructing a classifier from any training set. Given a specific training set T, from bootstrap training set T_k, construct classifiers $h(x, T_k)$ and let these vote from the bagged predictor. For each y, x in the training set, aggregate the votes only over those classifiers for which T_k does not contain y, x. Call this the out-of-bag classifiers. Then the out-of-bag estimate for generalization error is the error rate of the out-of-bag classifier on the training set (Breiman, 2001).

Note that in this section they are using OOB error to choose the final set of features, not to obtain unbiased estimates of the error rate of this rule. Because of the iterative approach, the OOB error is biased down and cannot be used to assess the overall error rate of the approach, for reasons analogous to those leading to "selection bias" (Ambroise & McLachlan, 2002). The bootstrap strategy will be used to assess prediction error rates (see subsection 3.2.2).

3.2.2 Estimation of Error Rates in Feature Selection

Besides feature selection procedure, Diaz-Uriarte & Andres (2006) also described the way to estimate error rates. To estimate the prediction error rate of all methods they used the .632+ bootstrap method as proposed by Efron & Tibshirani (1997) and Ambroise & McLachlan (2002). The .632+ bootstrap method uses a weighted average of the re-substitution error (the error when a classifier is applied to the training data) and the error on samples not used to train the predictor (the "leave-one-out" bootstrap error); this average is weighted by a quantity that reflects the amount of over fitting. To calculate the prediction error rate, the .632+ bootstrap method is applied to the complete procedure, and thus the samples used to compute the leave-one-out bootstrap error used in the .632+ method are samples that are not used when fitting the random forest, or

carrying out variable selection. The .632+ bootstrap method was also used when evaluating the competing methods.

3.3   Partial Least Squares Discriminant Analysis (PLS-DA)

Partial Least Squares regression makes it possible to relate a set of dependent variables $Y = \{Y_1, \dots, Y_p\}$ to a set of independent variables $X = \{X_1, \dots, X_M\}$ when the number of independent and/or dependent variables is much larger than the number of observations. PLS regression consists in carrying out a principal components analysis of the set of variables X subject to the constraint that the (pseudo-) principle components of X_j are as "explanatory" as possible to the set of variables Y. It is then possible to predict Y_k from X_j by better separating the signal from the variable (Tenenhaus, Gauchi, & Ménardo, 1995). The goal of PLS regression is to provide a dimension reduction strategy in a situation where we want to relate a set of response variables Y to a set of predictor variables X.

The PLS regression algorithm has been explained by some authors, such as Tenenhaus (1998) and Tenenhaus, Gauchi, & Ménardo (1995). The starting point is two data matrices X and Y. X is an $N \times M$ matrix and Y is an $N \times P$ matrix. Before the algorithm starts, the matrices should be scaled. The procedures are:

(1) Set $X_0 = X$ and $Y_0 = Y$

(2) Construct a linear combination of u_1 from the Y columns and a linear combination of t_1 from the X columns that maximize $cov(u_1, t_1) = cor(u_1, t_1). \sqrt{var(u_1). \ var(t_1)}$. We obtain then two variables u_1 and t_1 that correlate and resume well the variables X and Y. We conduct then the regression:

$$X_0 = t_1 p'_1 + X_1$$
$$Y_0 = t_1 r'_1 + Y_1$$

(3) Re-process step 2 by replacing X_0 and Y_0 with X_1 and Y_1. We will obtain new components: u_2 as a linear combination from the Y_1 columns and t_1 as a linear combination from the X_1 columns. After conducting the regression, we will obtain the decomposition as follows.
$$X_0 = t_1 p'_1 + t_2 p'_2 + X_2$$
$$Y_0 = t_1 r'_1 + t_2 r'_2 + Y_2$$

Iterate the procedure until the components obtained $t_1, t_2, \dots, t_A$ explain sufficiently the variables Y. The components t_h are the linear combinations of X columns, which is also uncorrelated. The regression equation is:

$$Y_k = \beta_{k0} + \beta_{k1} X_1 + \dots + \beta_{kM} X_M + Y_{hk}$$

Partial least squares discriminant analysis (PLS-DA) is a partial least squares regression of a set Y of binary variables describing the categories of a categorical variable on a set X of predictor variables (Pérez-Enciso & Tenenhaus, 2003). Although PLS was not originally designed for classification and discrimination purpose, some authors routinely use PLS for that purpose and there is substantial empirical evidence to suggest that it performs well in that role (Barker & Rayens, 2003). It is a compromise between the usual discriminant analysis and a discriminant analysis on the significant principal components of the predictor variables. This technique is specially suited to deal with a much larger number of predictors than observations and with multicolinearity, two of the main problems encountered when analyzing "omics" (such as transcriptomics, proteomics, and metabolomics) expression data.

3.3.1 Feature Selection based on Variable Importance in the Projection

When the number of independent variables is very large, it will give impact for PLS-DA analysis, even though we create components. That is because the impact of noisy data as well as redundancy data. Because of this reason, we still need to select several important variables before creating the components. Besides that, a fundamental requirement for PLS to yield meaningful answers is some preliminary variable selection. Enciso and Tenenhaus (2003) did this feature selection technique by selecting the variables on the basis of the Variable Importance in the Projection (VIP) for each variable.

By PLS regression model written as:

$$Y_k = \sum_{h=1}^{H}(Xw_h^*)c_h + e$$

Where w_h^* is a p dimension vector containing the weights given to each original variable in the h-th component, and c_h is the regression coefficient of Y_k.

VIP_j is a popular measure in the PLS literature and it is defined for variable j as:

$$VIP_j = \left\{ p \sum_{h=1}^{H} \sum_{k} R^2(y_k, t_h)w_{hj}^2 / \sum_{h=1}^{H} \sum_{k} R^2(y_k, t_h) \right\}^{1/2}$$

For each j-th predictor variable $j = 1,2, \ldots, M$, where $R^2(a, b)$ stands for the squared correlation between items in vector a and b, and $t_h = X_{h-1}w_h$, X_{h-1} is the residual matrix in the regression of X on components $t_1, \ldots t_{h-1}$ and w_h is a vector of norm 1. Note that w_{hj}^2 measures the contribution of each variable j on the h-th PLS component. Thus, VIP_j quantifies the influence on the response of each variable summed over all components and categorical response, relative to the sum of squares of the model. This make the *VIP* as an intuitively appealing measure of the global effect of each variable.

In this study, we will analyze all three cases using PLS DA variables selection based on VIP. To select the number of variables selected, we use the criteria of R^2, Q^2, and accuracy rate. Q_h^2 is the value of Q^2 for component h, and it can bedefined as:

$$Q_h^2 = 1 - \frac{PRESS_h}{RESS_{h-1}}$$

Where $PRESS_h$ is the predicted sum of squares of a model containing h components, that also written as $PRESS_h = \sum_{i=1}^n (y_i - \hat{y}_{h(-i)})^2$ and $RESS_{h-1}$ is the residual sum of squares of a model containing h-1 components where $RESS_h = \sum_{i=1}^n (y_i - \hat{y}_{hi})^2$. Q_h^2 is also used for selecting the number of components in the model, the number of PLS components will be selected if a new component satisfied $Q_h^2 \geq 0.05$.

R^2 (or cofficient of determination) is the fraction of the total variability explained by the model. R^2 is defined as $R^2 = 1 - RESS_{h=1,..,H}/SST_{h=1,...,H}$, while $SST_h = \sum_{i=1}^n (y_i - \bar{y}_{hi})^2$. Beside that, Q^2 is a measurement of the predictive ability of the model and it is obtained by:

$$Q^2 = 1 - \prod_{h=1}^H \frac{PRESS_h}{RESS_{h-1}} \quad or \quad Q^2 = 1 - \prod_{h=1}^H (1 - Q_h^2)$$

3.3.2 Sparse PLSDA (sPLS-DA)

The sparse PLS proposed by Lê Cao, Rossouw, Robert-Granié, & Besse (2008) was initially designed to identify subsets of correlated variables of two different types coming from two different data sets X and Y of sizes $(n \times p)$ and $(n \times q)$ respectively. The original approach was based on Singular Value Decomposition (SVD) of the cross product $M_h = X_h^T Y_h$. Any real r-rank matrix M $(p \times q)$ can decomposed into three matrices U, Δ, V as $M = U\Delta V^T$. One interesting property that will be used in sparse PLS method is that the columns vectors of U or u_h and V or v_h (called left and right singular vectors) correspond to the PLS loadings of X and Y if $M = X^T Y$.

Sparse loading vectors are then obtained by applying Lasso penalization on both u_h and v_h to perform variable selection. Indeed, one interesting property of PLS is the direct interpretability of the loading vectors as a measure of the relative importance of the variables in the model. The optimization problem of the sparse PLS minimizes the Frobenius norm between the current cross product matrix and the loading vectors:

$$\min_{u_h, v_h} \| M_h - u_h v_h' \|_F^2 + P_{\lambda 1}(u_h) + P_{\lambda 2}(v_h)$$

Where $P_{\lambda 1}(u_h) = sign(u_h)(|u_h| - \lambda_1)_+$ and $P_{\lambda 2}(v_h) = sign(v_h)(|v_h| - \lambda_2)_+$ are applied componentwise in the vectors u_h and v_h and are the soft thresholding functions that approximate Lasso penalty functions. They are simultaneously applied on both loading vectors. The procedures of Sparse PLS are:

1. $X_0 = X$ and $Y_0 = Y$

2. For h in 1 until H:
   (a) Set $\widetilde{M}_{h-1} = X_{h-1}^T Y_{h-1}$
   (b) Decompose $\widetilde{M}_{h-1}$ and extract the first pair of singular vectors $u_{old} = u_h$ and $v_{old} = v_h$
   (c) Until convergence of u_{new} and v_{new}:
      i. $u_{new} = P_{\lambda 2}(\widetilde{M}_{h-1} v_{old})$, normalize u_{new}
      ii. $v_{new} = P_{\lambda 1}(\widetilde{M}_{h-1} u_{old})$, normalize v_{new}
      iii. $u_{old} = u_{new}, v_{old} = v_{new}$
   (d) $\xi_h = X_{h-1} u_{new}/u'_{new} u_{new}$
       $\omega_h = Y_{h-1} v_{new}/v'_{new} v_{new}$
   (e) $c_h = X_{h-1}^T \xi_h/\xi'_h \xi_h$
       $d_h = Y_{h-1}^T \xi_h/\xi'_h \omega_h$
       $e_h = Y_{h-1}^T \omega_h/\omega'_h \omega_h$
   (f) $X_h = X_{h-1} - \xi_h c'_h$
   (g) Regression mode: $Y_h = Y_{h-1} - \xi_h d'_h$
       Canonical mode: $Y_h = Y_{h-1} - \omega_h e'_h$

In the case where there is no sparsity constraint ($\lambda_1 = \lambda_2 = 0$) we obtain same results as in a classical PLS.

The extension of sparse PLS to a supervised classification framework is straightforward. It is possible to make an analysis of sparse PLS for discrimination purpose. The response matrix Y of size $(n \times K)$ is coded with dummy variables to indicate the class membership of each sample (Lê Cao, Boitard, and Besse, 2011).

Note that in this specific framework, we will only perform variable selection on the X data set, i.e., we want to select the discriminative features that can help predicting the classes of the samples. The Y dummy matrix remains unchanged. Therefore, we set $M_h = X_h^T Y_h$ and the optimization problem of the sPLS-DA can be written as:

$$\min_{u_h, v_h} \| M_h - u_h v'_h \|_F^2 + P_{\lambda 1}(u_h)$$

with the same notation as in sPLS. The procedure used is exactly same as sPLS procedure, except there is no change of v_h (there is no v_{old} or v_{new}) and we can delete procedure *2c(ii)* in sPLS procedure. Therefore, the penalization parameter to tune is λ. For practical reasons, sPLS-DA algorithm has been implemented to choose the number of variables to select rather than parameter λ.

CHAPTER 4
METHOD OF ANALYSIS

In chapter 4, the method of analysis is described. Besides presentation of datasets used in this study, we also explain the statistical analysis including pre-processing data, R-programming, and interpretation.

4.1   Presentation of Datasets

Several datasets are needed to make sure the stability performance of our algorithms used, as mentioned that our objective is to compare the performance of four feature selection techniques in classification purpose. In term of this purpose, we use three datasets in this study containing human urines test, rat's urines test, and rat's plasma test. The first dataset contains 28 human urines which are analyzed by LC-MS (Metabolomic machine). There are two groups in this datasets, where group called "0" represents 14 human urines and group called "1" represents the same 14 human urines in which 30 molecules are added. In this experiment, we measure 1271 features that we will then select some of which contributing the best in the classification.

The second dataset contains 20 rat's urines which are also analyzed by LC-MS. Two groups named "contaminated" represents 10 rat's urines occurring from rats contaminated in their drinking water by natural uranium and "not contaminated" represents 10 normal rat's urines. In this experiment, 1376 features are measured and it will be selected several most important features. Last dataset contains the data obtained from 2x10 rat's plasma with 810 features measured. It was collected from rats contaminated or not by natural natrium as described above. Overall, we use two type rats' samples and one human sample and we use two type urines samples and one plasma sample.

4.2   Statistical Analysis

Statistical analysis steps include pre-processing data, R-programming, and interpretation. Step of pre-processing data explains the pre-action before analyzing using four features selection techniques for classification mentioned in chapter 3. R-programming step describes the R *package* used and the main idea of several functions created in software R. And last, step of interpretation explains two sides of interpretation view: statistical and biological interpretation.

4.2.1   Pre-Processing Data

The function of pre-processing data is to understand more deeply about data characteristics since skimming the data to overview is not sufficient. This step is also

useful for data treatment because each classification technique needs different requirement such as data centering, scaling, and transformation. As explained that R-SVM and RFFS don't require any data transformation in analyzing procedure, it's different with PLS technique, where both PLS-DA feature selection based on VIP and sPLS-DA, require pre-treatment data.

A recommended data transformation used for PLS-DA VIP and sPLS-DA is log10-pareto transformation, where we need two times data transformation. We transform the data to log10 and we use then Pareto scaling in log10 transformation data. Pareto scaling is defined as $\tilde{x}_{ij} = (x_{ij} - \bar{x}_i)/\sqrt{s_i}$ and it aims to reduce the relative importance of large values, but keep data structure partially intact (van den Berg, Hoefsloot, Westerhuis, Smilde, & van der Werf, 2006).

4.2.2 R-Programming

The process of statistical classification techniques is facilitated by software R. R is a language and environment for statistical computing and graphics. R provides a wide variety of statistical and graphical techniques, and is highly extensible. R is available as Free Software under the terms of the Free Software Foundation's GNU General Public License in source code form. It compiles and runs on a wide variety of UNIX platforms and similar systems (including FreeBSD and Linux), Windows and MacOS. R can be extended easily via *packages*. There are about eight packages supplied with the R distribution and many more are available through the CRAN family of Internet sites covering a very wide range of modern statistics (R-Foundation, 2014). Example of R screenshots is presented in Figure 4.1.
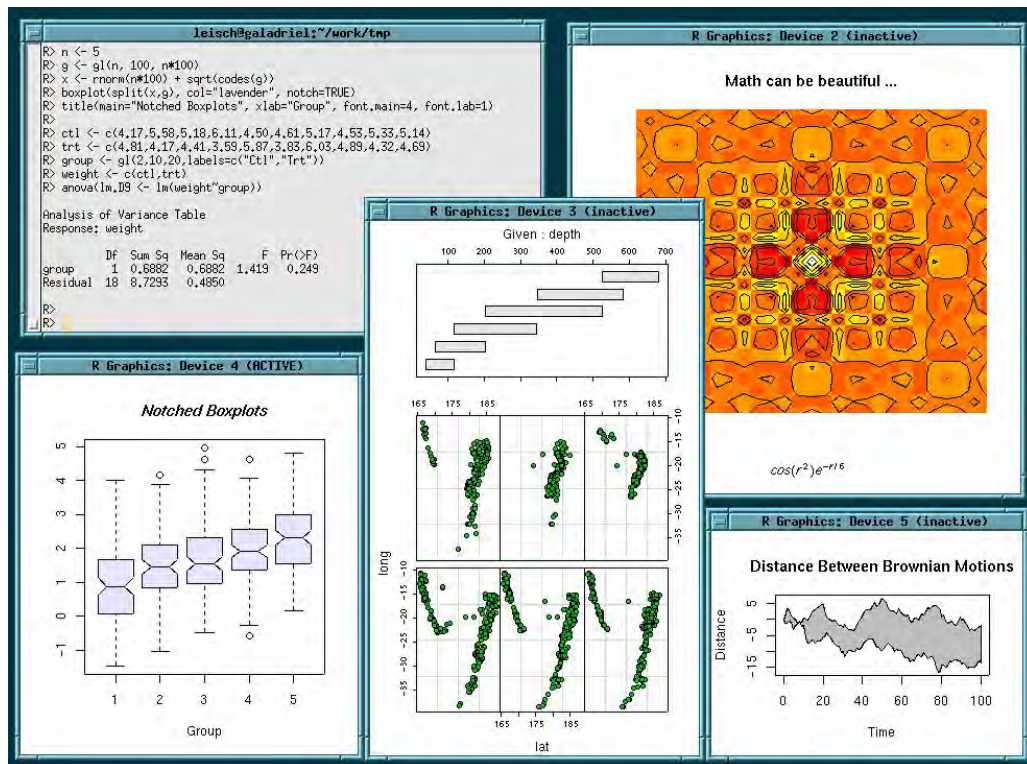
Figure 4.1. R Screenshots (retrieved from http://www.r-project.org/)

Before analyzing to main algorithms, we firstly create a decreasing ladder. The purpose of creating ladder is to define the variable selected based on decreasing function. In case 1 (human urines) for example, there are 26 iterations of features selected: 1271, 1017, 814, 651, 521, 417, 334, 267, 214, 171, 137, 110, 88, 70, 56, 45, 36, 29, 23, 18, 14, 11, 9, 7, 6, and 5 features. It is also carried out for rat's urines dataset (26 iterations) and rat's plasma dataset (24 iterations).

In this study, we use four core packages; it is one package for each technique. R-SVM: *e107*, RFFS: *VarSel*, PLS-DA feature selection based on VIP: *DiscriMiner*, and sPLS-DA: *mixOmics*. Package *e107* is functions for latent class analysis, short time Fourier transform, fuzzy clustering, support vector machines, shortest path computation, bagged clustering, and also naive Bayes classifier (Meyer, et al., 2014). We use this package for developing the algorithm of Recursive SVM created by Zhang, et al., (2006). In analyzing of random forest for feature selection, package *VarSel* is proposed because this package is specially created by Diaz-Uriarte (2010) for this purpose. Unlike package *e107* that we should develop the algorithm for analysis of R-SVM purpose, package *VarSe*l is able to facilitate the users to analyze the datasets using RFFS directly without re-programming.

DiscriMiner is an R package created by Sanchez & Determan (2013) that has functions for discriminant analysis and classification purposes covering various methods such as descriptive, geometric, linear, quadratic, PLS, as well as qualitative discriminant analyses. We develop feature selection of PLS-DA based on VIP using this package. The main idea of this algorithm is the same as the idea of R-SVM algorithm. The different side is in the criteria of features ranking method. Package *mixOmics* is a package that provides statistical integrative techniques and variants to analyze highly dimensional data sets like sparse PLS-DA. This package is developed by (Dejean, Gonzalez, & Lê Cao, 2014). All R-code developed in this study is presented in Appendix 1.

4.2.3 Interpretation

Interpretation is the next step after programming the all feature selection techniques. There are two sides of insight presented in this study: statistical and biological interpretation. In the view of statistical interpretation, we focus on the statistical criteria and assumption. The purpose is to ensure that there are no many mistakes of statistical procedures. After that, it is also purposed that this study result will be a good reference for next study. However, statistical interpretation is not sufficient to make a great study while it always needs an importance implementation insight. In term of this reason, we also study the biological interpretation to make our study better. The result of statistical analysis will be verified by biological knowledge. Therefore, the combination of statistical and biological interpretation will complete the analysis required in this study.

CHAPTER 5
ANALYSIS RESULTS

Chapter 5 describes the results of analysis, which are divided into several sections. Section 1-4 explains the results obtained by each statistical technique. After that, section 5 shows the comparison performance of all techniques proposed. Finally, section 6 guides the readers in the biological interpretation and give the meanings of the study.

5.1   Analysis Using R-SVM

Recursive SVM (R-SVM) is an algorithm to recursively classifies the sample using SVM rules and selects the variables according to their weight in the SVM classifiers (Zhang, et al., 2006). The main objective of R-SVM is to select a subset of features with maximum discriminatory power between two classes. Since the feature dimension is large and the sample size is small, there are usually many combinations of features that can give zero error on the training data. Therefore, the "minimal error" cannot work. Intuitively, it is desirable to find a set of features that give the maximum separation between two classes of samples.

Table 5.1. Selection of Parameter C

Case 1		Case 2		Case 3	
	cost error		cost error		cost error
1	2.000000e-04 0		2.000000e-04 0.15		2.000000e-04 0.25
2	2.000000e-03 0		2.000000e-03 0.15		2.000000e-03 0.25
3	2.000000e-02 0		2.000000e-02 0.15		2.000000e-02 0.25
4	2.000000e-01 0		2.000000e-01 0.15		2.000000e-01 0.25
5	2.000000e-01 0		2.000000e-01 0.15		2.000000e-01 0.25
6	6.324555e-01 0		6.324555e-01 0.15		6.324555e-01 0.25
7	9.283178e-01 0		9.283178e-01 0.15		9.283178e-01 0.25
8	1.124683e+00 0		1.124683e+00 0.15		1.124683e+00 0.25
9	5.000000e+00 0		5.000000e+00 0.15		5.000000e+00 0.25
10	6.666667e+00 0		6.666667e+00 0.15		6.666667e+00 0.25
11	1.000000e+01 0		1.000000e+01 0.15		1.000000e+01 0.25
12	2.000000e+01 0		2.000000e+01 0.15		2.000000e+01 0.25
13	2.000000e+01 0		2.000000e+01 0.15		2.000000e+01 0.25
14	2.000000e+02 0		2.000000e+02 0.15		2.000000e+02 0.25
15	2.000000e+03 0		2.000000e+03 0.15		2.000000e+03 0.25
16	2.000000e+04 0		2.000000e+04 0.15		2.000000e+04 0.25

The SVM regularization parameter C is another challenge in SVM model selection. The parameter C determines the trade-off between minimizing the training error and reducing the model complexity. The range of C depends on the underlying SVM learning algorithm being used, but we see that the most appropriate C is the lowest. It is suggested that the most appropriate C range for SVM is between 10^{-2} and 10^4 (Huang, Lee, Lin, and Huang, 2007).

In this study, we use the range of parameter C between 2.10^{-4} and 2.10^4 for all three cases. By comparing the error rate, we can select the lowest possible of parameter C that

obtain the lowest error rate. In this C interval selected, there is no error rates change, it means that there is no different error rates obtained by using all point between 2.10^{-4} and 2.10^4 and we should select the lowest parameter C that is coloured red in Table 5.1, i.e., 2.10^{-4} for each case.
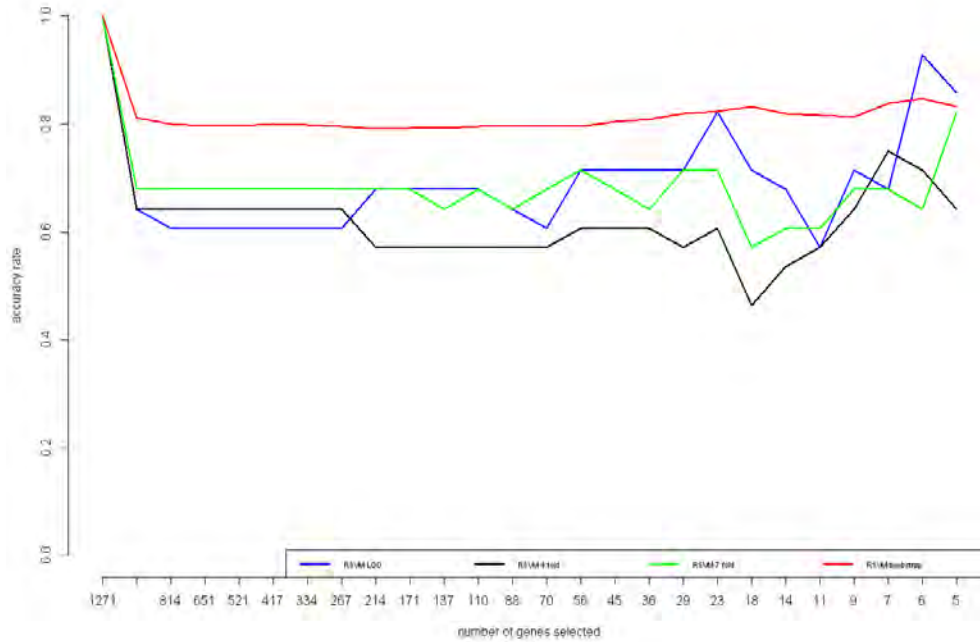
Figure 5.1. Human Urines Datasets Performance

Analyzing Human urines datasets (case 1), we will compare four cross-validation (CV) methods, i.e., leave-one-out (LOO), 4-fold, 7-fold, and bootstrap. Figure 5.1 represents the accuracy rates based on the number of features selected and CV methods used. The performance of 4-fold is the worst since it tends to obtain lowest accuracy rates. After that, 7-fold CV accuracy rates obtained is also quite poor even though it is better than 4-fold CV performance. The most stable and quite good is the performance of bootstrap since it leads to 80 percent and there is no much fluctuation. However, since it fluctuates, the performance of LOO is able to reach around 90 percent that has never been achieved by other CV methods.

We can see the accuracy rates for each CV used and features selected in Table 5.2. Using all features leads to obtain maximum accuracy rates, the result that quite contradict to our hypotheses. To better known, we will show the conclussion after describing all results. In this step, we see there is a significant decreasing performance using between all features (1271 features) and the second level (1017 features).

Since we have two objectives: (1) to identify small sets of features (under 15 features) that could be used for diagnostic purpose in clinical practice and (2) to select a much wider set of variables (50-100 features) that detailed the outcome to be explained, we will choose the features selected in to two levels, called low-level for small sets features selected and middle-level, for another purpose.

Table 5.2. Human Urines Accuracy Performance

	LOO	fold4	fold7	bootstrap
Features:1271	1.0000000	1.0000000	1.0000000	1.0000000
Features:1017	0.6428571	0.6428571	0.6785714	0.8115827
Features:814	0.6071429	0.6428571	0.6785714	0.7992860
Features:651	0.6071429	0.6428571	0.6785714	0.7973027
Features:521	0.6071429	0.6428571	0.6785714	0.7965093
Features:417	0.6071429	0.6428571	0.6785714	0.7992860
Features:334	0.6071429	0.6428571	0.6785714	0.7980960
Features:267	0.6071429	0.6428571	0.6785714	0.7961127
Features:214	0.6785714	0.5714286	0.6785714	0.7913526
Features:171	0.6785714	0.5714286	0.6785714	0.7925426
Features:137	0.6785714	0.5714286	0.6428571	0.7925426
Features:110	0.6785714	0.5714286	0.6785714	0.7961127
Features:88	0.6428571	0.5714286	0.6428571	0.7953193
Features:70	0.6071429	0.5714286	0.6785714	0.7949226
Features:56	0.7142857	0.6071429	0.7142857	0.7953193
Features:45	0.7142857	0.6071429	0.6785714	0.8036493
Features:36	0.7142857	0.6071429	0.6428571	0.8092027
Features:29	0.7142857	0.5714286	0.7142857	0.8187227
Features:23	0.8214286	0.6071429	0.7142857	0.8242761
Features:18	0.7142857	0.4642857	0.5714286	0.8322094
Features:14	0.6785714	0.5357143	0.6071429	0.8187227
Features:11	0.5714286	0.5714286	0.6071429	0.8163427
Features:9	0.7142857	0.6428571	0.6785714	0.8135660
Features:7	0.6785714	0.7500000	0.6785714	0.8385561
Features:6	0.9285714	0.7142857	0.6428571	0.8472828
Features:5	0.8571429	0.6428571	0.8214286	0.8322094

In human urines dataset (Table 5.2), we can choose six features (LOO), seven features (4-Fold), five features (7-Fold), and six features (bootstrap) for low-level, while selecting 56 features (LOO, 4-Fold, and 7-Fold) and 88 features (bootstrap) for middle-level. In this case, we know that the performance of low-level is higher than the performance of middle-level, even for high-level. By using 6 features, the accuracy performance obtained by LOO CV could achieve 93 percent, that means there are 26 samples classified correctly, and there are just two samples classified incorrectly.
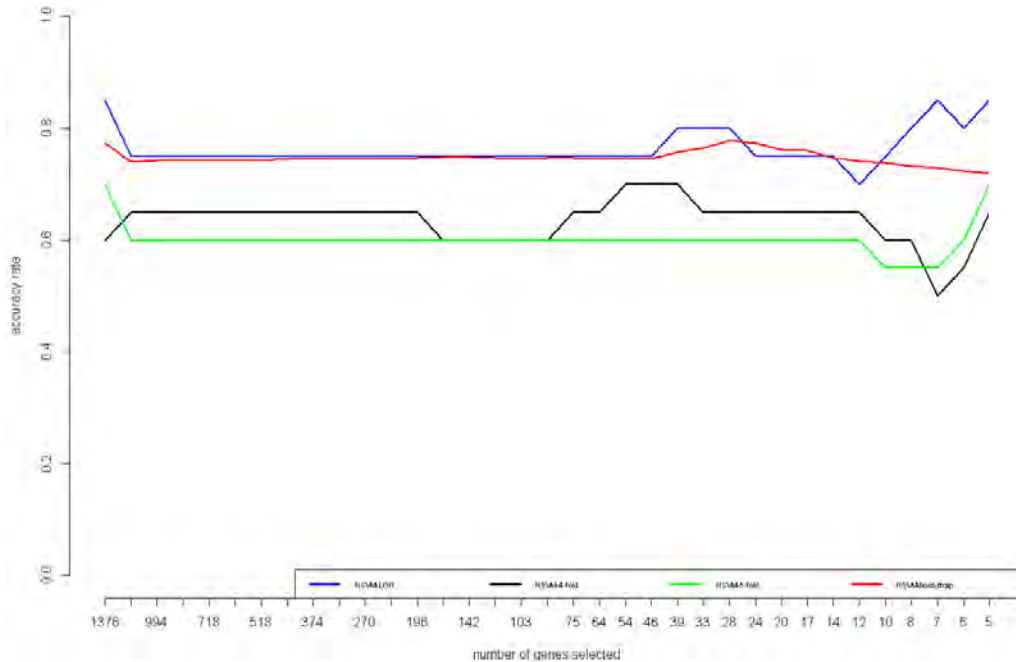
Figure 5.2. Rat's Urines Datasets Performance

In case 2, Rat's urines dataset, the performance of accuracy using RSVM is poorer than Human urines dataset although it is more stable. The performance of 4-Fold and 5-fold seems similar, while it looks also similar for the performance of bootstrap and LOO. The best accuracy rates is achieved by LOO in around 85 percent (See Figure 5.2).

Table 5.3. Rat's Urines Accuracy Performance

	LOO	fold4	fold5	bootstrap
Features:1376	0.85	0.60	0.70	0.7727019
Features:1170	0.75	0.65	0.60	0.7392758
Features:994	0.75	0.65	0.60	0.7431755
Features:845	0.75	0.65	0.60	0.7426184
Features:718	0.75	0.65	0.60	0.7431755
Features:610	0.75	0.65	0.60	0.7431755
Features:518	0.75	0.65	0.60	0.7431755
Features:440	0.75	0.65	0.60	0.7448468
Features:374	0.75	0.65	0.60	0.7448468
Features:318	0.75	0.65	0.60	0.7448468
Features:270	0.75	0.65	0.60	0.7448468
Features:230	0.75	0.65	0.60	0.7459610
Features:196	0.75	0.65	0.60	0.7470752
Features:167	0.75	0.60	0.60	0.7481894
Features:142	0.75	0.60	0.60	0.7476323
Features:121	0.75	0.60	0.60	0.7470752
Features:103	0.75	0.60	0.60	0.7454039
Features:88	0.75	0.60	0.60	0.7465181
Features:75	0.75	0.65	0.60	0.7470752
Features:64	0.75	0.65	0.60	0.7459610
Features:54	0.75	0.70	0.60	0.7465181
Features:46	0.75	0.70	0.60	0.7459610
Features:39	0.80	0.70	0.60	0.7565460
Features:33	0.80	0.65	0.60	0.7643454
Features:28	0.80	0.65	0.60	0.7771588
Features:24	0.75	0.65	0.60	0.7727019
Features:20	0.75	0.65	0.60	0.7610028
Features:17	0.75	0.65	0.60	0.7604457
Features:14	0.75	0.65	0.60	0.7470752
Features:12	0.70	0.65	0.60	0.7403900
Features:10	0.75	0.60	0.55	0.7387187
Features:8	0.80	0.60	0.55	0.7320334
Features:7	0.85	0.50	0.55	0.7286908
Features:6	0.80	0.55	0.60	0.7236769
Features:5	0.85	0.65	0.70	0.7192201

There is like a confusing in classification while using thousand features, when the features are highly corelated and redundant. It can be shown that the result using many features do not always tend to achive a better accuracy than using few features. If we could select the most appropriate features, although it is not many, we could achive a good or even better than using high-level features.

In rat's urines dataset (Table 5.3), we can choose five features (LOO, 4-fold, and 5-fold) and 10 features (bootstrap) for low-level, while selecting 54 features (LOO, 4-Fold, and 5-Fold) and 88 features (bootstrap) for middle-level. In this case, we know that the performance of low-level is not always higher than the performance of middle-level like in Human urines dataset. The highest accuracy is obtained by LOO CV that achieve 85 percent of accuracy classification by just using five features. By all 20 samples, there are 17 samples classified correctly and there are three samples classified incorrectly.
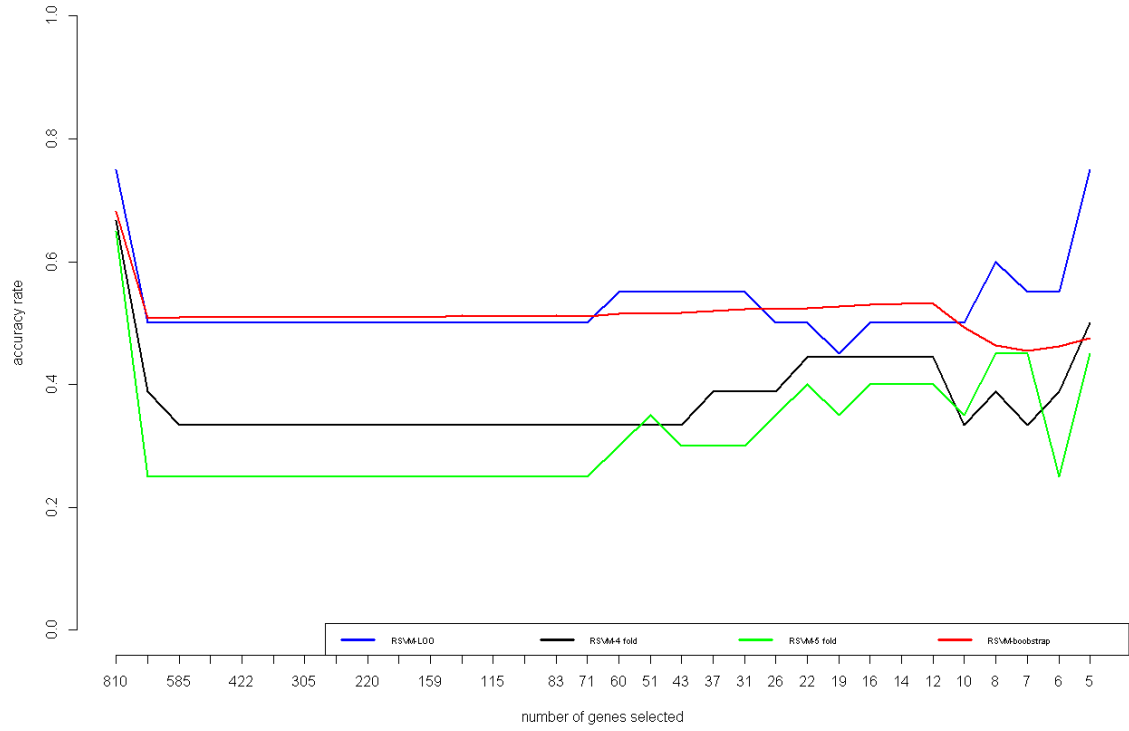
Figure 5.3. Rat's Plasma performance

Recursive SVM as a selection variable technique based on SVM rules could not achieve a quite good in our last dataset, i.e, Rat's plasma dataset (case 3). The accuracy performance is always under 80 percent. Besides that, by observing all three cases, we can see that the performance of N-fold is always poorer than bootstrap and LOO. The performance of bootstrap and LOO is quite similar, but LOO always leads to the best accuracy (See Figure 5.3).

In Rat's plasma dataset (Table 5.4), it is shown that RSVM couldn't achieve a good results since the accuracy is around 50 percent. The highest accuracy is obtained by LOO CV that achieve just 75 percent, that means 15 samples are classified correctly, and 5 samples are not. For low-level, we can choose five features (LOO, 4-fold, and 5-fold), and twelve features (bootstrap), while selecting 51 features (LOO, 4-Fold, and 5-Fold) and 60 features (bootstrap) for middle-level.

By accuracy rates, N-fold shows it's concistenty in obtaining the lowest value. Besides that, it is indicated that LOO is the most appropriate cross-validation method to use in RSVM classification because it always leads the highest performance, comparing to the other CV methods used. Although bootstrap's accuracy rates is more stable than LOO, since it consumes much more computational time and achieve similar accuracy to (or even less than) LOO, we suggest to choose LOO.

Table 5.4. Rat's Plasma Accuracy Performance

```
                    LOO     fold4   fold5 bootstrap
Features:810 0.75 0.6666667  0.65 0.6817193
Features:688 0.50 0.3888889  0.25 0.5081610
Features:585 0.50 0.3333333  0.25 0.5087051
Features:497 0.50 0.3333333  0.25 0.5097933
Features:422 0.50 0.3333333  0.25 0.5097933
Features:359 0.50 0.3333333  0.25 0.5092492
Features:305 0.50 0.3333333  0.25 0.5092492
Features:259 0.50 0.3333333  0.25 0.5092492
Features:220 0.50 0.3333333  0.25 0.5097933
Features:187 0.50 0.3333333  0.25 0.5097933
Features:159 0.50 0.3333333  0.25 0.5097933
Features:135 0.50 0.3333333  0.25 0.5119695
Features:115 0.50 0.3333333  0.25 0.5103373
Features:98  0.50 0.3333333  0.25 0.5108814
Features:83  0.50 0.3333333  0.25 0.5130577
Features:71  0.50 0.3333333  0.25 0.5114255
Features:60  0.55 0.3333333  0.30 0.5152339
Features:51  0.55 0.3333333  0.35 0.5146899
Features:43  0.55 0.3333333  0.30 0.5163221
Features:37  0.55 0.3888889  0.30 0.5201306
Features:31  0.55 0.3888889  0.30 0.5223069
Features:26  0.50 0.3888889  0.35 0.5228509
Features:22  0.50 0.4444444  0.40 0.5244831
Features:19  0.45 0.4444444  0.35 0.5277476
Features:16  0.50 0.4444444  0.40 0.5304679
Features:14  0.50 0.4444444  0.40 0.5321001
Features:12  0.50 0.4444444  0.40 0.5321001
Features:10  0.50 0.3333333  0.35 0.4929271
Features:8   0.60 0.3888889  0.45 0.4630033
Features:7   0.55 0.3333333  0.45 0.4553863
Features:6   0.55 0.3888889  0.25 0.4619151
Features:5   0.75 0.5000000  0.45 0.4760609
```

To compare all CV method, it is shown two types point of view, which are in accuracy rates comparison point of view (Table 5.5) and features selected point of view (Figure 5.4). We would like to describe the low-level analysis, while middle-level analysis is presented in Appendix 2d.

Human urines datasets is able to be classified well by using RSVM since it is confirmed that the two classes are able to be differentiated easily. That is why the accuracy rates obtained is highest. In other side, Rat's plasma dataset achieves poorest accuracy since the two clases are more complicated. Comparing to the other CV methods, LOO always leads the best performance, quite different with other CV methods achieved.

Table 5.5. Summary of RSVM low-level Result

Human Urines		Rat's Urines		Rat's Plasma	
CV	F|A	CV	F|A	CV	F|A
LOO	6|93	LOO	5|85	LOO	5|75
4-Fold	8|75	4-Fold	5|65	4-Fold	5|50
7-Fold	5|82	5-Fold	5|70	5-Fold	5|45
Bootstrap	5|85	Bootstrap	11|74	Bootstrap	12|53

Notes : CV: Cross-Validation, F: Number of Features selected, A=Accuracy rates (%)
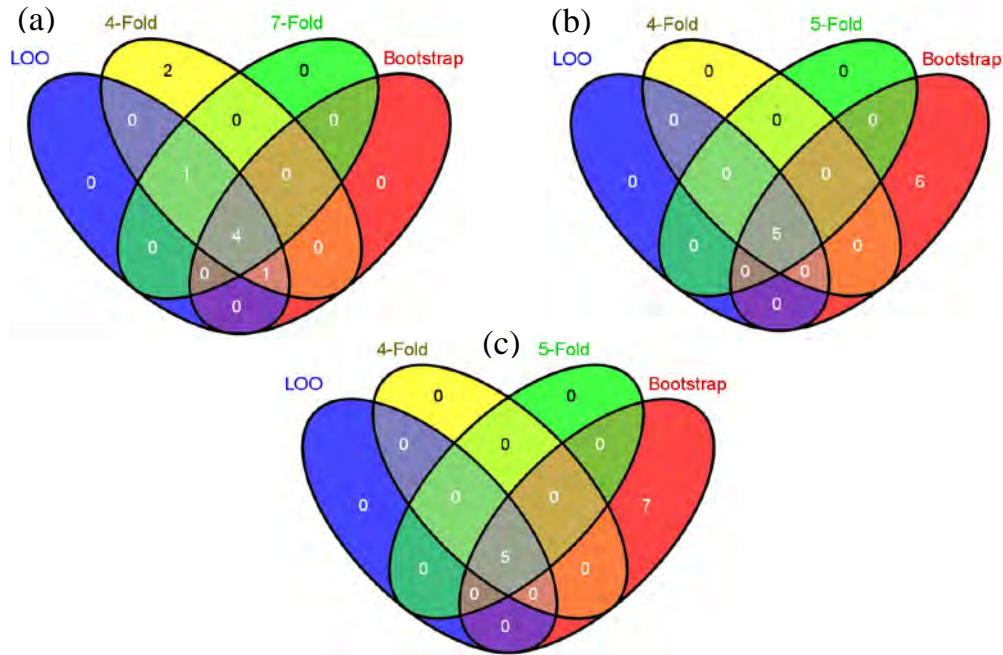
Figure 5.4. Comparison of features selected in : (a) Human Urines datasets, (b) Rat's Urines datasets, and (c) Rat's Plasma datasets

To better understand about the meaning, we make a venn diagram of features selected that is devided by CV methods used for each case (See Figure 5.4). The features selected are quite similar for 4 CV methods used in each case (detailed in Table 5.6). Figure 5.4a shows that LOO selects four features that are also selected by other three cv methods (even there are 5 features same with 4-fold and 7 fold, and also 5 features which are same with bootstrap). In addition, LOO selects the features which are exactly selected by other three cv methods. It means that LOO is really the most appropriate CV methods in low-level analysis because they use similar features to classify the dataset and LOO is able to achieve much higher.

Table 5.6. The Common Features Selected by RSVM

Human Urines	M302T478	M310T428	M207T97	M169T41	
Rat's Urines	M180T194	M194T256	M338T217	M340T269	M206T158
Rat's Plasma	M875T275_1	M494T437	M496T473	M524T539	M524T551

5.2 Analysis Using Random Forest for Feature Selection (RFFS)

Random forest is an algorithm for classification developed by Leo Breiman that uses an ensemble of classification trees. Each of the classification trees is built using a bootstrap sample of data, and each split the candidate set of variables is a random subset of the variables. Thus, random forest uses both bagging and random variable selection for tree building. The algorithm yields an ensemble that can achieve both low bias and low variance.

Random forest returns several measures of variable importance. The most reliable measure is based on the decrease of classification accuracy when values of the variable

in a node of tree are permuted randomly. To select features, we iteratively fit random forest, at each iteration building a new forest after discarding those variables with the smallest variable importance. The selected set of features is the one that yields the smallest Out-of-Bag (OOB) error rate.

Table 5.7. Analysis using Feature Selection Random Forest

	features1	OOB1	features2	OOB2	features3	OOB3
[1,]	1271	0	1376	0.25	810	0.25
[2,]	1017	0	1101	0.25	648	0.25
[3,]	814	0	881	0.20	518	0.20
[4,]	651	0	705	0.20	414	0.20
[5,]	521	0	564	0.20	331	0.15
[6,]	417	0	451	0.15	265	0.10
[7,]	334	0	361	0.15	212	0.10
[8,]	267	0	289	0.15	170	0.10
[9,]	214	0	231	0.15	136	0.15
[10,]	171	0	185	0.10	109	0.10
[11,]	137	0	148	0.10	87	0.10
[12,]	110	0	118	0.15	70	0.10
[13,]	88	0	94	0.05	56	0.10
[14,]	70	0	75	0.10	45	0.10
[15,]	56	0	60	0.10	36	0.10
[16,]	45	0	48	0.05	29	0.10
[17,]	36	0	38	0.05	23	0.10
[18,]	29	0	30	0.05	18	0.10
[19,]	23	0	24	0.05	14	0.10
[20,]	18	0	19	0.05	11	0.10
[21,]	14	0	15	0.05	9	0.10
[22,]	11	0	12	0.05	7	0.05
[23,]	9	0	10	0.05	6	0.05
[24,]	7	0	8	0.05	5	0.05
[25,]	6	0	6	0.05		
[26,]	5	0	5	0.05		

As shown in Table 5.7, when using high-level features, the OOB error rates obtained are quite poor. It indicates that using high-level features (even all features) is not proposed. OOB error rates decrease almost linearly with the decreasing of features level selected. After all, we choose five features for each case (human urines, rat's urines, and rat's plasma) because it leads to get minimum error rates. Besides that, we should see that the performance of middle-level is not better than low-level performance.

Since it is confirmed that two classes of Human Urines datasets could be differentiated easily, the error rates obtained is always zero, even for low level. It means that, by just using five features, all 28 samples are classified correctly. Although RFFS could not achieve zero error rates, the performance is still satisfied since it achieves 0.05 error rates for Rat's Urines datasets and Rat's Plasma datasets. Analyzing these two datasets, it means RFSS is able to classify the 19 samples correctly, and just one sample is not.

RFSS is a classification technique that returns small sets of features selected because this technique will not return sets of features that are highly correlated, because they are redundant. That's why RFSS achieves a better accuracy in low-level, i.e., five features for our three cases. This method will be most useful under scenario when considering the design of diagnostic tools, when having small sets features is often desirable.

5.3   PLS-DA Feature Selection based on VIP

Partial Least Squares regression makes it possible to relate a set of dependent variables to a set of independent variables when the number of independent and/or dependent variables is much larger than the number of observations. However, when the number of independent variables is very large compare to the number of observation, it will give impact for PLS-DA analysis, even though we create components. That is because the impact of noisy data as well as redundancy data. Because of this reason, PLS-DA is not sufficient and we still need to select several important variables before creating the components. Besides that, a fundamental requirement for PLS to yield meaningful answers is some preliminary variable selection. Enciso and Tenenhaus (2003) did this feature selection technique by selecting the variables on the basis of VIP for each variable.

After using machine learning analysis, in this part, we would like to show the result analysis of our three datasets treatment using supervised classification, i.e., PLS-DA feature selection technique based on VIP and sPLS-DA for next section.

Not like machine learning rules that use only accuracy rates as performance criteria, we use also two other criteria, i.e., R^2, Q^2, since the estimation of accuracy is overestimed in PLS-DA. It is also shown in Figure 5.5 that this method could reach 100 percent for almost all feature levels selected for all three cases. After all, to select the number of features, we select the levels that produce the largest R^2, Q^2, and accuracy rate, with the closest distance between R^2 and Q^2.
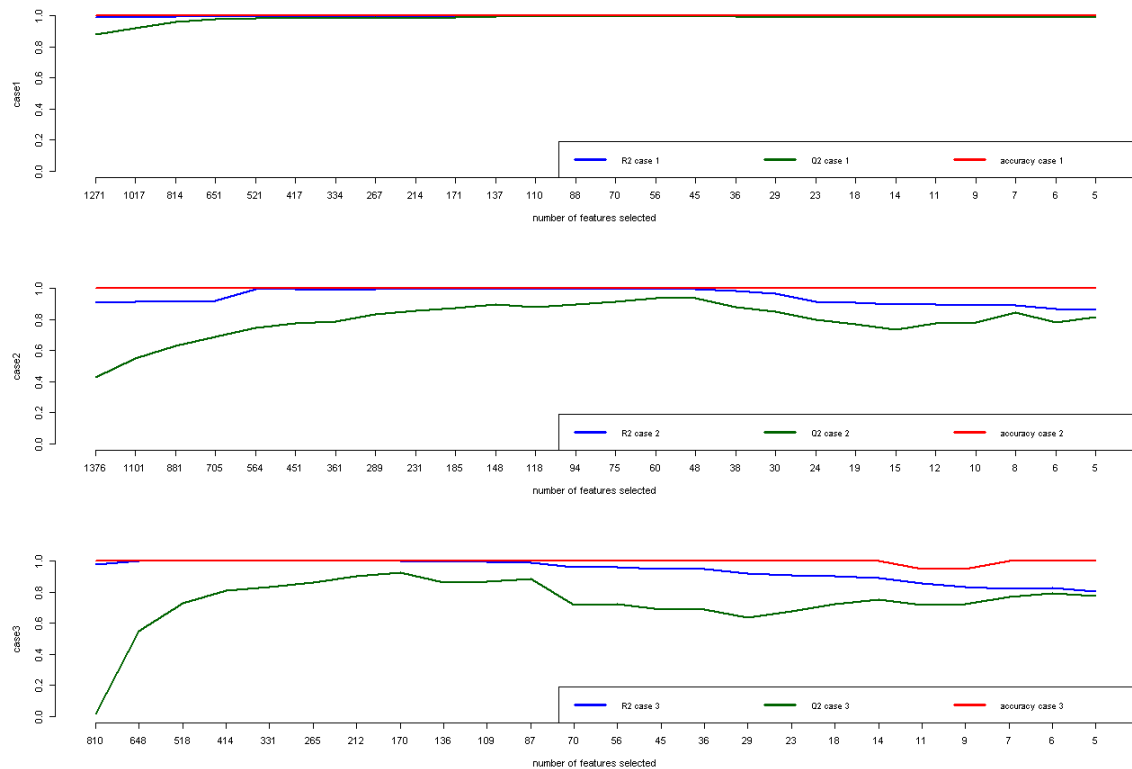

Figure 5.5. The Score of R^2, Q^2, and Accuracy Rate for each Case

28

To obtain the optimum result, we choose different number of components selected for each level using Q^2_j criterion (the PLSDA model having the highest Q^2 was kept). This idea is logic where each level of selected features defines different number of selected components.

Analyzing human urines dataset, the accuracy rate obtained is very perfect when using all features (see Table 5.8). However, we can see in Figure 5.5 that there is a relatively large distance between R^2 and Q^2 in this level which indicates selecting all features is not the best choice, as also shown in RFSS analysis. We surmise that there is a significant effect of noisy data in this level. This phenomena is also happened in rat's urines and rat's plasma dataset (see Figure 5.5), where the largest distance is placed in the first level when using all features.

The distance go downly by the decreasing number of variables selected. As stated that Human urines dataset is clearly differentiated both in two classes, so the performance is always well in almost all levels. Analyzing rat's urines and rat's plasma datasets, we can see that about 200 features selected, the distance fluctuates smoothly in low value. We indicates that the best choice is to select the features around five and 200 features.

Table 5.8. Human Urines Dataset's Selected Components and It's Performance

	Nb of comp	R2	Q2	Q2-R2	accuracy
Features:1271	2	0.9908164	0.8813312	0.109485176	1
Features:1017	2	0.9909920	0.9221698	0.068822203	1
Features:814	4	0.9990854	0.9604008	0.038684572	1
Features:651	4	0.9991370	0.9772226	0.021914473	1
Features:521	4	0.9991994	0.9866695	0.012529855	1
Features:417	3	0.9978066	0.9831169	0.014689738	1
Features:334	3	0.9975167	0.9868576	0.010659106	1
Features:267	4	0.9989806	0.9912973	0.007683324	1
Features:214	2	0.9958153	0.9859041	0.009911242	1
Features:171	4	0.9990407	0.9928112	0.006229523	1
Features:137	4	0.9993084	0.9946179	0.004690510	1
Features:110	4	0.9995425	0.9958015	0.003741019	1
Features:88	3	0.9990705	0.9937978	0.005272757	1
Features:70	3	0.9989248	0.9942969	0.004627895	1
Features:56	3	0.9992077	0.9949546	0.004253066	1
Features:45	4	0.9995507	0.9974800	0.002070734	1
Features:36	3	0.9994636	0.9974189	0.002044759	1
Features:29	2	0.9989607	0.9914811	0.007479609	1
Features:23	2	0.9989907	0.9933156	0.005675067	1
Features:18	3	0.9991712	0.9980733	0.001097953	1
Features:14	2	0.9978050	0.9907847	0.007020271	1
Features:11	2	0.9978277	0.9909598	0.006867869	1
Features:9	2	0.9976242	0.9922309	0.005393260	1
Features:7	2	0.9966577	0.9932432	0.003414526	1
Features:6	2	0.9960863	0.9925722	0.003514102	1
Features:5	2	0.9959755	0.9944453	0.001530175	1

The number of components selected is varied from two to four components (See Table 5.8), where the the high-level (over 1000 features) and low-level (under 15 features) choose the lowest number of components. It is clearly shown that our first case is well classified, by looking at criteria of R^2, Q^2 and accuracy that is always higher than 99 percent, except Q^2 score at high level (more than 200 features, but it is still very good). By using two components from five features selected, 99.6 percent of human urines

classes variability is explained by the model with predictive ability that's equal to 99.4 percent. It does not have a big difference with middle level using three components from 56 features selected (the model can explain 99.9 percent of total variability and has 99.5 percent of predictive ability).

Table 5.9. Rat's Urines Dataset's Selected Components and It's Performance

	Nb of comp	R2	Q2	Q2-R2	accuracy
Features:1376	2	0.9111545	0.4285984	0.48255608	1
Features:1101	2	0.9129031	0.5480000	0.36490310	1
Features:881	2	0.9156964	0.6318897	0.28380664	1
Features:705	2	0.9194253	0.6872336	0.23219164	1
Features:564	4	0.9950133	0.7489174	0.24609596	1
Features:451	4	0.9947185	0.7745323	0.22018616	1
Features:361	3	0.9914623	0.7904004	0.20106193	1
Features:289	4	0.9949143	0.8366565	0.15825786	1
Features:231	4	0.9943987	0.8557423	0.13865636	1
Features:185	4	0.9947324	0.8760232	0.11870929	1
Features:148	4	0.9951128	0.8958061	0.09930672	1
Features:118	4	0.9942080	0.8818587	0.11234932	1
Features:94	4	0.9945935	0.8968175	0.09777598	1
Features:75	4	0.9938828	0.9149509	0.07893188	1
Features:60	4	0.9949043	0.9406060	0.05429831	1
Features:48	4	0.9938825	0.9371156	0.05676693	1
Features:38	4	0.9873298	0.8790258	0.10830406	1
Features:30	3	0.9650909	0.8488803	0.11621058	1
Features:24	2	0.9138793	0.8008479	0.11303142	1
Features:19	2	0.9067631	0.7723893	0.13437379	1
Features:15	2	0.8963762	0.7373421	0.15903413	1
Features:12	2	0.8982126	0.7776854	0.12052713	1
Features:10	2	0.8901992	0.7811275	0.10907165	1
Features:8	2	0.8911702	0.8444107	0.04675957	1
Features:6	2	0.8695033	0.7835420	0.08596128	1
Features:5	2	0.8646273	0.8157507	0.04887652	1

Like Human urines dataset, the number of components selected of rat's urines dataset (Table 5.9) has a particular pattern like Bell-shape, where it starts selecting with small number of components for high level, increase in middle level, and decrease in low level. It is also happening for our rat's plasma dataset (shown in Table 5.10).

Analyzing rat's urines dataset, we consider eight features (two components) for low level and 60 features (four components) for middle level. For rat's plasma dataset, we consider six features (two components) for low level and 87 (three components) features for middle level. Not like human urines dataset, the performance of low-level is lower than the permormance of middle-level in rat's urines and rat's plasma dataset. In rat's urines dataset, the performance of low-level is approximately 10 percent lower than the middle-level. Furthermore, performance of low-level in rat's plasma dataset is approximately 17 percent lower for R^2 and 9 percent lower for Q^2.

Table 5.10. Rat's Palsma Dataset's Selected Components and It's Performance

```
              Nb of comp      R2         Q2       Q2-R2 accuracy
Features:810        2  0.9810562 0.01432494 0.96673124     1.00
Features:648        4  0.9993895 0.54955150 0.44983800     1.00
Features:518        4  0.9995372 0.72917677 0.27036044     1.00
Features:414        4  0.9997921 0.81180479 0.18798734     1.00
Features:331        4  0.9997323 0.83211360 0.16761868     1.00
Features:265        4  0.9997163 0.86565951 0.13405681     1.00
Features:212        4  0.9997041 0.90573943 0.09396464     1.00
Features:170        4  0.9997552 0.92613263 0.07362257     1.00
Features:136        3  0.9975110 0.86284282 0.13466820     1.00
Features:109        3  0.9938217 0.86643182 0.12738983     1.00
Features:87         3  0.9924037 0.88590651 0.10649715     1.00
Features:70         2  0.9610839 0.72031958 0.24076432     1.00
Features:56         2  0.9593909 0.72418570 0.23520523     1.00
Features:45         2  0.9515053 0.68990226 0.26160300     1.00
Features:36         2  0.9519781 0.69097187 0.26100624     1.00
Features:29         2  0.9213534 0.63742029 0.28393310     1.00
Features:23         2  0.9092161 0.68031272 0.22890338     1.00
Features:18         2  0.9059875 0.72648845 0.17949906     1.00
Features:14         2  0.8921305 0.75407547 0.13805500     1.00
Features:11         2  0.8550647 0.71689113 0.13817360     0.95
Features:9          2  0.8322472 0.72404498 0.10820220     0.95
Features:7          2  0.8234507 0.76763838 0.05581234     1.00
Features:6          2  0.8269796 0.79589797 0.03108159     1.00
Features:5          2  0.8065974 0.77880733 0.02779012     1.00
```

PLS-DA feature selection based on VIP is a technique that could achieve well the accuracy rate, as well as the score of R^2 and Q^2. In low level, This model can explain 89 percent total variability of rat's urines classes and 83 percent of rat's plasma classes, while it has a predictive ability of 84 percent for rat's urines dataset and 80 percent for rat's plasma dataset. List of features selected is presented in Appendix 3.

5.4   Sparse PLS-DA (sPLS-DA)

sPLS-DA is based on Partial Least Squares (PLS) Regression for discrimation analysis, but a lasso penalization has been added to select variables. Lê Cao, Boitard, and Besse (2011) showed that sPLS-DA is extremely competitive to the wrapper methods. In addition, the computational efficiency of sPLS-DA as well as the valuable graphical outputs that provide easier interpretation of the results make sPLS-DA a great alternative to other types of variables selection techniques in a supervised classification framework. In this part, we would like to show the the result of sPLS-DA analysis in our three datasets.

Our Human urines dataset is always well classified as confirmed that both of two classes are differentiated contrastly. In this analysis, we use three criteria to judge the performance, i.e., accuracy rate, R^2, and Q^2, because the only accuracy rate is not sufficient since it is overestimated in PLS-based analysis. Human urines classification performance always closes to a hundred percent (see Figure 5.6), while the score of R^2 and Q^2 is relatively varied for rat's urines and rat's plasma datasets, since the two classes is quite complicated. Compared to PLS-DA VIP (see Figure 5.5), the performance obtained by sPLS-DA looks quite better.
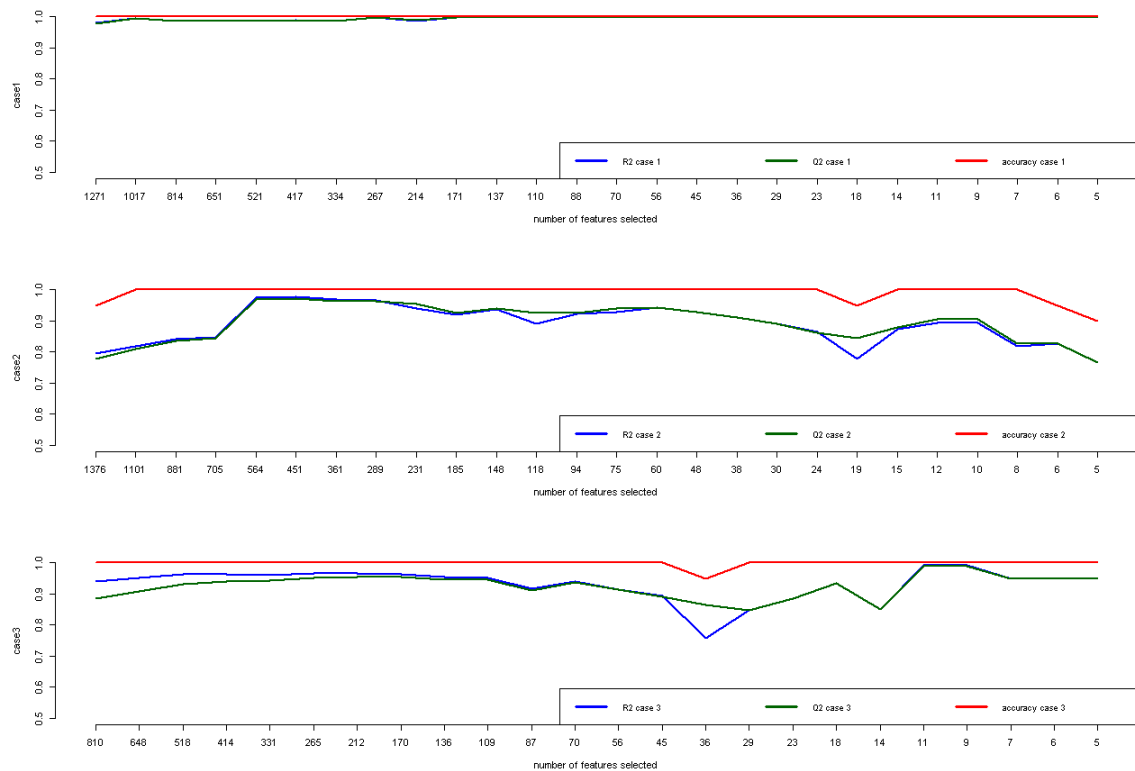
Figure 5.6. The Performance Using sPLS-DA Analysis

Since sPLS-DA works with loading factors to select the features, it optimizes the selection of features by computing the PLS-DA model. This method affects to the better selection of PLS-DA model for each level, indicated by similar value obtained of R^2 and Q^2 in Figure 5.6.

Table 5.11 shows the performance of human urines classification. sPLS-DA is able to choose the features based on the loading factors from each iteration. This technique is quitely surprising since it produce very good performance, in either variability explanation performance, predictive ability, or accuracy rate. Unlike PLS-DA feature selection based on VIP, the number of components selected in sPLS-DA does not follow Bell-shape. Because the performance is very well for all levels (the criteria is always more than 98 percent), it doesn't matter to choose the lowest level of feature selection. Basicly, the different class is differentiated clearly so that a low level of features selected is sufficient.

To better understand about contaminated and not contaminated rat's urines using sPLS-DA, we show the result of analysis in Table 5.12. In the high level of features used, the performance is not quite good. These three score criteria are relatively increased based on the iteration and it decreases again in low level.

Table 5.11. Human Urines Dataset's Selected Components and It's Performance

```
               Nb of comp       R2        Q2      Distance accuracy
Features:1271         4  0.9914684 0.9905652 9.032142e-04      100
Features:1017         4  0.9908951 0.9902920 6.030576e-04      100
Features:814          5  0.9931671 0.9941041 9.370245e-04      100
Features:651          5  0.9929531 0.9953848 2.431629e-03      100
Features:521          5  0.9965659 0.9967131 1.471710e-04      100
Features:417          5  0.9976423 0.9975635 7.881294e-05      100
Features:334          2  0.9874981 0.9874643 3.380471e-05      100
Features:267          2  0.9846424 0.9902706 5.628178e-03      100
Features:214          5  0.9974809 0.9974001 8.075442e-05      100
Features:171          2  0.9898469 0.9898306 1.630340e-05      100
Features:137          2  0.9914221 0.9916403 2.182398e-04      100
Features:110          2  0.9916366 0.9916253 1.129203e-05      100
Features:88           2  0.9904584 0.9904472 1.122496e-05      100
Features:70           2  0.9912234 0.9912120 1.138931e-05      100
Features:56           2  0.9900832 0.9900704 1.285769e-05      100
Features:45           2  0.9908182 0.9908077 1.054361e-05      100
Features:36           5  0.9979260 0.9979112 1.473533e-05      100
Features:29           5  0.9985902 0.9985437 4.648603e-05      100
Features:23           5  0.9979404 0.9979055 3.493097e-05      100
Features:18           5  0.9986579 0.9986031 5.481605e-05      100
Features:14           3  0.9969203 0.9968867 3.360147e-05      100
Features:11           4  0.9971737 0.9971634 1.030550e-05      100
Features:9            3  0.9967897 0.9967710 1.869424e-05      100
Features:7            5  0.9968659 0.9968639 2.039298e-06      100
Features:6            5  0.9968659 0.9968639 2.039298e-06      100
Features:5            5  0.9968659 0.9968639 2.039298e-06      100
```

In low-level, we can be satisfied by selecting ten features (5 components), while we consider to choose 60 features (4 components) in middle-level. By using ten features, the model can explain 90 percent of the total variability of rat's urines different classes with 91 percent of predictive ability. Therefore, sPLS-DA model is able to explain 94 percent of the total variability of rat's urines different classes with 94 percent of predictive ability in middle-level (see Table 5.12).

Table 5.12. Rat's Urines Dataset's Selected Components and It's Performance

```
               Nb of comp       R2        Q2      Distance accuracy
Features:1376        2  0.7963323 0.7787452 1.758708e-02       95
Features:1101        2  0.8192211 0.8090468 1.017433e-02      100
Features:881         2  0.8413899 0.8346795 6.710410e-03      100
Features:705         2  0.8483283 0.8442695 4.058802e-03      100
Features:564         5  0.9753704 0.9696540 5.716398e-03      100
Features:451         5  0.9774576 0.9725566 4.900957e-03      100
Features:361         5  0.9678506 0.9642446 3.605962e-03      100
Features:289         5  0.9648615 0.9634974 1.364186e-03      100
Features:231         5  0.9408892 0.9554292 1.453997e-02      100
Features:185         5  0.9201309 0.9265619 6.431029e-03      100
Features:148         5  0.9371352 0.9413072 4.172054e-03      100
Features:118         5  0.8923623 0.9250527 3.269038e-02      100
Features:94          4  0.9233456 0.9250527 1.707073e-03      100
Features:75          4  0.9293306 0.9403069 1.097625e-02      100
Features:60          4  0.9433932 0.9431716 2.216252e-04      100
Features:48          3  0.9284022 0.9283512 5.098846e-05      100
Features:38          4  0.9101804 0.9100668 1.135532e-04      100
Features:30          2  0.8896898 0.8896892 6.221405e-07      100
Features:24          3  0.8635025 0.8632007 3.017785e-04      100
Features:19          3  0.7766210 0.8433913 6.677036e-02       95
Features:15          5  0.8731695 0.8801314 6.961984e-03      100
Features:12          5  0.8951543 0.9064202 1.126589e-02      100
Features:10          5  0.8951543 0.9064202 1.126589e-02      100
Features:8           3  0.8183291 0.8278871 9.558053e-03      100
Features:6           2  0.8283333 0.8278871 4.461617e-04       95
Features:5           2  0.7671218 0.7660001 1.121708e-03       90
```

33

Rat's plasma dataset is also analyzed using sPLS-DA technique in features selection purpose. The result is presented in Table 5.13. Like rat's urines, this classification is diffetentiated by two classes, contaminated and not contaminated plasma. By all four criteria, we choose nine features in low level (five components) and 70 features (three components) in middle level that is presented in Appendix 3.

In low level, the model can explain 99 percent of the total variability of rat's plasma different classes with 99 percent of predictive ability. Therefore, sPLS-DA model is able to explain 94 percent of the total variability of rat's plasma different classes with 94 percent of predictive ability in middle-level.

sPLS-DA working way by selecting the features and creating the components from loading factors achieves very well the accuracy rate, as well as the score of R^2 and Q^2. In low level, This model can explain at least 90 percent total variability and it has a predictive ability more than 94 percent. In addition, this model could achive a hundred percent of accuracy rate for all three cases.

Table 5.13. Rat's Plasma Dataset's Selected Components and It's Performance

	Nb of comp	R2	Q2	Distance	accuracy
Features:810	4	0.9392172	0.8863339	5.288327e-02	100
Features:648	4	0.9510196	0.9086320	4.238763e-02	100
Features:518	4	0.9630113	0.9308490	3.216239e-02	100
Features:414	4	0.9634885	0.9394199	2.406851e-02	100
Features:331	4	0.9615811	0.9435682	1.801293e-02	100
Features:265	4	0.9656913	0.9528627	1.282854e-02	100
Features:212	4	0.9662258	0.9549681	1.125775e-02	100
Features:170	4	0.9629563	0.9534406	9.515736e-03	100
Features:136	4	0.9547934	0.9473049	7.488473e-03	100
Features:109	5	0.9518681	0.9470862	4.781907e-03	100
Features:87	2	0.9168406	0.9125077	4.332925e-03	100
Features:70	3	0.9415654	0.9371960	4.369415e-03	100
Features:56	3	0.9153494	0.9132783	2.071121e-03	100
Features:45	3	0.8933687	0.8922832	1.085568e-03	100
Features:36	4	0.7577367	0.8660217	1.082851e-01	95
Features:29	3	0.8474206	0.8473994	2.116510e-05	100
Features:23	3	0.8861185	0.8860339	8.462267e-05	100
Features:18	3	0.9355897	0.9351846	4.050375e-04	100
Features:14	4	0.8517874	0.8517631	2.427292e-05	100
Features:11	5	0.9914826	0.9904862	9.963312e-04	100
Features:9	5	0.9914826	0.9904862	9.963312e-04	100
Features:7	5	0.9489667	0.9480525	9.141503e-04	100
Features:6	5	0.9489667	0.9480525	9.141503e-04	100
Features:5	5	0.9489667	0.9480525	9.141503e-04	100

5.5 Methods Comparison

As stated, a key difficulty in the Metabolomic study is the noisy nature of the data (Rakotomamonjy, 2003 and Zhang, et al., 2006), which can be caused by the intrinsic complexity of the biological problem, as well as experimental and technical biases. Another difficulty arises from the high dimensionally of the data while the training samples are very scarce, even after pre-processing steps such as peak and/or biomarker detection, the dimensionality is usually much larger than the sample size.

Both in machine learning and supervised classification techniques used in this study, one solution proposed is to reduce the dimensionality of the data either by performing features selection, or by introducing artificial variables (i.e. latent variables). The purpose of the features or variables selection is to eliminate irrelevant variables to enhance the generalization performance of a given algorithm (Rakotomamonjy, 2003) as well as to gain some insight about the concept learned (Diaz-Uriarte & Andres, 2006). Other advantages of feature selection and introducing artificial variables include cost reduction of data gathering and storage, and also on computational speedup. We choose the machine learning technique, i.e., RSVM and RFFS, as proposed techniques in features selection and supervised classification technique, i.e., PLS-DA VIP and sPLS-DA, as the technique of introducing artificial variables.

After one-by-one technique interpretation section, we would like to compare all performance techniques used in classifying our three datasets. We consider three point of views to evaluate our three techniques : (1) Accuracy rates (R^2 and Q^2 for supervised classification techniques), (2) Computational time needed, and (3) Similarity of features selected. After all, one main point, we would like to evaluate our four techniques by biological interpretation for the next section to gain some insight and meaning about the concept learned.

Table 5.14. Low-level Performance Comparison

Method	Human Urines	Rat's Urines	Rat's Plasma
RSVM-LOO	6 (92,86%)	5 (85,00%)	5 (75,00%)
RSVM-Bootstrap	6 (84,73%)	10 (73,87%)	12 (53,21%)
RSVM-4Fold	7 (75,00%)	5 (65,00%)	5 (50,00%)
RSVM-7(5)Fold	5 (82,14%)	5 (70,00%)	5 (45,00%)
RFFS	5 (100%)	5 (95,00%)	5 (95,00%)
PLS-DA VIP	5 (99,60%|99,44%)	8 (89,12%|84,44%)	6 (82,70%|79,59%)
sPLS-DA	5 (99,69%|99,69%)	10 (89,52%|90,64%)	9 (99,15%|99,05%)

Notes : number of features selected (accuracy) – for machine learning; number of features selected (R^2 | Q^2) – for supervised classfication

In machine learning techniques proposed, the performance of RFFS is always leading since it reachs more than 95 percent correctly classified samples, either in low-level analysis (Table 5.14) or middle-level analysis (Table 5.15). The performance of RSVM is quite good for LOO, but it is poor for other three CV methods used, particularly these three methods performance is very poor for middle-level analysis. Analyzing supervised classification techniques, PLS-DA VIP and sPLS-DA could achieve a good performance since both of them could reach more than 90 percent for R^2 and 80 percent Q^2 in low-level (Table 5.14). In addition, these techniques could reach more than 94 percent for R^2 and 89 percent Q^2 in middle-level (Table 5.15). sPLS-DA performance is quite better than PLS-DA VIP in low-level, but they are both similar in middle-level.

RFFS consistently shows their performance in low-level analysis, not only in accuracy rates, but it could also show it is able to find smallest set of features selected (i.e., five

features). Despite RSVM N-Fold could achieve small set (five features), it is not an appropriate method in this study because of their poor performance. The number of features selected in supervised classification techniques (low-level analysis, see Table 5.13) is similar, although PLS-DA could achieve little smaller set of features.

Table 5.15. Middle-level Performance Comparison

Method	Human Urines	Rat's Urines	Rat's Plasma
RSVM-LOO	56 (71,43%)	54 (75,00%)	51 (55,00%)
RSVM-Bootstrap	88 (79,53%)	88 (76,45%)	60 (51,52%)
RSVM-4Fold	56 (60,71%)	54 (70,00%)	51 (33,33%)
RSVM-7(5)Fold	56 (71,43%)	54 (60,00%)	51 (35,00%)
RFFS	56 (100%)	94 (95,00%)	56 (90,00%)
PLS-DA VIP	56 (99,92%|99,49%)	60 (99,49%|94,06%)	87 (99,24%|88,59%)
sPLS-DA	56 (99,01%|99,01%)	60 (94,34%|94,32%)	70 (94,16%|93,72%)

Notes : number of features selected (accuracy) – for machine learning; number of features selected ($R^2 \mid Q^2$) – for supervised classfication

The number of features selected by machine learning techniques, we can say, is smaller than the number of features selected by supervised classification techniques, except for RSVM-bootstrap. In the middle-level analysis (Table 5.15), N-fold and LOO RSVM achieve the smallest set, while bootstrap RSVM achieves the biggest set of features selected. The number of features selected by RFFS is a little bigger than N-Fold and LOO RSVM, but it is still in the normal range. As low-level analysis, the number of features selected in supervised classification techniques is similar. However, sPLS-DA could achieve now little smaller set of features, as an opposite of low-level analysis.

Table 5.16. Computational Time Comparison (Minutes)

Method	Human Urines	Rat's Urines	Rat's Plasma
RSVM-LOO	8	4	12
RSVM-Bootstrap	29	10	32
RSVM-4Fold	1	1	1
RSVM-7Fold (5Fold)	1	1	1
RFFS	1	1	1
PLS-DA VIP	8	12	15
sPLS-DA	4	3	6

As one of the performance measurements, computational time needed is aslo recorded to compare since it is important for efficiency purpose. N-fold RSVM and RFFS is the fastest approach (See Table 5.16). N-fold RSVM is not necessarily the one that performs the best, but is certainly the most efficient on large data sets. In contrary, RFFS chould achieve a very good performance as well as computational efficiency. The second is sPLS-DA, that could also performs a very good performance as well as less consumed computational time. Bootstrap RSVM is the approach that consumes much more time. Compare to other RSVM methods, bootstrap sampling needs much more iteration than LOO and N-fold and it causes much time consuming.

Figure 5.7. The number of features selected for low-level analysis based on technique used for :
(a) Human urines dataset, (b) Rat's urines dataset, and (c) Rat's plasma dataset

Having compared the accuracy (R^2 | Q^2), number of features selected, as well as computational time, it is also important to know the similarity of features selected based on each teachnique proposed. In section 5.1, we did the comparison of features selected based on CV methods used in RSVM and it shows the strong similarity of features selected. Because of this reason, we use only RSVM-LOO in this section since it performs best for RSVM method for low-level analysis.

Analyzing Figure 5.7a, the features selected by each technique is strongly independent since they choose different features based on their own way. This result is followed by other cases (see Figure 5.7b and Figure 5.7c) although it is not as strongly different as human urines features selection performance. Since low-level analysis presents fewer than 15 features for each technique, it is important to know their similarity in middle-level.

Table 5.17. The Similarity of Features Selected in Middle-level Analysis (%)

Method	Human Urines	Rat's Urines	Rat's Plasma
RSVM	18	18	14
RFFS	50	51	70
PLS-DA VIP	77	82	51
sPLS-DA	57	35	26

Observing the percentage of the similarity of features selected in Table 5.17, PLS-DA VIP is the most common (except for Rat's plasma). RFFS is the second one and their similarity could be achieved more than 50 percent from total features that they select. As described more clearly in Figure 5.8, most of features selected by RSVM are independent, which is why the similarity is always under 20 percent.
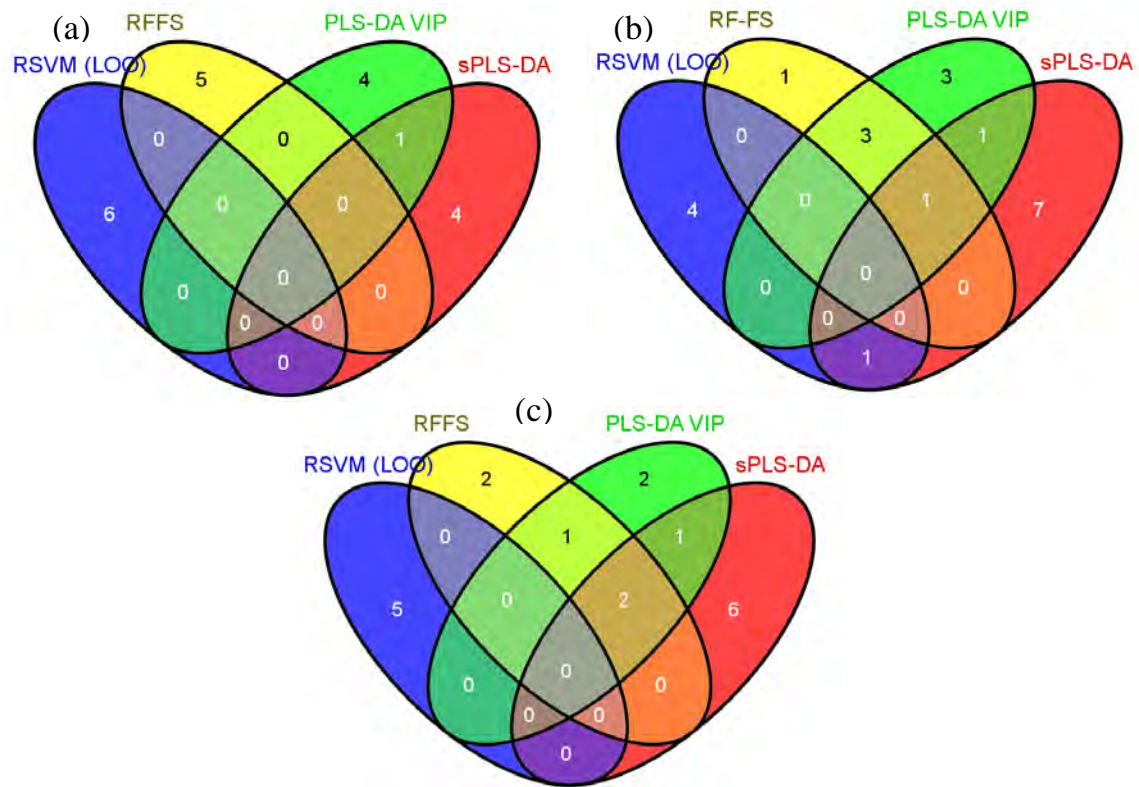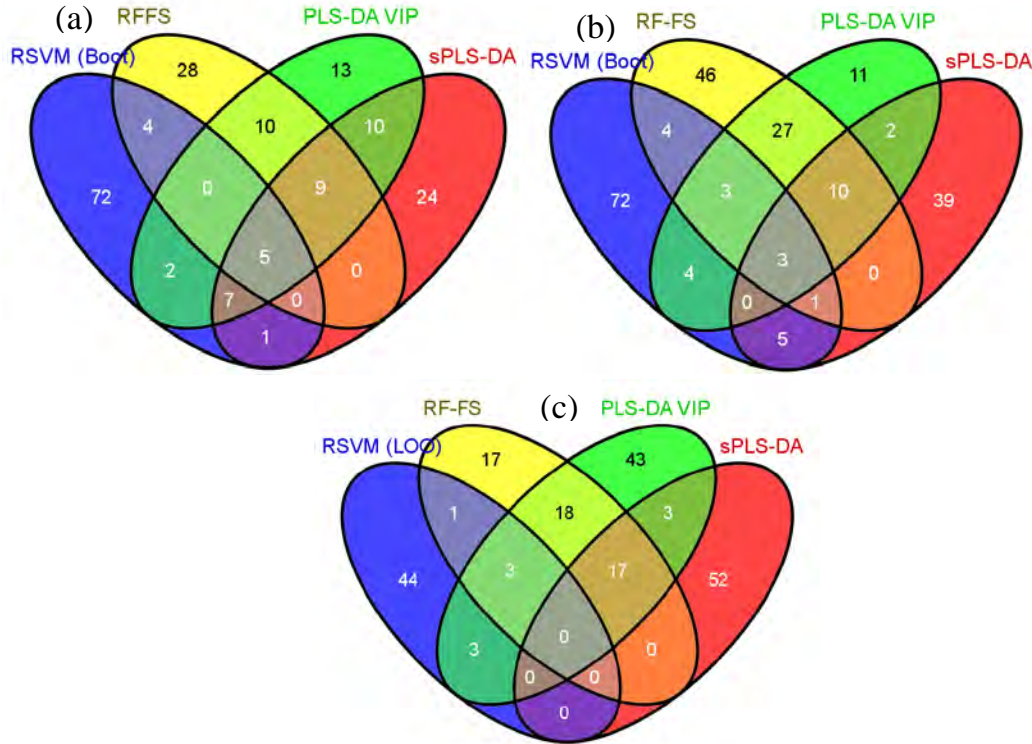

Figure 5.8. The number of features selected for middle-level analysis based on technique used for : (a) Human urines dataset, (b) Rat's urines dataset, and (c) Rat's plasma dataset

Another challenge of Metabolomic study that is mostly large dimension is their many combinations of features selected that obtain good performance. In term of this reason, biological interpretation is needed to clarify the result in order to give better recommendation.

5.6   Biological Interpretation

A study about our Rat's urines and Rat's plasma datasets has been conducted in the past, and we want to know how they could associate with our study. However, there is no previous study about our Human urines dataset. It doesn't matter since it is confirmed that both two classes are differentiated clearly. In addition, the similarity of features selected in middle-level analysis of Human urines datasets is also quite better than our two other datasets. In this section, we would like to overview our results study by the study of Grison, et al., (2013) to get more insight in biological interpretation.

Previous Rat's urines datasets study identified the most 40 top discriminating features based on PLS-DA analysis. These 40 features were ranked by their VIP score (above 1.8). This previous study is quite different with our PLS-DA VIP since we use recursive

technique to choose the features (not directly selecting the features based on VIP score) and we select different number of components used for each level. In term of analysis technique, previous study of our Rat's plasma datasets was also analyzed by PLS-DA. In this dataset study, they identified the most 38 features based on their VIP score. To get more input, we compare our middle-level analysis to these two past studies. Analyzing the similarity in Rat's urines datasets, RFFS as the common one could select 17 same features with the previous study. PLS-DA VIP as the second could select 13 same features, while RSVM and sPLS-DA got under 10 features (See Figure 5.9a).
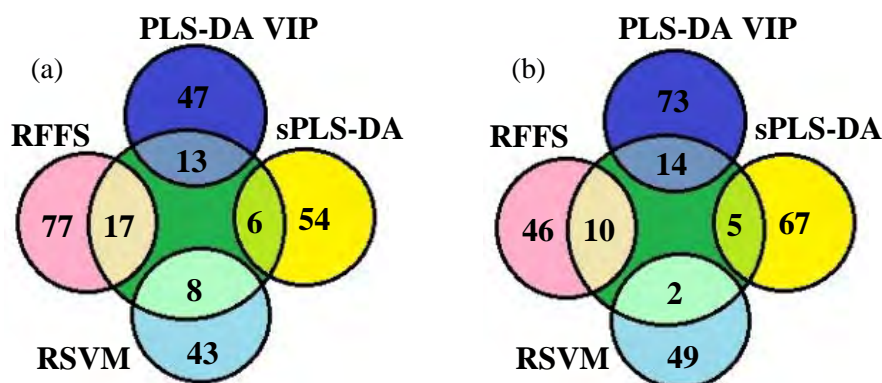
Figure 5.9. The number of features selected for middle-level analysis compare to previous study for : (a) Rat's urines dataset and (b) Rat's plasma dataset

In Rat's plasma datasets, their similarity result is not far from Rat's urines datasets comparison, where RFFS and PLS-DA is more common and RSVM and sPLS-DA select under 10 same features (See Figure 5.9b). These results, actually, associate with our 5.5 section interpretation, where PLS-DA and RFFS could achieve higher features selected similarity than RSVM and sPLS-DA. Listed of features selected are presented in Appendix 3.

Table 5.18. Features Selected Comparison

*Confirmed features		M137T30	M136T46	M132T32	M162T107	M146T272
RSVM	Low-level	-				
	Middle-level	M137T30	M162T107			
RFFS	Low-level	M137T30				
	Middle-level	M137T30	M132T32	M162T107		
PLS-DA VIP	Low-level	M137T30				
	Middle-level	M137T30	M136T46	M162T107		
sPLS-DA	Low-level	M137T30				
	Middle-level	M137T30				

(*): Based on previous study

Features description:
M137T30   : N1-methylnicotinamide
M136T46   : Nl-Methyl-2-pyridone-5carboxamide
M132T32   : Creatine
M162T107   : 4.6-Dihydroxy quinoline
M146T272   : 5-Hydroxyindoleacetic acid

Grison, et al., (2013) has analyzed their 40 discriminatory metabolites in urine samples. They stated that 36 were tentatively annotated with the MZedDB database browser, and 28 had a corresponding KEGG ID. Fourteen of these 28 were mapped in pathways according to the KEGG Mapper search: 8 in the tryptophan metabolism pathway, 7 in global metabolic pathways, and 2 in the nicotinate and nicotinamide metabolism pathway.

Grison, et al., (2013) has also found five confirmed metabolites based on biological study, which is presented in Table 5.18, with several metabolites that were also selected by our techniques either in low-level or middle-level analysis. Except low-level analysis of RSVM, N1-methylnicotinamide is always selected since it is confirmed as the most discriminatory metabolite, found at a concentration seven times higher in the urine of control versus contaminated rats. N1-methylnicotinamide has a function in the nicotinate and nicotinamide metabolism, regulate thrombotic as well as inflammatory processes in the cardiovascular system. It is important to select this metabolites because of their power. Nevertheless, RSVM couldn't be able to select this important feature although this technique could achieve 93 percent of accuracy rate.

After that, 4.6-Dihydroxy quinoline that has a function in tryptophan metabolism is also the common feature selected in middle-level analysis, i.e., RSVM, RFFS, and PLS-DA VIP. However, 5-Hydroxyindoleacetic acid (for features M146T272) is never selected in our study since their rank in previous study is also quite low, where in the bottom 10 (32th). RFFS and PLS-DA VIP have their better performance since they could select three metabolites of five confirmed metabolites that couldn't be achieved by RSVM and sPLS-DA.

The previous study is quite helpful to get several other insights in the biological information. However, there is still possibility for features selected in our four techniques that they may contain other biological meanings, such as the same molecules name but different code in the feature, or different molecules name that has also discriminatory power. Beside analyzing and comparing to the previous study, it is also needed to conduct biological study of features selected for future study.

CHAPTER 6
CONCLUSION AND PERSPECTIVE

After all interpretation, this chapter introduces the conclusion of this study and several recommendations for future study.

6.1   Conclusion

Selection of relevant variables for sample classification is a common task in most features expression studies, including this study. When there are much larger features than the number of sample(s), this problem may undermine the success of classification techniques that is strongly affected by data quality: redundant, noisy, and unreliable information as well as a confusing selection of relevant variables. This study facilitates four proposed techniques, which are two machine learning techniques (i.e., RSVM and RFFS) and two supervised classification techniques (i.e., PLS-DA VIP and sPLS-DA), to classify our three datasets, i.e., human urines, rat's urines, and rat's plasma datasets.

To identify small sets of features (fewer than 15 features) that could be used for diagnostic purpose in clinical practice, we conduct low-level analysis for both each technique used and our three datasets. RSVM-LOO always leads the accuracy performance compare to the other two cross-validation methods, i.e., bootstrap and N-fold. It reaches 93 percent for human urines datasets, 85 percent for rat's urines datasets, and 75 percent for rat's plasma datasets, much higher than other CV methods accuracy rates that they achieve. However, this RSVM results is not much better since RFFS could achieve 100 percent for human urines datasets and 95 percent for two others. In supervised classification technique, PLS-DA and sPLS-DA could reach good performance, but sPLS-DA could achieve quite better performance than PLS-DA VIP either for variability explanation or predictive ability.

In term of the number of features selected, RFFS consistently shows their performance in low-level analysis since it is able to find smallest set of features selected for our three cases (i.e., five features), while the other techniques select higher. RFFS is a classification technique that returns small sets of features selected because this technique will not return sets of features that are highly correlated, because they are redundant. This method will be most useful under scenario when considering the design of diagnostic tools, when having small sets features is often desirable.

When we want to identify much more relevant features, we did another purpose that is to identify a larger set of features; this involves obtaining a set of variables (around 50-100 features) that are related to the outcome of interest. RSVM could not achieve the accuracy performance well in this middle-level analysis since their performance is quite poor, particularly for rat's plasma classification. Nevertheless, RFFS is still able to

reach more than 90 percent in their accuracy performance. The performance of PLS-DA VIP and sPLS-DA in this middle-level are both similar. Comparing to low-level analysis, accuracy performance of low-level analysis is better for machine learning methods. In supervised classification techniques, middle-level performance is quite better since they are developed under PLS rules, where the higher number of variables selected could increase the coefficient of determination as well as predictive ability score.

As one of the performance measurements, computational time needed is aslo recorded to compare since it is important for efficiency purpose. As the fastest approach, RFFS is necessarily the one that performs the best, and it is certainly the most efficient on large data sets. The second is sPLS-DA, that could also performs a very good performance as well as less consumed computational time.

In biological interpretation, we did the comparison with previous study and we got several insights. RFFS and PLS-DA are more common to find the same features selected than RSVM and sPLS-DA. This is also confirmed in the statistical comparison when RFFS and PLS-DA could lead the similarity percentage of features selected. Besides that, N1-methylnicotinamide is always selected (except low-level analysis of RSVM) since it is confirmed as the most discriminatory metabolite, found at a concentration seven times higher in the urine of control versus contaminated rats. Furthermore, RFFS and PLS-DA VIP have their better performance since they could select three metabolites of five confirmed metabolites from previous study that couldn't be achieved by RSVM and sPLS-DA.

6.2   Perspective

Selection of relevant variables for sample classification is a common task in most Metabolomic study since their key difficulties are: redundant, noisy, and unreliable information as well as a confusing selection of relevant variables arising from the high dimensionally data. In this study, we did several works which is very helpful and useful for Metabolomic study. For future study, we would like to treat a lot of other datasets to make better generalization, particularly for multiclass problem. Since RSVM is less appropriate and it couldn't analyze for multiclass problem, there will be three comparison methods for future study, i.e., RFFS, PLS-DA VIP, and sPLS-DA. Besides that, to gain more biological meaning as well as identify the metabolites, it is also needed to conduct biological study of all features selected in this study.

REFERENCES

Ambroise, C., & McLachlan, G. (2002). Selection bias in gene extraction on the basis of microarray gene-expression data. *PNAS, 99*(10), 6562-6566.

Barker, M., & Rayens, W. (2003). Partial least squares for discrimination. *Journal of Chemometrics, 17*, 166-173.

Bhardwaj, N., Langlois, R., Zhao, G., & Lu, H. (2005). Kerenel-based machine learning protocol for predicting DNA-binding proteins. *Nucleic Acids Research, 33*(20), 6486-6493.

Breiman, L. (2001). Random Forest. *Machine Learning, 45*, 5-32.

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning, 20*, 273-297.

Dai, Y., Li, Z., Xue, L., Dou, C., Zhou, Y., Zhang, L., et al. (2010). Metabolomics study on the anti-depression effect of xiaoyaosan on rat model of chronic unpredictable mild stress. *Journal of Ethnopharmacology*, 482-489.

Dejean, S., Gonzalez, I., & Lê Cao, K.-A. (2014). *Omics Data Integration Project.* CRAN R.

Diaz-Uriarte, R. (2010). *Variable selection using random forests.* CRAN R.

Diaz-Uriarte, R., & Andres, S. (2006, January 06). Gene selection and classification of Microarry data using random forest. *BMC Bioinformatics, 7*(3).

Dudoit, S., & Fridlyand, J. (2003). Classification in microarray experiments. In T. Speed, *Statistical analysis of gene expression microarray data* (pp. 93-159). New York: Chapman and Hall.

Efron, B. (1979). Another look at the jacknife. *The Annals of Statistics, 7*(1), 1-26.

Efron, B., & Tibshirani, R. (1997). Improvements on cross-validation: the .632+ bootstrap method. *Journal of American Statistical Association, 92*, 548-560.

Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., et al. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science, 286*, 531-537.

Grison, S., Favé, G., Maillot, M., Manens, L., Delissen, O., Blanchardon, E., et al. (2013). Metabolomics identifies a biological response to chronic low-dose natural uranium contamination in urine samples. *Metabolomics, 9*, 1168-1180.

Gu, H., Chen, H., Pan, Z., Jackson, A., Talaty, N., Xi, B., et al. (2007). Monitoring diet effects via biofluids and their implication for metabolomics studies. *Analytical Chemistry, 79*(1), 89-97.

Gunn, S. (1998). *Support Vector Machines for classification and regression.* Southampton: Image Speech and Intellligent Systems Group.

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning, 46*, 389-422.

Huang, Chien-Ming., Lee, Yuh-Jye., Lin, Dennis K.J., Huang, Su-Yun. (2007). Model selection for support vector machines via uniform design. *Computational Statistics and Data* Analysis, 52, 335-346.

Kind, T., Tolstikov, V., Fiehn, O., & Weiss, R. (2007). A comprehensive urinary metabolomic approach for identifying kidney cancer. *Analytical Biochemistry*, 185-195.

Lê Cao, K.-A., Boitard, S., & Besse, P. (2011). Sparse PLS discriminant analysis: biologically relevant feature selection and graphical displyas for multiclass problems. *BMC Bioinformatics, 12*(253), 1-16.

Lê Cao, K.-A., Rossouw, D., Robert-Granié, C., & Besse, P. (2008). Sparse PLS: Variable selection when integrating Omics data. *Statistical Application and Molecular Biology, 7*(1), 37-59.

Liaw, A., & Wiener, M. (2002). Classification and regression by random forest. *R News, 2*(3), 18-22.

Mahadevan, S., Shah, S., Marrie, T., & Slupsky, C. (2008). Analysis of metabolomic data using support vector machines. *Analytical Chemistry, 80*(19), 7562-7570.

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., Leisch, F., Chang, C.-C., et al. (2014). *Misc Functions of the Department of Statistics (e1071), TU Wien.* CRAN R.

Nicholson, J., Lindon, J., & Holmes, E. (1999, July 05). "Metabonomics": understanding the metabolic respones of living systems to pathophysiological stimuli via multivariate statistical analysis of biological NMR spectroscopic data. *Xenobiotica, 29*(11), 1181-1189.

Pérez-Enciso, M., & Tenenhaus, M. (2003). Prediction of clinical outcome with microarray data: a partial least squares discriminant analysis (PLS-DA) approach. *Hum Genet, 112*, 581-592.

Rakotomamonjy, A. (2003). Variable selection using SVM-based criteria. *Journal of Machine Learning Research, 3*, 1357-1370.

R-Foundation. (2014). *What is R?* Retrieved June 23, 2014, from R Homepage: http://www.r-project.org/about.html

Sanchez, G., & Determan, C. (2013). *Tools of the Trade for Discriminant Analysis.*CRAN R.

Scalbert, A., Brennan, L., Fiehn, O., Hankemeier, T., Kristal, B., Ommen, B., et al. (2009, June 12). Mass-spectrometry-based metabolomics: limitations and recommendations for future progress with particular focus on nutrition research. *Metabolomics*, 435-458.

Suhre, K., Meisinger, C., Döring, A., Altmaier, E., Belcredi, P., Gieger, C., et al. (2010). Metabolic footprint of diabetes: a multiplatform metabolomics study in an epidemiological setting. *Plos One, 5*(11), 1-11.

Svetnik, V., Liaw, A., Tong, C., & Wang, T. (2004). Application of Breiman's random forest to modelinf stucture-activity relationships of pharmaceutical molecules. *Multiple Classier Systems, Fifth International Workshop MSC 2004* (pp. 334-343). Cagliary, Italy: Springer.

Tenenhaus, M. (1998). *La régression PLS, théorie et pratique.* Paris: Editions Technip.

Tenenhaus, M., Gauchi, J.-P., & Ménardo, C. (1995). Régression PLS et applications. *Rev. Statistique Appliquée*, 7-63.

Van den Berg, R., Hoefsloot, H., Westerhuis, J., Smilde, A., & Van der Werf, M. (2006). Centering, scaling, and transformations: improving the biological information content of metabolomics data. *BMC Genomics, 7*, 142-156.

Wu, B., Abbott, T., Fishman, D., McMurray, W., Mor, G., Stone, K., et al. (2003). Comparison of statistical methods for classification of varian cancer using mass spectrometry data. *Bioinformatics, 19*, 1636-1643.

Zhang, X., & Wong, W. H. (2001). Recursive sample classification and gene selection based on SVM: method and software description. *Technical Report*, 1-5.

Zhang, X., Lu, X., Shi, Q., Xu, X.-q., Leung, H.-c. E., Harris, L. N., et al. (2006). Recursive SVM feature selection and sample classification for mass-spectrometry and microarray data. *BMC Bioinformatics, 7*(197), 1-13.

APPENDIX

There are three appendixes, which are R Code contains R program used for analysis, middle level venny diagram for R-SVM analysis, and list of features selected for each features techniques and cross-validation selected.

Appendix 1. R Code

a. R Code of R-SVM

```
#Code of Recursive SVM using R
#Author : Achmad Choiruddin
#Department of Applied Mathematics in Social Science, University of Aix-Marseille
#Based on Dr.Xin Lu, Biostatistics Departemnt, Harvard School of Public Health

library(rJava)
library(xlsxjars)
library(xlsx)
library(e1071)

#Read in SVM formated data
dataset=read.xlsx("C:/Documents and
Settings/pirisi/Bureau/ProjetPludisiplinaire/Report/data.xlsx", sheetName = "SIMCA")
x2=as.matrix(dataset[,2:ncol(dataset)])
y2=as.factor(dataset[,1])
ret=list(x=x2, y=y2)

par.c=function(n)
{
d=vector()
k=vector()
l=vector()
m=vector()
            for (i in 1:n)
            {
            d[i]=2*10^i
            k[i]=2*10^(-i)
            l[i]=2*10^1/i
            m[i]=2*10^(-1/i)
            }
c=rbind(k,d,l,m)
w=sort(c)
}

c=par.c(4)
para2=tune.svm(x2,y2,cost=c,scale=F,type="C-classification",
kernel="linear",gamma=0,cross=5)

#Create a decreasing ladder for recursive feature elimination
CreatLadder=function(Ntotal,pratio,Nmin)
{
            d=vector()
            d[1]=Ntotal
            for (i in 1:100)
            {
```

```
                        pp=round(d[i]*pratio)
                        if (pp==d[i])
                        {pp=pp-1}
                        if (pp>=Nmin)
                        {d[i+1]=pp}
                        else
                        {break}
                        }
d
}

#Recursive SVM core code
RSVM=function(x,y,ladder,CVtype,CVnum=0)
{
                #check if y is binnary response
                Ytype=names(table(y))
                if (length(Ytype)!=2)
                {
                print ("ERROR!! R-SVM can only deal with two-class problem")
                return
                }

                #Class mean
                m1=apply(x[which(y==Ytype[1]),],2,mean)
                m2=apply(x[which(y==Ytype[2]),],2,mean)
                md=m1-m2

                yy=vector()
                yy[which(y==Ytype[1])]=1
                yy[which(y==Ytype[2])]=-1
                y=yy

                #Check ladder
                if (min(diff(ladder))>=0)
                {
                print("Error!!Ladder must be monotonously decreasing")
                return (0)
                }
                if (ladder[1]!=ncol(x))
                {ladder=c(ncol(x),ladder)}

                nSample=nrow(x)
                nGene=ncol(x)
                Sampind=seq(1,nSample)

                if(CVtype=="LOO")
                {CVnum=nSample}
                else
                {
                        if (CVnum==0)
                        {CVnum=nSample}
                }


                #Vector for test error and number of tests
                ev=vector(length=length(ladder))
                names(ev)=paste("Features:",ladder,sep="")
                ntest=0

                selfreq=matrix(0,nrow=nGene, ncol=length(ladder))
```

```
colnames(selfreq)=paste("Features:",ladder,sep="")


#For each CV
#SPlit data
if (CVtype=="LOO")
{
        for(i in 1:CVnum)
        {
        testind=i
        trainind=Sampind[-testind]

        ntest=ntest+length(testind)

        #in each level, train a SVM model and record test error
        xtrain=x[trainind,]
        ytrain=y[trainind]
        xtest=x[testind,]
        ytest=y[testind]

        Selind=seq(1,nGene)
                for (glevel in 1:length(ladder))
                {
                        selfreq[Selind,glevel]=selfreq[Selind,glevel]+1

                        #Train SVM Model and error
                        svmres=svm(xtrain[,Selind], ytrain, scale=F, type="C-
classification", kernel="linear",cost=2e-04)
                        svmpred=predict(svmres,matrix(xtest[Selind],nrow=1))
                        ev[glevel]=ev[glevel]+sum(svmpred !=ytest)

                        #weight vector

        s=t(svmres$coefs*ytrain[svmres$index])%*%svmres$SV*md[Selind]
                        rks=rank(s)

                        if (glevel<length(ladder))
                        {Selind=Selind[which(rks>(ladder[glevel]-
ladder[glevel+1]))]}
                }
        }
}

else
{
if (CVtype=="bootstrape")
{
        for(i in 1:250)
        {
        trainind=sample(Sampind,nSample,replace=T)
        testind=Sampind[which (!(Sampind %in% trainind))]

        ntest=ntest+length(testind)

        #in each level, train a SVM model and record test error
        xtrain=x[trainind,]
        ytrain=y[trainind]
        xtest=x[testind,]
        ytest=y[testind]
```

```r
                Selind=seq(1,nGene)
                for (glevel in 1:length(ladder))
                {
                        selfreq[Selind,glevel]=selfreq[Selind,glevel]+1

                        #Train SVM Model and error
                        svmres=svm(xtrain[,Selind], ytrain, scale=F, type="C-
classification", kernel="linear",cost=2e-04)
                        svmpred=predict(svmres,xtest[,Selind])
                        ev[glevel]=ev[glevel]+sum(svmpred !=ytest)

                        #weight vector

        s=t(svmres$coefs*ytrain[svmres$index])%*%svmres$SV*md[Selind]
                        rks=rank(s)

                        if (glevel<length(ladder))
                        {
                        Selind=Selind[which(rks>(ladder[glevel]-ladder[glevel+1]))]
                        }
                }
                }
        }

        else
        {
        a=1
                for(i in 1:CVtype)
                {
                        #NFold
                        testind=Sampind[a:(a+(nSample/CVtype)-1)]
                        trainind=Sampind[which (!(Sampind %in% testind))]
                        a=a+(nSample/CVtype)

                        ntest=ntest+length(testind)

                        #in each level, train a SVM model and record test error
                        xtrain=x[trainind,]
                        ytrain=y[trainind]
                        xtest=x[testind,]
                        ytest=y[testind]

                        Selind=seq(1,nGene)
                        for (glevel in 1:length(ladder))
                        {
                                selfreq[Selind,glevel]=selfreq[Selind,glevel]+1

                                #Train SVM Model and error
                                svmres=svm(xtrain[,Selind], ytrain, scale=F, type="C-
classification", kernel="linear",cost=2e-04)
                                svmpred=predict(svmres,xtest[,Selind])
                                ev[glevel]=ev[glevel]+sum(svmpred !=ytest)

                                #weight vector

        s=t(svmres$coefs*ytrain[svmres$index])%*%svmres$SV*md[Selind]
                                rks=rank(s)

                                if (glevel<length(ladder))
```

```
                                                {Selind=Selind[which(rks>(ladder[glevel]-
ladder[glevel+1]))]}
                                    }
                            }
                    }
                    }

ret=list(ladder=ladder, error=ev/ntest, selfreq=selfreq)

}




SummarySVM=function(RSVMres)
{
            ERind=max(which(RSVMres$error==min(RSVMres$error)))
            Minlevel=RSVMres$ladder[ERind]
            Freqvec=RSVMres$selfreq[,ERind]
            selind=which(rank(Freqvec)>=(RSVMres$ladder[1]-Minlevel))

            #print("Minimum CV error of ",min(RSVMres$error), "at ", Minlevel, "genes")

            ret=list(Miner=min(RSVMres$error), Minlevel=Minlevel, selind=selind)
}


ladder=CreatLadder(1376,0.85,5)

LOORSVM=RSVM(x2,y2,ladder,"LOO",CVnum=0)
LOO=1-LOORSVM$error

bootRSVM=RSVM(x2,y2,ladder,"bootstrape",CVnum=0)
bootstrap=1-bootRSVM$error

Fold4RSVM=RSVM(x2,y2,ladder,4,CVnum=0)
fold4=1-Fold4RSVM$error

Fold5RSVM=RSVM(x2,y2,ladder,5,CVnum=0)
fold5=1-Fold5RSVM$error

accuracy=cbind(LOO,fold4,fold5,bootstrap)
matplot(accuracy,type='l',axes=FALSE,xlab='number of genes selected',ylab='accuracy
rate',ylim=c(0,1),col=c("blue","black","green","red"),lwd=2,lty=1)
axis(1, c(1:length(ladder)), labels = ladder)
axis(2)
legend("bottomright",lty=1,legend=c('RSVM-LOO','RSVM-4 fold','RSVM-5 fold','RSVM-
boobstrap'),horiz=TRUE,cex=0.6,col=c("blue","black","green","red"),lwd=3)

varsel1=LOORSVM$selfreq[,ncol(LOORSVM$selfreq)]
varloo=seq(1,ncol(x2))[which(varsel1!=0)]
xloo=x2[,varloo]
svmloo=svm(xloo, y2, scale=F, type="C-classification", kernel="linear",cross=20)

loo2=LOORSVM$selfreq[,21]
vloo=seq(1,ncol(x2))[which(loo2>=16)]
xloo2=x2[,vloo]

varsel2=Fold4RSVM$selfreq[,35]
var4fold=seq(1,ncol(x2))[which(varsel2>=2)]
x4fold=x2[,var4fold]
```

```
svm4fold=svm(x4fold, y2, scale=F, type="C-classification", kernel="linear",cross=4)

fold4=Fold4RSVM$selfreq[,21]
v4fold=seq(1,ncol(x2))[which(fold4>=3)]
x4fold2=x2[,v4fold]

varsel3=Fold5RSVM$selfreq[,ncol(Fold5RSVM$selfreq)]
var5fold=seq(1,ncol(x2))[which(varsel3!=0)]
x5fold=x2[,var5fold]
svm5fold=svm(x5fold, y2, scale=F, type="C-classification", kernel="linear",cross=5)

fold5=Fold5RSVM$selfreq[,21]
v5fold=seq(1,ncol(x2))[which(fold5>=4)]
x5fold2=x2[,v5fold]

varsel4=bootRSVM$selfreq[,25]
varboot=seq(1,ncol(x2))[which(varsel4>=114)]
xboot=x2[,varboot]


boot=bootRSVM$selfreq[,21]
vboot=seq(1,ncol(x2))[which(boot>=4)]
xboot2=x2[,vboot]


nSample=nrow(x2)
Sampind=seq(1,nSample)
ntest=0
ev=0
for(i in 1:250)
{
            trainind=sample(Sampind,nSample,replace=T)
            testind=Sampind[which (!(Sampind %in% trainind))]

            ntest=ntest+length(testind)

            xtrain=xboot[trainind,]
            ytrain=y2[trainind]
            xtest=xboot[testind,]
            ytest=y2[testind]

            #Train SVM Model and error
            svmres=svm(xtrain, ytrain, scale=F, type="C-classification", kernel="linear")
            svmpred=predict(svmres,xtest)
            ev=ev+sum(svmpred !=ytest)

}
errorbot=ev/ntest
acbot=1-errorbot
```

b. R Code of RFFS

```
library(rJava)
library(xlsxjars)
library(xlsx)
library(MASS)
library(stats)
library(randomForest)
library(varSelRF)
```

```
#Read data
mydata=read.table('C:/Documents and
Settings/pirisi/Bureau/ProjetPludisiplinaire/Report/data.txt',header=T)
x1=as.matrix(mydata[,1:ncol(mydata)-1])
y1=as.factor(mydata[,ncol(mydata)])
ret1=list(x=x1, y=y1)

dataset=read.xlsx("C:/Documents and
Settings/pirisi/Bureau/ProjetPludisiplinaire/Report/data.xlsx", sheetName = "SIMCA")
x2=as.matrix(dataset[,2:ncol(dataset)])
y2=as.factor(dataset[,1])
ret2=list(x=x2, y=y2)

dataset3=read.xlsx("C:/Documents and
Settings/pirisi/Bureau/ProjetPludisiplinaire/Report/data3.xlsx", sheetName = "SIMCA")
x3=as.matrix(dataset3[,2:ncol(dataset3)])
y3=as.factor(dataset3[,1])
ret3=list(x=x3, y=y3)

rf1=randomForest(x1,y1)
rf2=randomForest(x2,y2)
rf3=randomForest(x3,y3)

#Variable selection from random forests using OOB error
recRF1=varSelRF(x1,y1)
features1=recRF1$selec.history$Number.Variables
OOB1=recRF1$selec.history$OOB
sdOOB1=recRF1$selec.history$sd.OOB
summary1=cbind(features1,OOB1)

recRF2=varSelRF(x2,y2)
features2=recRF2$selec.history$Number.Variables
OOB2=recRF2$selec.history$OOB
sdOOB2=recRF2$selec.history$sd.OOB
summary2=cbind(features2,OOB2)

recRF3=varSelRF(x3,y3)
features3=recRF3$selec.history$Number.Variables
OOB3=recRF3$selec.history$OOB
sdOOB3=recRF3$selec.history$sd.OOB
summary3=cbind(features3,OOB3)

var1mid=recRF1$selec.history$Vars.in.Forest[15]
var1low=recRF1$selec.history$Vars.in.Forest[26]

var2mid=recRF2$selec.history$Vars.in.Forest[13]
var2low=recRF2$selec.history$Vars.in.Forest[26]

var3mid=recRF3$selec.history$Vars.in.Forest[13]
var3low=recRF3$selec.history$Vars.in.Forest[24]
```

c.   R Code of PLS-DA

```
library(rJava)
library(xlsxjars)
library(xlsx)
library(MASS)
library(mclust)
library(gtools)
```

```
library(gplots)
library(mvtnorm)
library(ellipse)
library(MetabolAnalyze)
library(lattice)
library(mixOmics)
library(DiscriMiner)

#Read the all three datasets
mydata=read.table('C:/Documents and Settings/pirisi/Bureau/Projet
Pludisiplinaire/Report/data.txt',header=T)
x1=as.matrix(mydata[,1:ncol(mydata)-1])
y1=as.factor(mydata[,ncol(mydata)])
yn1=as.matrix(mydata[,ncol(mydata)])
xlog1=log(x1+max(x1))
xn1=scaling(xlog1, type = "pareto")
ret=list(x=x1, y=y1)

dataset2=read.xlsx("C:/Documents and Settings/pirisi/Bureau/Projet
Pludisiplinaire/Report/data.xlsx", sheetName = "SIMCA")
x2=as.matrix(dataset2[,2:ncol(dataset2)])
y2=as.factor(dataset2[,1])
yn2=as.matrix(dataset2[,1])
xlog2=log(x2+max(x2))
xn2=scaling(xlog2, type = "pareto")
ret=list(x=xn2, y=y2)

dataset3=read.xlsx("C:/Documents and Settings/pirisi/Bureau/Projet
Pludisiplinaire/Report/data3.xlsx", sheetName = "SIMCA")
x3=as.matrix(dataset3[,2:ncol(dataset3)])
y3=as.factor(dataset3[,1])
yn3=as.matrix(dataset3[,1])
xlog3=log(x3+max(x3))
xn3=scaling(xlog3, type = "pareto")
ret3=list(x=xn3, y=y3)

#count VIP score (Variable Importance in the Projection)
LOOm1=plsDA(xn1, y1, autosel = FALSE, comps=2, cv = "LOO")
vip1 = LOOm1$VIP[,3]
LOOm2=plsDA(xn2, y2, autosel = FALSE, comps=2, cv = "LOO")
vip2 = LOOm2$VIP[,3]
LOOm3=plsDA(xn3, y3, autosel = FALSE, comps=2, cv = "LOO")
vip3 = LOOm3$VIP[,3]

#Creating the decreasing ladder that contains the number of features selected
CreatLadder=function(Ntotal,pratio,Nmin)
{
        d=vector()
        d[1]=Ntotal
        for (i in 1:100)
        {
        pp=round(d[i]*pratio)
        if (pp==d[i])
        {pp=pp-1}
        if (pp>=Nmin)
        {d[i+1]=pp}
        else
        {break}
        }
d
```

```
}

ladder1=CreatLadder(1271,0.8,5)
ladder3=CreatLadder(810,0.8,5)
ladder2=CreatLadder(1376,0.8,5)

#Main Code of PLSDA vip using package DiscriMiner
components=function(x,y,ladder,cvtype,k)
{
nGene=ncol(x)
ac.plsdavip=matrix(0, nrow = length(ladder), ncol = 1)
comp=matrix(0,nrow=length(ladder),ncol=2)
rownames(comp)=paste("Features:",ladder,sep="")
Q2=matrix(0, nrow = length(ladder), ncol = k)
rownames(Q2)=paste("Features:",ladder,sep="")
R2=matrix(0, nrow = length(ladder), ncol = 1)
rownames(R2)=paste("Features:",ladder,sep="")
Selind=seq(1,nGene)

for (i in 1:length(ladder))
{
        if (cvtype=="loo")
        {
                #Train PLS-DA Model and accuracy rate
                plsdavip=plsDA(x[,Selind], y, autosel = FALSE, comps=k, cv = "LOO")

                Q2[i,]=plsdavip$Q2[,ncol(plsdavip$Q2)]

                #VIP
                vip=plsdavip$VIP[,k+1]
                rvip=rank(vip)

                if (i<length(ladder))
                {
                Selind=Selind[which(rvip>(ladder[i]-ladder[i+1]))]
                }
        }
        else
        {
                #Train PLS-DA Model and accuracy rate
                plsdavip=plsDA(x[,Selind], y, autosel = FALSE, comps=k, cv = "KLO",
k=cvtype)

                Q2[i,]=plsdavip$Q2[,ncol(plsdavip$Q2)]

                #VIP
                vip=plsdavip$VIP[,k+1]
                rvip=rank(vip)

                if (i<length(ladder))
                {
                Selind=Selind[which(rvip>(ladder[i]-ladder[i+1]))]
                }
        }
}

                for (m in length(ladder):1)
                {
                for (a in k:1)
                {
```

```
                              if (Q2[m,a]<=0.049)
                              {comp[m,1]=a}
                              }
                                      if (comp[m,1]>2)
                                      {comp[m,2]=comp[m,1]-1}
                                      else
                                      {comp[m,2]=2}
                              }
ret=list(ladder=ladder, Q2=Q2, comp=comp[,2])
}

comp.loo1=components(xn1,y1,ladder1,"loo",5)
comp.loo2=components(xn2,y2,ladder2,"loo",5)
comp.loo3=components(xn3,y3,ladder3,"loo",5)

comp1=cbind(comp.loo1$Q2,comp.loo1$comp)
rownames(comp1)=paste("Features:",ladder1,sep="")
comp2=cbind(comp.loo2$Q2,comp.loo2$comp)
rownames(comp2)=paste("Features:",ladder2,sep="")
comp3=cbind(comp.loo3$Q2,comp.loo3$comp)
rownames(comp3)=paste("Features:",ladder3,sep="")

colnames(comp1)=paste(c("R2","nb of components"))
colnames(comp2)=paste(c("R2","nb of components"))
colnames(comp3)=paste(c("R2","nb of components"))

PLSDAVIP=function(x,y,ladder,cvtype,k,comp)
{
nGene=ncol(x)
ac.plsdavip=matrix(0, nrow = length(ladder), ncol = 1)
Q2=matrix(1, nrow = length(ladder), ncol = 1)
R2=matrix(0, nrow = length(ladder), ncol = 1)
Q2.R2=matrix(0, nrow = length(ladder), ncol = 1)
Selind=seq(1,nGene)

for (i in 1:length(ladder))
{
        if (cvtype=="loo")
        {
                #Train PLS-DA Model and accuracy rate
                plsdavip=plsDA(x[,Selind], y, autosel = FALSE, comps=comp[i], cv =
"LOO")

                ac.plsdavip[i,]=(1-plsdavip$error_rate)*100

                for (a in 1:comp[i])
                {
                if (plsdavip$Q2[a,ncol(plsdavip$Q2)]<0)
                {plsdavip$Q2[a,ncol(plsdavip$Q2)]=1}
                Q2[i,]=Q2[i,]*plsdavip$Q2[a,ncol(plsdavip$Q2)]
                }

                R2[i,]=plsdavip$R2[comp[i],ncol(plsdavip$R2)]
                Q2.R2[i,]=abs((1-Q2[i,])-R2[i,])

                #VIP
                vip=plsdavip$VIP[,comp[i]+1]
                rvip=rank(vip)

                if (i<length(ladder))
                {
```

```
                        Selind=Selind[which(rvip>(ladder[i]-ladder[i+1]))]
                        }
                }
        else
        {
                #Train PLS-DA Model and accuracy rate
                plsdavip=plsDA(x[,Selind], y, autosel = FALSE, comps=comp[i], cv =
"KLO", k=cvtype)

                ac.plsdavip[i,]=(1-plsdavip$error_rate)*100

                for (a in 1:comp[i])
                {
                if (plsdavip$Q2[a,ncol(plsdavip$Q2)]<0)
                {plsdavip$Q2[a,ncol(plsdavip$Q2)]=1}
                Q2[i,]=Q2[i,]*(1-plsdavip$Q2[a,ncol(plsdavip$Q2)])
                }

                R2[i,]=plsdavip$R2[comp[i],ncol(plsdavip$R2)]
                Q2.R2[i,]=abs((1-Q2[i,])-R2[i,])

                #VIP
                vip=plsdavip$VIP[,comp[i]+1]
                rvip=rank(vip)

                if (i<length(ladder))
                {
                Selind=Selind[which(rvip>(ladder[i]-ladder[i+1]))]
                }
        }
}

ret=list(ladder=ladder, accuracy=ac.plsdavip, Q2=1-Q2, R2=R2, distance=Q2.R2)
}

#PLSDA vip data running
#Since Q2 is only calculated by LOO, it is not proposed to run KLO cv method
plsdavip.loo1=PLSDAVIP(xn1,y1,ladder1,"loo",5,comp.loo1$comp)
plsdavip.loo2=PLSDAVIP(xn2,y2,ladder2,"loo",5,comp.loo2$comp)
plsdavip.loo3=PLSDAVIP(xn3,y3,ladder3,"loo",5,comp.loo3$comp)

case1=cbind(plsdavip.loo1$R2,plsdavip.loo1$Q2,(plsdavip.loo1$accuracy/100))
rownames(case1)=paste("Features:",ladder1,sep="")
colnames(case1)=paste(c("R2","Q2","accuracy"))
case2=cbind(plsdavip.loo2$R2,plsdavip.loo2$Q2,(plsdavip.loo2$accuracy/100))
rownames(case2)=paste("Features:",ladder2,sep="")
colnames(case2)=paste(c("R2","Q2","accuracy"))
case3=cbind(plsdavip.loo3$R2,plsdavip.loo3$Q2,(plsdavip.loo3$accuracy/100))
rownames(case3)=paste("Features:",ladder3,sep="")
colnames(case3)=paste(c("R2","Q2","accuracy"))

Case1=cbind(plsdavip.loo1$R2,plsdavip.loo1$Q2,plsdavip.loo1$distance,(plsdavip.loo1
$accuracy/100))
rownames(Case1)=paste("Features:",ladder1,sep="")
colnames(Case1)=paste(c("R2","Q2","Q2-R2","accuracy"))
Case2=cbind(plsdavip.loo2$R2,plsdavip.loo2$Q2,plsdavip.loo2$distance,(plsdavip.loo2
$accuracy/100))
rownames(Case2)=paste("Features:",ladder2,sep="")
colnames(Case2)=paste(c("R2","Q2","Q2-R2","accuracy"))
```

```
Case3=cbind(plsdavip.loo3$R2,plsdavip.loo3$Q2,plsdavip.loo3$distance,(plsdavip.loo3
$accuracy/100))
rownames(Case3)=paste("Features:",ladder3,sep="")
colnames(Case3)=paste(c("R2","Q2","Q2-R2","accuracy"))

par(mfrow=c(3,1))
matplot(case1,type='l',axes=FALSE,xlab='number of features
selected',col=c("blue","darkgreen","red"),lwd=2,lty=1)
axis(1, c(1:length(ladder1)), labels = ladder1)
axis(2)
legend("bottomright",lty=1,legend=c('R2 case 1','Q2 case 1',"accuracy case
1"),horiz=TRUE,cex=0.8,col=c("blue","darkgreen","red"),lwd=3)

matplot(case2,type='l',axes=FALSE,xlab='number of features
selected',col=c("blue","darkgreen","red"),lwd=2,lty=1)
axis(1, c(1:length(ladder2)), labels = ladder2)
axis(2)
legend("bottomright",lty=1,legend=c('R2 case 2','Q2 case 2',"accuracy case
2"),horiz=TRUE,cex=0.8,col=c("blue","darkgreen","red"),lwd=3)

matplot(case3,type='l',axes=FALSE,xlab='number of features
selected',col=c("blue","darkgreen","red"),lwd=2,lty=1)
axis(1, c(1:length(ladder3)), labels = ladder3)
axis(2)
legend("bottomright",lty=1,legend=c('R2 case 3',"Q2 case 3","accuracy case
3"),horiz=TRUE,cex=0.8,col=c("blue","darkgreen","red"),lwd=3)

rvip1=rank(vip1)
rvip2=rank(vip2)
rvip3=rank(vip3)

var1loo=seq(1,ncol(x1))[which(rvip1>(ncol(x1)-5))]
varsel1loo=xn1[,var1loo]
var2loo=seq(1,ncol(x2))[which(rvip2>(ncol(x2)-8))]
varsel2loo=xn2[,var2loo]
var3loo=seq(1,ncol(x3))[which(rvip3>(ncol(x3)-6))]
varsel3loo=xn3[,var3loo]

var1big=seq(1,ncol(x1))[which(rvip1>(ncol(x1)-70))]
varsel1big=xn1[,var1big]
var2big=seq(1,ncol(x2))[which(rvip2>(ncol(x2)-60))]
varsel2big=xn2[,var2big]
var3big=seq(1,ncol(x3))[which(rvip3>(ncol(x3)-87))]
varsel3big=xn3[,var3big]
```

d.   R Code of s-PLSDA

```
library(rJava)
library(xlsxjars)
library(xlsx)
library(MASS)
library(mclust)
library(gtools)
library(gplots)
library(mvtnorm)
library(ellipse)
library(MetabolAnalyze)
library(lattice)
library(mixOmics)
```

```
library(DiscriMiner)

#Read the all three datasets
mydata=read.table('C:/Documents and
Settings/pirisi/Bureau/ProjetPludisiplinaire/Report/data.txt',header=T)
x1=as.matrix(mydata[,1:ncol(mydata)-1])
ynum1=mydata[,ncol(mydata)]
y1=as.factor(mydata[,ncol(mydata)])
xlog1=log(x1+max(x1))
xn1=scaling(xlog1, type = "pareto")
ret=list(x=x1, y=y1)

dataset2=read.xlsx("C:/Documents and Settings/pirisi/Bureau/Projet
Pludisiplinaire/Report/data.xlsx", sheetName = "SIMCA")
x2=as.matrix(dataset2[,2:ncol(dataset2)])
ynum2=dataset2[,1]
y2=as.factor(dataset2[,1])
xlog2=log(x2+max(x2))
xn2=scaling(xlog2, type = "pareto")
ret=list(x=xn2, y=y2)

dataset3=read.xlsx("C:/Documents and Settings/pirisi/Bureau/Projet
Pludisiplinaire/Report/data3.xlsx", sheetName = "SIMCA")
x3=as.matrix(dataset3[,2:ncol(dataset3)])
ynum3=dataset3[,1]
y3=as.factor(dataset3[,1])
xlog3=log(x3+max(x3))
xn3=scaling(xlog3, type = "pareto")
ret3=list(x=xn3, y=y3)

#Creating the decreasing ladder that contains the number of features selected
CreatLadder=function(Ntotal,pratio,Nmin)
{
                d=vector()
                d[1]=Ntotal
                for (i in 1:100)
                {
                pp=round(d[i]*pratio)
                if (pp==d[i])
                {pp=pp-1}
                if (pp>=Nmin)
                {d[i+1]=pp}
                else
                {break}
                }
d
}

ladder1=CreatLadder(1271,0.8,5)
ladder3=CreatLadder(810,0.8,5)
ladder2=CreatLadder(1376,0.8,5)

#-----------------------------------------------------------------------------------
spls_comp=function(x,ynum,ladder,k)
{
Q2=matrix(0, nrow = length(ladder), ncol = k)
comp=matrix(0,nrow=length(ladder),ncol=2)

for (i in 1:length(ladder))
{
```

```
spls=spls(x, ynum, ncomp = k, mode = c("regression"),keepX = rep(ladder[i],k))
resume=valid(spls, validation = c("loo"))
Q2[i,]=resume$Q2
}

                        for (m in length(ladder):1)
                        {
                        for (a in k:1)
                        {

                        if (Q2[m,a]<=0.049)
                        {comp[m,1]=a}

                        }
                                if (comp[m,1]>2)
                                {comp[m,2]=comp[m,1]-1}
                                else
                                {
                                if (comp[m,1]==0)
                                {comp[m,2]=k}
                                else
                                {comp[m,2]=2}
                                }
                        }

ret=list(ladder=ladder, Q2=Q2, comp=comp[,2])

}

comp1=spls_comp(xn1,ynum1,ladder1,5)
comp2=spls_comp(xn2,ynum2,ladder2,5)
comp3=spls_comp(xn3,ynum3,ladder3,5)

#-----------------------------------------------------------------------------------------------
splsda_new=function(x,y,ynum,ladder,comp)
{
#sPLS-DA using package mixOmics
ac.splsda=matrix(NA, nrow = length(ladder), ncol = 1)
d=round(ladder/comp)*comp
var=matrix(0, nrow =ncol(x), ncol=length(ladder))
colnames(var)=paste("Features:",ladder,sep="")
Q2=matrix(1, nrow = length(ladder), ncol = 1)
distance=matrix(1, nrow = length(ladder), ncol = 1)
R2=matrix(NA, nrow = length(ladder), ncol = 1)
col.class = as.numeric(y)
col.class[col.class == 1] <- 'red'
col.class[col.class == 2] <- 'blue'


for (i in 1:length(ladder))
{
sPLSDA=splsda(x, y, ncomp = comp[i], keepX = rep(round(ladder[i]/comp[i]),comp[i]))
splsn=spls(x, ynum, ncomp = comp[i], mode = c("regression"),keepX =
rep(round(ladder[i]/comp[i]),comp[i]))
ac=valid(sPLSDA,method="max.dist",validation="LOO")
ac.splsda[i, ]=(1-ac[comp[i],])*100
k=summary(splsn,what=c("all"),keep.var=TRUE)
var[1:length(k$keep.var$X),i]=k$keep.var$X
resume=valid(splsn, validation = c("loo"))
R2[i,]=resume$R2[,comp[i]]
```

```
                                for (a in 1:comp[i])
                                {
                                if (resume$Q2[,a]<0)
                                {resume$Q2[,a]=0}
                                if (resume$Q2[,a]==1)
                                {resume$Q2[,a]=0}
                                Q2[i,]=Q2[i,]*(1-resume$Q2[,a])
                                }
distance[i,]=abs(1-Q2[i,]-R2[i,])
}
ret=list(ladder=ladder, accuracy=ac.splsda, Q2=1-Q2, R2=R2, variabel=var, distance=distance)
}

model1=splsda_new(xn1,y1,ynum1,ladder1,comp1$comp)
model2=splsda_new(xn2,y2,ynum2,ladder2,comp2$comp)
model3=splsda_new(xn3,y3,ynum3,ladder3,comp3$comp)

spls_1=cbind(comp1$comp,model1$R2,model1$Q2,model1$distance,model1$accuracy)
rownames(spls_1)=paste("Features:",ladder1,sep="")
colnames(spls_1)=paste(c("Nb of comp","R2","Q2","Distance","accuracy"))

spls_2=cbind(comp2$comp,model2$R2,model2$Q2,model2$distance,model2$accuracy)
rownames(spls_2)=paste("Features:",ladder2,sep="")
colnames(spls_2)=paste(c("Nb of comp","R2","Q2","Distance","accuracy"))

spls_3=cbind(comp3$comp,model3$R2,model3$Q2,model3$distance,model3$accuracy)
rownames(spls_3)=paste("Features:",ladder3,sep="")
colnames(spls_3)=paste(c("Nb of comp","R2","Q2","Distance","accuracy"))

case1=cbind(model1$R2,model1$Q2,model1$accuracy/100)
case2=cbind(model2$R2,model2$Q2,model2$accuracy/100)
case3=cbind(model3$R2,model3$Q2,model3$accuracy/100)

par(mfrow=c(3,1))
matplot(case1,type='l',axes=FALSE,xlab='number of features
selected',ylim=c(0.5,1),col=c("blue","darkgreen","red"),lwd=2,lty=1)
axis(1, c(1:length(ladder1)), labels = ladder1)
axis(2)
legend("bottomright",lty=1,legend=c('R2 case 1','Q2 case 1',"accuracy case
1"),horiz=TRUE,cex=0.9,col=c("blue","darkgreen","red"),lwd=3)

matplot(case2,type='l',axes=FALSE,xlab='number of features
selected',ylim=c(0.5,1),col=c("blue","darkgreen","red"),lwd=2,lty=1)
axis(1, c(1:length(ladder2)), labels = ladder2)
axis(2)
legend("bottomright",lty=1,legend=c('R2 case 2','Q2 case 2',"accuracy case
2"),horiz=TRUE,cex=0.9,col=c("blue","darkgreen","red"),lwd=3)

matplot(case3,type='l',axes=FALSE,xlab='number of features
selected',ylim=c(0.5,1),col=c("blue","darkgreen","red"),lwd=2,lty=1)
axis(1, c(1:length(ladder3)), labels = ladder3)
axis(2)
legend("bottomright",lty=1,legend=c('R2 case 3',"Q2 case 3","accuracy case
3"),horiz=TRUE,cex=0.9,col=c("blue","darkgreen","red"),lwd=3)

#Select features
model1$var
#-------------------------------------------------------------------------------------------
```

Appendix 2. Middle Level Venn Diagram

a. Human Urines Dataset (R-SVM)
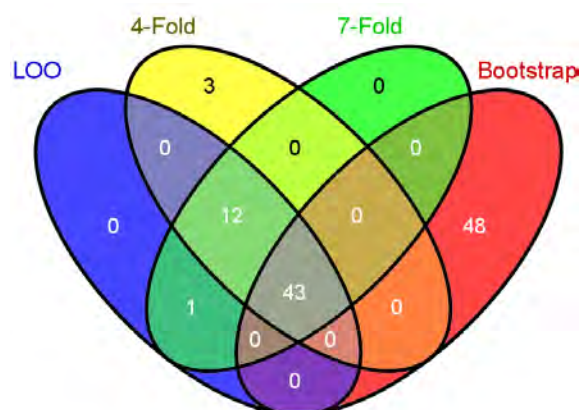
Common elements in "LOO", "4-Fold", "7-Fold" and "Bootstrap": (43 features)
M365T539 M302T478 M176T379 M279T379 M355T587 M303T473
M304T317 M310T428 M167T65 M207T97 M344T370 M342T360 M318T138
M330T414 M197T101 M265T283 M305T28 M183T56 M121T102 M326T464
M467T632 M145T104 M326T327 M384T81 M535T616 M134T423 M505T537
M274T258 M166T43 M202T44 M144T59 M169T41 M153T53 M136T52
M215T46 M175T47 M162T33 M170T490 M231T481 M314T501 M275T30
M259T29 M123T30

Common elements in "LOO" and "7-Fold": (1 feature)
M185T259

Common elements in "LOO", "4-Fold" and "7-Fold": (12 features)
M225T257 M267T335 M288T410 M271T536 M413T800 M483T450 M182T85
M267T490 M243T29 M130T334 M233T514 M145T124

b. Rat's Urines Dataset (R-SVM)
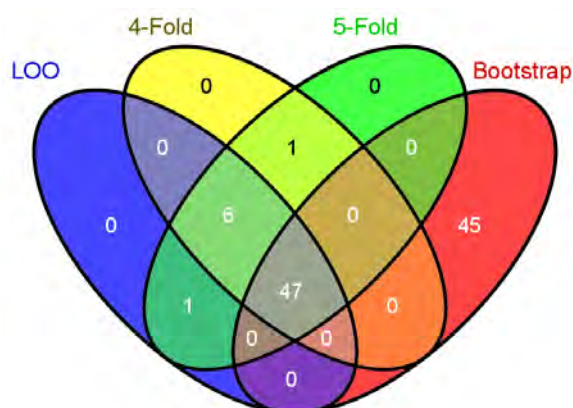
Common elements in "LOO", "4-Fold", "5-Fold" and "Bootstrap": (47 features)
M180T194M194T256M338T217M340T269M231T44M297T45M271T321M130T321M233T3
21M356T327M431T329M211T428M171T428M431T428M377T271M447T410M170T413M4
53T302M206T158M143T32M245T35M220T68M461T347M281T130M243T71M197T137M2

57T384M180T90M328T280M131T407M243T446M436T446M299T496M437T246M183T80
M397T80M305T253M227T66M162T300M162T107M227T188M180T216M188T158M137T3
0M144T137M217T138M481T253

Common elements in "4-Fold" and "5-Fold":
M384T64

Common elements in "LOO" and "5-Fold":
M594T431

Common elements in "LOO", "4-Fold" and "5-Fold": (6 features)
M255T321M496T661M149T296M475T455M210T89M354T333

c. Rat's Plasma Dataset (R-SVM)

Common elements in "LOO", "4-Fold", "5-Fold" and "Bootstrap": (51 features)

M289T27	M431T29	M227T29	M175T30_2	M159T30	M91T30	M140T34
M203T34	M118T35	M383T35	M875T275_1	M466T334	M240T356	M468T425
M542T433	M494T437	M495T441	M482T453	M568T457	M569T460	M496T473
M570T476	M454T481	M519T484	M523T485_1	M546T488	M522T490	M523T495
M544T500	M482T502	M524T539	M482T546	M524T551	M282T552	M538T587
M552T623	M606T684	M760T685	M785T685	M257T685	M759T685	M781T685
M406T687	M310T712	M640T754	M596T760	M554T769	M530T773	M339T785
M338T785	M360T785					

Elements only in "Bootstrap": (5 features)
M1469T301   M552T551_2   M510T573M256T635M807T685

d) Summary of RSVM Middle-level Analysis

Human Urines		Rat's Urines		Rat's Plasma	
CV	F|A	CV	F|A	CV	F|A
LOO	56|71	LOO	54|75	LOO	51|55
4-Fold	56|61	4-Fold	54|70	4-Fold	51|33
7-Fold	56|71	5-Fold	54|60	5-Fold	51|35
Bootstrap	88|80	Bootstrap	88|75	Bootstrap	60|52

Notes : CV: Cross-Validation, F: Number of Features, A=Accuracy rates (%)

Appendix 3. List of Features Selected

RSVM

CV	Var	Human Urines	Rat's Urines	Rat's Plasma
LOO	Case 1 : [56]‖[6] Case 2 : [54]‖[5] Case 3 : [51]‖[5]	M365T539 M302T478 M225T257 M176T379 M279T379 M355T587 M303T473 M304T317 M310T428 M167T65 M207T97 M344T370 M342T360 M318T138 M330T414 M197T101 M265T283 M305T28 M183T56 M185T259 M121T102 M326T464 M467T632 M145T104 M267T335 M326T327 M384T81 M288T410 M535T616 M271T536 M413T800 M134T423 M505T537 M274T258 M166T43 M202T44 M483T450 M144T59 M169T41 M153T53 M136T52 M215T46 M175T47 M162T33 M182T85 M267T490 M170T490 M231T481 M314T501 M275T30 M259T29 M243T29 M123T30 M130T334 M233T514 M145T124	M180T194 M194T256 M338T217 M340T269 M231T44 M297T45 M271T321 M130T321 M255T321 M233T321 M356T327 M431T329 M211T428 M171T428 M431T428 M377T271 M447T410 M170T413 M453T302 M206T158 M143T32 M245T35 M220T68 M496T661 M461T347 M281T130 M149T296 M243T71 M197T137 M257T384 M475T455 M180T90 M328T280 M210T89 M131T407 M243T446 M436T446 M299T496 M354T333 M437T246 M183T80 M397T80 M305T253 M227T66 M162T300 M162T107 M227T188 M180T216 M188T158 M137T30 M144T137 M217T138 M594T431 M481T253	M289T27 M431T29 M227T29 M175T30_2 M159T30 M91T30 M140T34 M203T34 M118T35 M383T35 M875T275_1 M466T334 M240T356 M468T425 M542T433 M494T437 M495T441 M482T453 M568T457 M569T460 M496T473 M570T476 M454T481 M519T484 M523T485_1 M546T488 M522T490 M523T495 M544T500 M482T502 M524T539 M482T546 M524T551 M282T552 M538T587 M552T623 M606T684 M760T685 M785T685 M257T685 M759T685 M781T685 M406T687 M310T712 M640T754 M596T760 M554T769 M530T773 M339T785 M338T785 M360T785
		M302T478 M310T428 M167T65 M207T97 M169T41 M170T490	M180T194 M194T256 M338T217 M340T269 M206T158	M875T275_1 M494T437 M496T473 M524T539 M524T551

CV	Var	Human Urines	Rat's Urines	Rat's Plasma
4-Fold	Case 1 : [58]‖[8] Case 2 : [54]‖[5] Case 3 : [51]‖[5]	M365T539 M302T478 M225T257 M176T379 M279T379 M355T587 M303T473 M304T317 M310T428 M167T65 M207T97 M344T370 M342T360 M318T138 M330T414 M197T101 M265T283 M305T28 M183T56 M121T102 M326T464 M467T632 M145T104 M267T335 M326T327 M384T81 M288T410 M535T616 M271T536 M413T800 M134T423 M505T537 M274T258 M159T42 M166T43 M202T44 M483T450 M144T59 M169T41 M153T53 M136T52 M215T46 M175T47 M162T33 M182T85 M267T490 M170T490 M231T481 M314T501 M275T30 M259T29 M243T29 M123T30 M344T336 M130T334 M233T514 M145T124 M595T467	M180T194 M194T256 M338T217 M340T269 M231T44 M297T45 M271T321 M130T321 M255T321 M233T321 M356T327 M431T329 M211T428 M171T428 M431T428 M377T271 M447T410 M170T413 M453T302 M206T158 M143T32 M245T35 M220T68 M496T661 M461T347 M281T130 M149T296 M243T71 M197T137 M257T384 M475T455 M384T64 M180T90 M328T280 M210T89 M131T407 M243T446 M436T446 M299T496 M354T333 M437T246 M183T80 M397T80 M305T253 M227T66 M162T300 M162T107 M227T188 M180T216 M188T158 M137T30 M144T137 M217T138 M481T253	M289T27 M431T29 M227T29 M175T30_2 M159T30 M91T30 M140T34 M203T34 M118T35 M383T35 M875T275_1 M466T334 M240T356 M468T425 M542T433 M494T437 M495T441 M482T453 M568T457 M569T460 M496T473 M570T476 M454T481 M519T484 M523T485_1 M546T488 M522T490 M523T495 M544T500 M482T502 M524T539 M482T546 M524T551 M282T552 M538T587 M552T623 M606T684 M760T685 M785T685 M257T685 M759T685 M781T685 M406T687 M310T712 M640T754 M596T760 M554T769 M530T773 M339T785 M338T785 M360T785
		M302T478 M310T428 M167T65 M207T97 M265T283 M169T41 M153T53 M170T490	M180T194 M194T256 M338T217 M340T269 M206T158	M875T275_1 M494T437 M496T473 M524T539 M524T551

CV	Var	Human Urines	Rat's Urines	Rat's Plasma
7-Fold [5-Fold]	Case 1 : [55]|[5] Case 2 : [54]|[5] Case 3 : [51]|[5]	M365T539 M302T478 M225T257 M176T379 M279T379 M355T587 M303T473 M304T317 M310T428 M167T65 M207T97 M344T370 M342T360 M318T138 M330T414 M197T101 M265T283 M305T28 M183T56 M185T259 M121T102 M326T464 M467T632 M145T104 M267T335 M326T327 M384T81 M288T410 M535T616 M271T536 M413T800 M134T423 M505T537 M274T258 M166T43 M202T44 M483T450 M144T59 M169T41 M153T53 M136T52 M215T46 M175T47 M162T33 M182T85 M267T490 M170T490 M231T481 M314T501 M275T30 M259T29 M243T29 M123T30 M130T334 M233T514 M145T124	M180T194 M194T256 M338T217 M340T269 M231T44 M297T45 M271T321 M130T321 M255T321 M233T321 M356T327 M431T329 M211T428 M171T428 M431T428 M377T271 M447T410 M170T413 M453T302 M206T158 M143T32 M245T35 M220T68 M496T661 M461T347 M281T130 M149T296 M243T71 M197T137 M257T384 M475T455 M384T64 M180T90 M328T280 M210T89 M131T407 M243T446 M436T446 M299T496 M354T333 M437T246 M183T80 M397T80 M305T253 M227T66 M162T300 M162T107 M227T188 M180T216 M188T158 M137T30 M144T137 M217T138 M594T431 M481T253	M289T27 M431T29 M227T29 M175T30_2 M159T30 M91T30 M140T34 M203T34 M118T35 M383T35 M875T275_1 M466T334 M240T356 M468T425 M542T433 M494T437 M495T441 M482T453 M568T457 M569T460 M496T473 M570T476 M454T481 M519T484 M523T485_1 M546T488 M522T490 M523T495 M544T500 M482T502 M524T539 M482T546 M524T551 M282T552 M538T587 M552T623 M606T684 M760T685 M785T685 M257T685 M759T685 M781T685 M406T687 M310T712 M640T754 M596T760 M554T769 M530T773 M339T785 M338T785 M360T785
		M302T478 M310T428 M207T97 M169T41 M170T490	M180T194 M194T256 M338T217 M340T269 M206T158	M875T275_1 M494T437 M496T473 M524T539 M524T551

CV	Var	Human Urines	Rat's Urines	Rat's Plasma
Bootstrap	Case 1: [91]|[5] Case 2: [92]|[11] Case 2 : [56]|[12]	M365T539 M361T538 M188T95 M302T478 M120T63 M202T232 M287T257 M377T324 M144T36 M300T432 M818T683 M286T387 M786T745 M195T218 M466T619 M114T34 M138T110 M232T58 M151T59 M415T744 M246T99 M171T448 M174T41 M268T41 M176T379 M279T379 M355T587 M290T53 M132T47 M303T473 M304T317 M310T428 M299T708 M170T437 M167T65 M207T97 M185T96 M344T370 M342T360 M318T138 M120T115 M330T414 M197T101 M272T263 M374T680 M265T283 M305T28 M183T56 M163T259 M121T102 M326T464 M467T632 M145T104 M165T44 M326T327 M182T46 M384T81 M535T616 M410T746 M134T423 M221T538_1 M505T537 M274T258 M208T324 M229T43 M166T43 M202T44 M357T742 M310T450 M125T450 M144T59 M358T744 M211T448 M169T41 M153T53 M136T52 M130T48 M215T46 M269T47 M175T47 M152T32 M162T33 M170T490 M231T481 M314T501 M275T30 M259T29 M123T30 M114T54 M136T45 M262T45	M180T194 M413T193 M105T194 M576T194_1 M194T256 M216T256 M338T217 M340T269 M229T44 M231T44 M297T45 M271T321 M130T321 M233T321 M162T264 M356T327 M431T329 M476T329 M261T266 M371T266 M211T428 M171T428 M431T428 M190T214 M377T271 M447T410 M197T324 M520T352 M123T28 M170T413 M415T302 M453T302 M255T508 M152T150 M337T26 M321T26 M206T158 M152T31 M212T32 M143T32 M245T35 M220T68 M372T74 M327T207 M461T347 M164T347 M281T130 M271T591 M209T615 M285T531 M243T71 M215T48 M197T137 M257T384 M180T90 M342T284 M328T280 M131T407 M243T446 M436T446 M299T496 M181T346 M437T246 M183T80 M397T80 M305T253 M185T351 M227T66 M303T182 M162T300 M162T107 M576T194_2 M441T256 M136T46 M402T328 M227T188 M180T216 M146T272 M153T47 M275T27 M301T149 M188T158 M456T31 M137T30 M283T131 M144T137 M217T138 M126T397 M229T33 M162T164 M481T253 M175T277	M289T27 M431T29 M227T29 M175T30_2 M159T30 M91T30 M140T34 M203T34 M118T35 M383T35 M875T275_1 M1469T301 M466T334 M240T356 M468T425 M542T433 M494T437 M495T441 M482T453 M568T457 M569T460 M496T473 M570T476 M454T481 M519T484 M523T485_1 M546T488 M522T490 M523T495 M544T500 M482T502 M524T539 M482T546 M552T551_2 M524T551 M282T552 M510T573 M538T587 M552T623 M256T635 M606T684 M760T685 M785T685 M257T685 M759T685 M781T685 M807T685 M406T687 M310T712 M640T754 M596T760 M554T769 M530T773 M339T785 M338T785 M360T785
		M302T478 M310T428 M167T65 M207T97 M169T41	M206T158 M180T194 M194T256 M338T217 M340T269 M231T44 M297T45 M130T321 M233T321 M356T327 M431T329	M227T29 M203T34 M875T275_1 M468T425 M494T437 M568T457 M496T473 M523T495 M524T539 M482T546 M524T551 M338T785

RFFS

CV	Var	Human Urines	Rat's Urines	Rat's Plasma
Bootstrap	Case 1: [56][5] Case 2: [94][5] Case 2: [56][5]	M208T324 M120T90 M213T539 M120T105 M849T538 M132T47 M437T746 M120T63 M377T324 M211T448 M961T617 M377T520 M381T529 M524T386 M510T437 M867T683 M478T619 M234T401 M702T325 M361T538 M93T110 M467T258 M423T324 M224T685 M245T111 M611T741 M729T619_2 M943T619 M268T41 M339T742 M507T685 M1001T617 M483T262 M376T742 M547T468 M475T447 M470T539 M227T620 M564T620 M652T683 M710T619 M786T745 M496T620 M493T683 M522T684 M972T619 M228T747 M243T444 M218T375 M244T619_1 M303T67 M359T365 M230T538_1 M165T44 M884T742 M375T343	M137T30 M402T328 M166T137 M227T66 M177T273 M158T35 M179T193 M111T325 M250T26 M325T208 M148T515 M340T269 M492T497 M139T383_2 M228T66 M147T365 M453T269 M143T32 M333T270 M164T188 M129T328 M253T603 M181T469 M184T48 M185T383 M128T323 M118T382 M98T68 M146T128 M260T34 M361T266 M146T131 M289T240 M267T222 M299T496 M395T662 M141T353 M305T146 M121T86 M426T56 M210T319 M209T66 M264T243 M162T107 M132T240 M146T109 M306T235 M243T240 M278T553 M581T38 M248T34 M138T33 M367T48 M373T270 M130T244 M274T186 M350T424 M340T110 M170T47 M374T276 M257T384 M175T269 M438T87 M405T276 M228T528 M213T33 M188T158 M288T529 M119T451 M242T188 M206T158 M157T276 M273T528 M322T269 M161T287 M168T32 M229T528 M260T195 M164T305 M342T284 M349T443 M348T300 M303T236 M378T269 M187T423 M132T32 M513T549 M346T497 M311T347 M314T515 M210T528 M162T238 M437T288 M158T68	M112T35 M606T275_1 M257T572 M853T276_1 M854T276_2 M371T664 M853T276_2 M311T712 M284T707 M243T29 M787T30 M559T664 M293T712 M761T274_2 M373T664 M310T712 M642T551 M289T497 M643T719 M332T712 M553T276 M840T30 M324T747 M123T30 M373T36 M431T29 M923T30 M767T486 M583T30 M438T500 M240T356 M169T34 M198T34 M562T462_2 M615T590 M347T323 M91T30 M159T30 M135T363 M257T36 M476T481 M336T724 M448T30 M424T386 M587T275_2 M482T603 M781T685 M191T550 M372T664 M198T374 M279T549 M269T516 M271T27 M532T29 M526T485_1 M763T533
		M208T324 M120T90 M213T539 M120T105 M849T538	M137T30 M402T328 M166T137 M227T66 M177T273	M112T35 M606T275_1 M257T572 M853T276_1 M854T276_2

PLS-DA

CV	Var	Human Urines	Rat's Urines	Rat's Plasma
LOO	case 1: [56]‖[5]　case 2: [60]‖[8]　case 3: [87]‖[6]	M361T538　M188T95　M120T63 M377T324　M818T683　M195T218 M466T619　M138T110　M415T744 M171T448　M132T47　M287T358 M374T680　M410T746　M215T685 M403T662　M510T621　M360T372 M218T375　M359T365　M377T411 M228T748　M849T538　M188T63 M461T323　M238T324　M867T683 M493T683　M864T683_2 M931T683　M462T683　M219T684 M245T683　M652T683　M884T742 M478T619　M496T620 M235T619_1 M244T619_1 M227T620　M710T619　M962T618 M943T619　M972T619　M211T448 M234T401　M393T142　M523T600 M228T747　M507T685　M224T685 M651T684　M521T686　M470T539 M474T619　M951T619	M179T193　M338T217 M340T269　M485T410 M143T32　M158T35 M328T280　M213T62 M248T34　M185T351 M227T66　M228T66 M374T276　M111T325 M129T328　M162T107 M274T186　M349T443 M177T273　M236T298 M347T225　M141T353 M181T469　M325T208 M210T319　M304T278 M267T222　M148T515 M453T269　M136T46 M402T328　M250T26 M137T30　M169T39 M166T137　M260T268 M157T276　M405T276 M119T451　M146T109 M340T110　M228T528 M229T528　M272T528 M273T528　M288T529 M386T528　M147T365 M139T383_2　M167T383 M185T383　M151T239 M132T240　M289T240 M128T272　M164T188 M175T269　M322T269 M333T270　M362T188	M289T27　M885T28　M817T28 M532T29　M243T29　M847T30_1 M448T30　M159T30　M583T30 M91T30　M923T30　M787T30 M120T32　M198T34　M116T34 M118T35　M112T35　M789T35 M317T35　M373T36　M257T36 M229T39　M147T170　M593T243_2 M412T256　M487T273　M761T274_2 M761T274_1　M886T274_1　M880T275_4 M606T275_1　M587T275_2　M854T276_2 M553T276　M853T276_1　M853T276_2 M155T297　M99T298　M130T314 M347T323　M240T356　M135T363 M223T375　M424T386　M357T387 M400T401　M357T448　M526T485_1 M767T486　M635T489　M267T498 M586T542　M279T549　M149T549 M191T550　M816T551_1　M809T551_1 M818T551　M582T551　M642T551 M371T554　M257T572　M336T595 M482T603　M374T615　M256T635 M376T656　M568T664　M372T664 M371T664　M559T664　M560T664 M373T664　M284T707　M293T712 M332T712　M311T712　M275T712 M310T712　M643T719　M336T724 M439T726　M638T730　M324T747 M610T747　M285T751　M322T785
		M361T538　M377T324　M466T619 M374T680　M651T684	M158T35　M129T328 M177T273　M148T515 M402T328　M250T26 M137T30　M166T137	M112T35　M854T276_2　M642T551 M257T572　M371T664　M439T726

sPLS-DA

LOO	case 1: [56]‖[5] case 2: [60]‖[10] case 3: [72]‖[9]	M361T538 M188T95 M120T63 M377T324 M143T35 M818T683 M195T218 M466T619 M415T744 M265T448 M171T448 M132T47 M231T47 M450T489 M287T358 M360T139 M114T128 M272T263 M374T680 M300T454 M258T50 M302T56 M181T129 M204T369 M215T685 M403T662 M325T472 M510T621 M360T372 M218T375 M330T340 M239T380 M359T365 M344T446 M340T147 M655T652 M381T211 M240T63 M316T233 M461T323 M238T324 M867T683 M493T683 M931T683 M245T683 M652T683 M478T619 M235T619_1 M244T619_1 M710T619 M943T619 M288T450 M540T58 M211T448 M392T129 M293T415 M702T325 M260T44 M376T122 M346T433 M486T94 M393T142 M523T600 M276T70 M304T130 M656T652 M651T684 M567T472 M304T163 M951T619	M190T214 M175T48 M321T26 M152T31 M143T32 M158T35 M226T295 M418T36 M307T312 M213T29 M278T454 M265T284 M342T284 M248T34 M443T23 M183T80 M360T81 M111T325 M129T328 M444T451 M304T276 M177T273 M277T209 M304T278 M148T515 M232T525 M560T194_2 M452T218 M527T217 M402T328 M411T27 M325T26 M391T26 M250T26 M456T31 M137T30 M174T35 M163T67 M374T373 M494T294 M291T312 M199T312 M458T312 M166T137 M260T268 M130T34 M511T79 M214T101 M211T480 M245T55 M257T342 M320T107 M177T418 M211T451 M404T25 M146T109 M134T110 M206T125 M230T124 M164T188	M328T28 M463T28 M531T28 M229T30 M364T30 M619T30 M823T30 M637T30 M164T33 M119T35 M112T35 M130T35 M317T35 M549T36 M105T246 M487T273 M761T274_2 M592T275_2 M590T275_2 M606T275_2 M854T276_2 M853T276_1 M853T276_2 M161T304 M250T369 M272T374 M424T386 M333T408 M295T431 M300T433 M428T442 M498T442 M539T461 M808T461 M558T461 M549T462 M548T462 M280T462 M821T462 M812T462 M341T463 M200T481 M526T485_1 M1021T485_1 M268T486 M317T487 M554T500 M272T500 M639T500 M361T523 M675T537 M553T551_1 M809T551_1 M642T551 M300T552 M291T552_2 M371T554 M257T572 M305T639 M371T664 M559T664 M355T664 M373T664 M699T704 M284T707 M654T708 M610T711 M571T715 M628T719 M643T719 M439T726 M324T747
		M466T619 M305T28 M135T421 M258T50 M443T261	M338T217 M165T90 M253T363 M330T286 M452T218 M303T427 M443T27 M250T26 M137T30 M187T40	M112T35 M761T274_2 M429T442 M492T517 M528T551 M642T551 M257T572 M538T574 M559T664