SOFTWARE SUPPORT FOR QUANTITATIVE NEAR-INFRARED ANALYSIS AND BENCHMARKING OF CHEMOMETRIC METHODS - A CASE STUDY ON SINGLE KERNEL SAMPLES

A thesis submitted to the University of Manchester for the Doctor of Philosophy degree in the Faculty of Science and Engineering

2020

Shupeng Hu

School of Engineering, Department of Computer Science

LIST OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF PUBLICATIONS

[1] L. Zhao, S. Hu, X. Zeng, Y. Wu, Y. Lin, J. Liu, et al., "An Integrated Software System for Supporting Real-Time Near-Infrared Spectral Big Data Analysis and Management," in Big Data (BigData Congress), 2017 IEEE International Congress on, 2017, pp. 97-104.

This paper first introduces the basic concepts of NIRS analysis intending to show its complexity. The paper then characterises the NIR spectral data using the “3H” of scientific big data, intending to show their challenges. Finally, the paper describes our initial effort on the development of an integrated software system to support efficient real-time NIRS data analysis and management. The paper claims that this development is an important contribution to tackling the challenges of scientific big data.

The integrated software system introduced in this paper was an initial version of the QNIRSA system that will be specified in this thesis. My role in that paper was to design the software system and spectral database and help to write a draft about the basic concepts of NIRS analysis and the features of NIR spectral data.

[2] Z. Xu, S. Fan, S. Hu, J. Liu, B. Liu, L. Tao, J. Wu, et al., "A calibration transfer optimized single kernel near-infrared spectroscopic method," Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy, vol. 220, p. 117098, 2019.

In this paper, a calibration transfer-optimized single kernel near-infrared spectroscopic method is proposed. This method aims to accurately detect the chemical composition of single seeds by using the calibration model of the corresponding dehusked seeds or seed flour. The proposed method was applied to the analysis of the protein content of a single rice kernel.

This paper reported a real-world application supported by the QNIRSA system, which will be illustrated in section 7.2 in this thesis. My role in that paper was to configure the QNIRSA system to support that application, including data collection, processing and result analysis, and help to code the proposed method in MATLAB as well.

[3] J. Liu, S. Hu, S. Yu, Y. Lin, P. Wei, Y. Yang, et al., "Analysis of biuret in urea fertiliser using a portable near-infrared spectrometer," Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy, Under review 2020.

This application is about analysing biuret in urea fertiliser using a portable near-infrared spectrometer. Some chemometric methods were compared and discussed.

This paper reported another real-world application entirely supported by the QNIRSA system, which will be illustrated in section 7.3 in this thesis. My role in that paper was to configure the QNIRSA system to support that application including data collection, processing and result analysis.

ABSTRACT

During the past decades, the technology of the Near-Infrared Spectroscopy (NIRS) has been widely adopted as a non-destructive analytical tool in various fields. In agriculture and chemometrics, NIRS analysis at single kernel level can improve not only the sample uniformity and purity but also the quality and economic value of a seed batch. However, many limitations and challenges exist in the single kernel Near-Infrared Spectroscopy (SKNIRS) analysis and applications. The first contribution of this PhD thesis was to develop an integrated software system to support data collection, processing and analysis for single kernel sample. IDEF0 Functional Modelling has been used to guide the development and implementation of the integrated software system. Two real-world applications supported by the integrated software system were reported as the validation of the integrated software system. Another contribution of this PhD thesis was to provide a benchmark of chemometric methods for comparative single kernel near-infrared spectroscopy analysis based on the proposed stepwise process. Sixteen methods including two dataset partition methods, three pre-processing methods, eight variable selection methods and three calibration methods were assessed and compared based on two statistics: root mean squared error of prediction (RMSEP) and coefficient of determination (R^2). Conclusions were discussed in detail based on the results of benchmarking analysis, which is appropriately general and may assist the choice of chemometric methods for SKNIRS.

Keywords: Near-Infrared Spectroscopy (NIRS), Chemometrics, Single Kernel Near-Infrared Spectroscopy (SKNIRS), Software System Development, IDEF0 Functional Modelling, Benchmarking of Chemometric Methods, Quantitative Analysis, Pre-processing, Variable Selection, Multivariate Calibration

DECLARATION

That no portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

COPYRIGHT STATEMENT

ACKNOWLEDGEMENTS

Thank you to my supervisor Dr Liping Zhao for her guidance and support in the past four years. Thanks to Prof. YueJin Wu, Dr Jing Liu, Dr Qi Wang, Dr Zhuopin Xu and PhD student Shuang Fan, from Key Laboratory of High Magnetic Field and Ion Beam Physical Biology, Hefei Institutes of Physical Science, Chinese Academy of Sciences, China, who provided research datasets and professional advice in near-infrared spectroscopy and chemometrics.

1 INTRODUCTION

1.1 Context

Near-infrared spectroscopy (NIRS) is a spectroscopic technique which utilises near-infrared (NIR) light region of the electromagnetic spectrum from 780 nanometres to 2500 nanometres [4]. Compared with other analytical and conventional chemical methods, NIRS is non-destructive, easy to use and has rapid response [5]. Due to those advantages, NIRS has been widely applied for diverse fields such as food science [6], agriculture [7], environmental science [8], human health [9], pharmaceutics [10], wood and paper science [11]. In the past decades, the number of NIRS applications have exponentially grown because of the rapid development of NIRS instrument and software. One kind of the most popular applications aims to use NIRS for the selection and classification of samples according to specific traits and attributes without alteration of their properties. Specifically, in this kind of application, the analysis is essential to utilise methods to construct an appropriate calibration model to correlate the underlying relationship between the spectral data and sample properties. For example, in the agricultural domain, the interesting properties of samples are water concentration, fat content, protein content [12] and so on. Therefore, the goal of NIRS for those agricultural applications is to correlate the relationship between the spectral data of agricultural commodities and the properties of those commodities. Chemometrics is an inter-discipline field involving mathematics, statistics, chemistry, biology, computer science and maybe even more. Quantitative NIRS analysis in chemometrics is one of the methodologies to quantify the numeric relationship between spectra and sample properties. A number of coherent processes employing methods provided by other fields construct steps of quantitative NIRS analysis. Eventually a NIR model will be established to reveal the potential relationship between the NIR spectral data and sample properties.

Generally, in agriculture, there are two types of solid samples for NIRS. One of the types is the bulk sample that every sample has similar weight about 200 grams to 250 grams on average consisting of some single kernels, provided by NIRS bulk sample analysers [13]. Another type of sample is the single kernel (e.g., a single seed or single rice). The NIR analysis for a single kernel is undertaken for every kernel of solid sample one by one. Therefore, analysis of the bulk sample is much easier than the investigation of unequable single kernels. However, bulk samples may cause the issue that the heritability of the desired characteristic may be low because difference among single kernels cannot be discriminated. On the contrary, though the investigation of the single kernel is complicated, it is possible to figure out the traits of the sample in current or even next generation [14]. An additional advantage given by single kernel NIRS

analysis (SKNIRS) is the improvement of sample uniformity and purity, and the quality and economic value of a kernel batch [15].

1.2 Research Problems

Bulk samples have been widely used in many NIRS applications, but SKNIRS has not reached its full potential yet. Due to the differences existing among single kernels, both reference methods and chemometric methods successfully apply for bulk samples may not have the same performance for single kernels [10]. According to the literature review of related work about applications of chemometric methods in section 3.7, most of the SKNIRS papers employed only 3 to 6 chemometric methods for one category of single kernel sample, which were not enough for carrying out a comparison study. Few numbers of chemometric methods for comparison are insufficient to provide a global view for NIRS benchmarking. Besides, the statistic criteria used for different papers sometimes may be different as well. A fair comparison should be undertaken in the same criteria. On the other hand, several review papers did comparative research between methods. However, they all focused on one category of methods such as pre-processing methods or variable selection methods only [16], [17], [18], [19]. No review has emphasised on the impact between different categories of methods yet. Additionally, those review papers mainly focused on the theory and principle but lack of applications or analysis on real-world data sets.

Before NIRS analysis at the single kernel level, the acquisition of spectral data for single kernels is necessary. Most of the current commercial spectrometers are used for bulk samples, though some of them have adapters for single kernels [13]. Some papers reported the issue that some current spectrometers with attached accessories available for bulk samples are not suitable for single kernel samples [20], [21]. Problems also exist in software corresponding to spectrometers. The first problem is that many spectrometers are available to control by their software only. As a result, if multiple spectrometers are required for investigation, all relevant software has to be installed and configured. When the spectrometer is changed, the corresponding software has to be swapped and reset, then re-connects with the spectrometer. This is time-consuming for comparative research. The second problem is that, software only responds to corresponding spectrometer but cannot control external sample-specific hardware accessories (e.g., adapters or external sensors) designed particularly for the scanning of single kernel sample. Thus, many NIR software are under restrictions in the SKNIRS analysis. The third problem is about the chemometric methods provided by the software. As for a large number of NIR software, only several methods are provided, which are not enough for NIR research, especially insufficient for a comparative study. As a result, NIRS analysis and spectra acquisition have to done separately on different software. The most current integrated software system is unavailable for the coherent process, including NIRS data collection and comparative analysis.

In summary, the first research problem is the lack of a comparative study for SKNIRS to provide a global view for the NIR benchmarking of chemometric methods. The

second research problem is the lack of an integrated system at the single kernel level, which can not only control sample-specific hardware accessories to support spectral data collection of single kernel sample but also provides a wide range of chemometric methods to support NIRS comparative analysis.

1.3 Research Questions, Aims and Objectives

Research questions are based on research problems summarised in previous section 1.2.

RQ 1. What is the most appropriate methodology for developing the integrated software system?
RQ 2. What is the best statistic criterion to assess and interpret the performance of chemometric methods?
RQ 3. How to classify the performance of chemometric methods based on statistic criterion?

Based on three research questions, the aims of this PhD thesis is to 1) develop an integrated software system to support spectral data collection of single kernel sample and single kernel near-infrared spectroscopy analysis and 2) provide a benchmark of the chemometric methods for single kernel near-infrared spectroscopy analysis. Benchmark processes for measurement and comparison of the performance of chemometrics methods will be interpreted in chapter 4, while the details about the development of the integrated software system will be interpreted in chapter 5. In order to realise two research aims, research objectives below should be reached:

Obj 1. To review the literature about software engineering, methodology and common methods and statistic criterion used for SKNIRS.
Obj 2. To design an architecture for the integrated software system.
Obj 3. To find out the most appropriate methodology for developing the integrated software system.
Obj 4. To develop a chemometric methods library providing desired chemometric methods for the integrated software system.
Obj 5. To figure out the best statistic criterion to assess and interpret the performance of desired chemometric methods.
Obj 6. To make a comparative study for desired chemometric methods on a single kernel sample as the benchmark for SKNIRS analysis.
Obj 7. To classify the performance of desired chemometric methods based on statistic criterion.

1.4 Research Methodology

The whole research of this PhD thesis has been done in steps inspired by evidence-based software engineering [22]. These steps are:

Step 1 Identify research problems from practical applications.
Step 2 Convert research problems into answerable research questions.

Step 3 Search the literature for the best available evidence to answer the research questions.
Step 4 Propose solutions based on evidence in step 3.
Step 5 Implement proposed solutions to solve research problems.
Step 6 Evaluate the proposed solutions in a pilot project (real-world application).

This PhD thesis is a practical NIRS application associated with collaborators from the Hefei Institute of Physical Science, Chinese Academy of Sciences. Thus, in terms of step 1 and 2, the research problems were addressed and converted into research questions illustrated in section 1.2 and 1.3. In step 3, the methodology for the literature review is that relevant review papers recommended by collaborators were reviewed at first. Reference papers cited by review papers concerning potentially available evidence were reviewed sequentially. Afterwards, other papers cited in those reference papers were reviewed as further reading. This literature review progress iterated until enough available evidence were found to answer the research questions. Methodology functional modelling by the IDEF0 approach was applied for the development of the proposed integrated system as a solution for software system development in step 4 and 5. Function modelling in systems engineering and software engineering is a structured representation of the functions within the modelled system or subject area [23]. The IDEF0 is a functional modelling language for describing functions and developing a software system. The reason why adopts such a stepwise functionalisation method is that NIRS analysis is a coherent, systematic process illustrated in section 2.3.2. The methodology for NIR benchmarking on the single rice sample refers to chapter 4 in sequential order:

1. Define the keyword ‘benchmark’ to clear the objectives.
2. Collect research data.
3. Figure out benchmarking criterion.
4. Design benchmarking process

In step 6, two real-world applications will be reported in chapter 7 as the validation for the integrated software system.

1.5 Summary of Research Contribution

There are two main contributions of this PhD thesis. The first contribution is to make a comparative study for various statistical models established by sixteen chemometric methods on the single rice sample as the benchmark of chemometric methods for single kernel near-infrared spectroscopy analysis. Compared with previous work which focused on the analysis of bulk rice samples, this exploratory research in addressing single rice samples leads to a more accurate assessment of properties such as protein content. This study is also an example and guidance to show analytical processes for assessing different statistical models on single kernel samples. Those processes of benchmarking can guide future research on how to design and implement the process on single kernel NIR analysis. Additionally, the comparative results of those models are a

useful reference for chemometric methods selection on single rice samples. They provide detailed assessments of sixteen methods, including not only the performance of those methods on three forms of single rice but also the optimal parameters of methods. Relevant research may briefly refer to this comparative study when it is related to methods selection, parameters tuning or model calibration on single rice samples.

Another contribution is to develop an integrated software system named QNIRSA system, which not only can control multiple spectrometers and sample-specific hardware for spectral data acquisition of single kernel sample, but also provides some chemometric methods for single kernel near-infrared spectroscopy analysis. The QNIRSA system can be regarded as a fully integrated functional platform which provides APIs for multiple spectrometers and hardware, a graphical user interface for users, and a chemometric methods library which provides some useful methods for single kernel near-infrared spectroscopy analysis as well. Two real-world NIRS applications have been solved by the QNIRSA system, which is reported in chapter 7. The QNIRSA system, including both Java codes for software and MATLAB codes for algorithms, was implemented by myself.

Chemometric methods involved in this PhD thesis can be divided into four categories (principles of these methods will be introduced in chapter 3):

1. Dataset Partition method. It is used to divide original data set into a training set and a validation set. Two sampling methods were assessed: 1) Kennard–Stone algorithm (KS) and 2) sample set partitioning based on joint x–y distances (SPXY).
2. Pre-processing method. It is used to remove physical phenomena in the spectra in order to improve the subsequent analysis. Three pre-processing methods were assessed: 1) multiplicative scatter correction (MSC), 2) standard normal variate (SNV), and 3) Savitzky-Golay polynomial derivative filters (SG).
3. Variable selection method. It is used to reduce the number of variables inside spectra which have noise or unrelated information, in order to improve the calibration model performance. Eight variable selection methods were assessed: 1) successive projections algorithm (SPA), 2) uninformative variable elimination (UVE), 3) genetic algorithm (GA), 4) simulated annealing (SA), 5) principal component analysis (PCA), 6) interval partial least squares (iPLS), 7) backward interval partial least squares (BiPLS) and 8) forward interval partial least squares (FiPLS).
4. Calibration method. It is also named the regression method, which is used to construct the calibration model based on its regression principle. Three calibration methods were assessed: 1) multiple linear regression (MLR), 2) principal component regression (PCR) and 3) partial least squares regression (PLSR).

1.6 Thesis Scheme

Besides the first chapter, there are seven more chapters in this PhD thesis. Chapter 2 is a literature review including the background of NIR technology, chemometrics and related works. Chapter 3 will specify the theory and principle of 16 chemometric methods involved in this PhD thesis. Chapter 4 and 5 will illustrate the proposed solutions for the design of the benchmark and development of the QNIRSA system, respectively. Chapter 6 will analysis the benchmark and provide a comparative study for those chemometric methods. Chapter 7 will report two real-world applications by using the QNIRSA system as the validation. Chapter 8 is the conclusion of this PhD thesis.

2 CONCEPTS AND BACKGROUND: NEAR-INFRARED SPECTROSCOPY AND CHEMOMETRICS

2.1 Overview

The near-infrared spectroscopy (NIRS) became a formal analytical tool utilised for practical application since the first application was reported by the pioneers of the NIRS, Hart, et al. [4]. Afterwards, a large number of NIRS applications have been developed in various fields. This chapter will specify the background of the near-infrared spectroscopy (NIRS) and chemometrics, including the principle of NIRS technology in section 2.2 and chemometric process in section 2.3. Specifically, section 2.2 elaborates the basics of NIRS, principles of NIRS instrument and features of NIRS data sequentially, while section 2.3 introduces the definition of chemometrics and steps of quantitative NIRS analysis (QNIRSA) with an overview of some popular chemometric methods.

2.2 Principle of the Near-Infrared Spectroscopy Technology

2.2.1 Basics of Near-Infrared Spectroscopy

Full name of the spectroscopy is overtone vibrational spectroscopy (OVS), which utilises light-electromagnetic radiation to analyse materials by describing the energy transfer between light and matter [5]. A spectrum reflecting the light intensity at a different wavelength on multi-dimensional spectral space is the product concerning the description of energy. The light-electromagnetic spectrum is divided into some regions, and each region indicates a particular category of molecular or atomic transition with a corresponding spectroscopic technique [5]. Near-infrared defines the region ranging from 780nm to 2500nm. Between this range, molecules can absorb infrared light without later reemission by exciting specific vibrational frequencies, while sample absorbs the frequencies of polychromatic light that corresponds to its molecular vibrational transitions [5]. Due to this feature, the near-infrared spectrum can be composed of absorptions. Moreover, every molecule containing hydrogen will have a measurable NIR spectrum and the ubiquitous distribution of hydrogen inside the molecule, which means a vast number of analyte are amenable to NIR analysis [24].

Compared with other analytical methods (e.g., wet chemical method) or other spectroscopic techniques (e.g., Raman), NIRS have six main advantages:

1. NIRS provides multi-constituent analysis on virtually any matrix with levels of accuracy and precision that are comparable to primary reference methods [5]. The comparison based on a set of criteria between the reference data and predictions by NIRS analysis is an essential step to evaluate the NIRS calibration model.
2. NIRS is a non-destructive and waste-free technique that does not require sample preparation or manipulation with hazardous chemicals, solvents, or reagents [5]. NIRS is attractive for straightforward, speedy characterisation of natural and synthetic products because it can collect and record spectral data for both solid and liquid samples without pre-treatment [17].
3. NIRS is faster than traditional laboratory analysis because NIRS allows several quality estimates to be performed within a manufacturing cycle in opposed to a single end of batch analysis [17]. This advantage may reveal potential problems early in the process and promote corrective actions as well.
4. NIRS is considered as a safe technique due to intrinsically safe measurement probes and fibre optics [17].
5. NIRS is flexible, which allows the determination (it means the whole process for measurement of the sample properties) of multiple values in a single measurement [5].
6. NIRS is sensitive to physical parameters such as size, density and colour. This feature is useful for the capture of those characteristics of samples.

However, there are also some limitations of NIRS. For example, the NIR spectra are often complex and usually possess broad overlapping NIR absorption bands that require special mathematical procedures for data analysis [5], which increases the difficulty of NIRS data analysis. Furthermore, with the development of the NIRS device, higher accuracy allows the instrument to produce a large number of spectral points as variables that even are more than the number of samples. A large number of variables often renders the prediction of a dependent variable unreliable [25]. Then proper variable selection method must be applied for the spectral data, which raises the analysis complexity as well. Although the sensitivity of NIRS to physical parameters brings the sixth advantage as the above context explained, it leads to an issue as well, that is, physical phenomena exist as noises in the spectral data. Thus, pre-processing methods, which increases the complexity of spectral data analysis as well, are necessary to be applied for the spectral data, in order to remove physical phenomena in the spectra. The non-destructive and waste-free features of NIRS attracts scholars' attention, but the complicated analysis and processing of spectral data stop direct NIRS measurement of samples. To overcome those disadvantages, a specialised field, named chemometrics, has been investigated for years, which will be introduced in the next section 2.3.

2.2.2 Principle of the Near-infrared Spectroscopy Instrument

The limited capability of the instrument has been the main reason that restricted most scholars' NISR research only in laboratory condition in the last century [26]. The lack of advanced computer chip has made the sensitive and low noise detectors in the form of silica photocells or lead sulphide photometers, and fibre optic cables difficult to control [24]. Significant development of computer technology has taken place since the 1990s. Nowadays, powerful computer-based elements are widely used to improve NIRS instrument precision and accuracy and make mathematical methods available for supporting the complex spectral data processing in diverse fields. Current NIRS instrument can be divided into two types according to its working modes. The two categories of working modes are transmittance mode and reflectance mode, respectively.

NIRS instrument utilises the transmittance mode, which mainly works on the region from 700 to 1100 nanometres. This kind of instruments measures the transmitted radiation through a fixed path length of a bulk sample of grains or beans, assuming that the decrease of the initial radiation in travelling through the sample is due to absorption [13]. The path length is optimised according to the commodity being measured and the instrument setup. Another type of NIRS instruments deploys the reflectance mode, which measures the diffusely reflected radiation from the sample. According to the illustration written by Agelet and Hurburgh [13], the diffusely reflected signal is a fraction of the original radiation source which after penetrating the sample few millimetres, has been interacting with the sample molecules, scattered in several directions, and travelled back to the surface. Only the diffuse fraction of the reflected radiation has interacted with the compound of interest, while other reflected fractions may only have interacted with the sample surface and thus does not contain chemical information related to the sample composition.

In terms of the techniques for spectrum measurement employed by NIR device, diode arrays (DA), Fourier Transform (FT), and chemical imaging units (CI) are most widely applied. Diode array instruments measure the signal from all the wavelengths simultaneously and are usually the cheapest and the most suitable for fast measurements in rougher environments because they do not contain mechanical parts [13]. Fourier Transform instruments, measuring all the wavelengths at the same time as well but in the frequency domain, are mostly seen as laboratory instruments because of its higher complexity. FT advantages over diode arrays are a higher signal to noise ratio (SNR), higher precisions and higher resolution. However, those benefits do not generally lead to significant over-performance when working in the NIR region, especially when working with agricultural samples [27]. Chemical imaging is a relatively newer technology in NIR spectroscopy, which provides an additional spatial dimension to the NIR multivariate data. It allows identifying and mapping NIR biochemical information. The feature has been proved to be useful when analysing oil content in corn kernels, while only pixels belonging to the germ region should be considered as most of the oil is located in that region [28]. However, the disadvantages of chemical imaging

technique are slower, have lower signal to noise ratios, and have lower penetration of the radiation in the sample [13].

2.2.3 Near-Infrared Spectral Data

When a sample is scanned by a spectrometer (NIR device), the spectrometer produces a single spectrum containing a large number of data points (absorptions measured at every single wavelength on multi-dimensional spectral space) as the result of spectral response. In other words, a measured NIR spectrum of a sample can be regarded as a massive number of absorption measurements that have to be performed at hundreds of wavelengths on different dimensions. Therefore, the near-infrared spectral data is multivariate, and its analysis is called multivariate analysis. The data point one at a wavelength is the independent variable of the spectral data (also named predictor in some papers), and the quantity of independent variable depends on the resolution of the spectrometer. Those data points equal to the specific term ‘observations’ in other fields like machine learning. Figure 2.1 shows a spectrum of a single rice sample, which has 936 variables, while Figure 2.2 displays the spectra consisting of 120 single rice samples. Every sample has 936 variables. The size of the spectra is a matrix of 120×936 in math. The X-axis presents the wavelength range while the Y-axis is about general absorbance.

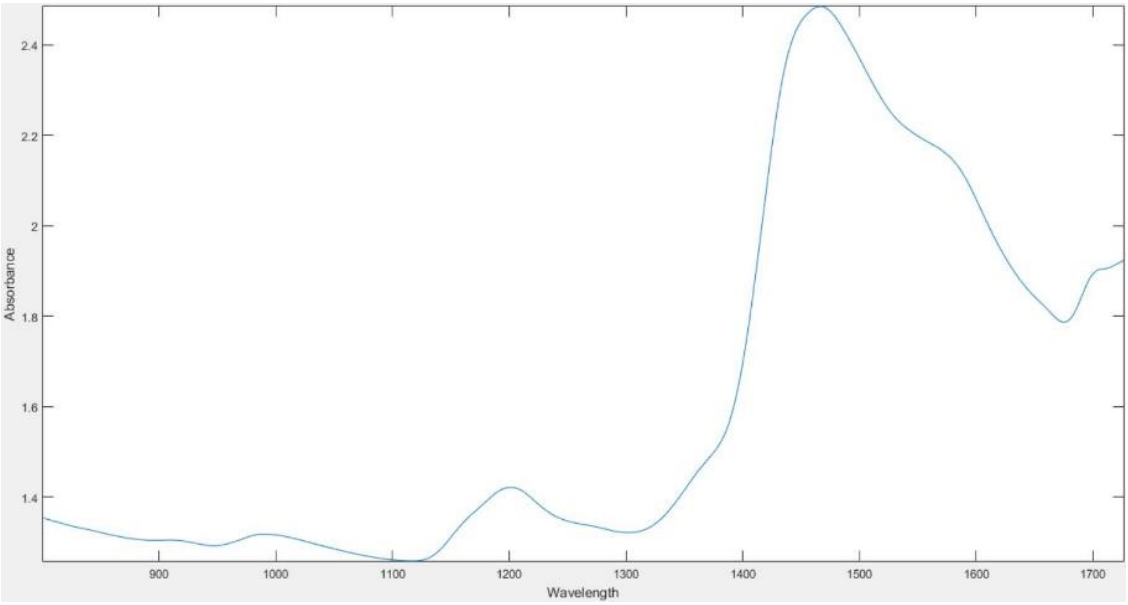


Figure 2.1: The spectrum of a single rice sample.

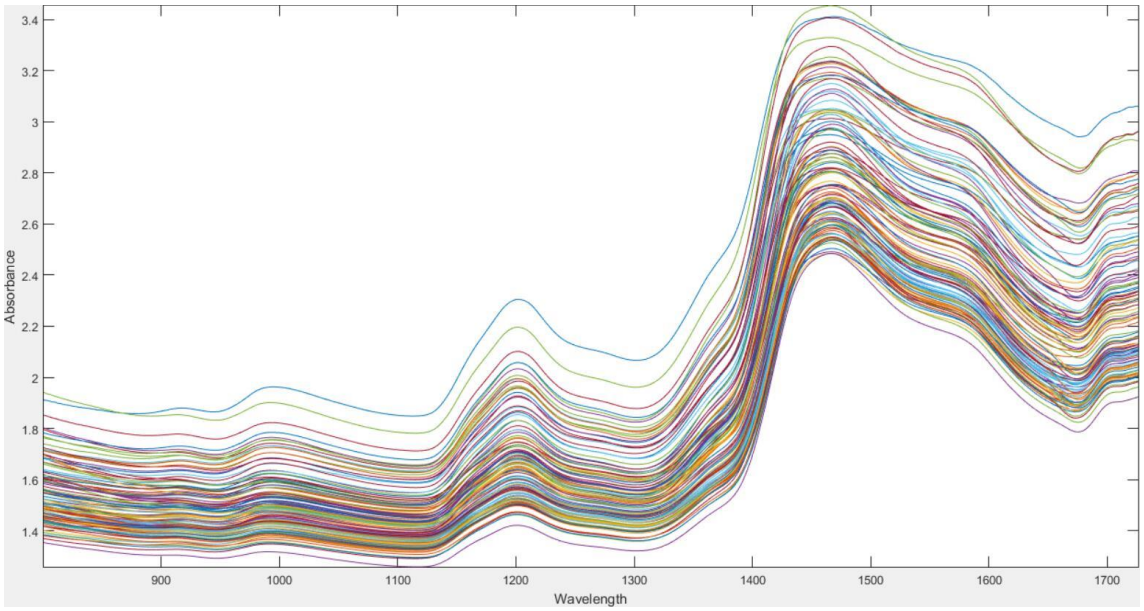


Figure 2.2: The spectra of 120 single rice samples.

Dependent variables should be those to be predicted. Specifically, sample property (e.g., the concentration of water), also called desirable interest in some papers, is the dependent variables in NIRS analysis. In machine learning, those dependent variables are also named targets. In model training, reference data (value of sample property used for model training and calibration) corresponding to the spectra are provided by the laboratory experiments based on reference methods. Precision and accuracy of NIRS calibrations will be impacted by the quality of the reference laboratory data.

An instance is shown in Figure 2.3 presents the relationship between the spectra of single rice samples (size: 120x936) and the reference data (protein content). As it is seen in Figure 3, the spectra and reference data follow the one-to-one relationship. Each spectrum of the spectra has its corresponding value of reference data and only has one value for every kind of reference data. For example, in Figure 2.3, there is one property (multi-properties are available but here takes one property, for instance), protein content. Therefore, every spectrum has one corresponding value of the protein content. Thus, the reference data set is constructed by 120 values of protein content. Regarding the multiple properties, if there are p properties, the size of the reference dataset should be a matrix of $120 \times p$ in math.

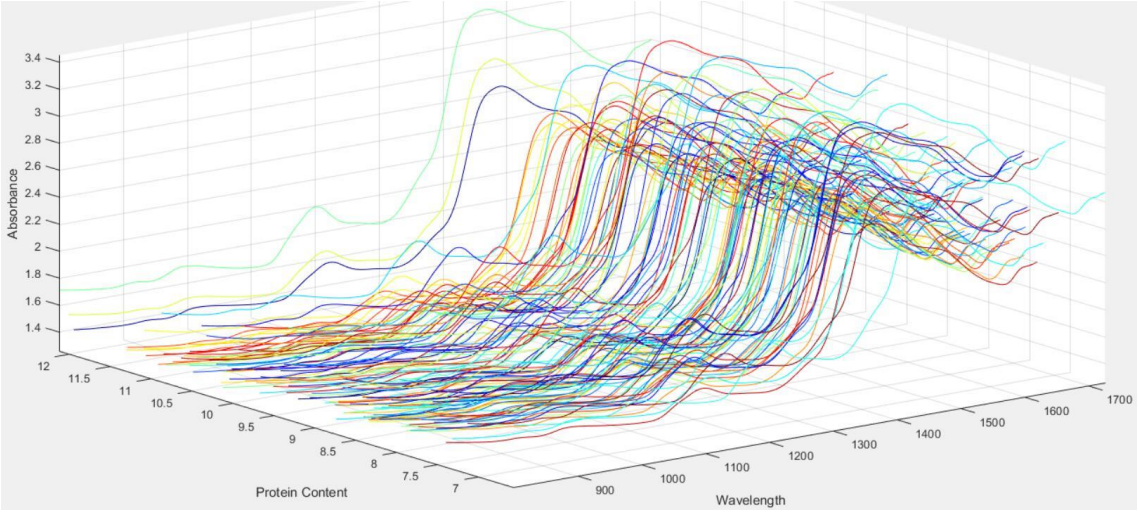


Figure 2.3: The spectra of 120 single rice samples with reference data.

In NIRS analysis, in order to make mathematical and statistical computation easier, the NIRS data have to be transformed into a matrix for future processes. The matrix transformed from the spectra is named the absorbance matrix (matrix *X*). An instance of the specific matrix *X* transformed from the spectra of 120 single grain samples (size: 120×960) shown in Table 1. In this table, every sample has its corresponding spectrum, while every spectrum has 936 values/data points as one value at one wavelength.

Table 2.1: matrix *X* transformed from the spectra of 120 single rice samples.

Samples	Spectra	Wavelengths				
		800.7(1st)	801.7(2nd)	802.7(3rd)	……	1726.1(936th)
Sample 1	Spectrum 1	1.4687	1.4679	1.4671	……	2.1623
Sample 2	Spectrum 2	1.4488	1.4484	1.4474	……	2.2160
……	……	……	……	……	……	……
Sample 120	Spectrum 120	1.7821	1.7812	1.7800	……	2.7752

In terms of reference data, a matrix *Y* contains all the values of it. Table 2 displays the matrix *Y* corresponding to the matrix *X* in Table 1. One of the NIRS analysis aims is to establish a model to investigate and reveal the correlation between matrix *X* and matrix *Y*.

Table 2.2: Matrix *Y* corresponding to the matrix *X* in Table 2.1.

Samples	Spectra	Properties		
		Protein Content	……	Water Concentration
Sample 1	Spectrum 1	9.0678	……	8%
Sample 2	Spectrum 2	11.9476	……	5%
……	……	……	……	
Sample 120	Spectrum 120	7.3899	……	10%

2.3 Chemometrics

2.3.1 Definition

An accepted definition of the chemometrics is that chemometrics is a chemical discipline that uses mathematics, statistics and formal logic to design or select optimal experimental procedures, provide maximum relevant chemical information by analysing chemical data and obtain knowledge about chemical systems [29]. Due to the development of machine learning, nowadays some machine learning methods (e.g., support vector machine and artificial neural network) are applied in chemometrics as well, in order to enrich the chemometric methods. Most of the chemometric methods are multivariate because they can obtain more information by considering multiple variables than that is obtained by considering each variable individually, which may omit the correlation between variables.

In terms of the NIRS domain, chemometrics applies mathematical and statistical approaches for multivariate NIR spectral data analysis to filter information that correlates to a specific property. Specifically, useful information is extracted from multivariate NIR spectral data, and undesired information (e.g., interferences or noise) is removed. The analysis of NIRS is divided into two types: quantitative analysis and qualitative analysis. NIRS qualitative analysis is also named NIRS identification or NIRS pattern recognition, which aims to classify or to cluster the spectral data into classes by comparing a sample spectrum to reference spectra of known materials [2]. NIRS quantification aims to build a mathematical model to predict the property of the sample. This mathematical model named the NIR model or calibration model as well, which is constructed to represent the correlation between measured NIR spectral data and reference data. Accurately, this model established by informative spectral data extracted by chemometric methods describes how the measured multivariate spectral features (data points/variables) are correlated to properties of the analyte (e.g., the water concentration of sample). The validation of that model is done by assessing how close

the predictive values by a model to the value of reference data. This PhD thesis focuses on the quantitative NIRS analysis (QNIRSA) of single rice so that only details about the QNIRSA will be illustrated in next sub-section 2.3.2.

2.3.2 Quantitative Near-Infrared Spectroscopy Analysis (QNIRSA)

In the past decades, in massive practical applications, biological scientists have already found relationship empirically existing between the NIR spectrum and constituents of samples [13]. The QNIRSA aims to construct a mathematical model by chemometric methods to quantify that relationship. However, two main problems perplex the QNIRSA:

1. Physical noise exists in the spectrum. In order to improve the predictive ability and robustness of the model, it is significant to select appropriate reference method and laboratory data because the error of reference data will be accounted into the NIR model prediction. However, it is challenging to obtain a useful reference data set because that data set needs not only small errors but also a homogeneous distribution covering the range of sample property. When it comes to single seed sample, some experts have already indicated that the determining and accounting for the error of reference data is a hard task [13]. One of the problems is the sample-to-sample difference, including size, shape and colour of the single seed. Besides, many reference methods are destructive for samples so that there is no chance to repeat consistent experiments to obtain sufficient data. Thus, physical noise existing in the spectrum is unavoidable.
2. Amount of variables are too large. The developing NIR device produces more and more independent variables in a single spectrum. As a result, another big problem is, the quantity of variables is usually much more than the number of samples, which easily gives rise to overfitting during the training. Additionally, a large amount of variables also causes a severe multicollinearity issue during the linear regression.

It is impossible to solve these problems only in one-step. Therefore, quantitative NIR analysis consists of two coherent stages:

Stage 1 **Data collection**. In this stage, spectral dataset and its corresponding reference dataset are collected in the NIR scanning and laboratory experiments procedures, respectively.

Stage 2 **Data processing**. This stage aims to process the spectral data systematically and finally construct a validated model, which can perform well in prediction.

There are five steps in stage 2:

Step 1: Dataset Partition. In section 2.2.3, it has been explained why the NIR spectral data is multivariate involving complex matrices. In this circumstance, experts have already indicated the difficulty that reproducing the composition variability of real

samples relying on optimised experimental designs [30]. It is hard to extract a representative training set to construct a model and reasonable validation set to assess model from the real natural samples in NIR cases [31], [32]. Therefore, the sampling procedure is supposed to solve the training set, and the validation set partitioning problem.

According to the review on the sampling methods, three sampling approaches are often employed in NIRS applications. They are random sampling, Kennard–Stone (KS) algorithm and sample set partitioning based on joint x–y distances (SPXY) respectively. Random sampling, like its literal meaning, divides the data set randomly into a training set and a validation set. This method is straightforward to implement, and it follows the statistical distribution of the whole data set. The drawback of this method is that it cannot ensure the samples on the boundaries of the entire set are inside the selected data set, which means this method does not guarantee the training set is representative [33]. The strategy of KS algorithm is that it uniformly covers the multi-dimensional space by maximising the Euclidean distances between the spectra vectors (x) (rows in the matrix X) of the selected samples [34]. This method has been used in some papers [32], [33], [34], but the case-by-case study for KS is strongly recommended by experts [33]. Dantas Filho et al. [35] published a paper and argued that a consideration of the dependent variables (y) (the information in matrix Y) deserved to add to the KS algorithm. His paper presents that the joint x-y consideration improves the predictive model performance compared with the model performance resulted by the classic KS algorithm. However, the data set partitioning in that paper was made on the calibration set, rather than on the entire data set. Galvao et al. [36] made a significant contribution for improving the KS algorithm and named their proposed solution sample set partitioning based on joint x–y distances (SPXY). The SPXY considers the variability in both x and y dimensions and encompasses both x- and y-differences in the calculation of inter-sample distances [36].

Step 2: Pre-processing. The sample-to-sample variations mainly caused by physical characteristics such as sample size, sample density and sample morphology, lead to light scattering effects that influence the measured NIR spectra and result in baseline shifts and scaling variations which are harmful to subsequent procedures [5]. Therefore, the pre-processing procedure is regarded as the first step of spectral data processing after sampling and gives the pre-treatment to NIR spectral data before variable selection and calibration. The aim of pre-processing is to remove the undesired physical phenomena in the spectra in order to facilitate the subsequent procedures [16]. Besides, pre-processing also helps to reduce the impact of kernel morphological characteristics, positioning, and orientation hiding in the spectra of single kernel sample [13]. For example, figure 2.4 is the pre-processed spectra of the original spectra (shown in figure 2.2) treated by standard normal variate (SNV), displaying the effect that pre-processing has made on the variation between samples for different wavelength regions. Compared with the original spectra in figure 2.2, variations at most of the wavelengths of the pre-processed spectra are much smaller because background offsets and slopes are removed.

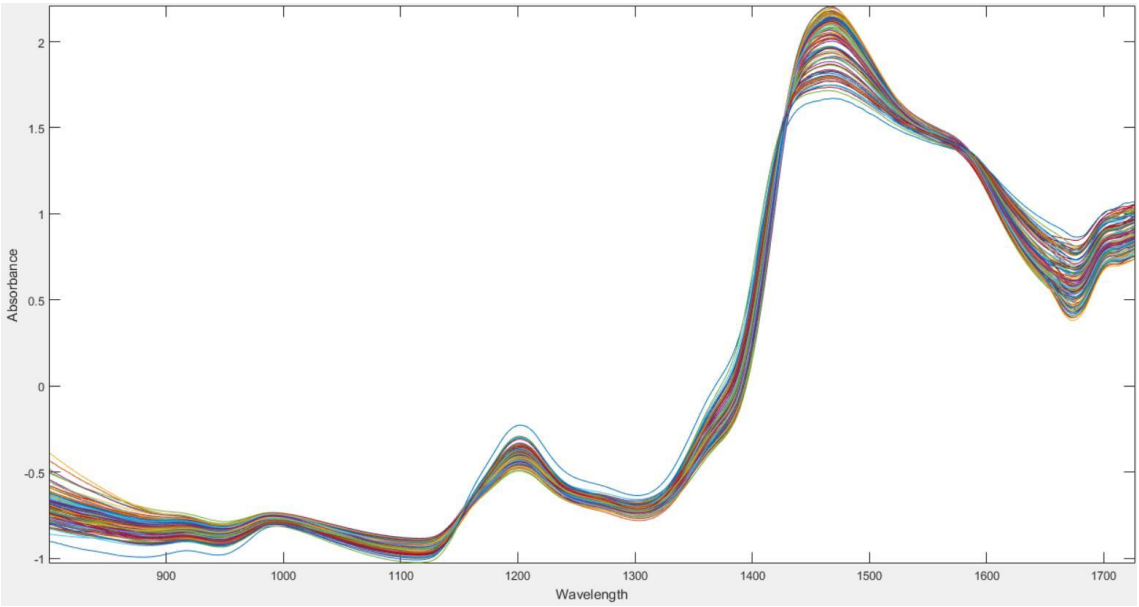


Figure 2.4: The pre-processed spectra of the original spectra treated by SNV.

The most optimal pre-processing method depends on not only the sample physical characteristics to be measured, or the sample property to be determined, but also the environment and instrument configuration [13]. Thus, empirically the most frequently used and standard pre-processing methods have been summarised by Rinnan et al. [16] in 2009. So far, that paper emphasising on the theoretical aspects of the pre-processing technique is still the only review paper about the pre-processing methods. Although that paper has introduced the principle and theory of most of the pre-processing methods clearly, there is no discussion on the effects that pre-processing methods impact on the subsequent procedures. Many other papers have not provided a comparative work for pre-processing as well, because the emphasis is often on the variable selection or calibration. It is commonly accepted that pre-processing is a necessary procedure because it can have a significant impact on the predictive model performance, but the optimal pretreatment is characterized by the lowest statistical errors obtained in the calibration step. Thus, the comparative study for pre-processing methods has to be finished until the end of calibration. Experts do not recommend combining several pre-processing methods because the risk of removing the variable information exists if too severe pre-processing is applied [16].

Pre-processing methods are divided into two categories: scatter-correction methods and spectral derivatives [16]. Scatter-correction methods are designed to reduce the physical variability between samples due to scatter and adjust baseline shifts between samples. Multiplicative scatter correction (MSC) and standard normal variate (SNV) are the two most popular methods in the category of scatter-correction methods. MSC was developed by Martens et al. [37] at first, then improved by Geladi et al. [38], expanded by Martens and Stark [39], Martens et al. [40], Thennadil and Martin [41], and Windig et al. [42]. Concepts behind SNV are as similar as MSC, except for that standard reference signal for SNV is not required [16], [43]. Spectral derivatives methods can

remove both additive and multiplicative effects (e.g., resolving nearby peaks) in the spectra. Two main spectral derivatives methods are Norris-Williams (NW) derivatives and Savitzky-Golay (SG) polynomial derivative filters.

The first derivative removes only the baseline, while the second derivative removes both baseline and linear trend. The NW derivatives method is a primary method developed to avoid noise inflation in differences, [44]. A moving window is used by the NW derivatives method for smoothing the data. Due to the span of the window, several points on both sides of data cannot be smoothened. In terms of the SG polynomial derivative filters method, it is popularized for the numerical derivation of a vector [45]. In order to find the derivative at the centre point *i*, a polynomial is fitted in a symmetric window on the raw data. Since the polynomial has been calculated, the derivative of any order of this function can be computed. This iterates for all points sequentially. However, the SG polynomial derivative filters method also has the problem that some points on both sides of data have to be neglected. For NW derivative method, the number of missing points equals the number of points used for smoothing plus the size of the gap minus one. For SG polynomial derivative filters method, the number of missing points equals the number of points used for smoothing minus one.

In contrast, the NW derivative method loses more points than the SG polynomial derivative filters method. Experts indicated that this issue is not important if there are more than 500 points in the spectra [16]. Nevertheless, for the case that the spectra have less than 500 points, there is no perfect solution for this issue yet [46].

Step 3: Variable Selection. The spectrum of each sample consists of independent variables at hundreds to thousands of wavelengths, which sums up in a large number of NIR spectral information that has to be processed. Thus the quantity of independent variables is usually much more than the number of samples, which easily gives rise to overfitting and unreliable prediction of the dependent variable during the data training [25]. In order to improve the predictive ability of the model, the number of independent variables should be reduced before multivariate calibration. Another challenge is the multi-collinearity among spectra. The multi-collinearity means the variables at wavelengths are correlated and not independent of each other [5], which can render bad prediction with linear regression model used in multivariate calibration. Besides the reduction of the number of variables, representative variables should be selected as well. These two issues are the reasons why variable selection procedure is needed before multivariable calibration. In a word, variable selection, based on the principle that selecting a small number of independent variables from the original set of variables will allow easier interpretation, is a crucial procedure to remove the non-informative information hiding inside the spectra, in order to obtain better predictions with simpler models in the multivariate calibration.

The early focus of variable selection has been given to support the multiple linear regressions (MLR) because the MLR is easier to interpret. The successive projections algorithm (SPA) is a variable selection method developed to solve collinearity problems

in multiple linear regression (MLR), which presents the advantage of finding a small representative set of spectral variables with a minimum level of collinearity [47]. However, the SPA may lead to low signal-noise ratio or be insufficient for multivariate calibration, which can affect the precision of the model prediction [17]. Another popular variable selection method is the uninformative variable elimination (UVE) which addresses an automatic approach to remove the uninformative variables from the data set based on either high noise or low detector response [48]. Because the UVE employs the regression coefficient vector, the partial least squares regression (PLSR) is often used with it (UVE-PLS). The main advantage of using UVE of modelling is that it can avoid model over-fitting and usually improve its predictive ability [17]. However, sometimes latent variables are still required a further elimination before modelling if the number of variables selected by UVE is still substantial. Thus, the other two papers reported the performance of combing UVE and Monte Carlo cross-validation (MCCV) [49], and SPA [50] respectively. By combining the UVE with MCCV, better prediction results have been displayed compared with only UVE [49]. On the other hand, fewer variables need to be sought and selected but with better prediction performance when UVE is applied after SPA, specified by Ye et al. [50].

Some experts consider the variable selection as an optimisation process, and they made efforts to find appropriate global optimisation methods. One of the typical optimisation methods is simulated annealing (SA), which is a probabilistic global optimisation technique that can traverse local optimums and to find the optimal global solution, firstly displayed by Kirkpatrick et al. [51], then employed for variable selection by Swierenga et al. [52], [53]. Details about SA will be introduced in section 3.4.3. Another method for the optimisation problem is the genetic algorithm (GA), which is a popular heuristic optimisation technique that employs a probabilistic, non-local search process in many fields. The GA is often used with PLS (GA-PLS) for NIR analysis, which combines both advantages of GA and PLS. For example, GA-PLS exhibits superiority over other variable selection methods in the PLS calibration, because it has been reported that there is no loss of prediction capacity by using a genetic algorithm [54], [55], [56]. The disadvantages of GA compared with other variable selection methods are time-consuming, and too many parameters affecting results need to be configured by the user. Methods like GA or SA adopts the selection strategy that selects the most useful wavelength variables possible but regardless of location.

Another strategy is to select continuous variables in a specific region of the original variable range. The representative method adopts the latter strategy is interval PLS (iPLS). The iPLS method partitions the original variable range into sub-intervals, then build the PLS model for every sub-interval and work out which sub-interval is the best region based on criteria [57]. This method may not provide a significantly better PLS model than full-spectrum PLS model, but it gives an overview of the spectral data and displays interesting spectral areas which may be selected [58]. Based on iPLS, other two methods backward interval partial least squares (BiPLS) and forward interval partial least squares (FiPLS) has been developed to improve the models selected by iPLS. The

first step is as same as what iPLS does, that is, iPLS divides the whole variable range into sub-intervals. However, BiPLS calculates the PLS models with each sub-interval left out in a sequence [59]. For example, if iPLS selects 30 sub-intervals, then each PLS model is based on 29 sub-intervals leaving out one sub-interval at a time by BiPLS. The first omitted sub-interval gives the poorest performing model concerning criteria. This procedure is continued until no more sub-intervals, which can be kicked out to improve the performance. On the contrary, FiPLS establishes PLS models by successively improving intervals with respect to criteria [59]. For instance, as for 30 intervals selected by iPLS, FiPLS produces the first model on the sub-interval, which has the best performance assessed by criteria. Afterwards, FiPLS adds one remaining sub-interval to the first model at a time. The entered sub-interval in the model is the one that when the PLS model based on combination of this sub-interval and those sub-intervals which has been entered before, gives the best performance. This procedure is continued until no more sub-intervals, which can be added to improve the performance.

Step 4: Multivariate Calibration. Multivariate Calibration is one of the most important techniques to ensure the quality of both quantitative and qualitative analyses. It aims to construct a mathematical regression model correlating the measurements of spectra to interesting properties of samples [60]. Mathematically, this model is developed through regressions of the measured NIR spectral data against the reference data values of analyte properties determined by reference analytical methods. For example, the multivariate calibration model for grain is the regression of the measured NIR spectra of grain against the reference data of grain properties such as protein content or water concentration. Then, the model is employed to predict the properties of samples, which are out of the calibration/training set. The reliability and accuracy of the predicted results rely on the quality of models. Therefore, multivariate calibration methods are significant for the establishment of the model.

Main linear calibration methods are multiple linear regression (MLR), principal component regression (PCR) and partial least squares regression (PLSR). The MLR is a linear statistical technique to predict one response (dependent variable) by two or more predictors (independent variables). However, a phenomenon named multi-collinearity may occur when the MLR is used. The multi-collinearity is that one predictor variable in an MLR model can be linearly predicted from the others with a substantial degree of accuracy [61]. In this manner, the coefficient estimates of the MLR may change erratically in response to small changes in the model or the data. The PCR is a solution to solve the multi-collinearity problem to some extent, based on the principal component analysis (PCA). It considers regression of the response on a set of predictors based on a standard linear regression model but uses PCA for estimating the unknown regression coefficients in the model. The PCR will select some principle components with high-variance in the regression step, which performs well in dimension reduction by decreasing the adequate number of parameters characterising the underlying model [62]. An appropriate number of principal components can lead to an efficient prediction of the response based on the proposed model. The PCR can be regarded as a

combination of the PCA and the MLR. It may not only deal with the collinearity but also handle the case that the number of variables is more than the number of samples. The PLSR expands the PCR to some extent. The PLSR selected principal components as well as what PCA does, but it considers the influence of reference value during the regression and employs partial least squares (PLS) rather than the MLR for regression [63]. Besides, cross-validation is often used to support the PLSR and the PCR for determining the optimal number of principal components [64].

Step 5: Model Validation. The validation set divided in step 1 is used to validate the initial model constructed in step 4. Root mean square error (RMSE) and the coefficient of determination (R^2) are the two most commonly used statistics to assess the quality of the model. These two statistics will be discussed in chapter 3.

2.4 Summary

Compared with other analytical methods (e.g., wet chemical method) or other spectroscopic techniques (e.g., Raman), the NIRS technology has many advantages such as non-destructive to samples and waste-free. NIRS spectral data is multivariate, and usually, the number of variables is more than the number of samples. QNIRSA, which aims to quantify the correlation between the measurement of the spectrum and interesting property of the sample, consists of some coherent processes mainly including dataset partition, pre-processing, variable selection and multivariate calibration. Chemometric methods related to various domains are applied to support one or more processes in the QNIRSA.

3 FOUNDATIONS AND RELATED WORK

3.1 Overview

This chapter will specify the theory and principle of chemometric methods and criteria used in this PhD thesis. They are dataset partition methods in section 3.2, pre-processing methods in section 3.3, and variable selection methods in section 3.4, multivariate calibration methods in section 3.5 and statistic criteria in section 3.6, respectively. Section 3.7 will present the related work about applications of chemometric methods, which is related to the first research problem in section 1.2. MATLAB codes for these methods can be found in Appendix 3.

3.2 Dataset Partition Methods

3.2.1 Kennard–Stone Algorithm

The Kennard–Stone (KS) algorithm adopts a stepwise procedure that new selections are taken in regions of the space far from the samples already selected, in order to give a uniform distribution along with the spectral data space [36]. Euclidean distances $d_x(p, q)$ between the x-vectors (rows in the matrix X) of two single spectrum p and q ($p, q \in [1, N]$, N is the quantity of samples), is

$$\mathrm{d}_x(\mathrm{p},\mathrm{q}) = \sqrt{\sum_{j=1}^{j}\left[x_p(j) - x_q(j)\right]^2} \qquad \text{(Eq.3-1)}$$

In Eq.3-1, j is the number of wavelengths, while $x_p(j)$ and $x_q(j)$ are the values at wavelength j for spectra p and q, respectively. Steps of KS algorithm are:

1. Calculate the Euclidean distance between every pair of single spectrums in matrix X.
2. Select two single spectrums p and q, which have the largest Euclidean distance $d_x(p, q)$ among the entire data set, as the first pair.
3. Calculate the minimum Euclidean distance between every remaining single spectrum and the first pair. For example, the Euclidean distance between a single spectrum C and the first pair (A, B), is 100 (C, A) and 200 (C, B) respectively. The minimum Euclidean distance between the single spectrum C and the first pair is 100 (C, A).
4. Select the third single spectrum, which has the largest minimum Euclidean distance concerning the first pair.

5. Select the next single spectrum, which has the largest minimum Euclidean distance concerning those already selected single spectrums. This step is iterated until the expected quantity of samples is selected.

3.2.2 Sample Set Partitioning Based on Joint X–Y Distances

The Sample Set Partitioning Based on Joint X–Y Distances (SPXY) adopts calculation as similar as the Eq.3-1, for the Euclidean distance $d_y\,(p,\,q)$ between the dependent variable y in the matrix Y, which is corresponding to the samples p and q (p, q $\in [1, N]$. N is the number of samples). The equation [36] is

$$\mathrm{d}_y(\mathrm{p},\mathrm{q}) = \sqrt{\left(y_p - y_q\right)^2} \qquad \text{(Eq.3-2)}$$

Then equal importance is assigned to the distribution of the samples in the x and y spaces by a division that both y- and x-distances $d_y\,(p,\,q)$ and $d_x\,(p,\,q)$ are divided by their maximum value in the data set [36]. The formula for xy-distance $d_{xy}\,(p,\,q)$ is

$$\mathrm{d}_{xy}(\mathrm{p},\mathrm{q}) = \frac{d_x(p,q)}{max d_x(p,q)} + \frac{d_y(p,q)}{max d_y(p,q)} \qquad \text{(Eq.3-3)}$$

Similarly, steps of SPXY are:

1. Calculate the xy-distance $d_{xy}\,(p,\,q)$ between every pair of single spectrums.
2. Select two single spectrums p and q, which have the largest xy-distance $d_{xy}\,(p,\,q)$ among the entire dataset, as the first.
3. Calculate the minimum xy-distance between every remaining single spectrum and the first pair. For example, the xy-distance between a single spectrum C and the first pair (A, B), is 100 (C, A) and 200 (C, B) respectively. The minimum xy-distance between the single spectrum C and the first pair is 100 (C, A).
4. Select the third single spectrum, which has the largest minimum xy-distance concerning the first pair.
5. Select the next single spectrum, which has the largest minimum xy-distance concerning those already selected single spectrums. This step is iterated until the expected quantity of samples is selected.

3.3 Pre-processing Methods

3.3.1 Multiplicative Scatter Correction

The Multiplicative Scatter Correction (MSC) assumes that unwanted scatter effect can be removed from the spectral data matrix X before constructing a model. The first step of MSC is to estimate the correction coefficients, and then spectra are corrected by the estimation. Eq.3-4 and Eq.3-5 show the formulas of MSC [16].

$$X = b_0 + b_{ref,1} \times X_{ref} + e \qquad \text{(Eq.3-4)}$$

$$X_{corr} = \frac{X - b_0}{b_{ref,1}} = X_{ref} + \frac{e}{b_{ref,1}} \qquad \text{(Eq.3-5)}$$

X is the original spectra; X_{ref} is a reference spectrum (the average spectrum of the training set) used for pre-processing of the entire dataset; X_{corr} is the corrected spectra; e is the un-modelled part of X; b_0 and $b_{ref,1}$ are scalar parameters which are different for every sample. The two scalar parameters can be calculated by the least squares regression fit between the reference spectrum and the spectrum corresponding to the sample.

3.3.2 Standard Normal Variate

The formula of the Standard Normal Variate (SNV) [65] is

$$X_{corr} = \frac{X - b_{mean}}{b_{std}} \qquad \text{(Eq. 3-6)}$$

In Eq.3-6, X is the original spectra while the X_{corr} is the corrected spectrum; b_{mean} is the average value of the sample spectrum, and b_{std} is the standard deviation of the sample spectrum.

3.3.3 Savitzky-Golay Polynomial Derivative Filters

The Savitzky-Golay Polynomial Derivative Filters (SG) utilises a polynomial fitting technique in the symmetric window on the raw data, in order to find the derivative of the point. Since the polynomial has been calculated, the derivative of any order of this function can be computed. This iterates for all points sequentially. Both the number of points used to calculate the polynomial (window size) and the degree of the fitted polynomial are decisions that need to be made. The highest derivative that can be determined depends on the degree of the polynomial used during the fitting (i.e., a second-order polynomial can be used to estimate up to the second-order derivative). The formulas of this method are very sophisticated [45], [46], so the main steps of SG are summarised:

1. Set up a window with a fixed-length including some points (e.g., a window of 7 points).
2. Set up a polynomial fit and choose the order of spectral derivative. The order of polynomial fit must be not less than the order of spectral derivative.
3. Move the window from left to right on the spectrum. The point in the middle of the window is corrected by the polynomial fit.
4. The techniques neglect some points at each end of the spectrum. The number of points lost equals the number of points used for smoothing minus one. If the spectral vector is long (i.e. more than 500 points), this issue is not important [16].

The spectral data used in this thesis contains 936 points so that this issue will not affect the analysis result seriously.

3.4 Variable Selection Methods

3.4.1 Successive Projections Algorithm

The Successive Projections Algorithm (SPA) is a variable selection method developed to reduce collinearity issues when the MLR is used. Araújo et al. reported that the SPA-MLR model has better prediction ability than the full-spectrum PLS/PCR models for some applications of chemical compounds [47]. According to that paper, the SPA adopts the forward variable selection strategy for multivariate calibration. The SPA uses simple projection operations in a vector space to obtain subsets of variables with minimal collinearity, and its principle is that the new variable selected is the one variable among all of the remaining variables that has the maximum projection value in the orthogonal subspace of the previously selected variable [17], [47]. Explicitly, first, set the maximum number of variables k to be selected before a start vector is chosen in the space of n-dimensions (where n is the number of original variables). Subsequently, in an orthogonal sub-space, the vector of higher projection is selected and becomes the new starting vector. The choice of the orthogonal sub-space at iteration is made in order to select only the non-collinear variables. The optimal initial variable and number of variables can be determined by criteria such as the smallest root mean square error of validation (RMSEV) with the validation set. In summary, steps of SPA are:

1. Set up iterations. In every iteration, the start column of variables is different, which ranges from the first column of variables to the last. With a given constant k, k columns of variables are selected in every iteration based on descending order of the projection operations. As a result, original dataset is divided into some candidate subsets according to the difference of the start column of variables.
2. Evaluate those candidate subsets based on criteria assessing the MLR regression. Find the best candidate subset.
3. Use F-test to get F-value and then use F-value to compute the threshold of criteria. Eliminate the variables in the best candidate subset, which value of criteria is not significantly larger than the threshold.

3.4.2 Uninformative Variable Elimination

The 3.4.2 Uninformative Variable Elimination (UVE) method was firstly investigated by Centner et al. [48], addressing an automatic approach to remove the uninformative variables from the dataset, based on either high noise or low detector response. The principle of the UVE is that random variables (e.g., an artificial random variable matrix X with minimal amplitude (e.g., 10^{-10})) are manually added to the training set as a reference so that those variables, which play a less important role than the random

variables, will be eliminated. Specialized term "stability" is used to measure the contribution of each variable during calibration. This term is defined as

$$s_j = \frac{mean(b)}{std(b)}, \;\; b = \left[b_1, \dots, b_j\right], \; j = 1,2, \dots, n \qquad \text{(Eq. 3-7)}$$

In Eq.3.7, *mean(b)* and *std(b)* are the mean and standard deviation of the *jth b* of variable *j*. *b* is the regression coefficient vector obtained from each iteration of the cross validation. The logic is that, the larger the stability, the more important the corresponding variable is. Therefore, those variables whose stability is less than a threshold should be eliminated as uninformative variables. The threshold named 'cut-off' is defined as the maximum of absolute value among the random variables as the Eq.3-8 (*k* is an arbitrary value such as 1 [48]).

$$\text{cutoff} = k \times max\left(abs(s_{noise})\right) \qquad \text{(Eq. 3-8)}$$

Steps of UVE are:

1. Create a noise matrix, which has the same size as the original spectral matrix.
2. Insert the noise matrix after the original spectral matrix to make a new matrix as input.
3. Use leave-one-out cross validation to calculate the regression coefficient vectors of each iteration.
4. Calculate the "stability" of the original spectral matrix and noise matrix separately based on Eq.3-7.
5. Calculate the cutoff based on Eq.3-8.
6. Eliminate uninformative variables if the stability of them is not larger than the cutoff.

3.4.3 Simulated Annealing

The Simulated Annealing (SA) is a probabilistic global optimisation technique referring to physical annealing process of solids firstly described by Kirkpatrick et al. [51]. Swierenga et al. introduced the SA to variable selection and illustrated the model robustness had been enhanced [52]. According to the theory of the SA, it assumes that an initial solution for a problem should be iteratively modified subject to some control parameter *T*. When the parameter *T* is reduced from the maximum value, the convergence criterion will be challenging to satisfy. At some point, if *T* is lowered sufficiently, the solution will be frozen at a local optimum. In order to avoid this issue, the SA should move slowly through the solution space by accepting non-improving moves with a certain probability that decreases as the algorithm progresses. In terms of variable selection for NIR spectral data, the SA selects *k* variables from the whole variable range to produce the calibration model and computes an error value presenting the predictive ability of the model. In order to find the optimal value *k*, which minimises the error value at all circumstances of parameter *T*, iterations of SA need to be executed.

The criterion is a Boltzman's probability distribution (Metropolis criterion) [66] as a function of parameter *T* is

$$\begin{cases} p(\Delta F) = exp\left(\frac{-\Delta F}{T}\right) \\ \Delta F = F(v') - F(v_i) \end{cases} \qquad \text{(Eq. 3-9)}$$

In Eq.3-9, F is the objective function, and ΔF is the increment of F, v_i is the current values, and v' is a randomly generated new solution in the neighbourhood of v_i. Steps of SA are:

1. Configure initial parameters including initial error, cooling ratio, initial temperature, temperature to stop, cooling schedule function, generator range and Markov chain.
2. Set up the SA iteration. Iteration starts from initial values. Result of the current iteration will be compared with the previous one. If the current result is smaller than the previous result, keep it to be compared with the next one. The number of comparisons is determined by the Markov chain. However, if the current result is larger, an accepted probability based on Eq. 3-9 gives a chance to keep it as well.
3. After all iterations, the optimal solution is generated.
4. Try different initial parameters and repeat step 1 to 3.

3.4.4 Genetic Algorithm

The genetic algorithm (GA) is a popular heuristic optimisation technique that employs a probabilistic, non-local search process in many fields. GA manipulates binary strings called chromosomes that contain genes that encode experimental factors or variables. As for NIR analysis, GA is used to select suitable variables to build calibration models [67]. Many applications regard GA as a comparative method on many spectral data sets and have been shown to provide better results than full-spectrum approaches [25], [55], [56], [58], [67], [68]. There are four steps to apply to GA. Firstly, variables are represented by binary code in a vector (chromosome) with one cell for each variable. The first chromosome generates some chromosomes randomly to form an initial population. Secondly, A PLS model is made for every chromosome, and all models are evaluated by cross validation. A criterion named fitness value, which reveals the quality of the model, is computed from the cross validation for guiding the GA to the global optimum. Thirdly, a new population is produced as the next generation made up by recombination of the original chromosomes. The chromosomes with a high fitness value have a higher probability to reproduce than a chromosome with the target to improve the overall fitness of the population. Lastly, configure the mutation which is an inversion of a gene in a chromosome, in order to overcome some problems that may occur. One of the problems is, if a variable is not selected from any of the original chromosomes, it will never be selected in the next generations. Iterations of the four steps will be stopped when a certain percentage of the chromosomes are identical. In summary, steps of GA are:

1. Configure initial parameters including the number of chromosomes, maximum generations, probability of single-point crossover and probability of mutation.
2. Compute the first generation. Firstly, Convert spectral data into binary code. 0 or 1 is used to represent every variable. 1 means this variable is selected. Both 0 and 1 have the same probability of occurrence for each variable. As a result, every chromosome has its corresponding genes, as a set of indices of selected variables. Calculate the fitness value of all chromosomes based on PLSR. Find the chromosome that has the best fitness. Compute average fitness for the first generation as well.
3. Set up iterations as evolutions of population. In every evolution, two chromosomes are selected as parents. The chromosome with a higher fitness value has a higher probability of reproducing. Then, execute the single-point crossover for the selected two chromosomes. A mutation is happened after crossover according to its probability. The mutation point is random. Afterwards, a new generation of the population is established. Find the chromosome that has the best fitness and compute average fitness for this generation as well.
4. Find the chromosome that has the best fitness among all generations. The average fitness of generations displays the trade of evolution.
5. Try different initial parameters and repeat step 1 to 4.

3.4.5 Interval Partial Least Squares

Nørgaard et al. designed a method to select variables which maintain continuity in the original variable domain [57]. This method is named interval partial least squares (iPLS), which divides the original range of variables into non-overlapping sub-intervals. Then PLSR is used to build the PLS model for every sub-interval and get relevant values of criteria (usually the RMSEP seen in section 3.6). By comparing the criteria, an overall picture of the relevant information of different sub-sections/intervals is produced. Important spectral regions, which have better performance of criteria, should be considered into calibration. Other regions with less useful information should be removed. Generally, full-spectrum PLSR model is used as a global model as a reference for the local model. In terms of local PLSR models generated from sub-intervals, they must have the same dimensions as each other for a fair comparison. Because the larger the spectral sub-interval, the higher the number of substances that are likely to absorb/interfere. A simple improvement for the iPLS is to revise the width of the best sub-interval. For example, when the best sub-interval is determined by the iPLS, this sub-interval width is changed one variable at a time on both sides and evaluated by same criteria provided by the application of the PLS regression to the sub-interval. In summary, the steps of iPLS are:

1. Set a range of the number of sub-intervals.
2. Step up iteration. Every iteration adopts one case of the number of sub-intervals from minimum to maximum. Seek the best spectral region in every iteration.

3. Compare the best spectral regions in different iterations and find the global optimum.
4. Some variables next to both sides of the global optimal spectral region may be possible to be added. Try to add those variables into that region one at a time. Then establish PLSR models to see if those variables can improve model performance compared with the original best sub-interval.

3.4.6 Backward Interval Partial Least Squares

The Backward Interval Partial Least Squares (BiPLS) firstly employs the iPLS to divide the original variable range into sub-intervals. Secondly, leave one sub-interval out at a time and use the remaining sub-intervals to build PLS models. For example, if there are 30 sub-intervals, a sub-interval is left at a time and then use 29 sub-intervals to establish PLS model. The first sub-interval to be kicked out should have the most inferior performance assessed by criteria. Then the second model is kicked out by the same principle. Iteration will stop when no more sub-intervals to be kicked out to improve the model performance.

3.4.7 Forward Interval Partial Least Squares

The Forward Interval Partial Least Squares (FiPLS) is contrary to the BiPLS. If there are 30 sub-intervals, the FiPLS will build one PLS model for every sub-interval. The first sub-interval to be selected should have the best performance assessed by criteria. Then, the selected model combines the remaining 29 sub-intervals one at a time. Next model based on the next sub-interval with the early-selected sub-interval should give the best performance as well. This repeated selection will not stop until no more sub-intervals can be selected to improve model performance.

3.4.8 Principal Component Analysis

The goals of the Principal Component Analysis (PCA) [62] are:

1. To extract the most crucial information from the original data set.
2. To compress the size and dimension of the original data set.
3. To analyse the structure of the observations and variables (this is usually for classification or discrimination analysis).

The two main issues the PCA may solve:

1. Multi-collinearity between independent variables.
2. The number of samples is less than the number of variables which may lead to over-fitting in calibration.

Briefly, the PCA eliminates those principal components with low-variance to avoid the first issue. Regarding the second problem, the PCA projects the variables into an M-dimensional subspace where *M* is much smaller than the number of variables, in most cases, smaller than the number of samples as well. These *M* projections are used as predictors to fit a linear regression model by least squares.

Principles about the PCA refers to Geladi and Kowalski's paper [63]. Assume an original data set is donated *X* comprising *n* observations described by *p* variables (the size of matrix *X: n*p*).

$$X_{original} = \begin{bmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{bmatrix} = (x_1, \cdots x_p) \qquad \text{(Eq.3-10)}$$

The PCA computes new variables called principal components, which are obtained as linear combinations of the original variables. Every principal component *F* is a kind of linear combination of the original variables. The maximum number of principal components equal to the number of variables.

$$\begin{cases} F_1 = a_{11}x_1 + a_{12}x_2 + \cdots + a_{1p}x_p \\ \vdots \\ F_p = a_{p1}x_1 + a_{12}x_2 + \cdots + a_{pp}x_p \end{cases} \qquad \text{(Eq.3-11)}$$

The matrix notation of Eq.3-11 is

$$\begin{cases} \mathrm{F} = \mathrm{Q}X_{transformed} \\ \mathrm{F} = \begin{bmatrix} F_1 \\ . \\ . \\ . \\ F_p \end{bmatrix} \\ Q = \begin{bmatrix} a_{11} \ldots a_{1p} \\ . \\ . \\ . \\ a_{p1} \ldots a_{jp} \end{bmatrix} \\ X_{transformaed} = \begin{bmatrix} x_1 \\ . \\ . \\ . \\ x_p \end{bmatrix} \end{cases} \qquad \text{(Eq.3-12)}$$

In Eq.3-12, *Q* (coefficient matrix of the principal component) is the coefficients of the linear combinations. Eq.3-12 should satisfy two conditions:

The variance of former principal components must be bigger than the latter one: $V_{F_1} > V_{F_2} > \cdots > V_{F_p}$.

1. Q should be orthogonal matrix: $QQ^T = I$. I is a unit matrix.

Calculate the covariance matrix of principal component:

$$V_F = V_{QX_{transformed}} = \left(QX_{transformed}\right) \cdot \left(QX_{transformed}\right)^T =$$

$$QX_{transformed}{X_{transformed}}^T Q^T = \Lambda = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_p \end{bmatrix} \quad \text{(Eq.3-13)}$$

Practically, the original data set will be centred at first. In this case, the covariance matrix V of the original data set equals to its correlation matrix R, notated as

$$\mathrm{R} = \mathrm{V} = \mathrm{X}X^T \quad \text{(Eq.3-14)}$$

Combining Eq.3-13 and Eq.3-14, we can obtain

$$\mathrm{Q}RQ^T = \Lambda \rightarrow \mathrm{R}Q^T = Q^T\Lambda \quad \text{(Eq.3-15)}$$

Expand the Eq.3-15 to

$$\begin{bmatrix} r_{11} & \cdots & r_{1p} \\ \vdots & \ddots & \vdots \\ r_{p1} & \cdots & r_{pp} \end{bmatrix} \cdot \begin{bmatrix} a_{11} & \cdots & a_{p1} \\ \vdots & \ddots & \vdots \\ a_{1p} & \cdots & a_{pp} \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & a_{p1} \\ \vdots & \ddots & \vdots \\ a_{1p} & \cdots & a_{pp} \end{bmatrix} \cdot \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_p \end{bmatrix} \quad \text{(Eq.3-16)}$$

According to the properties of the matrix, take the calculation involving the first column of Q^T at both sides of the equation for instance; we can get a set of homogeneous equations:

$$\begin{cases} (r_{11} - \lambda_1)a_{11} + r_{12}a_{12} + \cdots + r_{1p}a_{1p} = 0 \\ r_{21}a_{11} + (r_{22} - \lambda_1)a_{12} + \cdots + r_{2p}a_{1p} = 0 \\ \vdots \\ r_{p1}a_{11} + r_{p2}a_{12} + \cdots + \left(r_{pp} - \lambda_1\right)a_{1p} = 0 \end{cases} \quad \text{(Eq.3-17)}$$

In order to work out the homogeneous equations, the coefficient matrix should be zero.

$$\begin{bmatrix} r_{11} - \lambda_1 & r_{12} & \cdots & r_{1p} \\ r_{21} & r_{22} - \lambda_1 & \cdots & r_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ r_{p1} & r_{p2} & \cdots & r_{pp} - \lambda_1 \end{bmatrix} = 0 \rightarrow \mathrm{R} - \lambda_1 I = 0 \quad \text{(Eq.3-18)}$$

Therefore, λ_1 is the eigenvalue while $Q_1 = \{a_{11}, a_{12}, \ldots a_{1p}\}$ is the corresponding eigenvector of λ_1. As for all columns of Q^T, we can summarise $R - \lambda_i I = 0$, and $\lambda_i$is the eigenvalue and its corresponding eigenvector is $Q_i = \{a_{i1}, a_{i2}, \ldots a_{ip}\}$. First k principal components will be selected when the quotient obtained by dividing the sum of p principal components' eigenvalues by the sum of k principal components' eigenvalues, is larger than the threshold set manually.

$$\frac{\sum_{i=1}^{k} V_{F_k}}{\sum_{i=1}^{j} V_{F_p}} = \frac{\sum_{i=1}^{k} \lambda_k}{\sum_{i=1}^{j} \lambda_p} > \text{threshold} \quad 1 < \text{k} < \text{p} \qquad \text{(Eq.3-19)}$$

The last step is to use the corresponding matrix of PCA scores *T* of the selected *k* principal components for the linear regression of the response *Y* on those selected components. ${Q_k}^T$ is the loading matrix.

$$\begin{cases} T = X_{original} * {Q_k}^T \\ Y = T * B + E \\ \hat{B} = (T^T \cdot T)^{-1} \cdot T^T \cdot Y \end{cases} \qquad \text{(Eq.3-20)}$$

In Eq.3-20, *B* is the vector of regression coefficients, and *E* is the error matrix.

3.5 Multivariate Calibration Methods

3.5.1 Multiple Linear Regression

In many practical applications, multiple predictor variables are affecting one response variable. Multiple Linear Regression (MLR) is a method to investigate the mathematical relationship between multiple predictors and one single response linearly. The MLR model with *k* predictor variables and one response variable *y* is denoted as

$$y = \beta_0 + \beta_1 x_1 + \cdots \beta_k x_k + \epsilon \qquad \text{(Eq.3-21)}$$

In Eq.3-21, ϵ is the residual, while β_k is the regression coefficient. Geometrically, an MLR model with *k* predictors and one response can be regarded as a k-dimensional surface in space [69]. The shape of this surface depends on the structure of the model. The observations are the points in space, and the surface is fitted to best approximate the observations. Thus, the general formula for *n* observations is

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots \beta_k x_{ik} + \epsilon_i \quad \text{i} = 1,2 \ldots \text{n} \qquad \text{(Eq.3-22)}$$

The matrix form of the Eq.3-22 is

$$Y = X\beta + E \quad \text{i} = 1,2 \ldots \text{n} \qquad \text{(Eq.3-23)}$$

Usually, the least squares (LS) approach is used to estimate the value of β. Then, we can obtain some equations:

$$\begin{cases} \hat{\beta} = (X^T X)^{-1} X^T Y \\ \hat{Y} = X\hat{\beta} = X(X^T X)^{-1} X^T Y = HY \\ E = Y - \hat{Y} = Y - HY \\ H = X(X^T X)^{-1} X^T \end{cases} \qquad \text{(Eq.3-24)}$$

The matrix *H* is named HAT-Matrix, which maps the vector of observed values *Y* onto the vector of fitted values $\hat{Y}$ that lie on the regression hyper-plane.

3.5.2 Principal Component Regression

The Principal Component Regression (PCR) is based on the dimension reduction technique PCA. When the vector of regression coefficients *B* is obtained according to the Eq.3-20, the PCR employs the MLR to build a regression model with respect to the Eq.3-23.

3.5.3 Partial Least Squares Regression

The most significant difference between the Partial Least Squares Regression (PLSR) and the PCR is that the PCR only considers the variable matrix *X*, while the PLSR involves the response matrix *Y* into the regression step [63]. Similarly, the PLSR makes linear transformation for the variable vectors as well but applies the same transformation for the response vector.

$$\begin{cases} X = TQ + E = \sum t_i q_i + E \\ Y = UP + F = \sum u_i p_i + F \end{cases} \quad \text{(Eq.3-25)}$$

In Eq.3-25, *T*, *Q* and *X* are as same as they defined in the PCR, while *U* and *P* are the score matrix and loading matrix of *Y*, respectively. The t_i and u_i are the principal components extracted from *X* and *Y*, respectively. In order to make t_i and u_i contain information as much as possible, two conditions should be satisfied:

1. The variance of t_i and u_i should be as large as possible: $\text{Variance}(t_i) \rightarrow max, \text{Variance}(u_i) \rightarrow max$
2. The correlation between t_i and u_i should be as large as possible: $\text{Correlation}(t_i, u_i) = \text{Covariance}(t_i, u_i) \rightarrow max$

The PLSR iterates the extraction of principal component and regression in sequential order. Specifically, the PLSR extracts the first principal component t_1 and u_1 from the linear combination of *X* and *Y*, respectively. Then the PLSR regresses *X* by t_1 and *Y* by u_1. If the regression results reach the desired accuracy, the algorithm will stop. Otherwise, residuals of first regression will be used for the extraction of the next component and regression. This iteration will not stop until the regression satisfies the required accuracy. Assume X_s and Y_s are the standardised matrixes of *X* and *Y*, respectively. In order to make the covariance between t_1 and u_1 as large as possible, based on relevant mathematical principles, we can deduce:

$$\begin{cases} X_s^T Y_s Y_s^T X_s w_1 = \theta_1^2 w_1 \\ Y_s^T X_s X_s^T Y_s c_1 = \theta_1^2 c_1 \\ t_1 = X_s w_1 \\ u_1 = Y_s c_1 \end{cases} \quad \text{(Eq.3-26)}$$

where θ_1^2 is the maximum eigenvalue while w_1 and c_1 are the unit eigenvector of the matrix $X_s^T Y_s Y_s^T X_s$ and $Y_s^T X_s X_s^T Y_s$ respectively. Calculate X_s and Y_s from Eq.3-26:

$$\begin{cases} X_s = t_1 q_1^T + E_1 \\ \mathrm{q}_1 = \frac{X_s^T t_1}{\|t_1\|^2} \\ Y_s = t_1 r_1^T + F_1 \\ r_1 = \frac{Y_s^T t_1}{\|t_1\|^2} \end{cases} \quad \text{(Eq.3-27)}$$

If this X_1 and Y_1 by the first component is not good enough, the second component will be extracted and repeat regression. Specifically, E_1 and F_1 is the residual matrix which will be used to replace X_s and Y_s respectively in Eq.3-26 and then compute the Eq.3-27 again to obtain X_2 and Y_2. Eq.3-27 and Eq.3-26 will be iterated until the regression reaches the expected accuracy. If the rank of X is A, then the general regression equations are:

$$\begin{cases} X_s = t_1 q_1^T + \cdots + t_A q_A^T \\ Y_s = t_1 r_1^T + \cdots + t_A r_A^T + F_A \end{cases} \quad \text{(Eq.3-28)}$$

The number of the principal component selected for regression usually is determined by the cross-validation (CV) [64]. The simplest one is the leave-one-out cross-validation (LOOCV). It uses one observation as the validation set and the remaining observations as the training set. This method is repeated in all ways to cut the original sample on a validation set of one observation and a training set of *n-1* observations. The predicted residual error sum of squares (PRESS) is a criterion of cross-validation used in regression analysis to provide a summary measure of the fit of a model to a sample of observations. It is calculated as the sums of squares of the prediction residuals for those observations.

$$\mathrm{PRESS} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \quad \text{(Eq.3-29)}$$

Therefore, the idea of using the CV to select the number of components for PLSR has three steps. The first step is to choose the first component for cross-validation as the first step and calculate its PRESS. Secondly, add the second component so that the first plus the second component is applied for cross-validation and obtain the new PRESS. Thirdly, add next component and iterate the steps until the latest PRESS becomes bigger than the last one.

3.6 Statistic Criteria

3.6.1 Standard Error of the Estimate

The Standard Error of the Estimate (SEE) is a classical statistic criterion for the measurement of the accuracy of predictions widely used in regression analysis. The SEE represents the average distance that the observed values fall from the regression line. In other words, the SEE shows how wrong the regression model is, on average, using the units of the response variable. Smaller values are better because it indicates that the observations are closer to the fitted line.

In many papers, the standard error of prediction is calculated on different dataset [70]. SEE renames into the standard error of the prediction (SEP) for the validation set and the standard error of the cross-validation (SECV) for cross-validation on the training set. In the NIRS domain, usually, the root mean squared error (RMSE) is preferred as the calculation of SEE. Thus, the root mean square error of prediction (RMSEP) is for validation set while the root mean square error of cross-validation (RMSECV) is for cross-validation on the training set. The formula is:

$$RMSE = SEE_{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}},\ \mathrm{i} = 1,2,\dots,\mathrm{n} \quad \text{(Eq.3-30)}$$

In Eq.3-30, a set of reference values *y* and predictions $\hat{y}$ have *n* samples. This RMSE is an estimate of the typical difference between reference value and prediction, which shows directly how good the calibration model is. A reasonable supplement for the RMSE is the average differences/bias between reference values and predictions. When the RMSE is large relatively, bias presents systematic errors that may be due to instrument, chemometric methods or reference. In this case, the SEE_{BIAS} (SEE corrected for BIAS) tells how good the calibration model will be if the BIAS problem can be solved.

$$\mathrm{BIAS} = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)}{n},\ \mathrm{i} = 1,2,\dots,\mathrm{n} \quad \text{(Eq. 3-31)}$$

$$\mathrm{SEE}_{BIAS} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i - BIAS)^2}{n-1}},\ \mathrm{i} = 1,2,\dots,\mathrm{n} \quad \text{(Eq. 3-32)}$$

The SEE_{RMSE} and the SEE_{BIAS} can be related by:

$$n \cdot SEE_{RMSE}^2 = (n-1) \cdot SEE_{BIAS}^2 + \mathrm{n} \cdot BIAS^2 \quad \text{(Eq. 3-33)}$$

Approximately, we can count *n-1* as *n* when n is big enough. Then:

$$SEE_{RMSE}^2 = SEE_{BIAS}^2 + BIAS^2 \quad \text{(Eq. 3-34)}$$

Thus SEE_{RMSE} consists of two independent parts SEE_{BIAS} and BIAS. If the BIAS is removed, the deviation term, SEE_{BIAS}, accounts for the dispersion around the 1:1 line in the prediction versus reference graph. Averaging the spectral measurements can reduce the deviation comes from random errors, whereas the BIAS caused by systematic errors can only be improved by establishing a calibration model as robust as possible [71].

3.6.2 Coefficient of Determination

The R^2 interprets how close the predictions against the references are to the fitted regression line. For a set of reference values y_i, associated with a set of prediction values $\hat{y}_i$, the sum of squares of residuals (SS_{res}) and the total sum of squares (SS_{tot}) respectively are

$$SS_{res} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 ,\ \mathrm{i} = 1,2,\dots,\mathrm{n} \quad \text{(Eq.3-38)}$$

$$SS_{tot} = \sum_{i=1}^{n}(y_i - \bar{y}_i)^2 ,\ \mathrm{i} = 1,2,\dots,\mathrm{n} \quad \text{(Eq.3-39)}$$

The R^2 is

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \quad \text{(Eq.3-40)}$$

3.7 Related Work: Applications of Methods

Due to the difference among sample properties, there is no point to make a comparison between different categories of samples. Besides, experts discouraged to combine reference data from different laboratories even if with the same reference method, because errors from different laboratories differ and precision and accuracy of NIRS calibrations will be determined by the quality of the reference laboratory data [13]. Besides, the gap of the NIRS instrument's precision and resolution affect the calibration model's predictive ability. It is difficult to compare different NIRS researches if they employed different reference methods or instruments because it is complicated to distinguish whether the results influenced by reference methods, instruments' precision or chemometric methods. As a result, few related research deserves a comparison between them and research works done by this PhD thesis. In this section, related works contain two types of researches. One is review papers offering a comparison among chemometric methods. Another is NIRS applications on single rice sample.

Roggo et al. provided a comprehensive review of the conventional pre-processing methods in NIRS [16]. That paper specified the principle, merit and demerit of 6 pre-processing methods with some expanded methods. A quantitative example by using six different instruments on moisture content and sugar content of 32 marzipan samples (bulk sample, legacy lab data set) were offered. The first conclusion of that paper was using pre-processing methods one at a time is much better than using a combination of pre-processing methods. Another conclusion was that all pre-processing methods make a slight improvement for model compared with the global model built without any pre-

processing methods. Thirdly, no single pre-processing method is useful for all either six instruments or two properties, but the optimal pre-processing method is changed when the circumstance is changed. The drawbacks of that paper are 1) the amount of samples utilized is relatively insufficient and 2) comparisons undertook between pre-processing methods only. Xiaobo et al. wrote a review paper for 12 variable selections methods with some expansion methods [17]. Chemical and physical basis and the principle of those methods have been illustrated in details. However, this review paper emphasised on the theory of methods and did not give an example to compare those methods. Mehmood et al. published a review paper for 11 PLSR-based variable selection methods [18]. Those two review papers presented and classified elaborately with relevant applications for further reading, but no common dataset offered for the comparison between those methods. Pasquini published a review paper for NIRS on three aspects of chemometrics, fundamentals and instrumentations [19] recently. That paper offered a summary of a wide range of methods with reference applications, but neither detailed principle nor example was provided. Balabin and Smirnov's paper is the only one according to the literature review provided a global view for benchmarking 16 variable selection methods but on rapeseed biodiesel fuel samples (bulk sample) [25]. In summary, based on the literature review, no review papers offered a real-world example for benchmarking a large number of methods. Only one paper constructed a comparative study to benchmark a large number of variable selection methods but for bulk sample.

Table 3.1 presents a summary of 28 quantitative NIRS applications on rice in the past 20 years (papers were searched on Google Scholar and Elsevier by keywords 'rice' and 'near-infrared spectroscopy'). Rice forms depend on which form of rice was scanned to acquire spectrum: 1) single rough rice (SRR), 2) single brown rice (SBR), 3) single milled rice (SMR), 4) bulk milled rice (BMR), 5) bulk rough rice (BRR), 6) bulk brown rice (BBR), and 7) rice flour (RF). The average number of methods used by those 28 applications is only three, which may be sufficient for a specific application but insufficient for a global comparative study. In figure 3.1, chart (a) displays the percentage of 28 publications in table 3.1 for three forms of rice. Almost half of those applications employed bulk rice sample for research. In contrast, about one-third of applications referred to single rice sample. Chart (b) displays the percentage of 28 publications in table 3.1 for different interesting properties. The most interested property is amylose, followed by protein because the concentration of these two chemical properties is easy to determine by reference methods.

Table 3.1: A summary of 28 quantitative NIRS applications on rice in the past 20 years.

Paper ID	Publication Year	Single Rice Form	Sample size	No. Methods	Interested Properties	Reference
1	1995	BMR	247	1	Amylose	[72]
2	2002	BMR	204	2	Amino acids	[73]
3	2003	SMR & SRR	150	1	Moisture & Protein	[74]
4	2003	SRR	222	3	Amylose	[75]
5	2004	SBR	100	4	Moisture & Protein	[76]
6	2004	SMR,SRR&SBR	474	2	Weight & Amylose	[77]
7	2007	BMR & RF	225	1	Starch	[78]
8	2007	BMR	90	1	Aroma, Appearance, Brightness, Taste, Stickiness, Hardness and Eating quality	[79]
9	2007	RF	586	3	Amylose, Gel consistency & Alkali spread value	[80]
10	2007	BBR	178	1	Amylose & Protein	[81]
11	2010	BMR	198	9	Surface lipid content	[82]
12	2011	RF	320	3	Starch & Protein	[83]
13	2011	BBR & RF	279	3	Amino acids	[84]
14	2013	BRR,BBR&BMR	106	5	Aflatoxigenic fungal contamination	[85]
15	2014	BMR	180	2	Freshness	[86]
16	2014	RF	519	3	Amylose & Protein	[87]
17	2015	SRR	160	2	Vigour	[88]
18	2016	BBR	173	3	Amylose & Protein	[89]
19	2017	RF	168	5	Amylose	[90]
20	2017	SMR	105	1	Amylose	[91]
21	2018	BRR	148	6	Sugar	[92]
22	2018	RF	168	7	Amylose	[93]
23	2019	SRR	288	2	Hardness	[94]
24	2019	BBR	832	2	Amylose	[95]
25	2019	BRR	164	3	Purity	[96]
26	2019	SRR	14	4	Moisture	[97]
27	2019	SRR	164	4	Purity	[98]
28	2019	SRR, SBR & RF	201	5	Protein	[2]

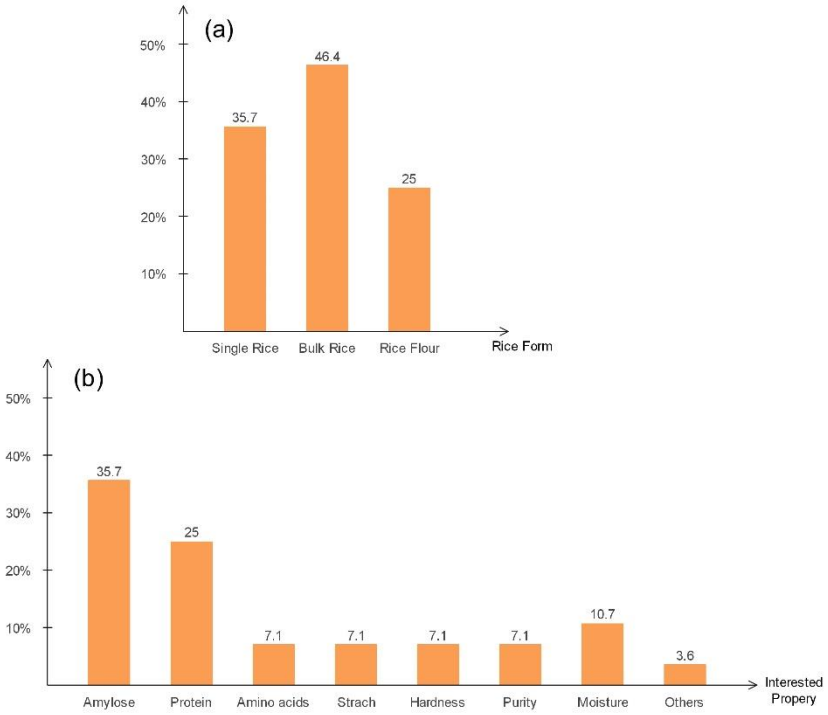


Figure 3.1: Percentages of rice forms (a) or interesting properties (b) among 28 publications.

3.8 Summary

Foundations about the principles and theories of sixteen methods and two statistic criteria have been illustrated in this chapter. Related work for the applications of those methods has been reviewed as well. Most application researches refer to rice focused on one method and employed two more other methods as a brief comparison, which could not provide a sufficient comparative study. However, most review papers gave a demonstration between methods but lacked a dataset to show a specific difference. There is still no comparative study for methods on a single rice sample yet. All of these are evidence to support the research problems mentioned in section 1.2, and they are motivations of this PhD thesis as well.

4 DESIGNING THE BENCHMARK FOR CHEMOMETRIC METHODS FOR ANALYSING SINGLE KERNEL SAMPLE

4.1 Overview

This chapter will illustrate how the benchmark has been designed. Definition, criteria, and process will be interpreted in detail from section 4.2 to 4.4, respectively.

4.2 Definition of the Benchmark for Chemometric Methods

In the Oxford dictionary, the explanation of the word 'benchmark' is "In business, the benchmark is a process in which a company compares its products and methods with those of the most successful companies in its field, in order to try to improve its performance." Although that explanation is more related to the business domain, the word 'benchmark' can be extended its meaning in other fields. The emphasis of the 'benchmark' is on the comparison between methods, products or models. The keyword 'benchmark' has been utilized in the title of some papers in NIRS domain, but it is hard to find out its earliest appearance and all of those paper did not provide a specific definition of benchmark when it is related to the NIRS or chemometrics. A common point among papers whose title contains word 'benchmark' is that they provided a comparative study. For example, among the papers which have the word 'benchmark' in the title, the paper with most citations used the 'benchmark' as the global comparison of variable selection methods in the NIRS domain on a biodiesel sample set [25]. Considering the domain difference between the NIRS and computer science, it is necessary to provide a specific definition of the word 'benchmark' for this PhD thesis at first in this chapter. Based on the original meaning of the word 'benchmark' and those relevant NIRS papers [13], [16], [17], [18], [25], which refers to the benchmark, the specialized definition of the benchmark in this PhD thesis is:

> The Benchmark for chemometric methods is a standard or point of reference for evaluating chemometric methods by assessing their corresponding calibration models with a set of statistic criteria.

This PhD thesis focuses on the QNIRSA so that four categories of chemometric methods have been assessed. They are dataset partition methods, pre-processing methods, variable selection methods and multivariate calibration methods respectively, and all have been specified in section 2.3.2 and chapter 3. The performance of both single chemometric method and the combination of two or more chemometric methods

will be measured and assessed relatively. Specifically, the measurement for the performance of chemometric methods is to assess calibration models corresponding to different chemometric methods by the statistic criteria, while the comparison for the performance of chemometric methods is a comparative study between those assessment results.

4.3 Benchmarking Criteria

Two statistic criteria for assessing the calibration model have already been illustrated in section 3.6, and they are: 1) coefficient of determination (R^2), and 2) root mean square errors of prediction (RMSEP). The RMSEP is an absolute criterion to measure the performance of the calibration model. The smaller the RMSEP, the better predictive performance the calibration model has. Therefore, the RMSEP will be the primary criterion to assess the performance of the calibration model in chapter 6. When the RMSEP results of different calibration models are closed (difference smaller than 0.01), it will be hard to conclude the best one. In this manner, the R^2 and may help the comparison. The R^2 is a statistical measure of how close the data are to the fitted regression line, which presents the percentage of the response variable variation that is explained by a linear model. The larger the R2, the better the model fits data; the more percentage of the response variable variation can be explained. For example, assume that the RMSEP of two calibration models A and B is 0.551 and 0.552, respectively. The difference between them is only 0.001, which is not a significant bias to say A is better than B convincingly. Meanwhile, assume that the R^2 of A and B is 0.85 and 0.9 respectively. In this manner, a possible conclusion is that B may be better than A because B fits data better than A due to the larger R^2 that B has.

Multiple chemometric methods may be used as a combination so that there will be a large number of possibilities for the comparative study. In order to make the comparison clear and logical, potential comparisons are divided into two levels: 1) global level and 2) local level. Comparison at the global level is executed at the level of a dataset. For example, to find the best combination of chemometric methods, whose calibration model has the best performance for the SRK data set is one of the global goals. The local level is on the steps of the QNIRSA. For instance, to figure out the impact a step makes to model is one of the local goals. Before listing all goals, reference model needs to be determined as a standard for comparison. Three full spectrum (FS) models have been established as the global reference models for three kinds of datasets, respectively presented in table 4.1. Full spectrum model means all variables are used to build the model without any variable selection methods. Spectral data remain original feature without any pre-processing methods. Data set is divided empirically and manually by experts [2] as a reference method for data partition (RMDP) (30% of the original dataset for validation set and 70% for training set). Calibration method for all models is the PLSR method specified in section 3.5.3. In terms of the three models in table 4.1, without any chemometric methods before calibration, the values of RMSEP can be standard criteria values for three data sets. If values of the RMSEP

corresponding to chemometric methods are larger than this standard value, those methods can be classified as ineffective. The decrement chemometric methods make to the standard values can be a criterion to distinguish the low-effective and highly-effective method, notated as:

$$\text{DRMSEP} = \text{RMSEP} - RMSEP_{standard} \quad \text{(Eq.4-1)}$$

Positive DRMSEP means the relevant chemometric methods are ineffective, while negative DRMSEP means the relevant chemometric methods are effective.

Table 4.1: Three global reference models.

Model ID	Data set	Data partition			Pre-process	Variable Selection		Calibration		Assessment	
		Method	Training set	Validation set	Method	Method	Selected variables	Method	Latent variable	RMSEP	R^2
1	SRK	RMDP	149	52	None	FS	936	PLSR	20	0.7418	0.817
2	SBK	RMDP	149	52	None	FS	936	PLSR	18	0.6393	0.864
3	RF	RMDP	149	52	None	FS	936	PLSR	13	0.5531	0.8982

Specifically, goals at the global level include:

1. To find the best combination of chemometric methods, whose calibration model has the best performance for the SRK. There will be 56 kinds of combinations of chemometric methods (2 x 4 x 7) for each rice form. All combinations will be tested and then find the best one for SRK samples by RMSEP and R^2.
2. To find the best combination of chemometric methods, whose calibration model has the best performance for the SBK. All combinations will be tested and then find the best one for SBK samples by RMSEP and R^2.
3. To find the best combination of chemometric methods, whose calibration model has the best performance for the RF. All combinations will be tested and then find the best one for RF samples by RMSEP and R^2.
4. If some combinations of chemometric methods available in the first to third global goal, classify them respectively based on RMSEP and R^2.

Local goals are:

1. To figure out the impact sampling makes to PLSR model. Utilise sampling method one at a time to build a model. Ensure no other methods else are used except PLSR. Assess that model to obtain relevant RMSEP and DRMSEP. Calculate and compare the average and optimal performance of each sampling method to figure out the impact sampling makes to the model. This comparison will be repeated for SRK, SBK and RF samples, respectively.

2. To figure out the impact pre-processing makes to model. Utilise pre-processing method one at a time to build a model. Ensure no other methods else are used except PLSR. Assess that model to obtain relevant RMSEP and DRMSEP. Calculate and compare the average and optimal performance of each pre-processing method to figure out the impact pre-processing makes to the PLSR model. This comparison will be repeated for SRK, SBK and RF samples, respectively
3. To figure out the impact variable selection makes to PLSR model. Utilise variable selection method one at a time to build a model. Ensure no other methods else are used except PLSR. Assess that model to obtain relevant RMSEP and DRMSEP. Calculate and compare the average and optimal performance of each variable selection method to figure out the impact variable selection makes to the model. This comparison will be repeated for SRK, SBK and RF samples, respectively.
4. To figure out the impact pre-processing makes to variable selection. Balabin and Smirnov made a benchmark of some variable selection methods for bulk fuel sample [25]. In their paper, a fixed pre-processing method was used before variable selection. Since the third local goal investigates the variable selection methods without pre-processing, the fourth local goal is to investigate variable selection methods with pre-processing methods. Compare the optimal, and average performance between the third local goal and fourth local goal to figure out the impact pre-processing makes to variable selection.
5. To figure out the impact calibration methods MLR, PCR, and PLSR makes to model respectively based on relevant RMSEP. This comparison will be repeated for SRK, SBK and RF samples, respectively.
6. To compare the impact each step makes to PLSR model. Compare the average, and optimal DRMSEP obtained in local goals 1, 2, and 3. The largest absolute value of the negative DRMSEP implies the most significant impact that step makes to model.

4.4 Benchmarking Process

4.4.1 Overview

Figure 4.1 shows the BPMN (Business Process Model and Notation) diagram for the designing of the benchmarking process for chemometric methods. The benchmarking process can be divided into three stages (dash line box is the stage while the blue box is the step. Black case with black arrow is output data while with the white arrow is input data for steps.): 1) data collection, 2) data processing and 3) real-world applications. Conduct reference experiments and scan samples are the first two steps in stage 1. The original dataset is the common input for these two steps. In terms of the second stage, reference data and spectra from stage 1 are imported to step 3 at first. Dataset is divided into a training set for step 4 and validation set for step 7. Step 4 pre-processes the training set, and exports pre-processed dataset to step 5. Step 5 selects variables for step 6. Step 6 construct a model, and it is validated by step 7 with the validation set. The validated model is used in step 8 for real-world applications refers to chapter 7.

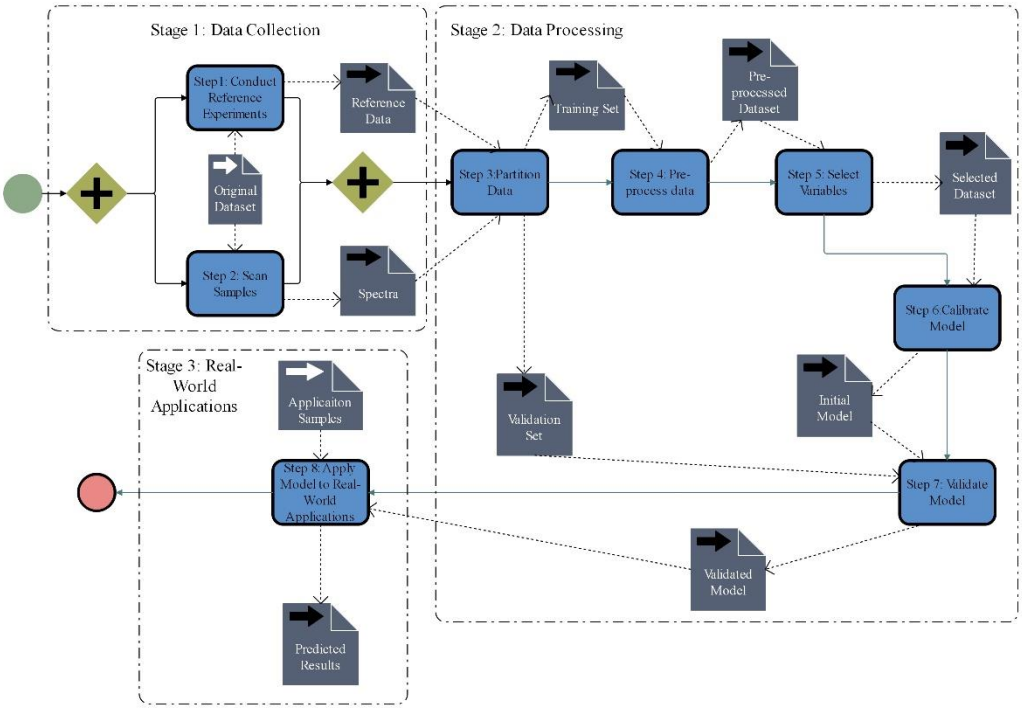
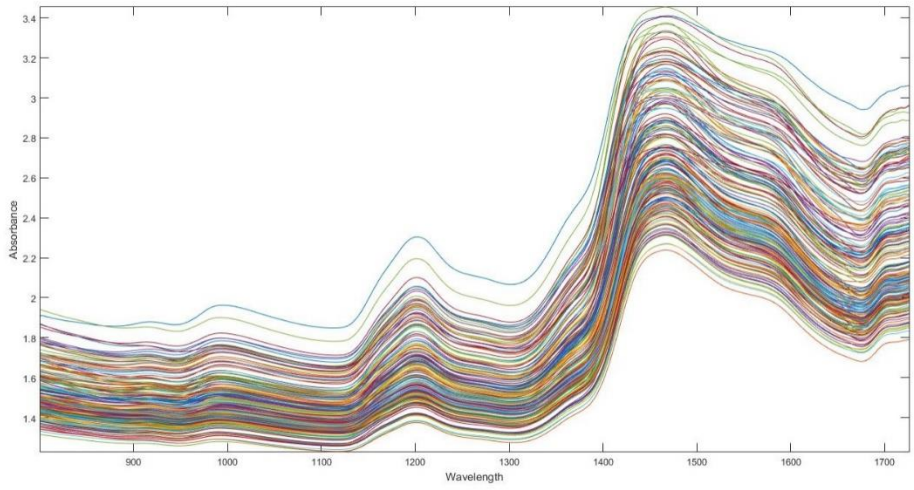


Figure 4.1: BPMN diagram for the designing of the benchmarking process for chemometric methods.

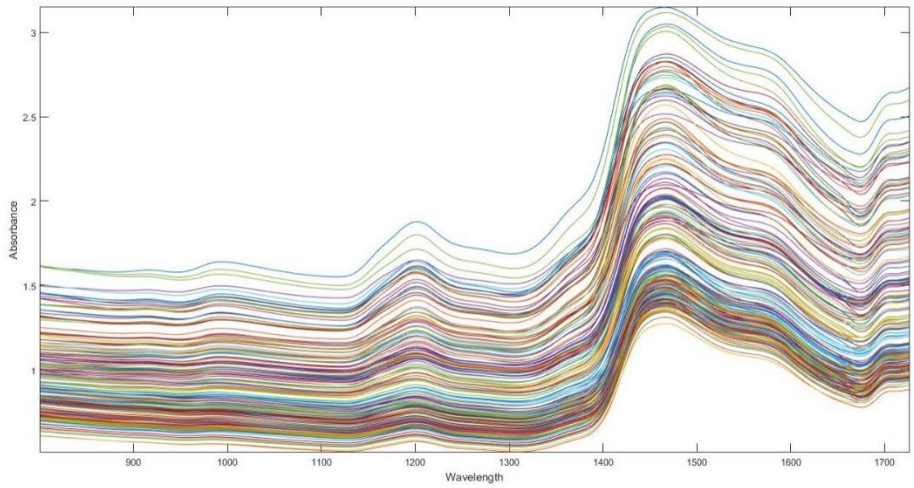
4.4.2 Stage 1: Data Collection

Some single kernel samples such as rice, corn, solid fertilisation have been collected in the past three years. However, there is no point to make a comparison between different categories due to the difference among sample properties. Additionally, compared with the bulk sample, it is quite difficult for single kernel samples to account for their laboratory error by reference methods [13]. At last, one category of those single kernel samples which has been successfully employed for the real-world application was selected. Totally 201 single rice kernels were selected for the NIR analysis. It follows 2018 Chinese national crop variety regional test materials, with 21 rice varieties (about 1–4 rice kernels per rice variety). The calibration samples were from 25 rice mutant varieties with different protein content (about 4–7 rice kernels per rice mutant variety). These mutant rice varieties were derived from the rice agronomic traits mutant library constructed by the laboratory established by collaborators from the Hefei Institute of Physical Science, Chinese Academy of Sciences. The mutant library was generated by

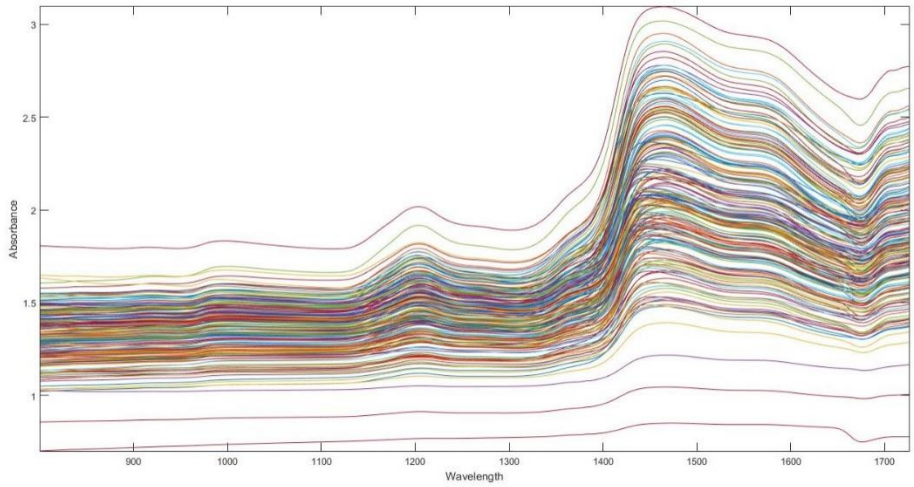
low energy and heavy-ion beam irradiation on rice variety ‘9311’ [99]. Three forms of



(a) spectra of the SRK sample



(b) spectra of the SBK sample



(c) spectra of the RF sample

Figure 4.2: The spectra (a), (b), and (c) of SRK, SBK and RF, respectively.

rice samples have been produced. The original form is single rice kernel (SRK) samples placed in an incubator for 24 h at 25 °C and relative humidity (r.h.) of 50% before spectra acquisition. The single brown rice kernel (SBK) samples were obtained by removing each SRK sample's glume manually. The rice flour (RF) samples were obtained by grinding each SBK sample manually with a small mortar and pestle.

The NIR transmission spectra were obtained by MPA Fourier transform near-infrared spectrometer (Bruker, Germany) in the full spectral range (800–1726 nm), with a spectral resolution of 1 nanometre. Every SRK and SBK sample was scanned for 20 times slowly and averaged spectrum was calculated for usage. Each RF sample was scanned for five times, and the averaged spectrum was calculated as well. Every spectrum of a single sample has 936 variables. As a result, there are three data sets for SRK, SBK, and RF samples, respectively. These three data sets are shown in figure 4.2, where chart (a), (b) and (c) are the spectra of SRK, SBK and RF samples, respectively.

After the spectra acquisition of three forms of rice samples, the protein contents of single rice kernels were analysed using the Dumas combustion method. This method has been proved to be effective in single seed spectral analysis for the minimum sampling weight was only a few milligrams [100]. However, the Dumas combustion method will destroy samples after analysis so we cannot get any information for other properties of rice. Each rice flour sample was weighed to 4.0 ± 0.2 mg using the electronic scale (Mettler Toledo, Switzerland) and wrapped in foil into a ball shape, and approximately 5 mg of pure benzene sulfonic acid was weighed and wrapped in foil, as the standard. All samples were analysed in an elemental analyser (Elementar, Germany). The protein content was calculated according to the instrument output of nitrogen (N% × 5.95).

4.4.3 Stage 2: Data Processing

As it has been illustrated in detail in the section 2.3.2, during the QNIRSA, chemometric methods mainly are utilised in four steps in the sequential order: 1) dataset partition, 2) pre-processing, 3) variable selection and 4) multivariate calibration. The dataset partition step is to solve the training set, and validation set partitioning problem, in order to extract a representative training set to construct a model and reasonable validation set to assess model from the original NIR spectral data set. The pre-processing step is to remove the undesired physical phenomena in the spectra mainly caused by sample-to-sample variations, in order to facilitate the subsequent procedures. Variable selection step aims to reduce the number of variables and select representative variables, which can render better prediction with the regression model used in multivariate calibration. Multivariate calibration methods are developed through regressions of the measured NIR spectral data against the reference data values of analyte properties determined by reference analytical methods. The calibration model is constructed in this step and validated in the next step. Chapter 3 demonstrated principles about sixteen chemometric methods involved in this PhD thesis. They are two dataset

partition methods KS and SPXY, three pre-processing methods MSC, SNV and SG, eight variable selection methods SPA, UVA, SA, GA, iPLS, BiPLS, FiPLS and PCA, and three calibration methods MLR, PCR and PLSR.

The original data set will be divided into training set and validation set in an empirically fixed ratio 7:3 [2]. Thus, as for all chemometric methods, the training set has 149 samples while validation set has 52 samples. In order to provide a fair comparative study for the different chemometric methods, each method should be optimised independently. Although the computing time for different optimisations can be rather large, only the final (the best possible) results can be compared without bias. In other words, only the best results of every method or the combination of multiple methods will be compared. In terms of some chemometric methods which have statistical randomness for the initial parameters (e.g., SA & GA), multiple initial parameters will be tested for many times. The best result will be regarded as the optimal result of those methods. Those experiments will be repeated for SRK, SBK and RF data set respectively. All the chemometric methods and criteria were coded on the MATLAB software version 2015a (Mathworks, USA).

The average spectrum of the training set is used as the reference spectrum for the MSC. Seventeen points smoothing and polynomial fit whose order is varied from 1 to 6 are set for both first derivative and second derivative in the SG.

The RMSECV of 5-fold cross-validation was minimised to detect the optimal models on training set for all variable selection methods. The number of sub-intervals was optimised for the iPLS, BiPLS and FiPLS in the range between 3 and 200. There is no limit on the expected number of variables selected by the SPA. As for the UVE, the arbitrary coefficient was set from 0.1 to 1.5, respectively. The noise matrix has the same size as the training set, and noisy values were set to $10^{-10,}$ which is relatively small enough. Due to the randomness of the noise matrix, 20 repetitions were done for each arbitrary coefficient. The cooling ratio for the SA was set from 0.5 to 0.9, respectively. 0.001 was set for the initial value of the parameter T for every different cooling ratio. Markov chain was set to 4000 in default, which means the maximum number of tries within one value of T is 4000. The threshold value of T to stop SA was set to a fixed value 10^{-6}. The size of chromosomes in the GA was set to 100, and no twins are allowed. There is no limit for the variables selected by GA. The maximum number of generations was set to 200, and the probability of single-point crossover was set from 0.5 to 0.9, respectively, while the probability of mutation was set from 0.05 to 0.1, respectively. Twenty iterations were made for different initial parameters.

The MLR is only used to support SPA as SPA-MLR principle required, while PCR is used only with PCA. PLSR will be the standard calibration method. 5-fold cross-validation is used to determine the optimal number of principal components/latent variable in both PCR and PLSR.

4.5 Summary

Spectral dataset of three forms of single rice and their relevant protein content has been collected. Two dataset partition methods, three pre-processing methods, eight variable selection methods, and three multivariate calibration methods have been optimised. The priority of criteria has been interpreted for measurement, and goals of comparison have been listed one by one. A BPMN diagram about the process of benchmarking for chemometric methods was designed to guide the benchmarking experiments.

5 DEVELOPMENT OF THE QNIRSA SYSTEM

5.1 Overview

This chapter will interpret the development and implementation of the QNIRSA system. Section 5.2 will illustrate the architecture of the QNIRSA system, while section 5.3 will specify how to use functional modelling (IDEF0 approach) to design and develop the QNIRSA system. Section 5.4 is the implementation of the QNIRSA system. Programming configuration and user interface dashboard for the user are illustrated, but full codes will be attached in appendix A.

5.2 The Architecture of the QNIRSA System

Figure 5.1 presents the architecture of the QNIRSA System. There are five layers for the architecture. The lowest layer is the data layer referring to spectral data and reference data. The second layer is the data manager layer, including two data storage carrier, spectral database and excel document. The third layer is the component layer providing three advantages: 1) as for different goals, relevant components can be invoked flexibly; 2) a component mainly designed for controlling spectrometer provides a connector for different devices according to their application program interface (APIs); 3) a library of chemometric methods are packaged and developed for universal use. The main components of this system are briefly described as follows:

- **Hardware Adapter**. It is used to operate the NIRS spectrometer and other sample-specific accessories. It is responsible for 1) configuring the spectrometer, 2) scanning the sample one at a time, 3) acquiring the sample spectrum. This component was implemented by the combination of the C programming language and Java for efficiency. The C programming language is mainly for the command to move hardware like the light source and conveyor belt, while Java is for setting up the interface of APIs for the connection between software and hardware.
- **Spectral Pre-processor**. It supports spectral pre-treatments. Pre-processing methods (e.g., SNV and MSC) were implemented by using the Java programming language.
- **Quantitative Multivariate Analyser**. It supports quantitative multivariate data analysis, which refers to the multivariate calibration. As for the on-line application, with an imported calibration model, this component employs the regression principles of multivariate calibration methods (e.g. PCR or PLSR) to predict the chemical or physical property in the sample. Regarding the off-line application, this component is used for constructing a calibration model. These methods were also implemented by using Java.

- **Spectral Visualizer**. It is used to display the analysis results, consisting of the analysed spectrum and the predicted chemical or physical property of the sample. This component is part of the user interface that was implemented by Java.
- **Data Manager**. This component supports two types of data storage. One is to stores data in Excel document, while another employs a database to store data. It is coded by Java as well.
- **Chemometric Methods Library**. It is used to store the implementations of chemometric methods. This component was implemented and packaged using MATLAB, which can be invoked by Java.
- **Variable selector**. It supports the variable/wavelength selection. This component is only for off-line application because it is not required for online application. The variable selector was implemented by using the Java programming language.
- **Dataset Partitioner**. This is only used for off-line application aiming to partition the dataset into a training set and a validation set. Partitioning methods are implemented by using Java programming language.

The fourth layer is the mode layer containing on-line mode and off-line mode. The sequence diagram in Figure 5.2 depicts the interactions between user, on-line mode and off-line mode in time sequence. The user selects methods, configures parameters and imports known data for off-line mode at first. Then off-line mode constructs models, returns them to the user, and imports them to on-line mode. With the import of unknown data by user, on-line mode utilises those models from off-line mode to predict relevant properties for unknown data and return predicted results to the user. Results are displayed in the user interface at the presentation layer.

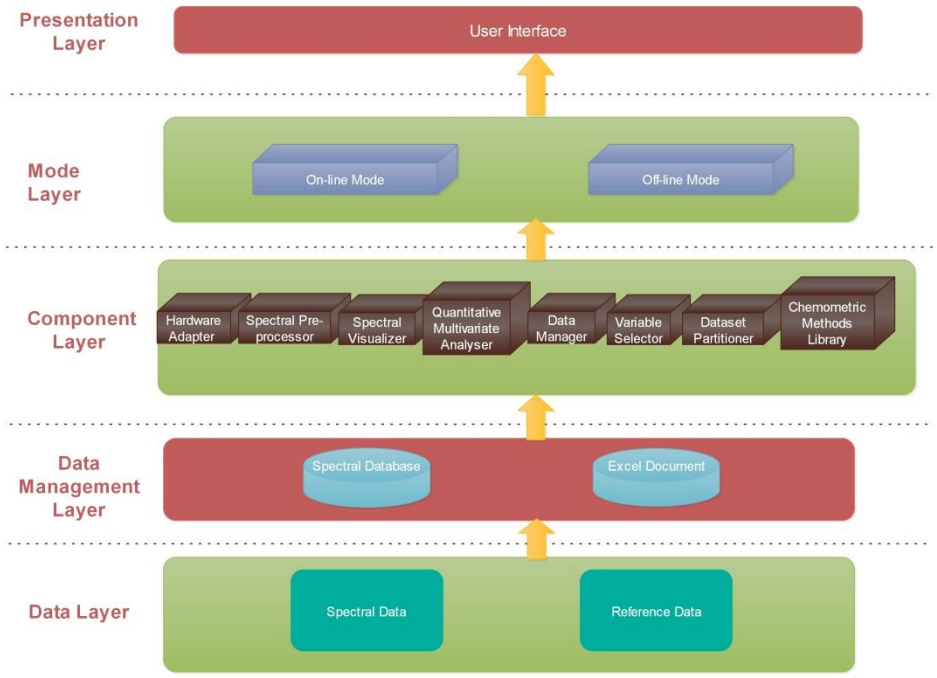


Figure 5.1: The architecture of the QNIRSA system.

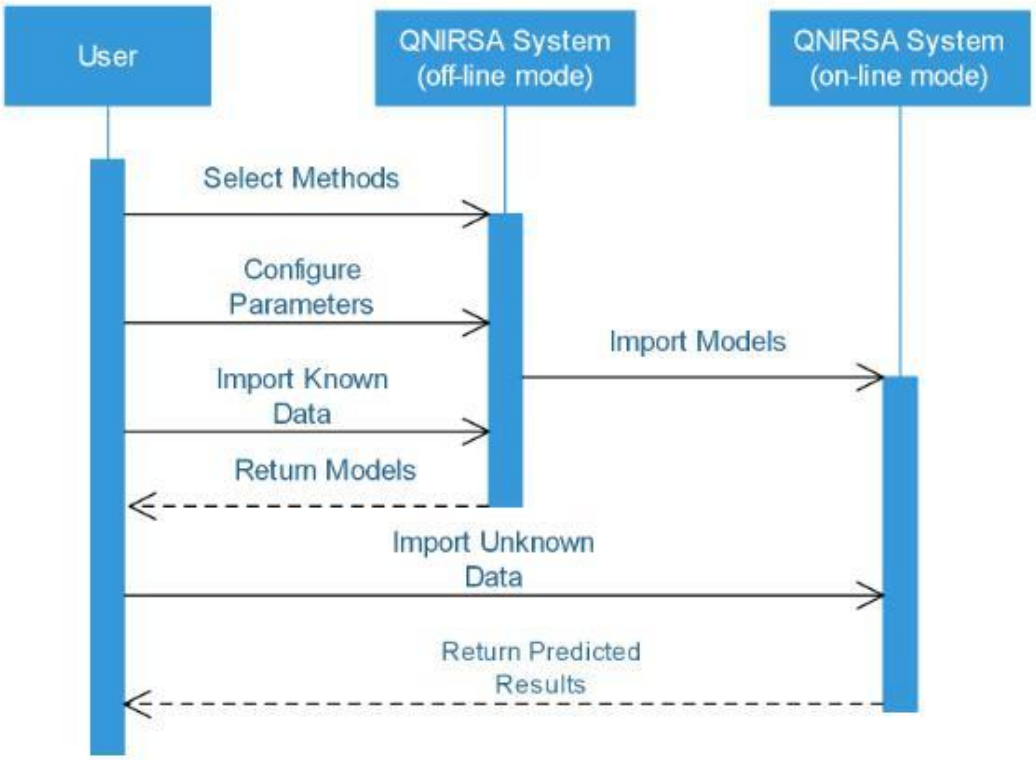


Figure 5.2: The sequence diagram for the mode layer.

5.3 Functional Modelling for the QNIRSA System

5.3.1 Overview

The key idea of the QNIRSA system is to modularise the steps of the proposed benchmarking process illustrated in section 4.4.1 by a stepwise functionalisation. Some requirements and functions of the QNIRSA system can be summarised according to the proposed benchmarking process:

1. The QNIRSA system can connect and control the external hardware, including different spectrometers and single-chip microcomputer by sending a command to them.
2. The QNIRSA system can acquire spectral data from the response of spectrometer.
3. The QNIRSA system can realise the quantitative near-infrared spectroscopy analysis, including the establishment of a calibration model and real-time analysis for the unknown sample.
4. The QNIRSA system can provide some chemometric methods to support the quantitative near-infrared spectroscopy analysis.
5. The QNIRSA system can store and extract spectra data obtained in the quantitative near-infrared spectroscopy analysis.

Functional modelling and IDEF0 approach are used to develop the QNIRSA system and model its functions. Figure 5.3 depicts the node tree diagram for the whole QNIRSA system. It has four levels. The top node, A-0, is the top-level context diagram setting a model's scope, that is, the boundaries of what may be included in that model. Figure 5.4 is the A-0 context diagram of a model named 'Develop QNIRSA System (DQA)'. The only function shown in A-0 is the A0 function whose node is under the A-0 node in the node tree. This A0 function represents the whole of the subject of the model; it is the unique parent of the entire modelled subject and thus the ancestor of all activities modelled. Requirements and samples are the input of the A0 function, while the output should be the QNIRSA system. Spectrometer and single-chip microcomputer are the external hardware outside the scope of this QNIRSA system. This QNIRSA system aims to connect and control them. A spectrometer is a NIR device to acquire spectral data from sample scanning, while single-chip microcomputer is used to control other hardware accessories. Those accessories are sample-specific in order to support the scanning of the sample, which makes the QNIRSA system possible for all kinds of solid agricultural samples. MATLAB (The Mathworks, USA) and Eclipse (Eclipse Foundation, open-source) is the software to program the QNIRSA system, and the third-party Java packages are used for serial port communication, Excel management and user interface design respectively. The viewpoint statement for a model shall be placed in the A-0 context diagram of the model, and it is a brief sentence that identifies a person or a personified role. The purpose statement for a model shall be placed in the A-0 context diagram of the model, and it is a brief sentence that identifies the question

addressed by a model. The purpose is the development and implementation of the QNIRSA system.

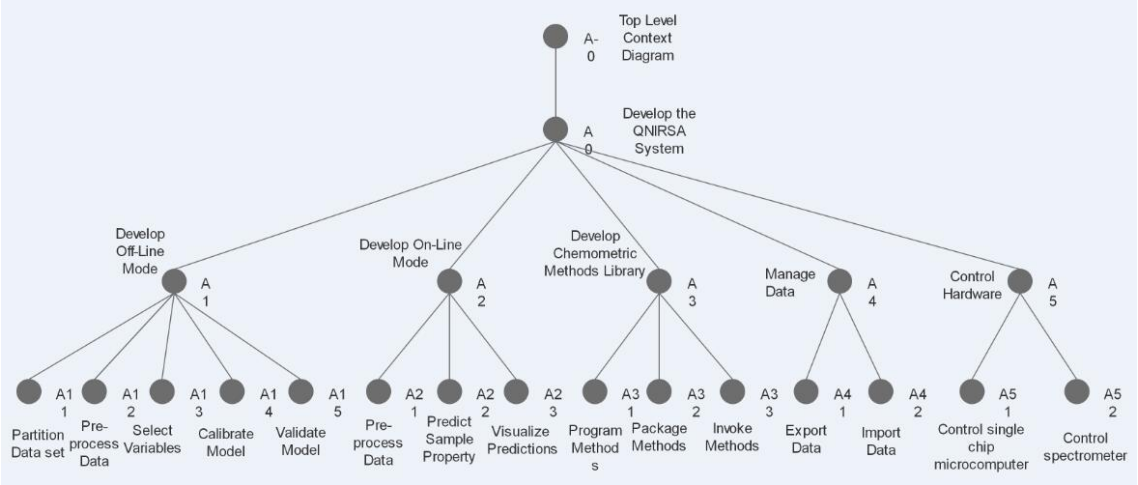


Figure 5.3: The node tree diagram for the QNIRSA system.

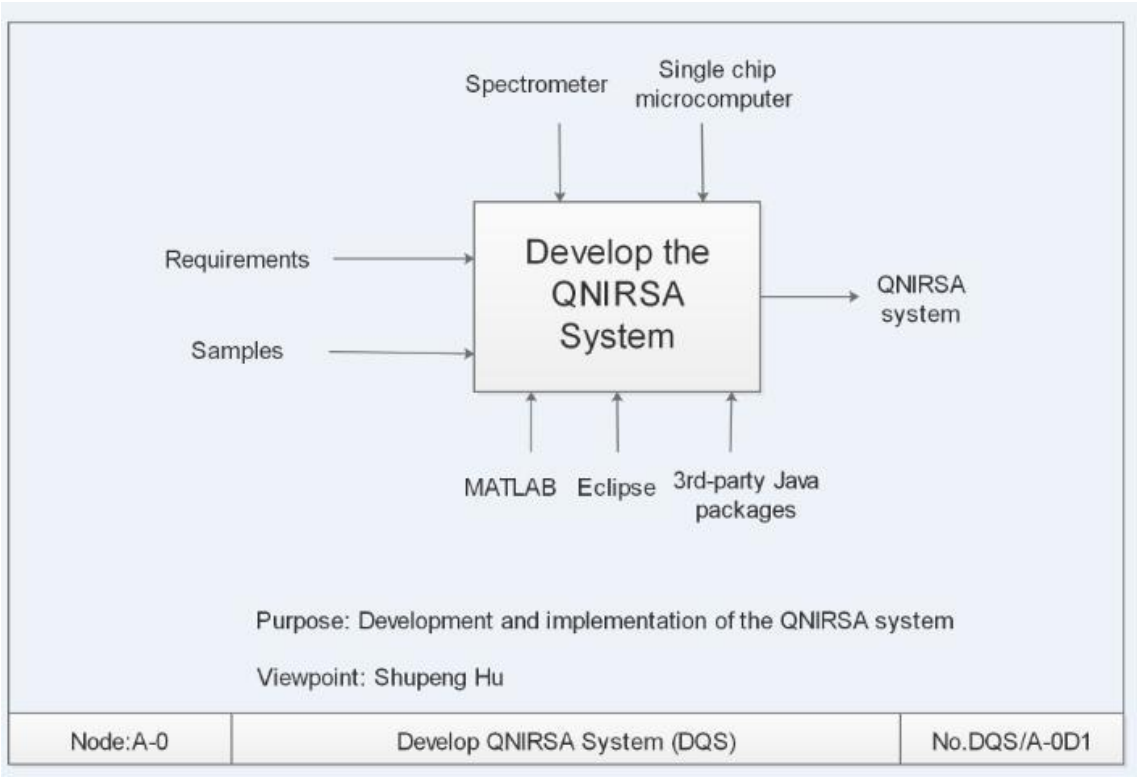


Figure 5.4: The top-level context diagram A-0.

The single function A0 represented in the A-0 context diagram is decomposed into its major sub-functions by creating its decomposition diagram. The node tree diagram depicts that A0 is decomposed into five lower-level decomposition diagrams from A1 to A5. A1 to A5 have their relevant decomposition diagrams at the lowest level as well. A3, A4 and A5 are exclusive for chemometric methods library, data manager and hardware adapter respectively at component layer.A5 is to connect and control hardware,

while A4 is to manage data obtained during the runtime. A3 aims to develop the chemometric methods library to provide methods for A1 and A2 by a combination programming of MATLAB and Java. A1 and A2 are at mode level, which can invoke components according to practical requirements. The third level nodes are all specific action based on the proposed benchmarking process

Figure 5.5 shows the specific IDEF0 diagram of A0. Numbers 1 to 5 are the process ID of functions, and A1 to A5 are the nodes concerning function. The A0 starts at the A5 (Control hardware) and ends at A2 (Develop online mode). A1 (Develop off-line mode) not only acquires input from A5's output but also outputs model as the input of A2. A3 (Develop chemometric methods library) provides chemometric methods for both A2 and A1. Outputs of A5, A1 and A2 are all inputs for the A4 (Manage data). Following sections from 5.3.2 to 5.3.6 will interpret the decomposition diagrams of A1 to A5, respectively.

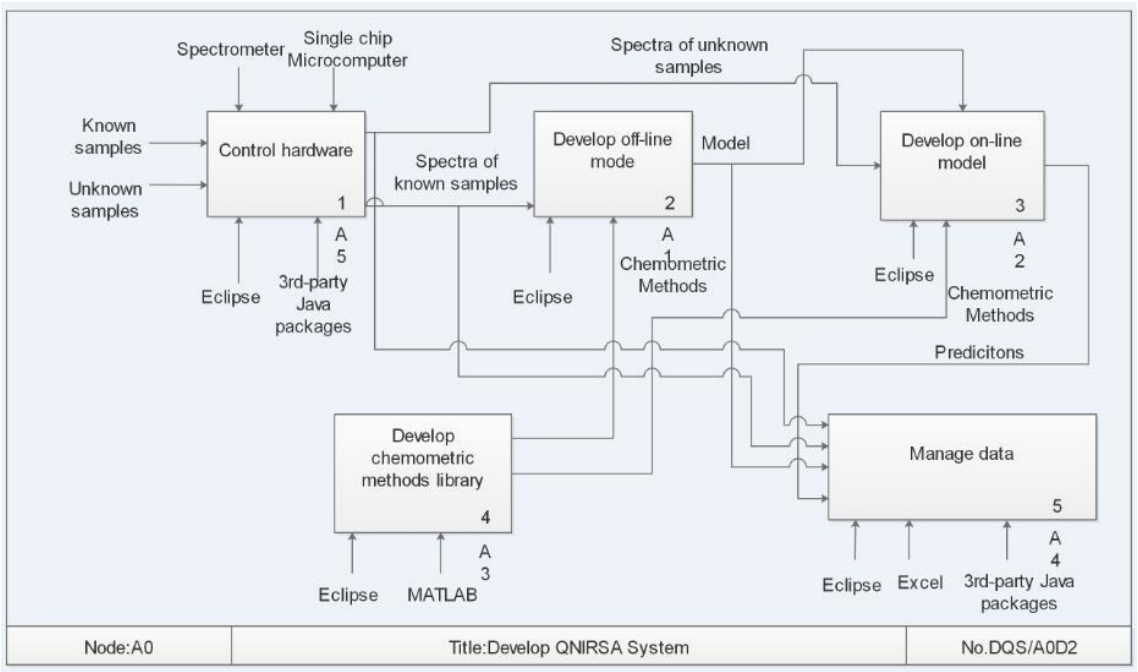


Figure 5.5: The IDEF0 diagram of A0 (Develop QNIRSA System).

5.3.2 Control of Hardware

Hardware system containing spectrometer, single-chip microcomputer and other accessories has been designed and developed by collaborators. Commands to control the single-chip microcomputer and application programming interface (API) have been provided. Thus, the QNIRSA system controls the single-chip microcomputer by commands to handle those sample-specific accessories. Sending commands to the single-chip microcomputer is supported by a third party Java package (RXTX, open-source)) and acquires spectral data from spectrometer via the corresponding API. Figure 5.6 is the IDEF0 diagram of A5, which shows the process for controlling the hardware.

Known samples for model calibration and unknown samples for real-time analysis are the input. By sending commands to the single-chip microcomputer, accessories are handled to move those samples to the desired position to be ready for scanning. During scanning, accessories will push samples one by one through the position under the light of spectrometer. The spectrometer will obtain those light reflectance information and generate a spectrum. Meanwhile, the QNIRSA system will acquire those real-time spectra via the spectrometer API. Outputs of A5 are spectra of known samples for off-line mode (A1) and spectra of unknown samples for on-line mode (A2).

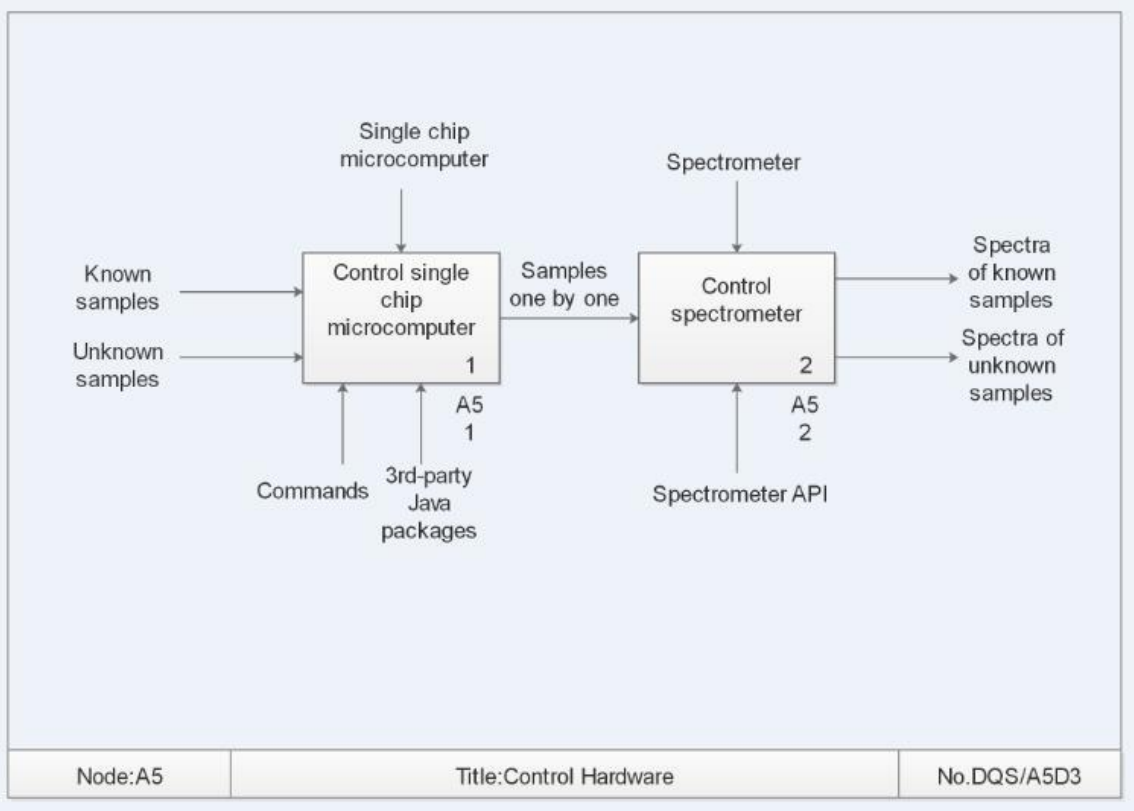


Figure 5.6: The IDEF0 diagram of A5 (Control Hardware).

5.3.3 Off-Line Mode

The goal of the Off-line mode is to build a calibration model, and it has five steps. Figure 5.7 shows the IDEF0 diagram of the off-line mode (A1). The off-line mode can be performed automatically in this sequence:

1. Data set of the spectra of known samples is divided into a training set and validation set based on the partition methods provided by the chemometric methods library (A3 in section 5.2.5).

2. Both the training set and validation set are pre-processed by pre-processing methods provided by the chemometric methods library as well.
3. Variable selection methods provided by the chemometric methods library are employed for both the pre-processed training set and validation set. However, the selected training set is the input of A14 while selected validation set is for A15.
4. The training set is used to establish a model by calibration methods provided by the chemometric methods library as well.
5. Model is validated by validation set based on some criteria provided by the chemometric methods library as well.

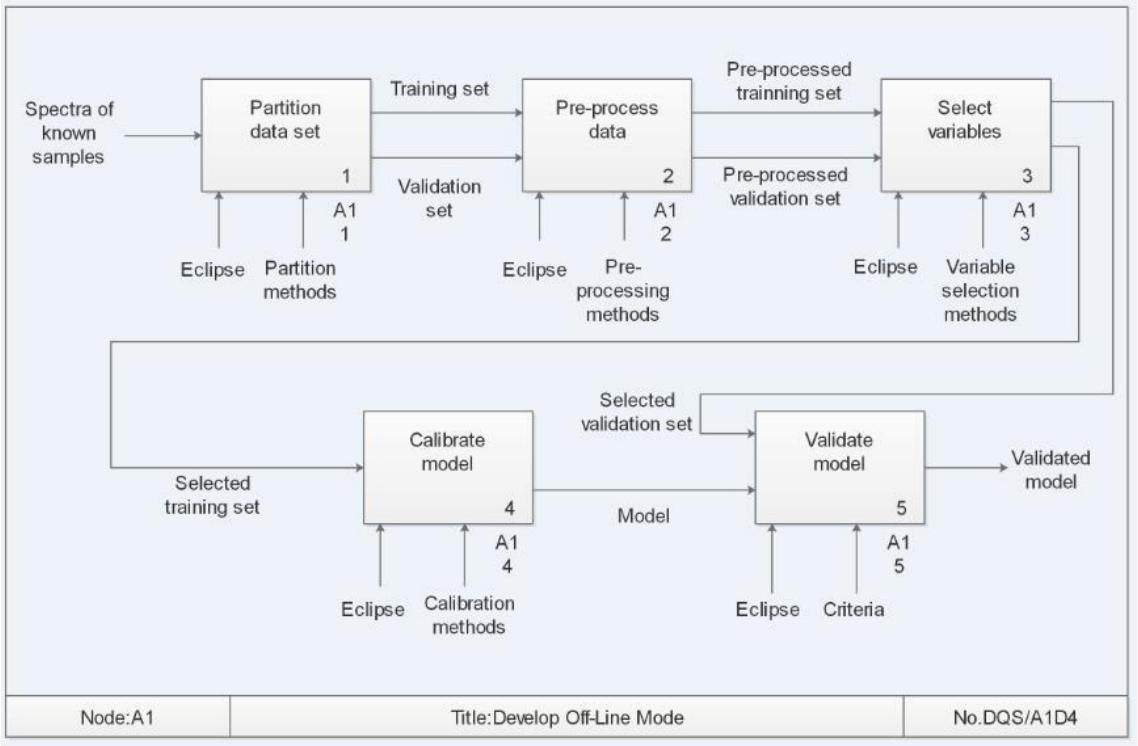


Figure 5.7: The IDEF0 diagram of A1 (Develop Off-Line Mode).

5.3.4 On-Line Mode

Figure 5.8 is the IDEF0 diagram of A2 (Develop On-Line Mode) which provides a real-time quantitative NIRS data analysis for unknown samples. This online mode has three steps, and they can be performed automatically in this sequence:

1. The spectra of unknown samples are pre-processed by pre-processing methods provided by the chemometric methods library.
2. The model established in A1 is used to predict sample property for the pre-processed spectra by calibration methods provided by the chemometric methods library as well. Calibration methods are the same in both A1 and A2, but the principles are different. In the model building (A1), it is to compute the regression coefficient vector by already known predictor and reference response. In terms of prediction for unknown sample (A2), regression coefficient vector and predictor are already known, and response is to be calculated.
3. Predications are visualised by displaying them on user interface developed by third-party Java package (Scene Builder, free to use).

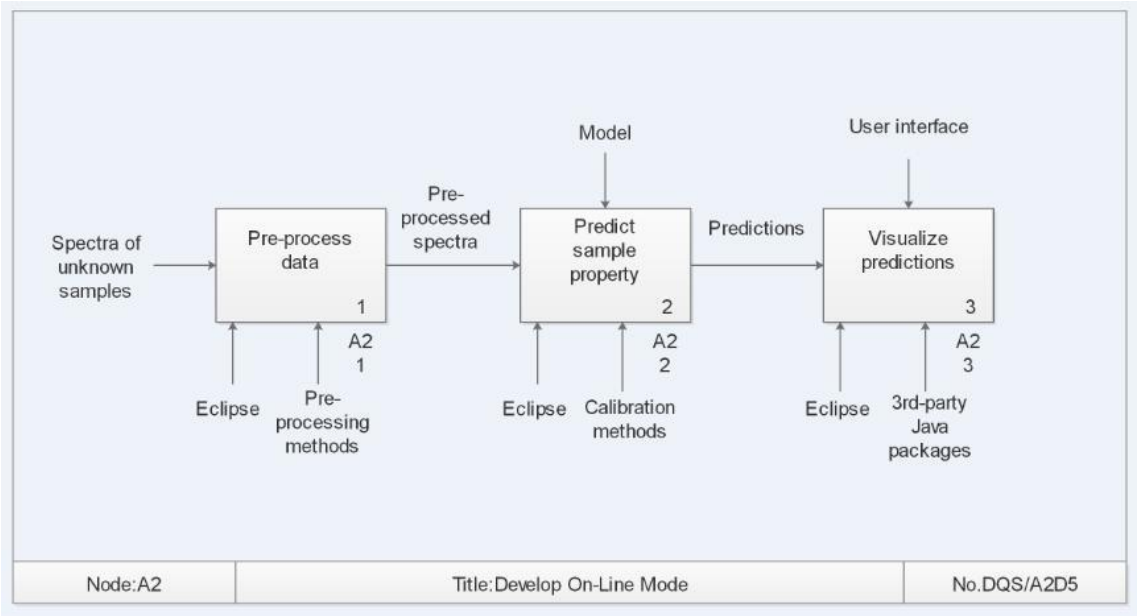


Figure 5.8: The IDEF0 diagram of A2 (Develop On-Line Mode).

5.3.5 Chemometric Methods Library

Figure 5.9 is the IDEF0 diagram of A3 (Develop Chemometric Methods Library), which illustrates how this library outputs chemometric methods. MATLAB is used to program methods and package those methods into Java-recognizable files (.jar file) one for each method. Whenever chemometric methods are required, those relevant files are invoked to provide chemometric methods.

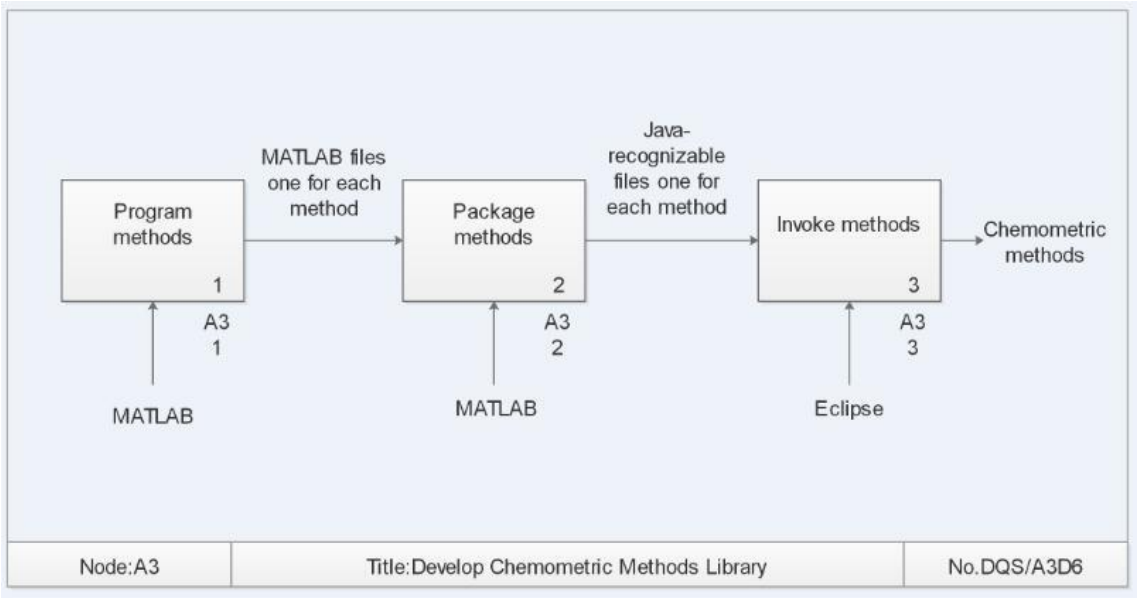


Figure 5.9: The IDEF0 diagram of A3 (Develop Chemometric Methods Library).

5.3.6 Data Management

Figure 5.10 is the IDEF0 diagram of A4 (Manage Data), which presents how the QNIRSA system imports and exports spectral data. The QNIRSA system is currently for research used in the laboratory, so the Excel document is determined to store the spectral data by storing the spectral matrix. Details about spectral matrix are explained in section 2.2.3. A third-party Java package (Apache POI, open-source) supports Java to access Excel files. We gave a try to designed an Entity-Relationship diagram for the NIR spectral database displayed in figure 5.11 [1].

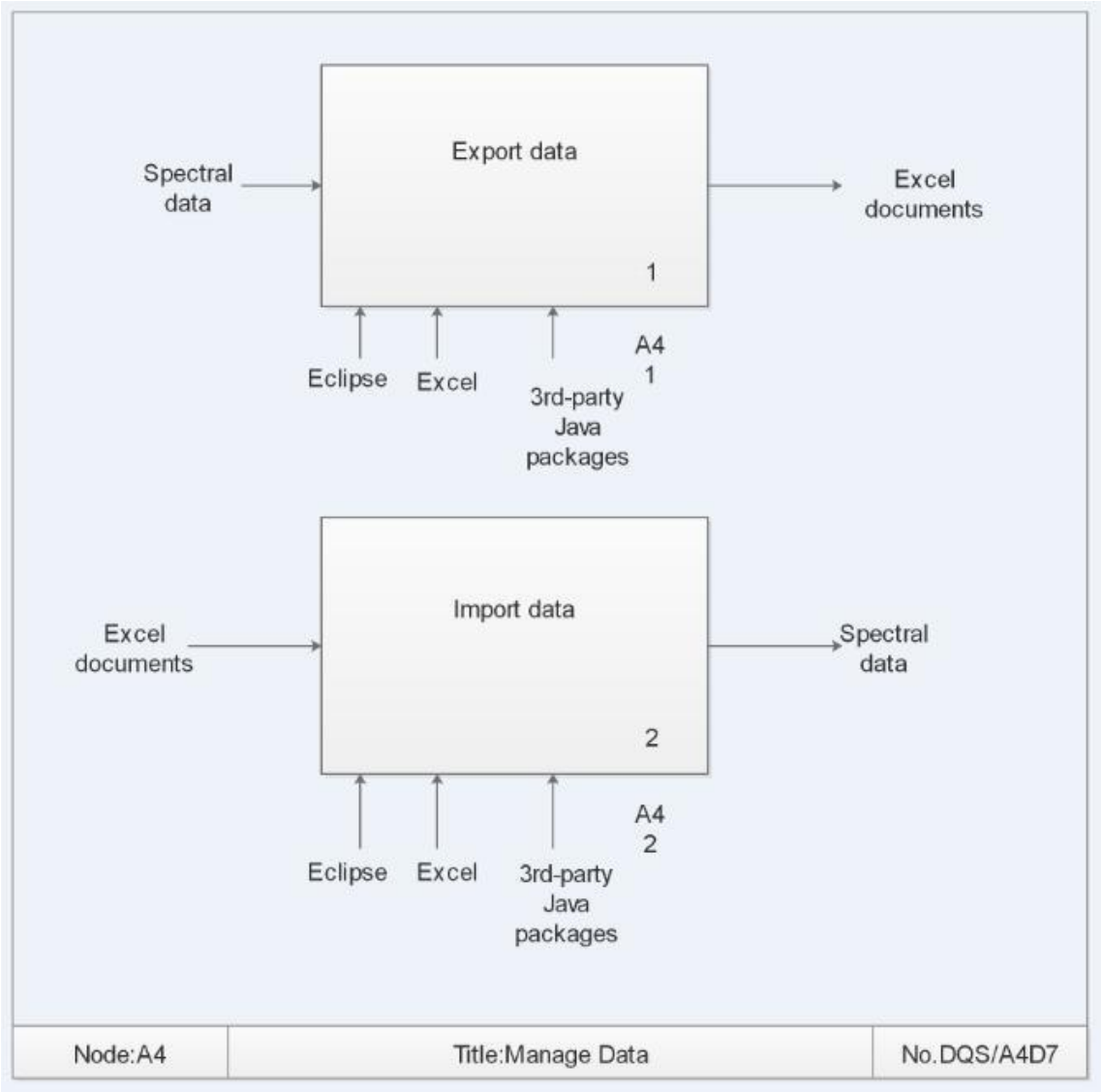


Figure 5.10: The IDEF0 diagram of A4 (Manage Data).

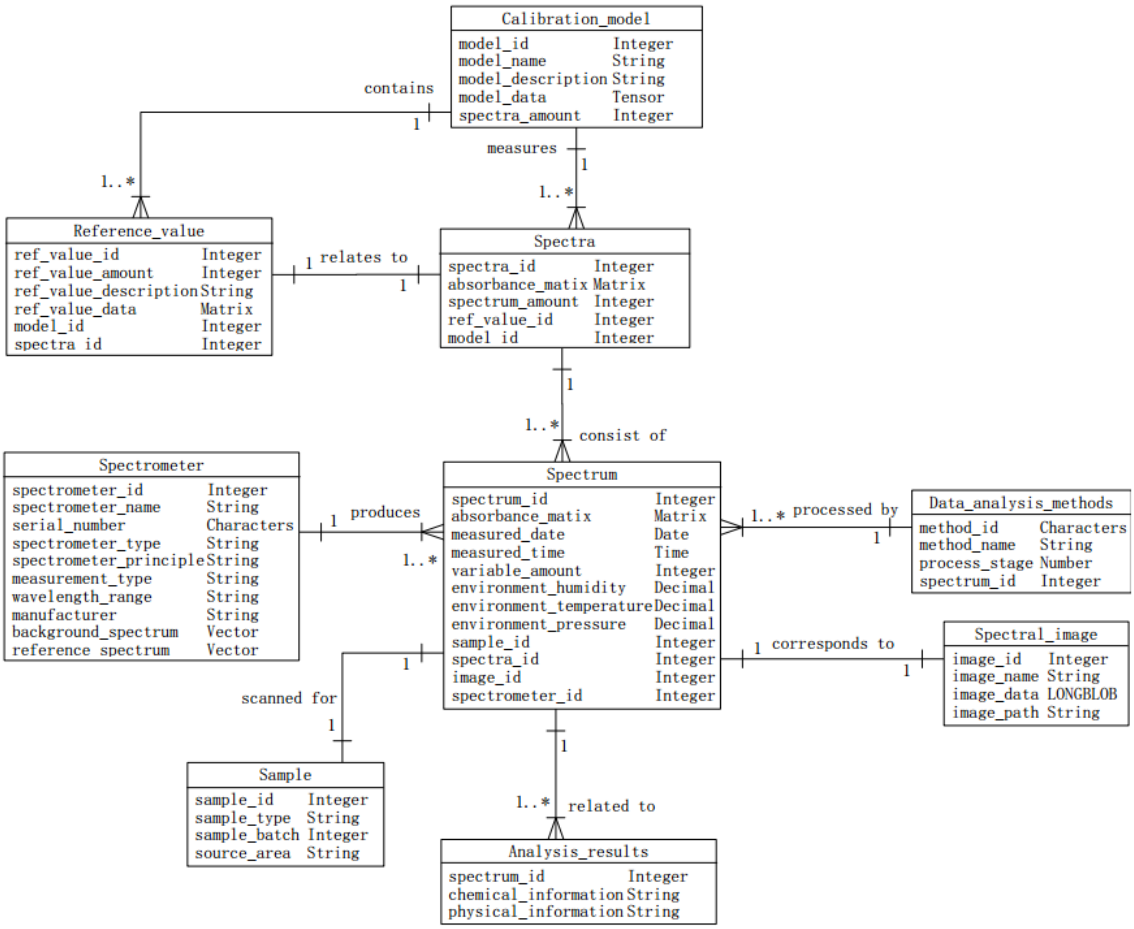


Figure 5.11: The Entity-Relationship Diagram for the NIR spectral database.

5.4 Implementation of the QNIRSA System

The QNIRSA System has three editions. The first edition was reported in the first publication mentioned in this thesis with respect to online mode. The second edition refers to the second and third publications. As for these two editions, MATLAB 2015a (The Mathworks, USA) was used for MATLAB programming while Eclipse Oxygen (Eclipse Foundation, open-source) was used for Java programming with the Java Development Kit (JDK) environment 1.7. Three third-party Java packages RXTX (open source), Scene Builder (free to use) and Apache POI (open source) were used for serial port communication, user interface development, Excel file management respectively.

Figure 5.12 is the user interface (UI) dashboard consisting of 6 areas. On the top of the UI, there are three areas Input, Methods and Predictions respectively from left to right. The Input area is a set of parameter required by the spectrometer. The Methods area provides some pre-processing methods and calibration methods for a user to choose. The Predictions area displays all prediction results. The area containing buttons next to the Predictions area is for running or stopping the on-line mode. In terms of the chart area, the left chart area depicts the real-time spectrum of a single sample, while the right chart area presents the fluctuation of predictions. When this dashboard is opened, user types inputs and selects desired methods in corresponding areas. Then clicks on the

‘Auto run’ button to activate the QNIRSA system and stops the system by clicking on the ‘stop’ button. Besides, inputs may be different for a different type of spectrometers, but current inputs are enough for three categories of spectrometers, which will be mentioned in chapter 7.

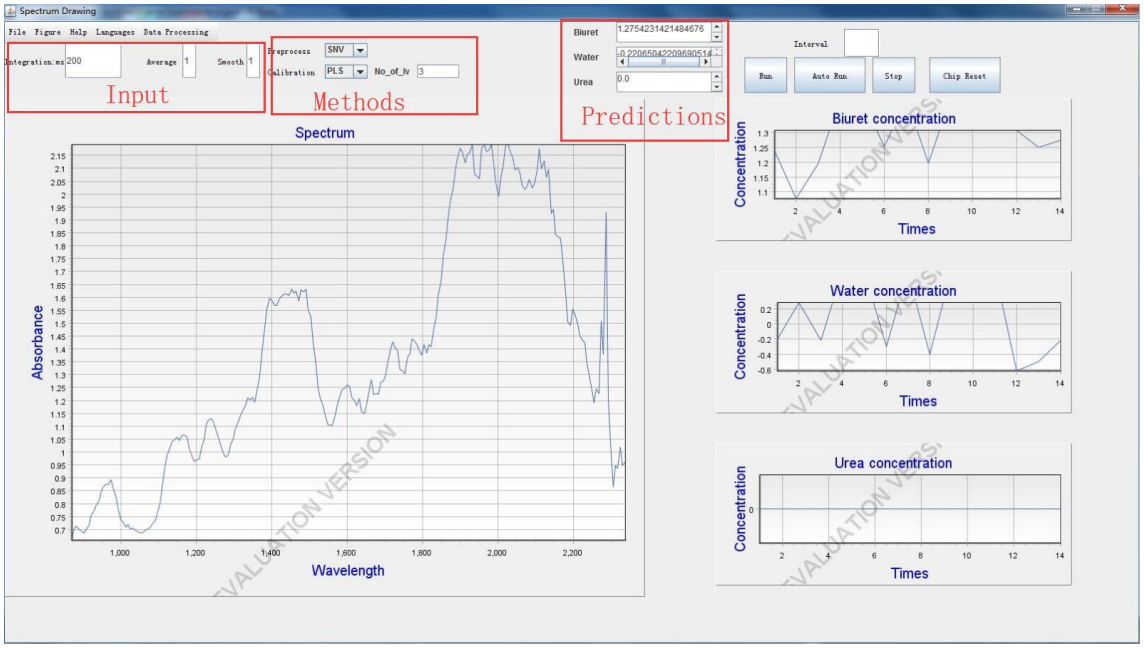


Figure 5.12: The user interface dashboard.

In order to make codes more readable and update the runtime environment of the QNIRSA system, the third edition is under implementation. Details can be found in Appendix 1.

5.5 Summary

The proposed architecture of the QNIRSA system has been illustrated, and this system was modularised and developed by functional modelling through IDEF0 approach. This system is possible for many kinds of agricultural samples because it can control the sample-specific hardware system. Implementation of user interface dashboard has been presented, and full codes can be found in appendix 1.

6 BENCHMARK RESULTS ANALYSIS

6.1 Overview

This chapter analyses the measurement and comparison between chemometric methods, referring to chapter 4. Local goals are analysed in section 6.2, and global goals are discussed in section 6.3.

6.2 Results Analysis for Local Goals

6.2.1 Determinations of Parameters of Methods

All necessary parameters of every method and their possible values for selection are displayed in Table 6.1. All of these parameters' values have been chosen based on previous research and related works, which have been mentioned in section 2.2 and chapter 3 already. Relevant key references are cited in Table 6.1 as well.

In order to provide a fair comparative study for the different chemometric methods, each method should be optimised independently. Although the computing time for different optimisations can be rather large, only the final (the best possible) results can be compared without bias. In other words, only the best results of every method or the combination of multiple methods will be compared. The best result will be regarded as the optimal result of those methods. Table 1, 2 and 3 in appendix 2.1, 2.2 and 2.3 display the optimal results of every combination of multiple methods, and all necessary parameters' values selected for those combinations (all fixed parameters' values can be found in Table 6.1, but tables in appendix only show the selection of parameters' values which have multiple possible values in Table 6.1).

Table 6.1 Parameters of Methods

Methods	Parameters	Selected Values	References
KS	Percentage of the training set and validation set	70% for training set while 30% for the validation set	[35]
SPXY	percentage of the training set and validation set	70% for training set while 30% for the validation set	[36]
MSC	reference spectrum to revise original spectra	the average spectrum of the training set	[16], [2]
SNV	no extra parameters		
SG 1st_der.	order of spectral derivative	1	
	order of polynomial fit (opf)	1 - 6	
	length of the smoothing window	17	
SG 2nd_der.	order of spectral derivative	2	
	order of polynomial fit (opf)	2 - 6	
	length of the smoothing window	17	
SPA	no extra parameters		[17], [25], [19]
UVE	an arbitrary value to control the cut-off	0.1- 1.5	
SA	initial energy/error value	5	
	cooling ratio (cr)	0.5 - 0.9	
	the initial value of T	0.001	
	the final value of T	10^{-6}	
iPLS	the desired number of sub-intervals (si)	3 - 200	
FiPLS	the desired number of sub-intervals (si)	3 - 200	
BiPLS	the desired number of sub-intervals (si)	3 - 200	
GA	number of chromosomes	100	
	maximum number of generations	200	
	probability of single-point crossover (pspc)	0.5 - 0.9	
	probability of mutation (pm)	0.05 - 0.1	
MLR	no extra parameters		[60], [19]
PCA & PCR	the number of selected principal components (pc)	decided by the percentage of the total variance explained by each principal component	
PLSR	the number of selected PLS components (PLSc)	decided by the percentage of the total variance explained by each PLS component	

6.2.2 Assessment for Dataset Partition Methods

Table 6.2 displays the descriptive statistics for protein content of single rice kernels by using three sampling methods, of which one is the reference method for dataset partition (RMDP) [2], and other two are KS and SPXY. As for the reference partition method, the protein content of the calibration set is distributed between 6.0 and 13.1, covering most of the protein content of the validation set except the minimum protein content. The means, standard deviation (SD), and standard errors of the mean (SEM) between calibration set and external validation set are closed, indicating that the protein distribution of the validation samples was well represented by the calibration set. In terms of the KS, the protein content of the calibration set is distributed between 5.6 and 12.8, covering most of the protein content of the validation set except the maximum protein content. The means, standard deviation (SD), and standard errors of the mean (SEM) between calibration set and external validation set are closed, indicating that the protein distribution of the validation samples was well represented by the calibration set. Regarding the SPXY, the protein content of the calibration set is distributed between 5.6 and 13.1, entirely covering the protein content of the validation set. The means, standard deviation (SD), and standard errors of the mean (SEM) between calibration set and external validation set are closed, indicating that the protein distribution of the validation samples was well represented by the calibration set. In summary, from the perspective of these statistics for protein content, SPXY＞KS≈RMDP.

Table 6.2: Descriptive statistics for the protein content of single rice.

Method	Training set					Validation set				
	Sample size	Range	Mean	SD	SEM	Sample size	Range	Mean	SD	SEM
RMDP	149	6.0-13.1	9.18	1.394	0.114	52	5.6-12.8	8.72	1.751	0.234
KS	149	5.6-12.8	9.00	1.490	0.122	52	6.0-13.1	9.25	1.540	0.217
SPXY	149	5.6-13.1	9.12	1.529	0.125	52	5.9-13.1	8.91	1.426	0.200

Figure 6.1 presents the values of RMSEP for three dataset partition methods on three forms of single rice (smaller RMSEP reveals the better performance the method made.). A first observation is that both KS and SPXY have better performance than RMDP. The second general comment is KS has smaller RMSEP than SPXY on single kernel data set SRK and SBK, while SPXY is better on rice flour. The RMSEP difference between KS and SPXY is small. Two conclusions are 1) both KS and SPXY are useful sampling methods for single rice spectral data; 2) KS may be better for single kernel data set, while SPXY may be better for rice flour.

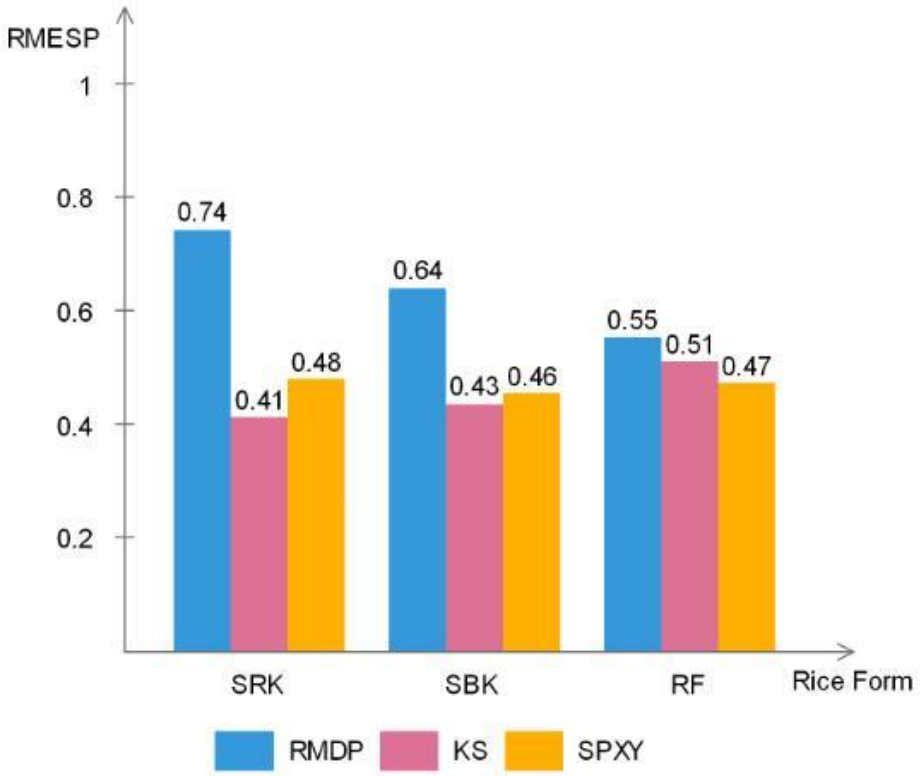


Figure 6.1: RMSEP values for three sampling methods on three forms of single rice.

6.2.3 Assessment for Pre-processing Methods

Figure 6.2 shows the values of RMSEP for four pre-processing methods on three forms of single rice (smaller RMSEP reveals the better performance the method made.). The first general observation is that most of the pre-processing methods do not provide improving performance. The average DRMSEP (in table 6.3) on SRK, SBK and RF are 0.0129, 0.0041 and 0.0642 respectively (positive DRMSEP means the relevant chemometric methods are ineffective, while negative DRMSEP means the relevant chemometric methods are effective), indicating that pre-processing makes few impacts even detrimental to model. The reason may be that PLSR considers both predictor and response for regression so that pre-processing cannot give impressive improvement for the PLSR model. However, the optimal pre-processing method is positive for SRK, SBK and RF. A significant example is that SNV has excellent performance on SBK. Therefore, conclusions are 1), not all pre-processing methods are useful for single rice spectral data; 2) from the average DRMSEP perspective, pre-processing cannot make a significant improvement to model, but there may be one pre-processing method is entirely appropriate for a specific single rice data set. Pre-processing methods still deserve a try.

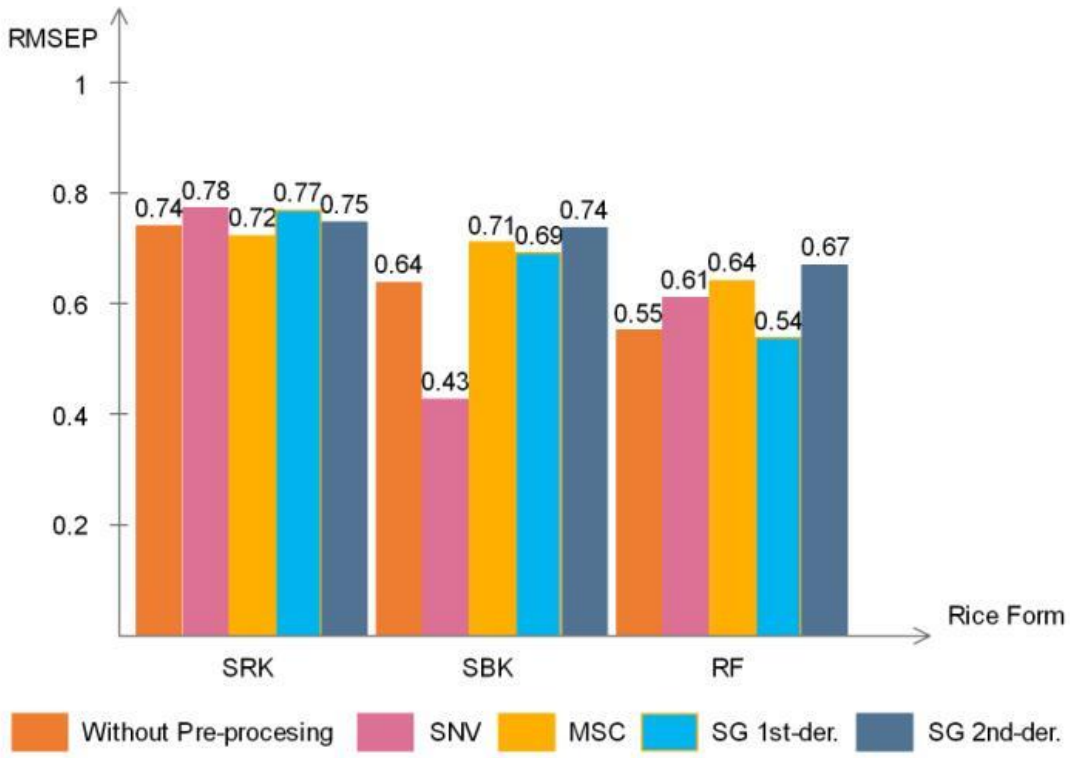


Figure 6.2: RMSEP values for four pre-processing methods on three forms of single rice.

Table 6.3: The optimal and average RMSEP and DRMSEP for pre-processing on three forms of single rice.

Rice Form	Step	Optimal			Average	
		RMSEP	DRMSEP	Method	RMSEP	DRMSEP
SRK	Pre-processing	0.7239	-0.0179	MSC	0.7547	0.0129
SBK	Pre-processing	0.4284	-0.2109	SNV	0.6343	0.0041
RF	Pre-processing	0.5438	-0.0093	SG-1st der.	0.6173	0.0642

6.2.4 Assessment for Variable Selection Methods

Figure 6.3 shows the RMSEP for seven variable selection methods without pre-processing on three forms of single rice (Smaller RMSEP reveals the better performance the method made). All variable selection methods are more or less effective for improving the model. Quantified evidence is the average DRMSEP for seven variable selection methods in table 6.4 (positive DRMSEP means the relevant chemometric

methods are ineffective, while negative DRMSEP means the relevant chemometric methods are effective). All the values of average DRMSEP are negative, indicating the variable selection is useful for improving the model. UVE may be the best variable selection method because it is the optimum for SRK and RF. Additionally, RMSEP of UVE is 0.485 on SBK, which is closed to 0.45, the RMSEP of GA, which is optimum on SBK. Conclusions are 1) variable selection is a useful step to improve model; 2) without pre-processing UVE may be the best variable selection method for single rice spectral data.

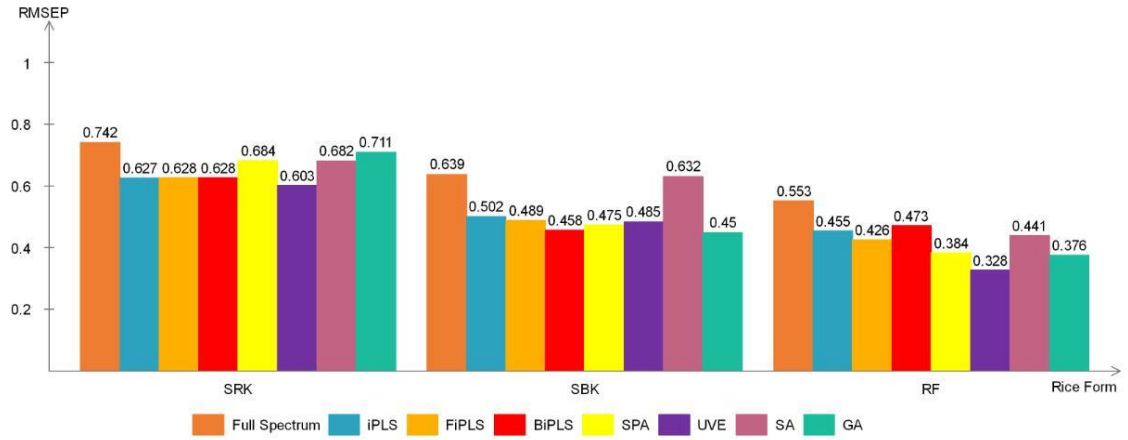


Figure 6.3: RMSEP values for seven variable selection methods without pre-processing on three forms of single rice.

Table 6.4: The optimal and average RMSEP and DRMSEP for pre-processing on three forms of single rice.

Rice Form	Step	Optimal			Average	
		RMSEP	DRMSEP	Method	RMSEP	DRMSEP
SRK	Variable selection	0.6033	-0.1384	UVE	0.6519	-0.0899
SBK	Variable selection	0.4502	-0.1891	GA	0.4988	-0.1405
RF	Variable selection	0.3279	-0.2252	UVE	0.4118	-0.1413

Figure 6.4 displays the RMSEP for seven variable selection methods with pre-processing on SRK (diagram (a)), SBK (diagram (n)) and RF (diagram (c)) respectively. A global observation for three diagrams is that the RMSEP of all variable selection methods decreased more or less when pre-processing methods were used in advance. So pre-processing plays an assist role in quantitative NIRS analysis. When pre-processing methods were used alone, no distinct improvement they made to the PLSR model.

However, the combination of pre-processing and variable selection shows a further improvement to the model compared with either using pre-processing alone or variable selection alone.

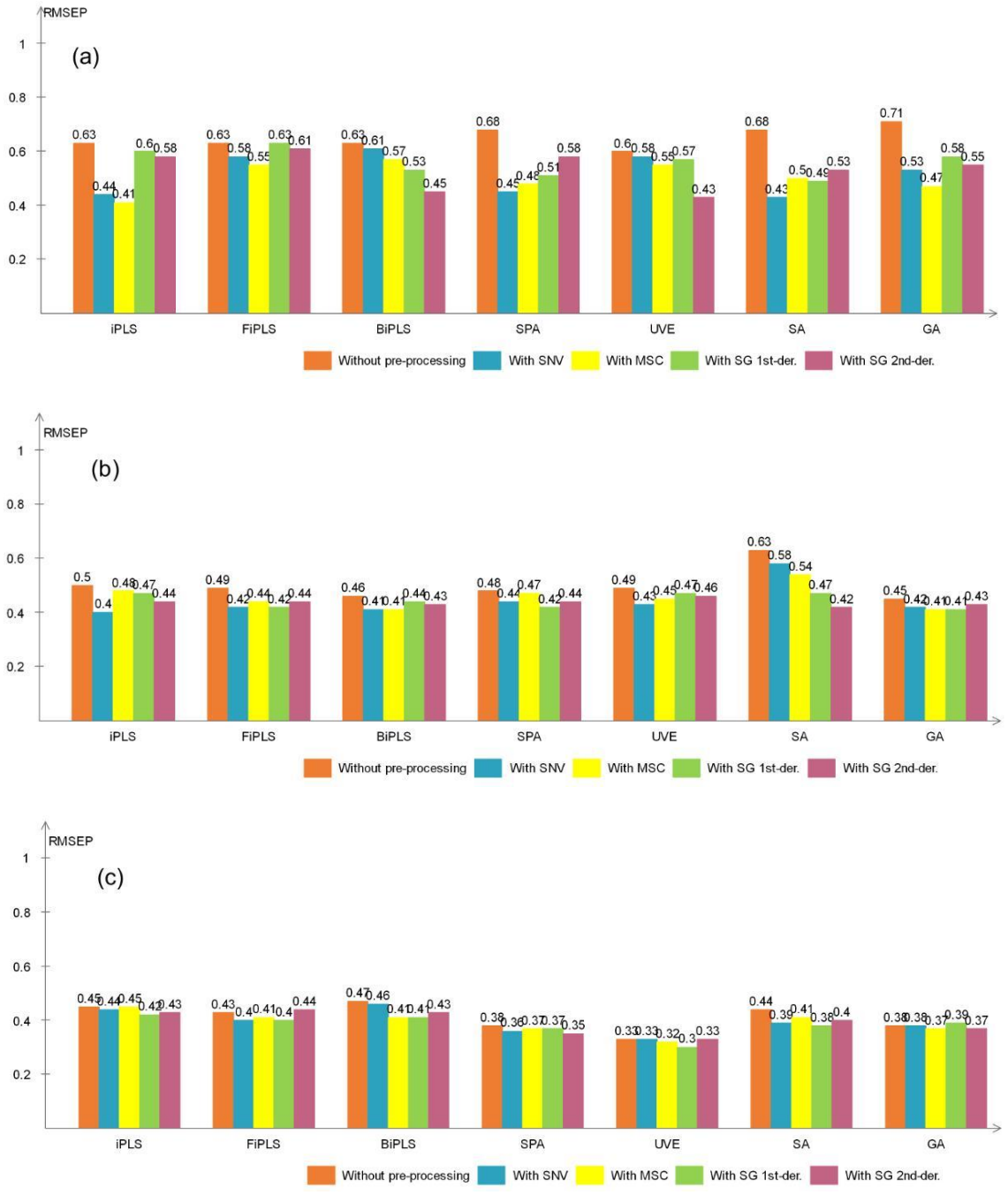


Figure 6.4 RMSEP for seven variable selection methods with pre-processing on SRK (a), SBK (b) and RF (c).

Table 6.5 presents optimal and average DRMSEP values for three steps on SRK, SBK and RF, respectively (positive DRMSEP means the relevant chemometric methods are ineffective, while negative DRMSEP means the relevant chemometric methods are effective). According to the average values, dataset partition is the most significant step

on SRK and SBK, but the variable selection is more critical on RF. As for optimal DRMSEP, dataset partition made the most significant improvement on SRK, but the optimal for three steps are similar on SBK, while variable selection made an outstanding improvement on RF.

Table 6.5 Optimal and average DRMSEP for three steps on SRK, SBK and RF. e

Rice Form	Step	DRMSEP	
		Optimal	Average
SRK	Dataset Partition	-0.3294	-0.2954
	Pre-processing	-0.0179	-0.0129
	Variable Selection	-0.1384	-0.0899
SBK	Dataset Partition	-0.2042	-0.1945
	Pre-processing	-0.2109	-0.0041
	Variable Selection	-0.1891	-0.1405
RF	Dataset Partition	-0.0805	-0.0615
	Pre-processing	-0.0093	-0.0642
	Variable Selection	-0.2252	-0.1413

6.2.5 Assessment for Calibration Methods

Figure 6.5 depicts the RMSEP for three calibration methods on three forms of single rice (smaller RMSEP reveals the better performance the method makes). Obviously, for three rice forms, the rank of performance of three calibration methods is PLSR>PCR>>MLR. That is the reason why PLSR is the most common calibration method used for NIRS. The poor performance of MLS should be caused by the multi-collinearity when the number of variables of spectra data is much more than the number of samples.

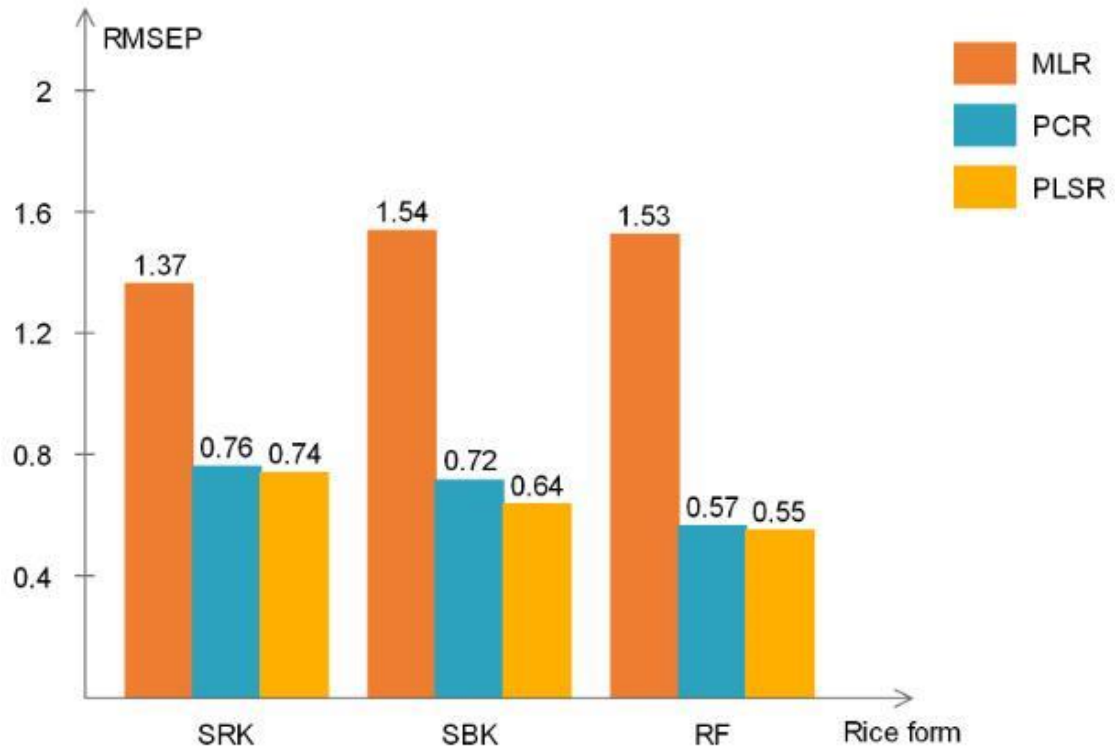


Figure 6.5: RMSEP for three calibration methods on three forms of single rice.

6.3 Results Analysis for Global Comparison

6.3.1 Effective Combinations of Methods

After a large number of tests for all combinations of 16 methods, three tables are used for presenting the performance of all combinations of methods on SRK, SBK, and RF, respectively. Table 6.6, 6.7 and 6.8 present the top 15 performance of all combinations of methods on SRK, SBK and RF respectively (PDRMSEP is the percentage of decrement of RMSEP that method combination made compared with the RMSEP of the global reference model). Full results can be found in Table 1 in Appendix 2.1, Table 2 in Appendix 2.2 and Table 3 in Appendix 2.3. Full results were attached in Appendix 2. The best model on SRK was built by a combination of KS, MSC, UVE and PLSR. The best model on SBK was built by a combination of KS, SNN, UVE and PLSR. The best model on RF was built by a combination of SPXY, MSC, GA and PLSR. Though performances of several combinations of methods were even worse than the performance of some single methods, generally speaking, the performance of most combinations of methods are better than the reference models in table 4.1.

Table 6.6 Top 15 of the performance of all combinations of methods on SRK.

Combination ID	Rice Form	Methods				Assessment	
		Dataset Partition	Pre-processing	Variable Selection	Calibration	RMSEP	PDRMSEP
1	SRK	KS	MSC	UVE	PLSR	0.2652	64.25%
2	SRK	SPXY	MSC	UVE	PLSR	0.2977	59.87%
3	SRK	KS	MSC	SPA	PLSR	0.3011	59.41%
4	SRK	KS	SNV	SPA	PLSR	0.3073	58.57%
5	SRK	SPXY	SNV	UVE	PLSR	0.3073	58.57%
6	SRK	SPXY	MSC	GA	PLSR	0.3125	57.87%
7	SRK	KS	SNV	UVE	PLSR	0.3128	57.83%
8	SRK	KS	SG 1st_der.	SA	PLSR	0.3145	57.60%
9	SRK	SPXY	SNV	SA	PLSR	0.3145	57.60%
10	SRK	KS	MSC	GA	PLSR	0.3214	56.67%
11	SRK	SPXY	MSC	SA	PLSR	0.3241	56.31%
12	SRK	KS	SG 1st_der.	UVE	PLSR	0.3279	55.80%
13	SRK	KS	SNV	GA	PLSR	0.3284	55.73%
14	SRK	KS	MSC	SA	PLSR	0.3294	55.59%
15	SRK	KS	SNV	SA	PLSR	0.3341	54.96%

Table 6.7 Top 15 of the performance of all combinations of methods on SBK.

Combination ID	Rice Form	Methods				Assessment	
		Dataset Partition	Pre-processing	Variable Selection	Calibration	RMSEP	PDRMSEP
1	SBK	KS	SNV	UVE	PLSR	0.2776	56.58%
2	SBK	KS	SNV	GA	PLSR	0.2874	55.04%
3	SBK	SPXY	SNV	GA	PLSR	0.2874	55.04%
4	SBK	KS	SNV	SA	PLSR	0.2942	53.98%
5	SBK	SPXY	SNV	UVE	PLSR	0.3031	52.59%
6	SBK	KS	MSC	GA	PLSR	0.3074	51.92%
7	SBK	SPXY	SNV	SA	PLSR	0.3075	51.90%
8	SBK	SPXY	SNV	SPA	PLSR	0.3127	51.09%
9	SBK	KS	SNV	SPA	PLSR	0.3154	50.66%
10	SBK	KS	MSC	SA	PLSR	0.3171	50.40%
11	SBK	SPXY	MSC	GA	PLSR	0.3179	50.27%
12	SBK	SPXY	SNV	BiPLS	PLSR	0.3247	49.21%
13	SBK	KS	MSC	SPA	PLSR	0.3341	47.74%
14	SBK	KS	SNV	BiPLS	PLSR	0.3354	47.54%
15	SBK	SPXY	MSC	SA	PLSR	0.3354	47.54%

Table 6.8 Top 15 of the performance of all combinations of methods on RF.

Combination ID	Rice Form	Methods				Assessment	
		Dataset Partition	Pre-processing	Variable Selection	Calibration	RMSEP	PDRMSEP
1	RF	SPXY	MSC	GA	PLSR	0.3187	42.38%
2	RF	SPXY	SNV	SA	PLSR	0.3274	40.81%
3	RF	SPXY	SNV	UVE	PLSR	0.3339	39.63%
4	RF	KS	SNV	SA	PLSR	0.33571	39.30%
5	RF	KS	MSC	UVE	PLSR	0.3374	39.00%
6	RF	KS	MSC	GA	PLSR	0.3387	38.76%
7	RF	KS	SNV	UVE	PLSR	0.3388	38.75%
8	RF	KS	MSC	SA	PLSR	0.3427	38.04%
9	RF	SPXY	SNV	SPA	PLSR	0.3471	37.24%
10	RF	SPXY	MSC	SA	PLSR	0.3478	37.12%
11	RF	SPXY	SG 1st_der.	GA	PLSR	0.3487	36.96%
12	RF	SPXY	MSC	UVE	PLSR	0.3505	36.63%
13	RF	SPXY	SG 2nd_der.	SA	PLSR	0.3517	36.41%
14	RF	KS	SNV	GA	PLSR	0.3571	35.44%
15	RF	KS	MSC	SPA	PLSR	0.3571	35.44%

6.3.2 Classify the Effective Combinations

Based on Tables 6.6, 6.7 and 6.8, all combinations of methods on SRK can be divided into four classes based on PDRMSEP relatively.

1. *Highly effective combinations*: PDRMSEP is higher than 60%.
2. *Well effective combinations:* PDRMSEP is from 50% to 59.9%.
3. *Effective combinations:* PDRMSEP is from 40% to 49.9%.
4. *Low effective combinations:* PDRMSEP is lower than 40%.

Therefore, for SRK in Table 6.6, Combination 1 is highly effective, Combinations 2-35 is well effective, and 36-56 is effective. In terms of SBK in table 6.7, Combinations 1-11 is well-effective, 12-33 is effective, and 34-56 is low effective. Regarding RF in table 6.8, combinations 1 and 2 are effective. The rest of the combinations are all low effective combinations. The effectiveness is relative so that low effectiveness dose not equal to ineffectiveness. There are 91 combinations of methods are effective or higher than effective. Figure 6.6 shows pie charts for the percentages of the number of effective combinations, which contains a specific method in total combinations. Pie chart (a), (b) and (c) are for dataset partition, pre-processing and variable selection methods respectively. In terms of sampling, the effective combinations containing SPXY is a bit more than that of KS, but they are closed and no clear evidence to judge which one is better. Effective combinations containing SNV is the most in pre-processing, followed by MSC. MSC and SNV should be a wide choice for pre-processing single rice spectral data. The percentage of effective combinations containing each variable selection method is closed, indicating no outstanding variable selection method over others. GA and SA share the first place in variable selection with a lead of 2% over the third method SPA and iPLS.

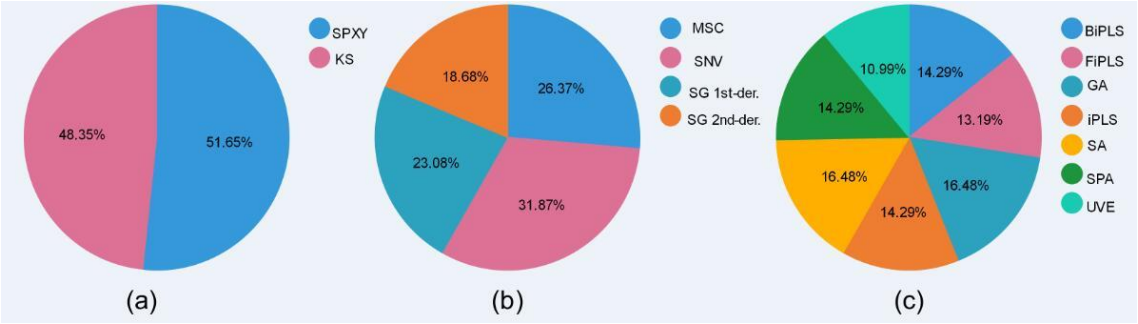


Figure 6.6: Percentages of methods among effective combinations.

6.4 Summary

6.4.1 Summary of Results Analysis

Specific assessments of combinations of methods have been done for SRK, SBK and RF, respectively. The following general conclusions can be drawn:

1. Both KS and SPXY are useful dataset partition methods for single rice spectral data. From the perspective of statistics for protein content in table 7, SPXY > KS. Nevertheless, if based on the RMSEP, the performances of SPXY and KS are closed. There is no clear evidence to judge which one is better.
2. Variable selection is a useful step to improve the PLSR model. The use of pre-processing methods before variable selection methods provides a further improvement for the PLSR model.
3. Dataset partition step has the most significant impact on the PLSR model.

4. Calibration methods ranking: PLSR>PCR>>MLR.
5. MSC and SNV should be a wide choice for pre-processing single rice spectral data.
6. Time consumed for the optimization variable selection methods: BiPLS (sub-intervals more than 80)>SA>GA> SPA>FiPLS>iPLS>UVE. BiPLS is not recommended when sub-intervals are more than 80. In that circumstance, BiPLS is quite time-consuming but cannot have leading superiority compared with other variable selection methods.

6.4.2 Contributions of the Benchmarking Works

The Benchmarking analysis works in chapter 6 constructed a comparative study for various statistical models established by sixteen chemometric methods on the single rice sample as the benchmark of chemometric methods for single kernel near-infrared spectroscopy analysis. Compared with previous work which focused on the analysis of bulk rice samples, this exploratory research in addressing single rice samples leads to a more accurate assessment of properties such as protein content. This study is also an example and guidance to show analytical processes for assessing different statistical models on single kernel samples. Those processes of benchmarking can guide future research on how to design and implement the process on single kernel NIR analysis. Additionally, the comparative results of those models are a useful reference for chemometric methods selection on single rice samples. They provide detailed assessments of sixteen methods, including not only the performance of those methods on three forms of single rice but also the optimal parameters of methods. Relevant research may briefly refer to this comparative study when it is related to methods selection, parameters tuning or model calibration on single rice samples.

Specifically, the benchmarking processes specified in chapter 4 and analytical processes presented in chapter 6 are representative examples of SKNIRS. They showed how to design coherent processes from data collection to statistical model assessment. This would help future researchers to design the process and experiment to select appropriate models on single kernel samples. In terms of single rice samples, those findings in chapter 6 are a useful reference for chemometric methods selection and statistical model assessment. For example, in section 6.4.1, the first finding suggests applying dataset partition method either SPXY or KS in before other methods, because results show both SPXY and KS made a crucial improvement on model performance. One more example is that the comparison results between calibration methods reveal the PLSR is the best one in the calibration stage because commonly the PLSR model has better performance than other calibration models.

Details about how the benchmarking works impacted two real-world applications and helped my collaborators will be specified in section 7.3.

7 TWO REAL-WORLD APPLICATIONS OF THE BENCHMARKING RESULTS AND THE QNIRSA SYSTEM

7.1 Overview

The QNIRSA system has been already successfully applied for real-world applications. This chapter reports those two real-world applications as the validation of the QNIRSA system. The first application is 'A calibration transfer optimised single kernel near-infrared spectroscopic method', which has been published [2]. This application will be addressed in section 7.2, which provided a calibration transfer optimised method to accurately detect the chemical composition of single seeds by using the calibration model of the corresponding dehusked seeds or seed flour. Another application is 'Analysis of biuret in urea fertiliser by using a portable near-infrared spectrometer', which introduced a fast and straightforward method for detection of biuret in urea fertiliser using a portable near-infrared spectrometer. The paper refers to this application is still under review [3]. In section 7.3, it will interpret how those benchmarking processes and results helped my collaborators to complete the two real-world applications.

7.2 Validation of the QNIRSA System through Two Real-World Applications

7.2.1 Application 1: A Calibration Transfer Optimized Single Kernel Near-Infrared Spectroscopic Method

The application 'A calibration transfer optimised single kernel near-infrared spectroscopic method' provided a calibration transfer optimised method to accurately detect the chemical composition of single seeds by using the calibration model of the corresponding dehusked seeds or seed flour [2]. The proposed method was applied to the analysis of the protein content of a single rice kernel. The near-infrared transmission spectra of three forms of rice (single rice kernel (SRK), single brown rice kernel (SBK) and rice flour (RF)) of 201 individual rice seeds and the corresponding protein content values were obtained. By comparing different pre-processing methods and spectral ranges, the spectral range 950–1250 nm, the standard normal variate transformation

(SNV), and 9 PLS latent variables were selected to construct the optimal PLSR models. Then, the protein content of single rice kernels was determined through two different methods. The direct method, in which single rice kernels were analysed using the single rice kernel model directly. Another is the proposed method, in which the spectra of single rice kernels were transferred into the spectra of single brown rice kernels and rice flours with a calibration transfer algorithm, spectral space transformation (SST), and were analysed by the respective calibration models. The external validation coefficient correlation (R) value of the direct method was 0.971, and the R values of the proposed method were 0.962 (SBK) and 0.975 (RF). The root mean square error of prediction (RMSEP) value of the direct method was 0.423, and the RMSEP of the proposed method was 0.480 (SBK) and 0.401 (RF). Besides, the transfer results among the spectra of three forms of rice were compared. By comparison, the results of the proposed method are relatively close to the results of the direct method. The results indicate that the spectra generated from one individual rice seed can be transferred freely among the three forms by means of calibration transfer. The proposed method is a promising way to overcome the challenges associated with analysing individual seeds and improving SKNIRS.

Figure 7.1 is the flow chart of single rice kernel protein content near-infrared spectral analysis via two methods. Those processes inside the black box were undertaken by the QNIRSA system. Figure 7.2 is the IDEF0 diagram for this application. The diagram (a) presents the whole application and indicates that not only spectra acquisition, sample pre-processing and chemical analysis for SRK, SBK, RF samples have been done by the QNIRSA system, but also the prediction by the direct method has been completed by this system. One of the outputs from the QNIRSA system, a validation set of SRK spectra is used for the proposed method which has been finished outside the QNIRSA system. Diagram (b) is the decomposition diagram of (a) displaying how the QNIRSA system has been utilised to support the whole application. Firstly, the QNIRSA system controlled the spectrometer, MPA (Bruker, Germany), single-chip microcomputer and other hardware accessories to scan SRK, SBK and RF samples and acquired spectra of them respectively. Secondly, the off-line mode is activated to establish SRK, SBK and RF models respectively by chemometric methods provided by the chemometric methods library. Outputs of the QNIRSA system are predictions by the direct method, SBK model, RF model and validation set of SRK spectra respectively.

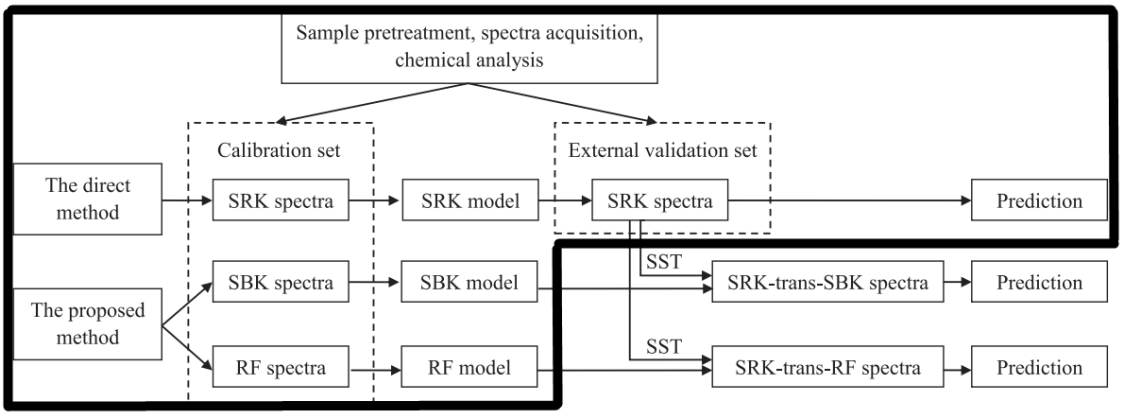


Figure 7.1: Analysis of single rice kernel protein content via two methods [2].

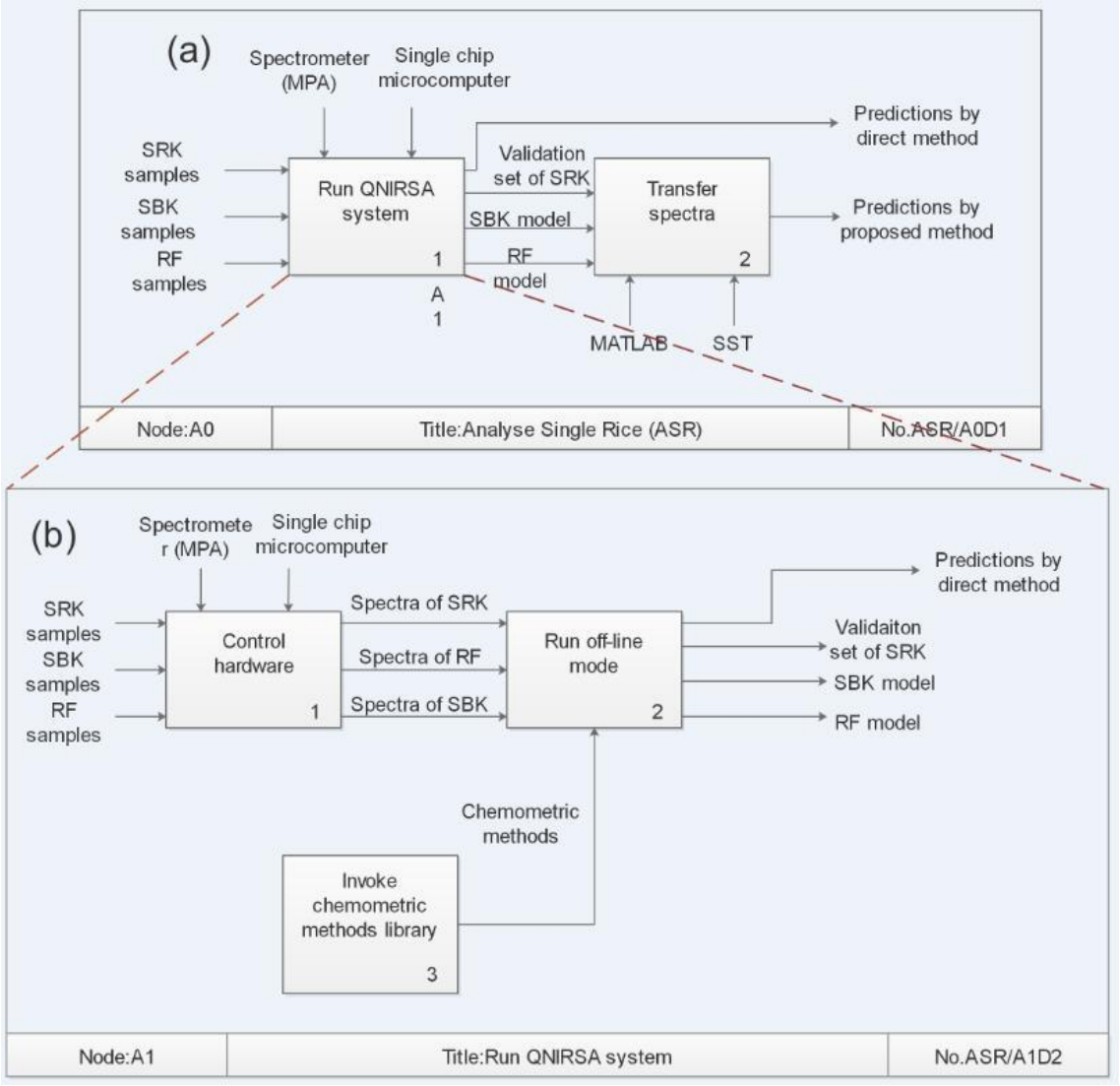


Figure 7.2: IDEF0 diagram of Application 1.

7.2.2 Application 2: Analysis of Biuret in Urea Fertilizer by Using a Portable Near-Infrared Spectrometer

This application is about analysing biuret in urea fertiliser using a portable near-infrared spectrometer [3]. One hundred thirty-five bulk samples of urea were collected, and reference biuret concentrations which ranged from 0.7% to 1.8% were measured by the AOAC Official Method 960.04 [101]. Some kernels weighing about 200 grams made up of every bulk sample. As for each bulk sample, 5 grams of it was used for biuret determination by reference method, while 150 grams of it was used for NIRS analysis. The concentration of the biuret is expressed in mass fraction varying from 0.7% and 1.8%. The mean and standard deviation value are 1.17% and 0.21% respectively. Spectral data were acquired by the QNIRSA system with a portable near-infrared spectrometer, NIRQuest512/1.7 (Ocean optics, USA). The spectral range is 900-1700nm. Outliers were eliminated by employing PLSR combining with a robust modelling strategy. The proposed method was used to determine biuret at concentrations from 0.78% to 1.78% in the calibration set and validation set. Experiment results have shown that the coefficient (R) of external validation set is 0.9868 with root mean square error (RMSE) of 0.0342, the ratio of performance deviation (RPD) value in calibration and validation set is 12.94 and 4.95 respectively. So, it can be concluded that this method can be potentially used as an alternative to traditional wet chemical methods due to its simplicity, sensitivity, and portability.

This paper provided a comparative experiment between the portable spectrometer and another commercial spectrometer, AOTF-3075 (Brimrose, USA), in order to evaluate the performance of that portable spectrometer. Therefore, so far, there are three spectrometers have been used with the QNIRSA system. It proves that the QNIRSA system is available for multiple spectrometers if they provide API. Besides the urea sample and rice sample involved in these two completed applications, other categories of agricultural samples such as corn and wheat have already been tested with the support from different hardware accessories. It gave the evidence to prove that the QNIRSA system can control multiple sample-specific hardware systems if they provide commands.

Figure 7.3 shows how the QNIRSA system supports this application. Diagram (a) is the IDEF0 diagram for the whole application while diagram (b) is the decomposition diagram of (a). By controlling hardware, urea samples were scanned, and the spectra of them were obtained. The off-line mode was activated to process the spectra of urea samples and produce the results by the proposed method provided by the chemometric methods Library. These steps were repeated for two spectrometers, respectively, for the comparative study. Besides, this application employed the sampling error profile analysis (SEPA) method, which was out of original chemometric methods library. Therefore, this method has been programmed and packaged by MATLAB. Then a Java-recognizable file for the SEPA is imported into that chemometric methods library before running the off-line mode. Generally, in terms of other methods which are out of

the current chemometric methods library, the QNIRSA system is possible to use them if the Java-recognizable files of them are imported.

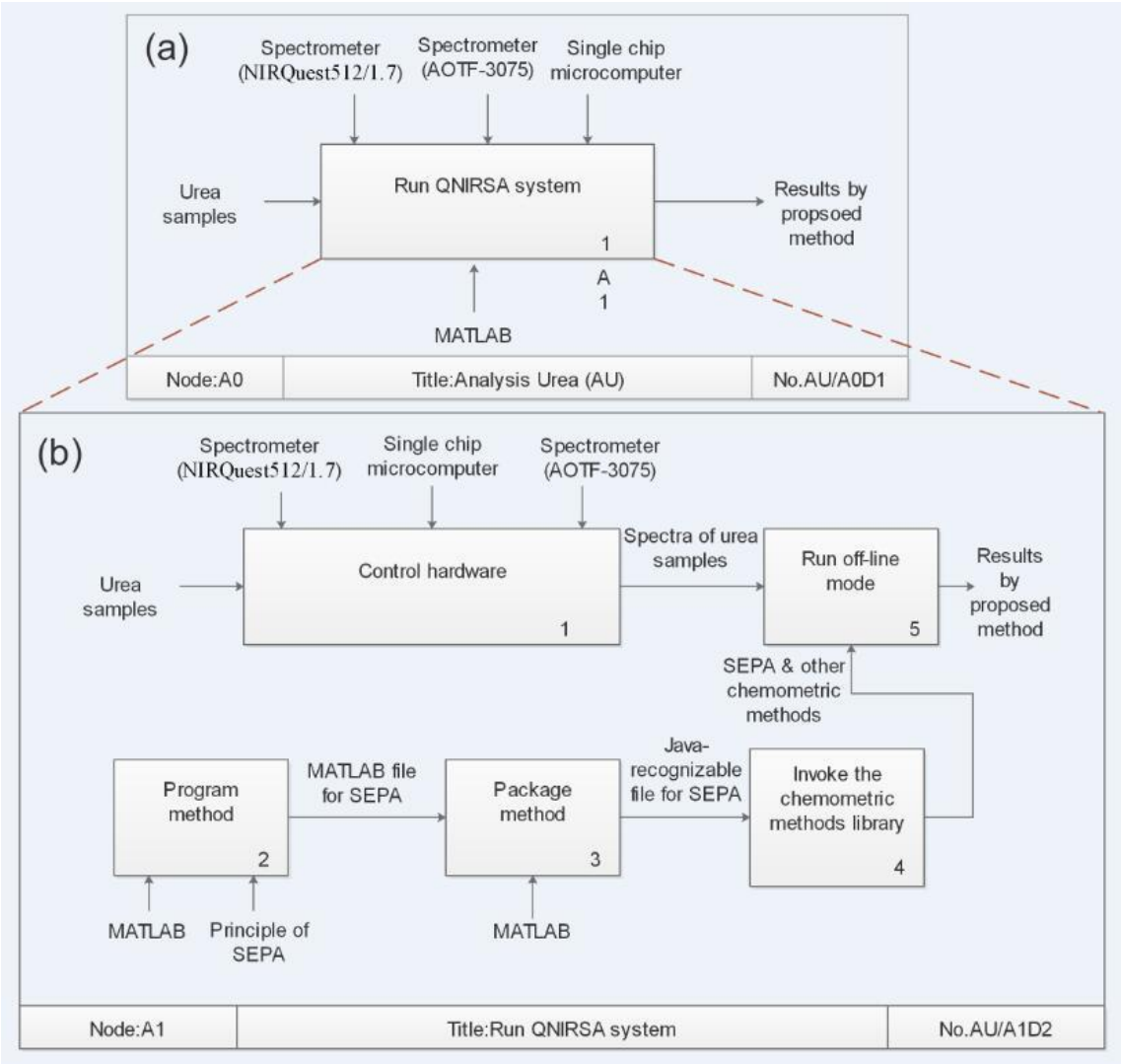


Figure 7.3: The IDEF0 diagram of Application 2.

7.3 Contributions of the Benchmarking Results for Two Real-world Applications

As it has been interpreted in section 6.4.2, contributions of the benchmarking results are mainly embodied in two aspects. The first aspect is that those coherent processes of benchmarking from data collection to model assessment could help researchers to design the process and experiment for SKNIRS. Specifically, take the Application 2 in 7.2.2, for instance. Although Application 2 did not use single rice samples, it took similar process pattern as what the benchmarking did in this thesis to analyse the biuret property of urea. After the data collection completed by a portable NIR spectrometer handled by the QNIRSA system, two pre-processing methods, three variable selection methods and PLSR were selected to establish six statistical models. RMSE is the main

criterion for model assessment and comparison as well. All of those methods and assessment were done by the QNIRSA system.

Another aspect is that the benchmarking results are a good reference for those researches on single rice samples. Take Application 1 in 7.2.1, for example. This application used the same samples which were used in this thesis. Therefore, the benchmarking results were used directly for that application. The selection of all methods, except the novel method reported in that paper, referred to the benchmarking results in this thesis. Statistical models built by those methods were imported from this thesis. Therefore, the last thing my collaborators needed to do was to compare the model established by that novel method with those statistical models. Additionally, the benchmarking results made it possible to investigate the difference between the three forms of single rice.

7.4 Summary

Two real-world applications have been reported in this chapter as the validation of the QNIRSA system. They prove that the QNIRSA system can undertake multiple applications of NIRS analysis. The QNIRSA system is available for various agricultural samples, spectrometers and chemometric methods. So far both completed applications emphasised on the establishment of an appropriate model so that only off-line mode have been utilised. Real-time Applications involving in on-line mode are investigating, but papers of them will be future works. On the other hand, interpretations about how those benchmarking processes and results helped my collaborators to complete the two real-world applications have been made as well.

8 CONCLUSION

8.1 Summary of Research Works

Near-Infrared Spectroscopy (NIRS) technology is a widely used non-destructive analytical tool in agriculture. This PhD thesis found two research problems in practical Single Kernel NIRS (SKNIRS) applications. The first research problem is the lack of a comparative study for SKNIRS to provide a global view for the benchmarking of chemometric methods. A literature review about NIRS applications on rice in the past 20 years was done by this PhD thesis as evidence to show the first research problem. Only 35.7% of those applications were related to a single rice sample, and the average number of methods those applications applied was three. No comparative study of methods on single rice has been investigated before progress made by this thesis. Another research problem is the lack of an integrated system at the single kernel level, which can not only controls sample-specific hardware accessories to support spectral data collection of single kernel sample but also provides a wide range of chemometric methods to support NIRS comparative analysis. This research problem is summarised based on the practical NIRS analysis applications at single kernel level and suggestions from relevant experts.

In order to solve the two research problems, the proposed solution of this PhD thesis was to design a stepwise process for benchmarking the chemometric methods. This proposed solution includes eight steps in three stages. The first stage is data collection that acquires both spectral data and reference data of single kernel samples. Spectral dataset and its reference data for rice's protein content of 201 single rice samples were collected and used for the research in this thesis. The second stage is data processing. In this stage, chemometric methods mainly are utilised in four steps in sequential order: 1) dataset partition, 2) pre-processing, 3) variable selection and 4) multivariate calibration. The dataset partition step is to solve the training set, and validation set partitioning problem, in order to extract a representative training set to construct a model and reasonable validation set to assess model from the original NIR spectral data set. The pre-processing step is to remove the undesired physical phenomena in the spectra mainly caused by sample-to-sample variations, in order to facilitate the subsequent procedures. Variable selection step aims to reduce the number of variables and select representative variables, which can render better prediction with the regression model used in multivariate calibration. Multivariate calibration methods are developed through regressions of the measured NIR spectral data against the reference data values of analyte properties determined by reference analytical methods. The calibration model is constructed in this step and validated in the next step. The third stage is to employ a validated model for real-world applications. In addition to the benchmarking of

chemometric methods involved in this thesis, the other two real-world applications were reported as well, as the instances of stage 3 of the proposed solution.

An integrated software system named QNIRSA system has been developed and illustrated by this thesis, in order to support the proposed solution. The key idea of the QNIRSA system is to modularise the steps of the proposed benchmarking process by a stepwise functionalisation. The architecture of the QNIRSA system consists of five layers. Five layers are data layer, data manager layer, component layer, mode layer and presentation layer. The data layer refers to spectral data and reference data, while the data manager layer includes two data storage carrier, spectral database and excel document. The component layer consists of some components. They are hardware adapter for manage hardware and spectrometer; spectral pre-processor for preprocessing spectral data; quantitative multivariate analyser for multivariate calibration, spectral visualiser for displaying spectral data; data manager for manager spectral data and reference data; chemometric methods library for providing desired chemometric methods for a comparative study of benchmarking; variable selector for selecting variables and dataset partitioner for dividing the dataset into a training set and a validation set. The QNIRSA system is available for multiple sample-specific hardware accessories and spectral device validated by real-world applications.

In terms of the benchmark for chemometric methods for analysing single kernel sample, firstly, single rice samples with threes forms including single brown rice kernel, single rice kernel and rice flour were collected. Spectral data of 201 single rice samples and their relevant reference data of rice's protein content were used for benchmarking. Different combinations of 16 methods were applied and compared, in order to provide a global view of how those methods performed on single rice sample.

Regarding the three research questions in the first chapter, the IDEF0 functional modelling approach has been adopted to develop the QNIRSA system because it is an appropriate tool to modularise the proposed solution. The RMSEP should be the best criterion to assess the performance of chemometric methods because it is an absolute criterion to measure the performance of the calibration model. The smaller the RMSEP, the better predictive performance the calibration model has. In terms of the classification of the performance of chemometric methods, PDRMSEP (the percentage of decrement of RMSEP that method combination made compared with the RMSEP of the global reference model) has been used to classify the combination of chemometric methods into four classes. *Highly effective* combinations that PDRMSEP is higher than 60%; *Well effective* combinations that PDRMSEP is from 50% to 59.9%; *Effective* combinations that PDRMSEP is from 40% to 49.9%; *Low effective* combinations that PDRMSEP is lower than 40%.

8.2 Summary of Contributions

There are two main contributions of this PhD thesis. The first contribution is to make a comparative study for various statistical models established by sixteen chemometric

methods on the single rice sample as the benchmark of chemometric methods for single kernel near-infrared spectroscopy analysis. Compared with previous work which focused on the analysis of bulk rice samples, this exploratory research in addressing single rice samples leads to a more accurate assessment of properties such as protein content. This study is also an example and guidance to show analytical processes for assessing different statistical models on single kernel samples. Those processes of benchmarking can guide future research on how to design and implement the process on single kernel NIR analysis. Additionally, the comparative results of those models are a useful reference for chemometric methods selection on single rice samples. They provide detailed assessments of sixteen methods, including not only the performance of those methods on three forms of single rice but also the optimal parameters of methods. Relevant research may briefly refer to this comparative study when it is related to methods selection, parameters tuning or model calibration on single rice samples.

Another contribution is to develop an integrated software system named QNIRSA system, which not only can control multiple spectrometers and sample-specific hardware for spectral data acquisition of single kernel sample, but also provides some chemometric methods for single kernel near-infrared spectroscopy analysis. The QNIRSA system can be regarded as a fully integrated functional platform which provides APIs for multiple spectrometers and hardware, a graphical user interface for users, and a chemometric methods library which provides some useful methods for single kernel near-infrared spectroscopy analysis as well. Two real-world NIRS applications have been solved by the QNIRSA system, which is reported in chapter 7. The QNIRSA system, including both Java codes for software and MATLAB codes for algorithms, was implemented by myself.

8.3 Limitations and Future Work

8.3.1 Limitations

The main limitation of the single rice data set used in this thesis is the single category of single kernel samples. Although three forms of rice have been discussed, no other categories of single kernel samples have been analyzed. Therefore, findings in chapter 6 are constrained in single rice samples. This is because it is quite difficult for single kernel samples to account for their laboratory error by reference methods. As a result, it is difficult to collect representative single kernel data sets. Details have been interpreted in section 4.4.2. However, the processes of benchmarking in this thesis are still can be an example for the design of analytical process on single kernel samples, but attention should be paid to the specific performance of statistical models when categories of single kernel samples are different.

Another limitation of the single rice data set is the small number of samples compared with a large number of variables. An average number of 200 to 300 single kernels

samples are often provided for SKNIRS. By contrast, the number of variables in a NIRS dataset reaches about 500 to several thousand depending on the precision of the spectral device. The number of variables is much more than the number of single rice samples, which may easily cause the over-fitting in model calibration. However, this is a common issue in agricultural NIRS applications. This issue would be alleviated gradually in the future when more single kernel samples can be provided with the development of NIRS technology.

One limitation objectively exists in the QNIRSA system, due to time limits, the spectral database was designed initially, but no further implementations have been made. Currently, the QNIRSA system utilises the Excel document to import and export spectral data. When the size of data becomes large, potential data security issues such as I/O exception may occur for the Excel document. Besides, the QNIRAS system provides a coherent process for SKNIRS but requires specialized knowledge from the user. Honestly, it is not friendly enough for the user who is out of the NIRS or chemometrics domain. Firstly, the user interface of the software displays several charts for the original spectrum, pre-processed spectrum, and variable selected spectrum, respectively. A professional and experienced user can check them if they may even have minor issues, but a general user who is not familiar with this domain is hard to distinguish them. On the other hand, though the software provides default parameters for every method, the professional and experienced user knows how to configure them appropriately.

This thesis assessed and compared eight variable selection methods, but more other methods are potential to consider. For example, the artificial neural network refers to deep learning may be able to improve the NIR model performance. Support vector machine is also a popular method in machine learning domain that may help the SKNIRS analysis. Another limitation is related to multivariate calibration. Calibration methods used in this thesis such as PLSR, PCR and MLR, are all linear regression techniques. Although the results showed that linear regression performed well in multivariate calibration, but they were not perfect. Non-linear regression tools may give an improvement based on current linear regression approaches.

8.3.2 Future Works

Future work may be undertaken in three aspects. With SKNIRS application goes, data will be massive. Data management and classification will be an important issue. Therefore, firstly the spectral database should be improved to support both real-time and off-line data import and export. Secondly, the QNIRSA system needs to optimize to be more user-friendly. Thirdly, more chemometric methods can be assessed and compared to enrich the benchmark of chemometric methods in this thesis. Non-linear regression methods and deep learning methods deserve investigation on single kernel samples. One of the foreseeable research issues is how to avoid the over-fitting problem when using

those advanced methods because NIRS spectral data is typical that the number of variables is much more than the number of samples.

REFERENCES

[1] L. Zhao, S. Hu, X. Zeng, Y. Wu, Y. Lin, J. Liu, *et al.*, "An Integrated Software System for Supporting Real-Time Near-Infrared Spectral Big Data Analysis and Management," in *Big Data (BigData Congress), 2017 IEEE International Congress on*, 2017, pp. 97-104.

[2] Z. Xu, S. Fan, J. Liu, B. Liu, L. Tao, J. Wu, *et al.*, "A calibration transfer optimized single kernel near-infrared spectroscopic method," *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy,* vol. 220, p. 117098, 2019.

[3] J. Liu, S. Hu, S. Yu, Y. Lin, P. Wei, Y. Yang, *et al.*, "Analysis of biuret in urea fertilizer using a portable near-infrared spectrometer," *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy,* Under review 2019.

[4] W. F. McClure, "204 years of near infrared technology: 1800–2003," *Journal of Near Infrared Spectroscopy,* vol. 11, pp. 487-518, 2003.

[5] M. AG, "NIR Spectroscopy A guide to near-infrared spectroscopic analysis of industrial manufacturing processes," *CH-9101 Herisau, Switzerland,* 2013.

[6] Y. Ozaki, W. F. McClure, and A. A. Christy, *Near-infrared spectroscopy in food science and technology*: John Wiley & Sons, 2006.

[7] V. Bellon, J. L. Vigneau, and F. Sévila, "Infrared and near-infrared technology for the food industry and agricultural uses: on-line applications," *Food Control,* vol. 5, pp. 21-27, 1994.

[8] C. Fernández, M. S. Larrechi, and M. P. Callao, "An analytical overview of processes for removing organic dyes from wastewater effluents," *TrAC Trends in Analytical Chemistry,* vol. 29, pp. 1202-1211, 2010.

[9] D. A. Boas, C. E. Elwell, M. Ferrari, and G. Taga, "Twenty years of functional near-infrared spectroscopy: introduction for the special issue," ed: Elsevier, 2014.

[10] Y. Roggo, P. Chalus, L. Maurer, C. Lema-Martinez, A. Edmond, and N. Jent, "A review of near infrared spectroscopy and chemometrics in pharmaceutical technologies," *Journal of pharmaceutical and biomedical analysis,* vol. 44, pp. 683-700, 2007.

[11] S. Tsuchikawa, "A review of recent near infrared research for wood and paper," *Applied Spectroscopy Reviews,* vol. 42, pp. 43-71, 2007.

[12] D. Wang, X. Zhou, T. Jin, X. Hu, J. Zhong, and Q. Wu, "Application of near-infrared spectroscopy to agriculture and food analysis," *Guang pu xue yu guang pu fen xi= Guang pu,* vol. 24, pp. 447-450, 2004.

[13] L. E. Agelet and C. R. Hurburgh, "Limitations and current applications of Near Infrared Spectroscopy for single seed analysis," *Talanta,* vol. 121, pp. 288-299, 2014.

[14] L. Silvela, R. Rodgers, A. Barrera, and D. Alexander, "Effect of selection intensity and population size on percent oil in maize, Zea mays L," *Theoretical and applied genetics,* vol. 78, pp. 298-304, 1989.

[15] B.-S. Yoon, B. W. Brorsen, and C. P. Lyford, "Value of increasing kernel uniformity," *Journal of agricultural and resource economics,* vol. 27, p. 481, 2002.

[16] Å. Rinnan, F. van den Berg, and S. B. Engelsen, "Review of the most common pre-processing techniques for near-infrared spectra," *TrAC Trends in Analytical Chemistry,* vol. 28, pp. 1201-1222, 2009.

[17] Z. Xiaobo, Z. Jiewen, M. J. Povey, M. Holmes, and M. Hanpin, "Variables selection methods in near-infrared spectroscopy," *Analytica chimica acta,* vol. 667, pp. 14-32, 2010.
[18] T. Mehmood, K. H. Liland, L. Snipen, and S. Sæbø, "A review of variable selection methods in partial least squares regression," *Chemometrics and Intelligent Laboratory Systems,* vol. 118, pp. 62-69, 2012.
[19] C. Pasquini, "Near infrared spectroscopy: A mature analytical technique with new perspectives–A review," *Analytica Chimica Acta,* vol. 1026, pp. 8-36, 2018.
[20] T. M. Baye, T. C. Pearson, and A. M. Settles, "Development of a calibration to predict maize seed composition using single kernel near infrared spectroscopy," *Journal of Cereal Science,* vol. 43, pp. 236-243, 2006.
[21] B. A. Orman and R. A. Schumann, "Nondestructive single-kernel oil determination of maize by near-infrared transmission spectroscopy," *Journal of the American Oil Chemists' Society,* vol. 69, pp. 1036-1038, 1992.
[22] T. Dyba, B. A. Kitchenham, and M. Jorgensen, "Evidence-based software engineering for practitioners," *IEEE software,* vol. 22, pp. 58-65, 2005.
[23] IEEE, "IEEE standard for functional modeling language—syntax and semantics for IDEF0," ed: Software Engineering Standards Committee of the IEEE Computer Society …, 1998.
[24] T. Davies, "The history of near infrared spectroscopic analysis: Past, present and future," *Analusis,* vol. 26, pp. 17-19, 1998.
[25] R. M. Balabin and S. V. Smirnov, "Variable selection in near-infrared spectroscopy: benchmarking of feature selection methods on biodiesel data," *Analytica chimica acta,* vol. 692, pp. 63-72, 2011.
[26] H. Cen and Y. He, "Theory and application of near infrared reflectance spectroscopy in determination of food quality," *Trends in Food Science & Technology,* vol. 18, pp. 72-83, 2007.
[27] L. Zhao, J. Li, L. Zhang, and Y. Yan, "Influence of FT-NIR spectrometer scanning requirements on the math model's precision," *Guang pu xue yu guang pu fen xi= Guang pu,* vol. 24, pp. 41-44, 2004.
[28] B. A. Weinstock, J. Janni, L. Hagen, and S. Wright, "Prediction of oil and oleic acid concentrations in individual corn (Zea mays L.) kernels using near-infrared reflectance hyperspectral imaging and multivariate analysis," *Applied spectroscopy,* vol. 60, pp. 9-16, 2006.
[29] P. Gemperline, *Practical guide to chemometrics*: CRC press, 2006.
[30] D. L. Massart, B. G. Vandeginste, L. Buydens, S. De Jong, P. J. Lewi, J. Smeyers-Verbeke, *et al.*, "Handbook of chemometrics and qualimetrics: Part A," *Applied Spectroscopy,* vol. 52, p. 302A, 1998.
[31] M. Daszykowski, B. Walczak, and D. Massart, "Representative subset selection," *Analytica chimica acta,* vol. 468, pp. 91-103, 2002.
[32] W. Wu, B. Walczak, D. Massart, S. Heuerding, F. Erni, I. Last, *et al.*, "Artificial neural networks in classification of NIR spectral data: design of the training set," *Chemometrics and intelligent laboratory systems,* vol. 33, pp. 35-46, 1996.
[33] K. Rajer-Kanduč, J. Zupan, and N. Majcen, "Separation of data on the training and test set for modelling: a case study for modelling of five colour properties of a white pigment," *Chemometrics and intelligent laboratory systems,* vol. 65, pp. 221-229, 2003.
[34] R. W. Kennard and L. A. Stone, "Computer aided design of experiments," *Technometrics,* vol. 11, pp. 137-148, 1969.

[35] H. A. Dantas Filho, R. K. H. Galvao, M. C. U. Araújo, E. C. da Silva, T. C. B. Saldanha, G. E. José, *et al.*, "A strategy for selecting calibration samples for multivariate modelling," *Chemometrics and intelligent laboratory systems*, vol. 72, pp. 83-91, 2004.

[36] R. K. H. Galvao, M. C. U. Araujo, G. E. Jose, M. J. C. Pontes, E. C. Silva, and T. C. B. Saldanha, "A method for calibration and validation subset partitioning," *Talanta*, vol. 67, pp. 736-740, 2005.

[37] H. Martens, S. Jensen, and P. Geladi, "Multivariate linearity transformation for near-infrared reflectance spectrometry," in *Proceedings of the Nordic symposium on applied statistics*, 1983, pp. 205-234.

[38] P. Geladi, D. MacDougall, and H. Martens, "Linearization and scatter-correction for near-infrared reflectance spectra of meat," *Applied spectroscopy*, vol. 39, pp. 491-500, 1985.

[39] H. Martens and E. Stark, "Extended multiplicative signal correction and spectral interference subtraction: new preprocessing methods for near infrared spectroscopy," *Journal of pharmaceutical and biomedical analysis*, vol. 9, pp. 625-635, 1991.

[40] H. Martens, J. P. Nielsen, and S. B. Engelsen, "Light scattering and light absorbance separated by extended multiplicative signal correction. Application to near-infrared transmission analysis of powder mixtures," *Analytical Chemistry*, vol. 75, pp. 394-404, 2003.

[41] S. Thennadil and E. Martin, "Empirical preprocessing methods and their impact on NIR calibrations: a simulation study," *Journal of Chemometrics: A Journal of the Chemometrics Society*, vol. 19, pp. 77-89, 2005.

[42] W. Windig, J. Shaver, and R. Bro, "Loopy MSC: a simple way to improve multiplicative scatter correction," *Applied spectroscopy*, vol. 62, pp. 1153-1159, 2008.

[43] M. Dhanoa, S. Lister, R. Sanderson, and R. Barnes, "The link between multiplicative scatter correction (MSC) and standard normal variate (SNV) transformations of NIR spectra," *Journal of Near Infrared Spectroscopy*, vol. 2, pp. 43-47, 1994.

[44] K. Norris and P. Williams, "Optimization of mathematical treatments of raw near-infrared signal in the," *Cereal Chem*, vol. 61, pp. 158-165, 1984.

[45] A. Savitzky and M. J. Golay, "Smoothing and differentiation of data by simplified least squares procedures," *Analytical chemistry*, vol. 36, pp. 1627-1639, 1964.

[46] P. A. Gorry, "General least-squares smoothing and differentiation by the convolution (Savitzky-Golay) method," *Analytical Chemistry*, vol. 62, pp. 570-573, 1990/03/15 1990.

[47] M. C. U. Araújo, T. C. B. Saldanha, R. K. H. Galvao, T. Yoneyama, H. C. Chame, and V. Visani, "The successive projections algorithm for variable selection in spectroscopic multicomponent analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 57, pp. 65-73, 2001.

[48] V. Centner, D.-L. Massart, O. E. de Noord, S. de Jong, B. M. Vandeginste, and C. Sterna, "Elimination of uninformative variables for multivariate calibration," *Analytical chemistry*, vol. 68, pp. 3851-3858, 1996.

[49] W. Cai, Y. Li, and X. Shao, "A variable selection method based on uninformative variable elimination for multivariate calibration of near-infrared spectra," *Chemometrics and intelligent laboratory systems*, vol. 90, pp. 188-194, 2008.

[50] S. Ye, D. Wang, and S. Min, "Successive projections algorithm combined with uninformative variable elimination for spectral variable selection," *Chemometrics and Intelligent Laboratory Systems,* vol. 91, pp. 194-199, 2008.
[51] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *science,* vol. 220, pp. 671-680, 1983.
[52] H. Swierenga, P. De Groot, A. De Weijer, M. Derksen, and L. Buydens, "Improvement of PLS model transferability by robust wavelength selection," *Chemometrics and Intelligent Laboratory Systems,* vol. 41, pp. 237-248, 1998.
[53] H. Swierenga, F. Wülfert, O. De Noord, A. De Weijer, A. Smilde, and L. Buydens, "Development of robust calibration models in near infra-red spectrometric applications," *Analytica Chimica Acta,* vol. 411, pp. 121-135, 2000.
[54] R. Leardi and A. L. Gonzalez, "Genetic algorithms applied to feature selection in PLS regression: how and when to use them," *Chemometrics and intelligent laboratory systems,* vol. 41, pp. 195-207, 1998.
[55] H. Abdollahi and L. Bagheri, "Simultaneous spectrophotometric determination of Vitamin K3 and 1, 4-naphthoquinone after cloud point extraction by using genetic algorithm based wavelength selection-partial least squares regression," *Analytica chimica acta,* vol. 514, pp. 211-218, 2004.
[56] J. Ghasemi, A. Niazi, and R. Leardi, "Genetic-algorithm-based wavelength selection in multicomponent spectrophotometric determination by PLS: application on copper and zinc mixture," *Talanta,* vol. 59, pp. 311-317, 2003.
[57] L. Nørgaard, A. Saudland, J. Wagner, J. P. Nielsen, L. Munck, and S. B. Engelsen, "Interval Partial Least-Squares Regression (i PLS): A Comparative Chemometric Study with an Example from Near-Infrared Spectroscopy," *Applied Spectroscopy,* vol. 54, pp. 413-419, 2000.
[58] O. Polgár, M. Fried, T. Lohner, and I. Bársony, "Comparison of algorithms used for evaluation of ellipsometric measurements random search, genetic algorithms, simulated annealing and hill climbing graph-searches," *Surface Science,* vol. 457, pp. 157-177, 2000.
[59] X. Zou, J. Zhao, and Y. Li, "Selection of the efficient wavelength regions in FT-NIR spectroscopy for determination of SSC of 'Fuji'apple based on BiPLS and FiPLS models," *Vibrational spectroscopy,* vol. 44, pp. 220-227, 2007.
[60] X. Shao, X. Bian, J. Liu, M. Zhang, and W. Cai, "Multivariate calibration methods in near infrared spectroscopic analysis," *Analytical Methods,* vol. 2, pp. 1662-1666, 2010.
[61] D. A. Freedman, *Statistical models: theory and practice*: cambridge university press, 2009.
[62] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics,* vol. 2, pp. 433-459, 2010.
[63] P. Geladi and B. R. Kowalski, "Partial least-squares regression: a tutorial," *Analytica chimica acta,* vol. 185, pp. 1-17, 1986.
[64] Q.-S. Xu and Y.-Z. Liang, "Monte Carlo cross validation," *Chemometrics and Intelligent Laboratory Systems,* vol. 56, pp. 1-11, 2001.
[65] R. Barnes, M. S. Dhanoa, and S. J. Lister, "Standard normal variate transformation and de-trending of near-infrared diffuse reflectance spectra," *Applied spectroscopy,* vol. 43, pp. 772-777, 1989.
[66] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The journal of chemical physics,* vol. 21, pp. 1087-1092, 1953.

[67] C. B. Lucasius, M. L. Beckers, and G. Kateman, "Genetic algorithms in wavelength selection: a comparative study," *Analytica Chimica Acta,* vol. 286, pp. 135-153, 1994.
[68] S. Gourvénec, X. Capron, and D. Massart, "Genetic algorithms (GA) applied to the orthogonal projection approach (OPA) for variable selection," *Analytica Chimica Acta,* vol. 519, pp. 11-21, 2004.
[69] X. Yan and X. Su, *Linear regression analysis: theory and computing*: World Scientific, 2009.
[70] T. Fearn, "Assessing Calibrations: SEP, RPD, RER and R 2," *NIR news,* vol. 13, pp. 12-13, 2002.
[71] V. Bellon-Maurel, E. Fernandez-Ahumada, B. Palagos, J.-M. Roger, and A. McBratney, "Critical review of chemometric indicators commonly used for assessing the quality of the prediction of soil attributes by NIR spectroscopy," *TrAC Trends in Analytical Chemistry,* vol. 29, pp. 1073-1081, 2010.
[72] S. R. Delwiche, M. M. Bean, R. E. Miller, B. D. Webb, and P. C. Williams, "Apparent amylose content of milled rice by near-infrared reflectance spectrophotometry," *Cereal Chemistry,* vol. 72, pp. 182-186, 1995.
[73] J. G. Wu, C. Shi, and X. Zhang, "Estimating the amino acid composition in milled rice by near-infrared reflectance spectroscopy," *Field Crops Research,* vol. 75, pp. 1-7, 2002/03/28/ 2002.
[74] S. Kawamura, M. Natsuga, K. Takekura, and K. Itoh, "Development of an automatic rice-quality inspection system," *Computers and Electronics in Agriculture,* vol. 40, pp. 115-126, 2003/10/01/ 2003.
[75] X. Xiao, Y. Chen, W. Luo, Y. Liu, X. Mao, and X. Li, "Determination of amylase content in single rice grain by near infrared transmittance spectroscopy (NITS)," *Chinese Journal of Rice Science,* vol. 3, 2003.
[76] R. Rittiron, I. Saranwong, and S. Kawano, "Useful tips for constructing a near infrared-based quality sorting system for single brown-rice kernels," *Journal of near infrared spectroscopy,* vol. 12, pp. 133-139, 2004.
[77] J. Wu and C. Shi, "Prediction of grain weight, brown rice weight and amylose content in single rice grains using near-infrared reflectance spectroscopy," *Field Crops Research,* vol. 87, pp. 13-21, 2004.
[78] J. Bao, Y. Shen, and L. Jin, "Determination of thermal and retrogradation properties of rice starch using near-infrared spectroscopy," *Journal of Cereal Science,* vol. 46, pp. 75-81, 2007/07/01/ 2007.
[79] L. Qingyun, C. Yeming, T. Mikami, M. Kawano, and L. Zaigui, "Adaptability of four-samples sensory tests and prediction of visual and near-infrared reflectance spectroscopy for Chinese indica rice," *Journal of Food Engineering,* vol. 79, pp. 1445-1451, 2007/04/01/ 2007.
[80] J. G. Wu and C. H. Shi, "Calibration model optimization for rice cooking characteristics by near infrared reflectance spectroscopy (NIRS)," *Food Chemistry,* vol. 103, pp. 1054-1061, 2007/01/01/ 2007.
[81] H.-j. Zhang, J.-h. Wu, L. I. Y. Luo Li-jun, H. Yang, X.-q. Yu, X.-s. Wang, *et al.*, "Comparison of Near Infrared Spectroscopy Models for Determining Protein and Amylose Contents Between Calibration Samples of Recombinant Inbred Lines and Conventional Varieties of Rice," *Agricultural Sciences in China,* vol. 6, pp. 941-948, 2007/08/01/ 2007.
[82] K. J. Chen and M. Huang, "Prediction of milled rice grades using Fourier transform near-infrared spectroscopy and artificial neural networks," *Journal of Cereal Science,* vol. 52, pp. 221-226, 2010/09/01/ 2010.

[83] Y. Shao, Y. Cen, Y. He, and F. Liu, "Infrared spectroscopy and chemometrics for the starch and protein prediction in irradiated rice," *Food Chemistry,* vol. 126, pp. 1856-1861, 2011/06/15/ 2011.

[84] B. Zhang, Z. Q. Rong, Y. Shi, J. G. Wu, and C. H. Shi, "Prediction of the amino acid composition in brown rice using different sample status by near-infrared reflectance spectroscopy," *Food Chemistry,* vol. 127, pp. 275-281, 2011/07/01/ 2011.

[85] C. Dachoupakan Sirisomboon, R. Putthang, and P. Sirisomboon, "Application of near infrared spectroscopy to detect aflatoxigenic fungal contamination in rice," *Food Control,* vol. 33, pp. 207-214, 2013/09/01/ 2013.

[86] Y.-K. Chuang, Y.-P. Hu, I. C. Yang, S. R. Delwiche, Y. M. Lo, C.-Y. Tsai, *et al.*, "Integration of independent component analysis with near infrared spectroscopy for evaluation of rice freshness," *Journal of Cereal Science,* vol. 60, pp. 238-242, 2014/07/01/ 2014.

[87] L. Xie, S. Tang, N. Chen, J. Luo, G. Jiao, G. Shao, *et al.*, "Optimisation of near-infrared reflectance model in measuring protein and amylose content of rice flour," *Food chemistry,* vol. 142, pp. 92-100, 2014.

[88] L. Song, Q. Wang, C. Wang, Y. Lin, D. Yu, Z. Xu, *et al.*, "Effect of γ-irradiation on rice seed vigor assessed by near-infrared spectroscopy," *Journal of Stored Products Research,* vol. 62, pp. 46-51, 2015/05/01/ 2015.

[89] T. B. Bagchi, S. Sharma, and K. Chattopadhyay, "Development of NIRS models to predict protein and amylose content of brown rice and proximate compositions of rice bran," *Food Chemistry,* vol. 191, pp. 21-27, 2016.

[90] P. Sampaio, A. Soares, A. Castanho, A. S. Almeida, J. Oliveira, and C. Brites, "Dataset of Near-infrared spectroscopy measurement for amylose determination using PLS algorithms," *Data in Brief,* vol. 15, pp. 389-396, 2017/12/01/ 2017.

[91] P. Siriphollakul, K. Nakano, S. Kanlayanarat, S. Ohashi, R. Sakai, R. Rittiron, *et al.*, "Eating quality evaluation of Khao Dawk Mali 105 rice using near-infrared spectroscopy," *LWT - Food Science and Technology,* vol. 79, pp. 70-77, 2017/06/01/ 2017.

[92] B. Das, R. N. Sahoo, S. Pargal, G. Krishna, R. Verma, V. Chinnusamy, *et al.*, "Quantitative monitoring of sucrose, reducing sugar and total sugar dynamics for phenotyping of water-deficit stress tolerance in rice through spectroscopy and chemometrics," *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy,* vol. 192, pp. 41-51, 2018/03/05/ 2018.

[93] P. S. Sampaio, A. Soares, A. Castanho, A. S. Almeida, J. Oliveira, and C. Brites, "Optimization of rice amylose determination by NIR-spectroscopy using PLS chemometrics algorithms," *Food Chemistry,* vol. 242, pp. 196-204, 2018/03/01/ 2018.

[94] M. Awanthi, B. Jinendra, S. Navaratne, and C. Navaratne, "Adaptation of visible and short wave Near Infrared (VIS-SW-NIR) common PLS model for quantifying paddy hardness," *Journal of Cereal Science,* vol. 89, p. 102795, 2019.

[95] E. O. Díaz, S. Kawamura, M. Matsuo, M. Kato, and S. Koseki, "Combined analysis of near-infrared spectra, colour, and physicochemical information of brown rice to develop accurate calibration models for determining amylose content," *Food chemistry,* vol. 286, pp. 297-306, 2019.

[96] F. Lei, Y. Yang, J. Zhang, J. Zhong, L. Yao, J. Chen, *et al.*, "Window optimisation PMSC–PLS with applications to NIR spectroscopic analyses," *Chemometrics and Intelligent Laboratory Systems,* vol. 191, pp. 158-167, 2019.

[97] M. Makky, Santosa, R. E. Putri, and K. Nakano, "Determination of moisture content in rice using non-destructive short-wave near infrared spectroscopy," in *AIP Conference Proceedings*, 2019, p. 020014.
[98] J. Zhang, M. Li, T. Pan, L. Yao, and J. Chen, "Purity analysis of multi-grain rice seeds with non-destructive visible and near-infrared spectroscopy," *Computers and Electronics in Agriculture*, vol. 164, p. 104882, 2019.
[99] C. Weimin, L. Binmei, Y. Yafeng, T. Liangzhi, and W. Yuejin, "Screening of Amylose and Protein Mutants and Correlation Analysis of Agronomic Character in Rice," *Acta Laser Biology Sinica*, vol. 25, pp. 356-361, 2016.
[100] N. Caporaso, M. B. Whitworth, and I. D. Fisk, "Protein content prediction in single wheat kernels using hyperspectral imaging," *Food chemistry*, vol. 240, pp. 32-42, 2018.
[101] R. Rund, "Report on fertilizers and agricultural liming materials," *Journal of the Association of Official Analytical Chemists,* 1980.

GLOSSARY

BBR	**Bulk Brown Rice**
BiPLS	Backward interval Partial Least Squares
BMR	Bulk Milled Rice
BPMN	Business Process Model and Notation
BRR	Bulk Rough Rice
CV	Cross-Validation
FiPLS	Forward interval Partial Least Squares
FS	Full Spectrum
GA	Genetic Algorithm
iPLS	interval Partial Least Squares
LOOCV	Leave-One-Out Cross-Validation
LS	Least Squares
MLR	Multiple Linear Regression
MSC	Multiplicative Scatter Correction
NIR	Near-Infrared
NIRS	Near-Infrared Spectroscopy
KS	Kennard–Stone algorithm
PCA	Principal Component Analysis
PCR	Principal Component Regression
PLS	Partial Least Squares
PLSR	Partial Least Squares Regression
PRESS	Predicted Residual Error Sum of Squares
QNIRSA	Quantitative Near-Infrared Spectroscopy Analysis
R^2	Coefficient of Determination
RF	Rice Flour

RMDP	Reference Method for Data Partition
RMSE	Root Mean Square Error
RMSEV	Root Mean Square Error of Validation
SA	Simulated Annealing
SBK	Single Brown Rice Kernel
SBR	Single Brown Rice
SG	Savitzky-Golay polynomial derivative filters
SMR	Single Milled Rice
SNV	Standard Normal Variate
SKNIRS	Single Kernel Near-Infrared Spectroscopy
SPA	Successive Projections Algorithm
SPXY	Sample set Partitioning based on joint X–Y distances
SRK	Single Rice Kernel
SRR	Single Rough Rice
UVE	Uninformative Variable Elimination

APPENDIX 1: QNIRSA SYSTEM

The QNIRSA System, including both Java codes for software and MATLAB codes for algorithms, was implemented by myself.

The third edition of the QNIRSA System can be accessed by the below Github link: https://github.com/ShupengHu/QNIRSA-System.git

Most codes in the third edition are moved from the previous two editions. Main differences are:

1. Java packages and classes are restructured to make them more readable.
2. User interface dashboard is re-designed to make it easier to use. Online mode and offline mode are divided into two graphical user interfaces (GUI). Functions in the previous two editions are all remained, but they are re-distributed on the GUI.
3. Intellij IDEA (Community version 2019.3, free to use) is used for Java programming, while MATLAB R2018 is used for coding algorithms.
4. Maven (version 3.6.3) is used for managing the open-source third-party Java package utilized in QNISA System.
5. Due to copyright reasons, some codes refer to hardware or spectral device have to be omitted.

APPENDIX 2.1
TABLE 1: THE PERFORMANCE OF ALL COMBINATIONS OF METHODS ON SRK.

Combination ID	Rice Form	Methods				Assessment	
		Dataset Partition	Pre-processing	Variable Selection	Calibration	RMSEP	PDRMSEP
1	SRK	KS	MSC	UVE (av: 1)	PLSR (PLSc: 20)	0.2652	64.25%
2	SRK	SPXY	MSC	UVE (av: 0.9)	PLSR (PLSc: 20)	0.2977	59.87%
3	SRK	KS	MSC	SPA	PLSR (PLSc: 20)	0.3011	59.41%
4	SRK	KS	SNV	SPA	PLSR (PLSc: 20)	0.3073	58.57%
5	SRK	SPXY	SNV	UVE (av: 1)	PLSR (PLSc: 20)	0.3073	58.57%
6	SRK	SPXY	MSC	GA (pspc: 0.7 pm: 0.1)	PLSR (PLSc: 20)	0.3125	57.87%
7	SRK	KS	SNV	UVE (av: 0.8)	PLSR (PLSc: 20)	0.3128	57.83%
8	SRK	KS	SG 1st_der (opf: 2)	SA (cr: 0.7)	PLSR (PLSc: 20)	0.3145	57.60%
9	SRK	SPXY	SNV	SA (cr: 0.7)	PLSR (PLSc: 20)	0.3145	57.60%
10	SRK	KS	MSC	GA (pspc: 0.7 pm: 0.1)	PLSR (PLSc: 20)	0.3214	56.67%
11	SRK	SPXY	MSC	SA	PLSR	0.3241	56.31%

				(cr: 0.6)	(PLSc: 20)		
12	SRK	KS	SG 1st_der (opf: 3)	UVE (av: 0.8)	PLSR (PLSc: 20)	0.3279	55.80%
13	SRK	KS	SNV	GA (pspc: 0.5 pm: 0.08)	PLSR (PLSc: 20)	0.3284	55.73%
14	SRK	KS	MSC	SA (cr: 0.8)	PLSR (PLSc: 20)	0.3294	55.59%
15	SRK	KS	SNV	SA (cr: 0.8)	PLSR (PLSc: 20)	0.3341	54.96%
16	SRK	SPXY	MSC	SPA	PLSR (PLSc: 20)	0.3341	54.96%
17	SRK	SPXY	MSC	BiPLS (si: 37)	PLSR (PLSc: 20)	0.3347	54.88%
18	SRK	KS	SG 1st_der (opf: 3)	GA (pspc: 0.5 pm: 0.08)	PLSR (PLSc: 20)	0.3357	54.75%
19	SRK	KS	MSC	FiPLS (si: 54)	PLSR (PLSc: 20)	0.3461	53.34%
20	SRK	SPXY	SG 1st_der (opf: 3)	UVE (av: 1)	PLSR (PLSc: 20)	0.3522	52.52%
21	SRK	KS	SG 1st_der (opf: 3)	SPA	PLSR (PLSc: 20)	0.3541	52.26%
22	SRK	SPXY	SNV	SPA	PLSR (PLSc: 20)	0.3547	52.18%
23	SRK	SPXY	MSC	FiPLS (si: 54)	PLSR (PLSc: 20)	0.3574	51.82%
24	SRK	SPXY	SG 1st_der (opf: 3)	SA (cr: 0.7)	PLSR (PLSc: 20)	0.3578	51.77%

25	SRK	SPXY	MSC	iPLS (si: 72)	PLSR (PLSc: 20)	0.3578	51.77%
26	SRK	SPXY	SG 2nd_der (opf: 4)	UVE (av: 1)	PLSR (PLSc: 20)	0.3597	51.51%
27	SRK	SPXY	SG 1st_der (opf: 2)	GA (pspc: 0.5 pm: 0.05)	PLSR (PLSc: 20)	0.3604	51.42%
28	SRK	KS	SG 1st_der (opf: 3)	FiPLS (si: 60)	PLSR (PLSc: 20)	0.3647	50.84%
29	SRK	KS	SG 2nd_der (opf: 5)	GA (pspc: 0.5 pm: 0.05)	PLSR (PLSc: 20)	0.3674	50.47%
30	SRK	SPXY	SNV	GA (pspc: 0.6 pm: 0.05)	PLSR (PLSc: 20)	0.3674	50.47%
31	SRK	SPXY	SNV	BiPLS (si: 42)	PLSR (PLSc: 20)	0.3677	50.43%
32	SRK	SPXY	SNV	FiPLS (si: 57)	PLSR (PLSc: 20)	0.3678	50.42%
33	SRK	SPXY	SG 2nd_der (opf: 5)	SA (cr: 0.8)	PLSR (PLSc: 20)	0.3678	50.42%
34	SRK	KS	SG 1st_der (opf: 3)	BiPLS (si: 46)	PLSR (PLSc: 20)	0.3684	50.34%
35	SRK	SPXY	SG 2nd_der (opf: 4)	GA (pspc: 0.8 pm: 0.07)	PLSR (PLSc: 20)	0.3698	50.15%
36	SRK	KS	SNV	FiPLS	PLSR	0.3746	49.50%

				(si: 64)	(PLSc: 20)		
37	SRK	SPXY	SG 1st_der (opf: 3)	SPA	PLSR (PLSc: 20)	0.3748	49.47%
38	SRK	SPXY	SNV	iPLS (si: 73)	PLSR (PLSc: 20)	0.3751	49.43%
39	SRK	KS	MSC	iPLS (si: 72)	PLSR (PLSc: 20)	0.3754	49.39%
40	SRK	KS	SG 2nd_der (opf: 4)	UVE (av: 0.9)	PLSR (PLSc: 20)	0.3766	49.23%
41	SRK	SPXY	SG 2nd_der (opf: 4)	SPA	PLSR (PLSc: 20)	0.3777	49.08%
42	SRK	SPXY	SG 2nd_der (opf: 2)	iPLS (si: 71)	PLSR (PLSc: 20)	0.3798	48.80%
43	SRK	SPXY	SG 2nd_der (opf: 5)	FiPLS (si: 66)	PLSR (PLSc: 20)	0.3812	48.61%
44	SRK	KS	SNV	iPLS (si: 75)	PLSR (PLSc: 20)	0.3828	48.40%
45	SRK	KS	MSC	BiPLS (si: 49)	PLSR (PLSc: 20)	0.3841	48.22%
46	SRK	KS	SG 2nd_der (opf: 4)	SA (cr: 0.7)	PLSR (PLSc: 20)	0.3874	47.78%
47	SRK	KS	SG 2nd_der (opf: 5)	FiPLS (si: 65)	PLSR (PLSc: 20)	0.3878	47.72%
48	SRK	KS	SG 2nd_der (opf: 4)	BiPLS (si: 51)	PLSR (PLSc: 20)	0.3946	46.81%

49	SRK	KS	SNV	BiPLS (si: 55)	PLSR (PLSc: 20)	0.3974	46.43%
50	SRK	SPXY	SG 1st_der (opf: 2)	FiPLS (si: 66)	PLSR (PLSc: 20)	0.3977	46.39%
51	SRK	KS	SG 1st_der (opf: 2)	iPLS (si: 68)	PLSR (PLSc: 20)	0.3987	46.25%
52	SRK	KS	SG 2nd_der (opf: 4)	iPLS (si: 69)	PLSR (PLSc: 20)	0.3997	46.12%
53	SRK	SPXY	SG 1st_der (opf: 2)	BiPLS (si: 48)	PLSR (PLSc: 20)	0.4021	45.79%
51	SRK	SPXY	SG 2nd_der (opf: 3)	BiPLS (si: 45)	PLSR (PLSc: 20)	0.4056	45.32%
55	SRK	KS	SG 2nd_der (opf: 4)	SPA	PLSR (PLSc: 20)	0.4084	44.94%
56	SRK	SPXY	SG 1st_der (opf: 3)	iPLS (si: 65)	PLSR (PLSc: 20)	0.4104	44.68%

APPENDIX 2.2
TABLE 2: THE PERFORMANCE OF ALL COMBINATIONS OF METHODS ON SBK.

Combination ID	Rice Form	Methods				Assessment	
		Dataset Partition	Pre-processing	Variable Selection	Calibration	RMSEP	PDRMSEP
1	SBK	KS	SNV	UVE (av: 1)	PLSR (PLSc: 20)	0.2776	56.58%
2	SBK	KS	SNV	GA (pspc: 0.8 pm: 0.1)	PLSR (PLSc: 20)	0.2874	55.04%
3	SBK	SPXY	SNV	GA (pspc: 0.8 pm: 0.1)	PLSR (PLSc: 20)	0.2874	55.04%
4	SBK	KS	SNV	SA (cr: 0.6)	PLSR (PLSc: 20)	0.2942	53.98%
5	SBK	SPXY	SNV	UVE (av: 0.9)	PLSR (PLSc: 20)	0.3031	52.59%
6	SBK	KS	MSC	GA (pspc: 0.7 pm: 0.1)	PLSR (PLSc: 20)	0.3074	51.92%
7	SBK	SPXY	SNV	SA (cr: 0.5)	PLSR (PLSc: 20)	0.3075	51.90%
8	SBK	SPXY	SNV	SPA	PLSR (PLSc: 20)	0.3127	51.09%
9	SBK	KS	SNV	SPA	PLSR (PLSc: 20)	0.3154	50.66%
10	SBK	KS	MSC	SA (cr: 0.7)	PLSR (PLSc: 20)	0.3171	50.40%

11	SBK	SPXY	MSC	GA (pspc: 0.8 pm: 0.07)	PLSR (PLSc: 20)	0.3179	50.27%
12	SBK	SPXY	SNV	BiPLS (si: 32)	PLSR (PLSc: 20)	0.3247	49.21%
13	SBK	KS	MSC	SPA	PLSR (PLSc: 20)	0.3341	47.74%
14	SBK	KS	SNV	BiPLS (si: 32)	PLSR (PLSc: 20)	0.3354	47.54%
15	SBK	SPXY	MSC	SA (cr: 0.6)	PLSR (PLSc: 20)	0.3354	47.54%
16	SBK	SPXY	SNV	FiPLS (si: 39)	PLSR (PLSc: 20)	0.337	47.29%
17	SBK	KS	SNV	FiPLS (si: 40)	PLSR (PLSc: 20)	0.3378	47.16%
18	SBK	KS	SNV	iPLS (si: 42)	PLSR (PLSc: 20)	0.3421	46.49%
19	SBK	SPXY	SNV	iPLS (si: 43)	PLSR (PLSc: 20)	0.3427	46.39%
20	SBK	SPXY	SG 1st_der (opf: 3)	SA (cr: 0.7)	PLSR (PLSc: 20)	0.3457	45.93%
21	SBK	KS	MSC	BiPLS (si: 37)	PLSR (PLSc: 20)	0.3512	45.06%
22	SBK	KS	MSC	FiPLS (si: 32)	PLSR (PLSc: 20)	0.3572	44.13%
23	SBK	SPXY	MSC	SPA	PLSR (PLSc: 20)	0.3671	42.58%
24	SBK	KS	MSC	iPLS (si: 45)	PLSR (PLSc: 20)	0.3674	42.53%

25	SBK	KS	SG 2nd_der (opf: 4)	BiPLS (si: 34)	PLSR (PLSc: 20)	0.3674	42.53%
26	SBK	SPXY	SG 1st_der (opf: 2)	SPA	PLSR (PLSc: 20)	0.3674	42.53%
27	SBK	SPXY	SG 1st_der (opf: 3)	GA (pspc: 0.6 pm: 0.06)	PLSR (PLSc: 20)	0.3674	42.53%
28	SBK	KS	SG 2nd_der (opf: 5)	iPLS (si: 44)	PLSR (PLSc: 20)	0.3714	41.91%
29	SBK	SPXY	SG 1st_der (opf: 3)	iPLS (si: 41)	PLSR (PLSc: 20)	0.3715	41.89%
30	SBK	SPXY	SG 2nd_der (opf: 4)	GA	PLSR (PLSc: 20)	0.3752	41.31%
31	SBK	SPXY	SG 1st_der (opf: 3)	FiPLS (si: 37)	PLSR (PLSc: 20)	0.3754	41.28%
32	SBK	KS	SG 1st_der (opf: 3)	SA (cr: 0.5)	PLSR (PLSc: 20)	0.3781	40.86%
33	SBK	SPXY	SG 1st_der (opf: 3)	BiPLS (si: 36)	PLSR (PLSc: 20)	0.3817	40.29%
34	SBK	KS	SG 2nd_der (opf: 4)	FiPLS (si: 40)	PLSR (PLSc: 20)	0.3842	39.90%
35	SBK	KS	SG 1st_der (opf: 2)	BiPLS (si: 37)	PLSR (PLSc: 20)	0.3871	39.45%
36	SBK	SPXY	SG 2nd_der (opf: 4)	iPLS (si: 43)	PLSR (PLSc: 20)	0.3872	39.43%
37	SBK	SPXY	MSC	iPLS (si: 44)	PLSR (PLSc: 20)	0.3874	39.40%
38	SBK	SPXY	SG 2nd_der (opf: 5)	SA (cr: 0.7)	PLSR (PLSc: 20)	0.3874	39.40%

39	SBK	KS	SG 2nd_der (opf: 5)	SA (cr: 0.7)	PLSR (PLSc: 20)	0.3878	39.34%
40	SBK	SPXY	SG 1st_der (opf: 3)	UVE (av: 0.8)	PLSR (PLSc: 20)	0.3884	39.25%
41	SBK	SPXY	SG 2nd_der (opf: 4)	UVE (av: 0.9)	PLSR (PLSc: 20)	0.3918	38.71%
42	SBK	KS	SG 1st_der (opf: 3)	GA (pspc: 0.7 pm: 0.07)	PLSR (PLSc: 20)	0.3941	38.35%
43	SBK	KS	SG 2nd_der (opf: 4)	SPA	PLSR (PLSc: 20)	0.3952	38.18%
44	SBK	SPXY	SG 2nd_der (opf: 5)	BiPLS (si: 35)	PLSR (PLSc: 20)	0.3972	37.87%
45	SBK	SPXY	MSC	BiPLS (si: 37)	PLSR (PLSc: 20)	0.3974	37.84%
46	SBK	SPXY	MSC	FiPLS (si: 38)	PLSR (PLSc: 20)	0.3975	37.82%
47	SBK	KS	SG 2nd_der (opf: 4)	GA (pspc: 0.8 pm: 0.08)	PLSR (PLSc: 20)	0.3984	37.68%
48	SBK	KS	SG 1st_der (opf: 3)	UVE (av: 1)	PLSR (PLSc: 20)	0.3994	37.53%
49	SBK	KS	SG 2nd_der (opf: 5)	UVE (av: 1)	PLSR (PLSc: 20)	0.3994	37.53%
50	SBK	KS	SG 1st_der (opf: 2)	SPA	PLSR (PLSc: 20)	0.4001	37.42%
51	SBK	KS	SG 1st_der (opf: 3)	FiPLS (si: 40)	PLSR (PLSc: 20)	0.4025	37.04%
52	SBK	SPXY	SG 2nd_der	FiPLS	PLSR	0.4027	37.01%

			(opf: 5)	(si: 39)	(PLSc: 20)		
53	SBK	SPXY	SG 2nd_der (opf: 4)	SPA	PLSR (PLSc: 20)	0.4027	37.01%
51	SBK	KS	SG 1st_der (opf: 3)	iPLS (si: 43)	PLSR (PLSc: 20)	0.4125	35.48%
55	SBK	SPXY	MSC	UVE (av: 0.9)	PLSR (PLSc: 20)	0.4265	33.29%
56	SBK	KS	MSC	UVE (av: 0.9)	PLSR (PLSc: 20)	0.4278	33.08%

APPENDIX 2.3
TABLE 3: THE PERFORMANCE OF ALL COMBINATIONS OF METHODS ON RF.

Combination ID	Rice Form	Methods				Assessment	
		Dataset Partition	Pre-processing	Variable Selection	Calibration	RMSEP	PDRMSEP
1	RF	SPXY	MSC	GA (pspc: 0.6 pm: 0.09)	PLSR (PLSc: 20)	0.3187	42.38%
2	RF	SPXY	SNV	SA (cr: 0.7)	PLSR (PLSc: 20)	0.3274	40.81%
3	RF	SPXY	SNV	UVE (av: 1)	PLSR (PLSc: 20)	0.3339	39.63%
4	RF	KS	SNV	SA (cr: 0.6)	PLSR (PLSc: 20)	0.33571	39.30%
5	RF	KS	MSC	UVE (av: 1)	PLSR (PLSc: 20)	0.3374	39.00%
6	RF	KS	MSC	GA (pspc: 0.7 pm: 0.07)	PLSR (PLSc: 20)	0.3387	38.76%
7	RF	KS	SNV	UVE (av: 1)	PLSR (PLSc: 20)	0.3388	38.75%
8	RF	KS	MSC	SA (cr: 0.8)	PLSR (PLSc: 20)	0.3427	38.04%
9	RF	SPXY	SNV	SPA	PLSR (PLSc: 20)	0.3471	37.24%
10	RF	SPXY	MSC	SA (cr: 0.6)	PLSR (PLSc: 20)	0.3478	37.12%
11	RF	SPXY	SG 1st_der	GA	PLSR	0.3487	36.96%

			(opf: 2)	(pspc: 0.5 pm: 0.05)	(PLSc: 20)		
12	RF	SPXY	MSC	UVE (av: 1)	PLSR (PLSc: 20)	0.3505	36.63%
13	RF	SPXY	SG 2nd_der (opf: 4)	SA (cr: 0.6)	PLSR (PLSc: 20)	0.3517	36.41%
14	RF	KS	SNV	GA (pspc: 0.6 pm: 0.08)	PLSR (PLSc: 20)	0.3571	35.44%
15	RF	KS	MSC	SPA	PLSR (PLSc: 20)	0.3571	35.44%
16	RF	SPXY	SNV	BiPLS (si: 65)	PLSR (PLSc: 20)	0.3574	35.38%
17	RF	KS	MSC	FiPLS (si: 70)	PLSR (PLSc: 20)	0.3621	34.53%
18	RF	KS	SNV	SPA	PLSR (PLSc: 20)	0.3645	34.10%
19	RF	SPXY	SNV	iPLS (si: 68)	PLSR (PLSc: 20)	0.3664	33.76%
20	RF	KS	SNV	FiPLS (si: 66)	PLSR (PLSc: 20)	0.3674	33.57%
21	RF	KS	MSC	iPLS (si: 71)	PLSR (PLSc: 20)	0.3674	33.57%
22	RF	SPXY	SG 2nd_der (opf: 5)	GA (pspc: 0.7 pm: 0.06)	PLSR (PLSc: 20)	0.3674	33.57%
23	RF	SPXY	MSC	SPA	PLSR (PLSc: 20)	0.3678	33.50%
24	RF	KS	MSC	BiPLS	PLSR	0.3715	32.83%

				(si: 64)	(PLSc: 20)		
25	RF	SPXY	SNV	FiPLS (si: 68)	PLSR (PLSc: 20)	0.3728	32.60%
26	RF	SPXY	SG 2nd_der (opf: 4)	SPA	PLSR (PLSc: 20)	0.3758	32.06%
27	RF	KS	SNV	BiPLS (si: 69)	PLSR (PLSc: 20)	0.3777	31.71%
28	RF	SPXY	MSC	FiPLS (si: 71)	PLSR (PLSc: 20)	0.378	31.66%
29	RF	KS	SNV	iPLS (si: 74)	PLSR (PLSc: 20)	0.3784	31.59%
30	RF	KS	SG 2nd_der (opf: 5)	iPLS (si: 73)	PLSR (PLSc: 20)	0.3789	31.50%
31	RF	SPXY	SG 1st_der (opf: 2)	FiPLS (si: 70)	PLSR (PLSc: 20)	0.3789	31.50%
32	RF	KS	SG 1st_der (opf: 3)	GA (pspc: 0.6 pm: 0.1)	PLSR (PLSc: 20)	0.3841	30.56%
33	RF	SPXY	MSC	iPLS (si: 74)	PLSR (PLSc: 20)	0.3871	30.01%
34	RF	SPXY	SG 1st_der (opf: 3)	SA (cr: 0.6)	PLSR (PLSc: 20)	0.3872	29.99%
35	RF	SPXY	SG 2nd_der (opf: 4)	GA (pspc: 0.8 pm: 0.06)	PLSR (PLSc: 20)	0.3897	29.54%
36	RF	SPXY	MSC	BiPLS (si: 70)	PLSR (PLSc: 20)	0.3927	29.00%
37	RF	KS	SG 2nd_der (opf: 4)	GA (pspc: 0.7	PLSR (PLSc: 20)	0.394	28.77%

				pm: 0.08)			
38	RF	SPXY	SG 1st_der (opf: 3)	iPLS (si: 69)	PLSR (PLSc: 20)	0.3957	28.46%
39	RF	KS	SG 1st_der (opf: 3)	SA (cr: 0.8)	PLSR (PLSc: 20)	0.3975	28.13%
40	RF	SPXY	SG 1st_der (opf: 2)	SPA	PLSR (PLSc: 20)	0.3998	27.72%
41	RF	KS	SG 1st_der (opf: 3)	FiPLS (si: 71)	PLSR (PLSc: 20)	0.4027	27.19%
42	RF	SPXY	SG 2nd_der (opf: 4)	iPLS (si: 74)	PLSR (PLSc: 20)	0.4075	26.32%
43	RF	KS	SG 1st_der (opf: 2)	SPA	PLSR (PLSc: 20)	0.4087	26.11%
44	RF	KS	SG 2nd_der (opf: 5)	SA (cr: 0.8)	PLSR (PLSc: 20)	0.4087	26.11%
45	RF	SPXY	SG 2nd_der (opf: 4)	UVE (av: 0.8)	PLSR (PLSc: 20)	0.4116	25.58%
46	RF	SPXY	SG 1st_der (opf: 3)	BiPLS (si: 68)	PLSR (PLSc: 20)	0.4128	25.37%
47	RF	SPXY	SG 2nd_der (opf: 4)	BiPLS (si: 67)	PLSR (PLSc: 20)	0.4158	24.82%
48	RF	KS	SG 1st_der (opf: 3)	iPLS (si: 72)	PLSR (PLSc: 20)	0.4187	24.30%
49	RF	KS	SG 2nd_der (opf: 4)	FiPLS (si: 69)	PLSR (PLSc: 20)	0.423	23.52%
50	RF	SPXY	SG 1st_der (opf: 3)	UVE (av: 0.7)	PLSR (PLSc: 20)	0.4255	23.07%
51	RF	SPXY	SG 2nd_der (opf: 5)	FiPLS (si: 68)	PLSR (PLSc: 20)	0.427	22.80%

52	RF	KS	SG 1st_der (opf: 3)	UVE (av: 0.9)	PLSR (PLSc: 20)	0.4272	22.76%
53	RF	KS	SG 2nd_der (opf: 5)	UVE (av: 0.8)	PLSR (PLSc: 20)	0.4286	22.51%
51	RF	KS	SG 1st_der (opf: 2)	BiPLS (si: 71)	PLSR (PLSc: 20)	0.4298	22.29%
55	RF	KS	SG 2nd_der (opf: 4)	BiPLS (si: 70)	PLSR (PLSc: 20)	0.435	21.35%
56	RF	KS	SG 2nd_der (opf: 5)	SPA	PLSR (PLSc: 20)	0.447	19.18%

APPENDIX 3: MATLAB CODES FOR METHODS

KENNARD-STONE ALGORITHM (KS)

```
function [index,distance,TsetX,TsetY,VsetX,VsetY] = KS(spectra,RefData,k)

%%        KS(Kennard-Stone Algorithm)


%----------------------Input------------------------

% spectra: spectral data matrix: n(samples) x m(variables)

% RefData: reference data: n(samples) x p(properties)

% k: number of samples to be selected for training set


%----------------------Output-----------------------

% index: indices (of rows in X) of the selected samples for training set

% distance: largest minimum Euclidean distance.

%          distance(1)=0

%          distance(2)=Euclidean distance between the 1st pair of samples selected by the algorithm

%                distance(i)=largest minimum Euclidean distance between the i-th selected sample and

%                      the previously selected samples(i>2)

% TsetX: spectra data of training set: k(samples) x m(variables)

% TsetY: reference of training set: k(samples) x p(properties)

% VsetX: spectral data validation set: n-k(samples) x m(variables)

% VsetY: reference of validation set: n-k(samples) x p(properties)


%%        Last modified by Shupeng Hu. July 5th 2019


% Check the input
```

```
[nSmp,~] = size(spectra);
if k >= nSmp
    k = nSmp;
end

%initialize the parameters
index = [];
distance=ones(k,1);
distance(1)=0;

%Calcuate the Euclidean distance between pairs of observations in X. For
%example, D(1,3)=D(3,1)= the Euclidean distance between 1st observation and 3rd
D = squareform(pdist(spectra));

%find the first pair of observations which has the largest Euclidean distance
[index(1),index(2)] = find(D==max(max(D)),1,'first');
distance(2)=max(max(D));

%stepwise selection
while length(index) < k
    %calculate the minimum Euclidean distances between every remaining samples and the selected samples
    minDist = min(D(index,:));
    %find the sample which has the largest minimum Euclidean distance
    [distance(length(index)+1),trIdx1] = max(minDist);
    index = [index,trIdx1];
end
```

```
%partition training set and validation set
TsetX=spectra(index,:); %spectra data of training set
spectra(index,:)=[];
VsetX=spectra; %spectral data validation set
TsetY=RefData(index,:); %reference of training set
RefData(index,:)=[];
VsetY=RefData; %reference of validation set

End
```

SAMPLE SET PARTITIONING BASED ON JOINT X-Y DISTANCES (SPXY)

```
function [index,distance,TsetX,TsetY,VsetX,VsetY] = SPXY(spectra,RefData,k)

%%      SPXY(Sample set Partitioning based on joint X-Y distances)


%----------------------Input-----------------------------------------

% spectra: spectral data matrix: n(samples) x m(varibales)

% RefData: reference data: n(samples) x p(properties)

% k: number of samples to be selected (minimum of 2)


%----------------------Output----------------------------------------

% index: indices of the selected samples for training set

% distance: largest minimum joint XY distance

%          distance(1) = 0;

%          distance(2) = joint XY distance between the 1st pair of samples selected by the algorithm

%          distance(i) = largest minimum joint XY distance between the i-th selected sample and

%          the previously selected samples (i > 2)

% TsetX: spectra data of training set: k(samples) x m(varibales)

% TsetY: reference of training set: k(samples) x p(properties)

% VsetX: spectral data validation set: n-k(samples) x m(varibales)

% VsetY: reference of validation set: n-k(samples) x p(properties)


%%      Last modified by Shupeng Hu. July 5th 2019


distance = zeros(1,k); % Initializes the vector of minimum distances

M = size(spectra,1); % Number of rows in X (samples)
```

```
samples = 1:M;
originalReference=RefData;

% Auto-scales the Y matrix
for i=1:size(RefData,2) % For each parameter in Y
   yi = RefData(:,i);
   RefData(:,i) = (yi - mean(yi))/std(yi);
end

D = zeros(M,M); % Initializes the matrix of X distances
Dy = zeros(M,M); % Initializes the matrix of Y distances

for i=1:M-1
   xa = spectra(i,:);
   ya = RefData(i,:);
   for j = i+1:M
     xb = spectra(j,:);
     yb = RefData(j,:);
     D(i,j) = norm(xa - xb);
     Dy(i,j) =  norm(ya - yb);
   end
end

Dmax = max(max(D));
Dymax = max(max(Dy));

D = D/Dmax + Dy/Dymax; % Combines the distances in X and Y
```

```
% D: Upper Triangular Matrix

% D(i,j) = Euclidean distance between objects i and j (j > i)


[maxD,index_row] = max(D); % maxD = Row vector containing the largest element of each column in D

                % index_row(n) = Index of the row with the largest element in the n-th column


[~,index_column] = max(maxD); % index_column = column corresponding to the largest element in matrix D


index = [];

index(1) = index_row(index_column);

index(2) = index_column;


distance(2) = D(index(1),index(2));


for i = 3:k

   % This routine determines the distances between each sample still available for selection and each of the samples already selected

   pool = setdiff(samples,index); % pool = Samples still available for selection

   dmin = zeros(1,M-i+1); % Initializes the vector of minimum distances between each sample in pool and the samples already selected

   for j = 1:(M-i+1) % For each sample xa still available for selection

      indexa = pool(j); % indexa = index of the j-th sample in pool (still available for selection)

      d = zeros(1,i-1); % Initializes the vector of distances between the j-th sample in pool and the samples already selected
```

```
        for p = 1:(i-1) % The distance with respect to each sample already selected is analyzed
            indexb =  index(p); % indexb = index of the k-th sample already selected
            if indexa < indexb
                d(p) = D(indexa,indexb);
            else
                d(p) = D(indexb,indexa);
            end
        end
        dmin(j) = min(d);
    end
    % The selected sample corresponds to the largest dmin
    [distance(i),trIdx1] = max(dmin);
    index(i) = pool(trIdx1);
end



%partition training set and validation set
TsetX=spectra(index,:); %spectra data of training set
spectra(index,:)=[];
VsetX=spectra; %spectral data validation set
TsetY=originalReference(index,:); %reference of training set
originalReference(index,:)=[];
VsetY=originalReference; %reference of validation set


end
```

STANDARD NORMAL VARIATE (SNV)

```
function [Xcorr] = SNV(spectra)
%%          SNV(Standard Normal Variate)

%------------------------------Input----------------------
% spectra: original spectra: n(samples) x m(varibales)

%------------------------------Output---------------------
% Xcorr: corrected spectra

%%      Last modified by Shupeng Hu. April 18th 2018

[m,n]=size(spectra);
Xcorr=spectra;
for i=1:m
   for j=1:n
      Xcorr(i,j)=(spectra(i,j)-mean(spectra(i,:),2))/std(spectra(i,:));
   end
end

end
```

MULTIPLICATIVE SCATTER CORRECTION (MSC)

```
function [Xcorr] = MSC( spectra,Xref )

%%        MSC(Multiplicative Scatter Correction)


%-----------------------------Input----------------------

% spectra: original spectra: n(samples) x m(varibales)

% Xref: reference spectra from calibration set. Usually calculate the average

%       spectrum of the calibration set as the reference spectrum.

%       size: n(samples) x m(varibales)


%-----------------------------Output----------------------

% Xcorr: corrected spectra


%%     Last modified by Shupeng Hu. April 18th 2018


[m,n]=size(spectra);

me=mean(Xref); %calculate the average spectrum of the calibration set

Xcorr=ones(m,n);


for i=1:m          %for each spectrum in spectra

   p=polyfit(me,spectra(i,:),1);        % least square fit between mean spectrum and each spectrum in spectra (first-degree polynomial)

   Xcorr(i,:)=(spectra(i,:)-p(1,2)*ones(1,n))./(p(1,1)*ones(1,n)); % each spectrum in spectra is corrected

end


end
```

SAVITZKY-GOLAY POLYNOMIAL DERIVATIVE FILTERS (SG)

```
function [Xcorr]=SG(spectra,wavelength,ddx,N,F)

%%                SG(Savitzky-Golay polynomial derivative filters)


%-----------------------------Input----------------------

% spectra: spectral data matrix: n(samples) x m(varibales)

% wavelength: the corresponding wavelengths of the spectra

% ddx: the order of spectral derivative: ddx=0 : smoothing

%                                ddx=1 : first derivative

%                                ddx=2 : second derivative

% N: order of polynomial fit, N>=ddx

% F: window length which must be an odd positive integer


%-----------------------------Output----------------------

% Xcorr: spectra corrected by SG. The number of points lost equals to F.


%%     Last modified by Shupeng Hu. July 2th 2019


[m,p] = size(spectra); %% m number of sample, p is spectrum wavelength span

dt=(wavelength(1)-wavelength(2));

HalfWin  = ((F+1)/2) -1;


[~,g] = sgolay(N,F);   % Calculate S-G coefficients


Xcorr = zeros(m,p);

v1=g(:,ddx+1)';
```

```
v2=repmat(v1,m,1);

for n = (F+1)/2:p-(F+1)/2

  v3=spectra(:,n - HalfWin: n + HalfWin);

switch ddx
   case 0
      Xcorr(:,n)=dot(v2,v3,2);
   case 1
      Xcorr(:,n)=dot(v2,v3,2);
   case 2
      Xcorr(:,n)=2*dot(v2,v3,2)';
end



end

switch ddx
   case 0
   case 1
     Xcorr = Xcorr/dt;
   case 2
     Xcorr = Xcorr/(dt*dt);
end

end
```

SUCCESSIVE PROJECTIONS ALGORITHM (SPA)

```
function [var_sel,sTsetX] = SPA(TsetX,TsetY,VsetX,VsetY)

%%          SPA(Successive Projections Algorithm)


% [var_sel,var_sel_phase2] = SPAa(Xcal,ycal,Xval,yval) --> Validated by validation set

% [var_sel,var_sel_phase2] = SPA(Xcal,ycal,[],[]) --> Cross-validation


%------------------------------Input------------------

% TsetX: spectral data matrix of training set

%       size:n(samples) x m(varibales),

% TsetY: reference data of training set: n(samples) x p(properties),p=1

% VsetX: spectral data matrix of validation set: n(samples) x m(varibales)

% VsetY: reference data of validation set: n(samples) x p(properties),p=1


%------------------------------Output-----------------

% var_sel: index set of selected variables

% sTsetX: variables-selected spectra for training set



%% Last modified by Shupeng Hu. on Sept.9th 2019



N = size(TsetX,1); % number of samples in training set

K = size(TsetX,2); % number of original variables

m_min = 1;

m_max = min(N-2,K);
```

```
%Step 1:Projection operations for the selection of candidate subsets

% The projections are applied to the columns of Xcal after mean-centering
for k = 1:K
   x = TsetX(:,k);
   Xcaln(:,k) = (x - mean(x));
end

SEL = zeros(m_max,K);
for k = 1:K
   SEL(:,k) = projections_qr(Xcaln,k,m_max);%Index set of the variables resulting from the projection operations
end

disp('Step 1 (projections) completed !')

%Step 2:Evaluation of the candidate subsets according to the PRESS criterion

PRESS = Inf*ones(m_max,K);
for k = 1:K
   for m = m_min:m_max
      var_sel = SEL(1:m,k); % index set of selected variables
      [~,~,press] = validation(TsetX,TsetY,VsetX,VsetY,var_sel);
      PRESS(m,k) = press;
   end
```

```
end

%PRESSmin is the minimum value of PRESS at each column, m_sel is the index of rows for the minimum value of PRESS

[PRESSmin,m_sel] = min(PRESS);

% the minimum value of PRESS among all cases, k_sel is the index of column

[~,k_sel] = min(PRESSmin);

%index of selected variables in the best candidate subset

var_sel_phase2 = SEL(1:m_sel(k_sel),k_sel);


disp('Step 2 (evaluation of variable subsets) completed !')


%----------------Step 3:Final elimination of variables------

%-------------Step 3.1:Calculation of the relevance index--


Xcal2 = [ones(N,1),TsetX(:,var_sel_phase2)];

b = Xcal2\TsetY; % MLR with intercept term

std_deviation = std(Xcal2);

relev = abs(b.*std_deviation');

relev = relev(2:end); % The intercept term is always included

% Sorts the selected variables in decreasing order of "relevance"

[~,index_increasing_relev] = sort(relev); % Increasing order

index_decreasing_relev = index_increasing_relev(end:-1:1); % Decreasing order


%----------------Step 3.2:Calculation of PRESS values----------------------


for i = 1:length(var_sel_phase2)
```

```
    [~,e,press]                                                                    =
validation(TsetX,TsetY,VsetX,VsetY,var_sel_phase2(index_decreasing_relev(1:i)) );
    PRESS_scree(i) =press;
end
RMSEP_scree = sqrt(PRESS_scree/length(e));

figure, grid, hold on
plot(RMSEP_scree)
xlabel('Number of variables included in the model'),ylabel('RMSE')

%----------------------Step 3.3:F-test criterion----------------------------
PRESS_scree_min = min(PRESS_scree);
alpha = 0.05;
dof = length(e); % Number of degrees of freedom
fcrit = finv(1-alpha,dof,dof); % Critical F-value
PRESS_crit = PRESS_scree_min*fcrit;
% Finds the minimum number of variables for which PRESS_scree
% is not significantly larger than PRESS_scree_min
i_crit = min(find(PRESS_scree < PRESS_crit));
i_crit = max(m_min,i_crit); % The number of selected variables must be at least m_min

var_sel = var_sel_phase2( index_decreasing_relev(1:i_crit) );
sTsetX=TsetX(:,var_sel); %variables-selected spectra for training set

title(['Final number of selected variables: ' num2str(length(var_sel)) '   (RMSE = '
num2str(RMSEP_scree(i_crit)) ')'])

% Indicates the selected point on the scree plot
```

```
plot(i_crit,RMSEP_scree(i_crit),'s')

disp('Step 3 (final elimination of variables) completed !')

% Presents the selected variables
% in the first object of the calibration set
figure,plot(TsetX(1,:));hold,grid
plot(var_sel,TsetX(1,var_sel),'s')
legend('Original variables','Selected variables')
xlabel('Variable index')

end


function chain = projections_qr(X,k,M)

% Projections routine for the Successive Projections Algorithm using the
% built-in QR function of Matlab
%
% chain = projections(X,k,M)
%
% X --> Matrix of predictor variables (# objects N x # variables K)
% k --> Index of the initial column for the projection operations
% M --> Number of variables to include in the chain
%
% chain --> Index set of the variables resulting from the projection operations
```

```
X_projected = X;

norms = sum(X_projected.^2);    % Square norm of each column vector
norm_max = max(norms); % Norm of the "largest" column vector

X_projected(:,k) = X_projected(:,k)*2*norm_max/norms(k); % Scales the kth column so that it becomes the "largest" column

[~,~,order] = qr(X_projected,0);
chain = order(1:M)';

function [yhat,e,PRESS] = validation(Xcal,ycal,Xval,yval,var_sel)

% [yhat,e] = validation(Xcal,ycal,Xval,yval,var_sel) --> Validation with a separate set
% [yhat,e] = validation(Xcal,ycal,[],[],var_sel) --> Cross-validation

N = size(Xcal,1); % Number of objects in the calibration set
NV = size(Xval,1); % Number of objects in the validation set

if NV > 0 % Validation with a separate set
   Xcal_ones = [ones(N,1) Xcal(:,var_sel)];
   b = Xcal_ones\ycal; % MLR with offset term (b0)
   yhat = [ones(NV,1) Xval(:,var_sel)]*b; % Prediction over the validation set
   e = yval - yhat; % Validation error
   PRESS=sumsqr(e);
else % Cross-validation
   yhat = zeros(N,1); % Setting the proper dimensions of yhat
```

```
    for i = 1:N
      % Removing the ith object from the calibration set
      cal = [[1:i-1] [i+1:N]];
      X = Xcal(cal,var_sel);
      y = ycal(cal);
      xtest = Xcal(i,var_sel);
      ytest = ycal(i);
      X_ones = [ones(N-1,1) X];
      b = X_ones\y; % MLR with offset term (b0)
      yhat(i) = [1 xtest]*b; % Prediction for the ith object
    end
    e = ycal - yhat; % Cross-validation error
     PRESS=sumsqr(e);
end
```

UNINFORMATIVE VARIABLE ELIMINATION (UVE)

```
Function[UV,var_sel,sTsetX,sVsetX]
=UVE(TsetX,TsetY,VsetX,componentNo,k)
%%          UVE(Uninformative Variable Elimination)

% this UVE method employs PLS to obtain regression coefficient vector. So
% before UVE, PLSR should be used to determine the suitable number of
% PLS principal components

%----------------------Input----------------------------
% TsetX: spectral data matrix of training set
%       size:n(samples) x m(varibales),
% TsetY: reference data of training set: n(samples) x p(properties)
% VsetX: spectral data matrix of validation set: n(samples) x m(varibales)
% componentNo: the number of selected PLS principal components
% k:(optional, default=1)an arbitrary value to control the cutoff by:
cutoff=k*max(abs(stability_noise))

%----------------------Output----------------------------
% UV:index set of uninformative variables
% var_sel: index set of informative variables
% sTsetX: variables-selected spectra for training set
% sVsetX: variables-selected spectra for validation set

%%      Last modified by Shupeng Hu. Sept. 10th 2019

if nargin<5
```

```
    k=1;
end

%--------Step 1:create the noise matrix--------------------
[n,m]=size(TsetX);
noise=1E-10*rand(n,m);
newX=[TsetX,noise];

%----------Step 2:leave-one-out cross-validation-----------
disp('leave-one-out cross-validation starts!');

B=ones(n,size(newX,2));
for i=1:n
    out=i;     % index of the leave-one
    in=1:1:n;
    in(out)=[];% index set except the leave-one
    X=newX(in,:);
    Y=TsetY(in,:);

    [~,~,~,~,betaPLS,~,~,~] = plsregress(X,Y,componentNo);
    %B:b-matrix contains all b-vectors obtained from every iteration of leave-one-out cross-validation
    B(i,:)=betaPLS(2:end,1)';

end

disp('regression coefficient matrix obtained');
```

```
%--------------------Step 3:calculate stability and cutoff-----------------
stability=mean(B)./std(B);
stability_X=stability(1,1:m);
stability_noise=stability(1,(m+1):end);

cutoff=k*max(abs(stability_noise));

%----Step 4:identify uninformative variables by cutoff-----
UV=find(abs(stability_X)<=cutoff);  %index set of uninformative variables
var_sel=find(abs(stability_X)>cutoff); % index set of informative variables
sTsetX=TsetX(:,var_sel);   %selected variables for training set
sVsetX=VsetX(:,var_sel);   %selected variables for validation set

%draw graph
figure,plot(TsetX(1,:));hold,grid
plot(var_sel,TsetX(1,var_sel),'s')
legend('Original variables','Selected variables')
xlabel('Variable index')

end
```

SIMULATED ANNEALING (SA)

```
function [finalVal_Sel,finalenergy,sTsetX,sVsetX,total]=SA
(TsetX,TsetY,VsetX,componentNo,initenergy,c,initialT,stopT)

%%          SA(Simulated Annealing)


% the number of principal components should be determined before SA. So use

% PLSR or PCR to obtain a fixed number of PCs prior to SA


% Objective function F = currentenergy - newenergy

%           energy = sqrt(sumsqr(TsetY-y)/size(y,1));


%-----------------------------Input----------------------

% TsetX: spectral data matrix of training set

%       size:n(samples) x m(varibales),

% TsetY: reference data of training set: n(samples) x p(properties)

% VsetX: spectral data matrix of validation set: n(samples) x m(varibales)

% VsetY: reference data of validation set: n(samples) x p(properties)

% componentNo: the number of selected PLS principal components

% initenergy: initial energy/error value, for NIR it is the initial MSE

%           from cross-validation

% c: cooling ratio.A constant between (0,1) for cooling schedule.

%    default=0.8

% initialT: initial value of T. default=0.005

% stopT: final value of T at which to stop. default=1e-6


%-----------------------------Output----------------------
```

```
% finalVal_Sel: optimal solution, for NIR it is the optimal index set of SA-selected variables
% finalenergy: RMSEP of the optimal solution
% sTsetX: variables-selected spectra for training set
% sVsetX: variables-selected spectra for validation set
% total: total tries for all cases of T

%%       Last modified by Shupeng Hu. on Sept.11th 2019


%----------------------Step 1:configure parameters--------

[~,n]=size(TsetX);

%identify inputs
 if nargin==5
    c=0.8;
    initialT=0.005;
    stopT=1e-6;
 elseif nargin==6
    initialT=0.005;
    stopT=1e-6;
 elseif nargin==7
    stopT=1e-6;
end

%cooling schedule: a function takes a scalar as input and returns a smaller
```

```
%but positive scalar as output. the constant 0<c<1 is a parameter to control
%cooling ratio
coolSched=@(T) (c*T);

%the number of selected variables need to be at least 1 higher than the
%number of principal components. Maximum equal to the nearest integer to
%half of the number of original variables in the training set
minVal_Sel=componentNo+1;
maxVal_Sel=round(n/2);
%Generator: a new solution to replace the old one. Solution for NIR is the
%index set of selected variables.
Generator=@(min,max) (randperm(max,randi([min,max],1,1)));

%set the Markov chain
maxTries=(maxVal_Sel-minVal_Sel)*10;%maximum number of tries within one temperature
maxConsRej=round(maxTries/5);%maximum number of consecutive rejections within one temperature
maxSuccess=round(maxTries/3);%maximum number of successes within one temperature

stopRMSEP=-Inf; %RMSEP at which to stop immediately. -Inf means SA almost will not stop by this value
k = 1; % default Boltzmann constant

%----------------------Step 2:initialize values-----------

tries = 0; %counter the number of tries within one T
```

```
success = 0;% counter the number of successes within one T

finished = 0;% flag to control the loop

consRej = 0;% counter the number of consecutive rejections within one T

total=0; % total tries for all cases of T


T = initialT;

initVal_Sel=1:1:n; % initial index set of selected variables=full spectrum

currentVal_Sel=initVal_Sel; %current solution, for NIR it is the current index set of selected variables

currentenergy = initenergy; %current energy= current RMSEP


%--------------------------Step 3: SA loop----------------


disp('SA loop starts!')

%loop for SA

while finished==0

    tries = tries+1;


  if tries >= maxTries || success >= maxSuccess|| consRej >= maxConsRej

     if T < stopT

        finished = 1;

        total = total + tries;

        break; % break the while loop

     else

        T = coolSched(T);  % decrease T according to cooling schedule

        disp(['Current T is  ',num2str(T)]);

        total = total + tries;
```

```
        tries = 1;
        success = 1;
        consRej=0;
    end
end

%compute the new solution and new energy
newVal_Sel = Generator(minVal_Sel,maxVal_Sel); %new solution
newX=TsetX(:,newVal_Sel); % new spectra by new solution
%[~,~,~,~,betaPLS,~,~,~] = plsregress(newX,TsetY,componentNo);
%y = [ones(size(newX,1),1),newX]*betaPLS;
%newenergy=sqrt(sumsqr(TsetY-y)/size(y,1)); % new RMSEV by new solution
regf=@(XTRAIN,ytrain,XTEST)(XTEST*regress(ytrain,XTRAIN,0.05));
mse = crossval('mse',[ones(size(newX,1),1),newX],TsetY,'kfold',5,'predfun',regf);
newenergy=sqrt(mse);

%break the while loop if new RMSEP is smaller than the RMSEP at which to stop
if (newenergy < stopRMSEP)
    currentVal_Sel = newVal_Sel;
    currentenergy = newenergy;
    break; %break the while loop
end
```

%compare the current energy with the new energy by Boltzman¡¯s probability distribution (Metropolis criterion)

% 1)if new energy is smaller enough (decrement>1e-6) than current energy, accepted probability=1(100%).

```
% 2)if new energy is larger than current energy, accepted probability=a single uniformly distributed random value between 0 and 1.
    if (currentenergy-newenergy > 1e-6)
       currentVal_Sel = newVal_Sel;
       currentenergy = newenergy;
       success = success+1;
       consRej = 0;
    else
       if (rand < exp( (currentenergy-newenergy)/(k*T) ))
          currentVal_Sel = newVal_Sel;
          currentenergy = newenergy;
          success = success+1;
       else
          consRej = consRej+1;
       end
    end

end

%------------------------Step 4: final results----------
%final results
finalVal_Sel = currentVal_Sel; % optimal solution, for NIR it is the optimal index set of SA-selected variables
finalenergy = currentenergy;  % RMSEP of the optimal solution
sTsetX=TsetX(:,currentVal_Sel); %selected variables for training set
sVsetX=VsetX(:,currentVal_Sel); %selected variables for validation set
%draw graph
figure,plot(TsetX(1,:));hold,grid
```

```
plot(finalVal_Sel,TsetX(1,finalVal_Sel),'s');
legend('Original variables','Selected variables');
xlabel('Variable index');

end
```

GENETIC ALGORITHM (GA)

```
function[bestFitValue,best_val_sel,sTsetX,sVsetX]=GA(TsetX,TsetY,VsetX,componentNo,chromosomeNo,maxGenerations,p_crossover,p_mutation)

%%              GA(Genetic Algorithm)


%-----------------------------Input-----------------

% TsetX: spectral data matrix of training set

%       size:n(samples) x m(varibales),

% TsetY: reference data of training set: n(samples) x p(properties)

% VsetX: spectral data matrix of validation set: n(samples) x m(varibales)

% VsetY: reference data of validation set: n(samples) x p(properties)

% chromosomeNo: number of chromosomes. default=100

% maxGenerations: maximum number of generations. default=200

% p_crossover: probability of single-point crossover. default=0.8

% p_mutation: probability of mutation. default=0.05


%-----------------------------Output-----------------

% bestFitValue: best fitness among all generations

% best_val_sel: best index set of selected varialbes

% sTsetX: variables-selected spectra for training set

% sVsetX: variables-selected spectra for validation set


%%   Last modified by Shupeng Hu. on Sept.12th 2019


%-------------------Step 1: configure parameters------
```

```
if nargin==4
    chromosomeNo=100;
    maxGenerations=200;
    p_crossover=0.8;
    p_mutation=0.05;
elseif nargin==5
    maxGenerations=200;
    p_crossover=0.8;
    p_mutation=0.05;
elseif nargin==6
    p_crossover=0.8;
    p_mutation=0.05;
elseif nargin==7
    p_mutation=0.05;
end

%-------------------Step 2: compute the first generation--

geneLength=size(TsetX,2); % number of variables = length of chromosome/ number of genes
allmax=ones(1,maxGenerations);
allmean=ones(1,maxGenerations);
allChro=ones(maxGenerations,geneLength);

%first generation of population.
generation=1;
[population,componentNo]=Code(chromosomeNo,geneLength,componentNo);          % binary code for spectral data
```

```
fitValue= Fitness(population,TsetX,TsetY,componentNo); % fitness of all chromosomes


%record the best fitness with corresponding chromosome,and average fitness for the first generation
[fmax,fmax_index]=max(fitValue);
allmax(generation)=fmax; %best fitness
allChro(generation,:)=population(fmax_index,:);%chromosome which has best fitness
allmean(generation)=mean(fitValue); %average fitness

disp('First generation completed!');

%-------------------Step 3: iteration of evolutions---

newChro_co=ones(chromosomeNo,geneLength);
newChro_mu=ones(chromosomeNo,geneLength);
while generation<maxGenerations
  for j=1:2:chromosomeNo
    %select two chromosomes
    seln=SelChro(fitValue);
    %single-point crossover
    chro_co=Crossover(population,seln,p_crossover,geneLength,componentNo);
    newChro_co(j,:)=chro_co(1,:);
    newChro_co(j+1,:)=chro_co(2,:);
    %mutation

newChro_mu(j,:)=Mutation(newChro_co(j,:),p_mutation,geneLength,componentNo);
```

```
newChro_mu(j+1,:)=Mutation(newChro_co(j+1,:),p_mutation,geneLength,componentNo);
  end

  generation=generation+1; %new generation
  population=newChro_mu;  %new generation of population
  %compute fitness for the new generation
  fitValue=Fitness(population,TsetX,TsetY,componentNo);
  %record the best fitness with corresponding chromosome, and average fitness for the new generation
  [fmax,fmax_index]=max(fitValue);
  allmax(generation)=fmax;
  allChro(generation,:)=population(fmax_index,:);
  allmean(generation)=mean(fitValue);

  disp(['Current generation is  ',num2str(generation)]);

end

%----------------------Step 4: final results---------

[bestFitValue,index]=max(allmax); % best fitness among all generations
bestChro=allChro(index,:); %chromosome which has best fitness among all generations
best_val_sel=find(bestChro(1,:)==1); %best index set of selected varialbes
sTsetX=TsetX(:,best_val_sel);%selected variables for training set
sVsetX=VsetX(:,best_val_sel);%selected variables for validation set
```

```
%draw graph
figure(1),grid,plot(TsetX(1,:));hold on
plot(best_val_sel,TsetX(1,best_val_sel),'s');
legend('Original variables','Selected variables');
xlabel('Variable index');

figure(2);grid
hand1=plot(1:generation,allmax);
set(hand1,'linestyle','-','linewidth',1.8,'marker','*','markersize',6)
hold on;
hand2=plot(1:generation,allmean);
set(hand2,'color','r','linestyle','-','linewidth',1.8,...
'marker','h','markersize',6)
xlabel('Generations');ylabel('Maximum/Average fitness');xlim([1 generation]);
legend('Maximum fitness','average fitness');

end


function [population,componentNo] = Code(chromosomeNo,geneLength,componentNo)
%% Generate binary codes for spectral data

% populaion size: chromosomeNo x geneLength.
% row = chromosome; Column = gene = either 1 or 0; 1 means this variable will
% be selected
```

```
population=ones(chromosomeNo,geneLength);
val_selNo=0; % number of selected variables
for i=1:chromosomeNo
   %the minimum number of selected variables must be at least one larger
   %than the number of PLS principal components;
   while val_selNo<=componentNo

   population(i,:)=round(rand(1,geneLength));
   val_sel=find(population(i,:)==1); %index set of genes whose value = 1
   val_selNo=size(val_sel,2);

   end
   val_selNo=0; %reset
end

end
```

```
function [fitValue] = Fitness(population,TsetX,TsetY,componentNo)
%% Compute the fitness value for all chromosomes by PLSR

% fitness value = 1/RMSEP
% fitness function = sqrt(sumsqr(VsetY-y)/size(y,1));

fitValue=ones(1,size(population,1));
for i=1:size(population,1) % i= chromosome
```

```
    val_sel=find(population(i,:)==1); % index set of selected variables
    newX=TsetX(:,val_sel); % variable-selected spectra for training set
    [~,~,~,~,betaPLS,~,~,~] = plsregress(newX,TsetY,componentNo);
    y = [ones(size(newX,1),1),newX]*betaPLS;
    RMSEV=sqrt(sumsqr(TsetY-y)/size(y,1));
    fitValue(i)=1/RMSEV;

end

end



function [seln] = SelChro(fitValue)
%% Selection two chromosomes for cross-over(reproduce)

% seln: index of two selected chromosomes

%calculate the selection probability of every chromosome; the chromosomes
%with a higher fitness value has a higher probability of reproducing
p_selection=fitValue./sum(fitValue);

seln=[0,0];
while seln(1)==seln(2) %two chromosomes cannot be same
```

```
seln=randsrc(1,2,[1:size(fitValue,2);p_selection]); %select two chromosomes based on the selection probability


end


end



function [chro_co] = Crossover(population,seln,p_crossover,geneLength,componentNo)
%%                  Single-point crossover

%chro_co: two new chromosomes after crossover

% flag=1, crossover; flag=0, no crossover
flag=randsrc(1,1,[1,0;p_crossover,1-p_crossover]);
if flag==1
   % number of selected variables
   val_selNo1=0;
   val_selNo2=0;
   %the minimum number of selected variables must be at least one larger
   %than the number of PLS principal components
   while val_selNo1<=componentNo || val_selNo2<=componentNo

   cp=round(rand*(geneLength-2))+1;  %randomly generate the crossover point between [1,geneLength-1]
   chro_co(1,:)=[population(seln(1),1:cp),population(seln(2),(cp+1):geneLength)];
   chro_co(2,:)=[population(seln(2),1:cp),population(seln(1),(cp+1):geneLength)];
```

```
  val_sel1=find(chro_co(1,:)==1); %index set of genes whose value = 1
  val_selNo1=size(val_sel1,2);
  val_sel2=find(chro_co(2,:)==1); %index set of genes whose value = 1
  val_selNo2=size(val_sel2,2);

  end
else
  chro_co(1,:)=population(seln(1),:);
  chro_co(2,:)=population(seln(2),:);
end

end
```

```
function [newChro_mu] = Mutation( newChro_co,p_mutation,geneLength,componentNo)
%% Mutation

% newChro_mu: chromosome after mutation

newChro_mu=newChro_co;
% flag=1, mutation; flag=0, no mutation
flag=randsrc(1,1,[1,0;p_mutation,1-p_mutation]);
if flag==1
  val_selNo=0; % number of selected variables
  %the minimum number of selected variables must be at least one larger
```

```
   %than the number of PLS principal components
   while val_selNo<=componentNo


   mp=round(rand*(geneLength-1))+1;  %randomly generate the mutation point between
[1,geneLength]
   newChro_mu(mp)=abs(newChro_co(mp)-1);
   val_sel=find(newChro_mu(1,:)==1); %index set of genes whose value = 1
   val_selNo=size(val_sel,2);


   end
end


end
```

INTERVAL PARTIAL LEAST SQUARES (IPLS)

```
function
[bestmse,bestinterval,bestintervalNo,bestmseSOBI,bestintervalSOBI]=iPLS(TsetX,Tset
Y,componentNo,intervalsNo)

%%          Interval Partial Least Squares (iPLS)


% number of PC should be the same for all local PLS models


%----------------------Input----------------------------

% TsetX: spectral data matrix of training set

%       size:n(samples) x m(varibales),

% TsetY: reference data of training set: n(samples) x p(properties)

% VsetX: spectral data matrix of validation set: n(samples) x m(varibales)

% componentNo: nubmer of PLS components

% intervalsNo: (optional) the desired number of intervals. default = auto-divide
intervals


%----------------------Output----------------------------

% globalmse: the mse of global PLS model

% bestmse£ºbest mse from cross-validation

% bestinterval: best interval which has best mse from cross-validation

% bestintervalNo: best number of intervals

% bestmseSOBI: best mse after SOBI

% bestintervalSOBI: best interval which has best mse after SOBI


%%      Last modified by Shupeng Hu. Sept. 14th 2019
```

```
%----------------------Step 1:configure inputs------------

[~,n]=size(TsetX);
min_intervals=3; % at least 3 intervals
max_intervals=floor(n/(componentNo+1)); % maximum number of intervals

if nargin==3
  intervalsNo=-1;
end

%-------------------Step 2:seek the best spectral regions--

bestmse=10;
if intervalsNo==-1
   for i=min_intervals:max_intervals
   disp(['Currently there are ',num2str(i),' intervals']);

   width=floor(n/i); % identical width for each interval
   intervals=IntervalPartition(width,i);
   allmse=iPLSMSE(intervals,componentNo,TsetX,TsetY);
   [localmse,bestmse_index]=min(allmse);
   if localmse<bestmse
   bestmse=localmse;
   bestinterval=intervals(bestmse_index,:);
   bestintervalNo=i;
   end
```

```
    end
else
    width=floor(n/intervalsNo); % identical width for each interval
    intervals=IntervalPartition(width,intervalsNo);
    allmse=iPLSMSE(intervals,componentNo,TsetX,TsetY);
    [localmse,bestmse_index]=min(allmse);
    if localmse<bestmse
    bestmse=localmse;
    bestinterval=intervals(bestmse_index,:);
    end
    bestintervalNo=intervalsNo;
end

%----------------------Step 3: simple optimization---------

[bestmseSOBI,bestintervalSOBI]                                                    =
SOBI(bestinterval,bestmse,componentNo,TsetX,TsetY);

disp('Simple optimization for iPLS completed');

%draw graph
figure,plot(TsetX(1,:));hold,grid
plot(bestinterval,TsetX(1,bestinterval),'s')
legend('Original variables','Selected variables')
xlabel('Variable index')

end
```

```
function [intervals] = IntervalPartition(width,intervalsNo)
%compute the intervals

intervals=ones(intervalsNo,width);
for j=1:intervalsNo
    intervals(j,:)=(width*(j-1)+1):1:(width*j);
end

end
```

```
function [allmse] = iPLSMSE(intervals,componentNo,TsetX,TsetY)
%calculate the PLS-mse for all intervals

[m,~]=size(intervals);
allmse=ones(1,m);
for i=1:m
    newX=TsetX(:,intervals(i,:));
    [~,~,~,~,betaPLS,~,~,~] = plsregress(newX,TsetY,componentNo);
    y = [ones(size(newX,1),1),newX]*betaPLS;
    mse=sumsqr(TsetY-y)/size(y,1);
    allmse(1,i)=mse;
end
```

```
end



function [bestmse,bestinterval] = SOBI(bestinterval,bestmse,componentNo,TsetX,TsetY)

%%      Simple Optimization of the Best Interval


% when the best interval is determined by iPLS, this interval width

% is changed one variable at a time on both sides and evaluated by

% same criteria provided by the application of PLS regression to the interval.



%expand interval on right side

while size(bestinterval)<size(TsetX,2)

[~,n]=size(bestinterval);

exinterval=bestinterval(1):1:(bestinterval(n)+1); %one variable at a time

newX=TsetX(:,exinterval);

[~,~,~,~,betaPLS,~,~,~] = plsregress(newX,TsetY,componentNo);

y = [ones(size(newX,1),1),newX]*betaPLS;

mse=sumsqr(TsetY-y)/size(y,1);

if mse<bestmse

bestmse=mse;

bestinterval=exinterval;

else

   break;

end
```

```
end

%expand interval on left side
while bestinterval(1)>1
[~,n]=size(bestinterval);
exinterval=(bestinterval(1)-1):1:bestinterval(n); %one variable at a time
newX=TsetX(:,exinterval);
[~,~,~,~,betaPLS,~,~,~] = plsregress(newX,TsetY,componentNo);
y = [ones(size(newX,1),1),newX]*betaPLS;
mse=sumsqr(TsetY-y)/size(y,1);
if mse<bestmse
bestmse=mse;
bestinterval=exinterval;
else
   break;
end
end



end
```

BACKWARD INTERVAL PARTIAL LEAST SQUARES (BIPLS)

```
function [finalmse,finalIntervals,bestintervalNo,sTsetX,sVsetX] = BiPLS(TsetX,TsetY,VsetX,componentNo,intervalsNo)

%%  Backward Interval Partial Least Squares (BiPLS)


% number of PC should be the same for all local PLS models


%----------------------Input-----------------------------% TsetX: spectral data matrix of training set

%  size:n(samples) x m(varibales),

% TsetY: reference data of training set: n(samples) x p(properties)

% VsetX: spectral data matrix of validation set: n(samples) x m(varibales)

% componentNo: nubmer of PLS components

% intervalsNo: (optional) the desired number of intervals. default = auto-divide intervals


%----------------------Output----------------------------

% finalmse: best mse of cross-validation

% finalIntervals: index set of selected variables whose has best mse of cross-validation

% bestintervalNo: best number of intervals

% sTsetX: variables-selected spectra for training set

% sVsetX: variables-selected spectra for validation set


%%  Last modified by Shupeng Hu. Sept. 15th 2019


%----------------------Step 1:configure inputs------------
```

```
[~,n]=size(TsetX);
min_intervals=3; % at least 3 intervals
%max_intervals=floor(n/(componentNo+1)); % maximum number of intervals
max_intervals=94;

if nargin==4
   intervalsNo=-1;
end

%-------------------Step 2:seek the best spectral regions--

finalmse=10;
leaveIndex=0;
counter=0;
flag=0;

if intervalsNo==-1
for i=min_intervals:max_intervals
disp(['Currently there are ',num2str(i),' intervals']);

width=floor(n/i); % identical width for each interval
intervals=IntervalPartition(width,i); %partition intervals
newX=TsetX(:,intervals);
[~,~,~,~,betaPLS,~,~,~] = plsregress(newX,TsetY,componentNo);
y = [ones(size(newX,1),1),newX]*betaPLS;
localmse=sumsqr(TsetY-y)/size(y,1);
```

```
%backward stepwise iterations
while size(intervals,1)>1
for j=1:size(intervals,1)
    interval=intervals;
    interval(j,:)=[];
    newX=TsetX(:,interval);
    [~,~,~,~,betaPLS,~,~,~] = plsregress(newX,TsetY,componentNo);
    y = [ones(size(newX,1),1),newX]*betaPLS;
    mse=sumsqr(TsetY-y)/size(y,1);
    if mse<localmse
        localmse=mse;
        leaveIndex=j;
    elseif j<size(intervals,1)
        counter=counter+1;
    elseif j==size(intervals,1) && counter==(size(intervals,1)-1)
        flag=1;
    end
end
counter=0;
if flag==1  %this interation completed
    flag=0;
    break;
end

intervals(leaveIndex,:)=[]; %% the interval should be kicked out at this interation
```

```
end

if localmse<finalmse %compare the mse between iterations
finalmse=localmse;
fi=intervals;
bestintervalNo=i;
end

end

else      % given a fixed desired number of intervals

width=floor(n/intervalsNo); % identical width for each interval
intervals=IntervalPartition(width,intervalsNo); %partition intervals
newX=TsetX(:,intervals);
[~,~,~,~,betaPLS,~,~,~] = plsregress(newX,TsetY,componentNo);
y = [ones(size(newX,1),1),newX]*betaPLS;
localmse=sumsqr(TsetY-y)/size(y,1);

%backward stepwise iterations
while size(intervals,1)>1
for j=1:size(intervals,1)
   interval=intervals;
   interval(j,:)=[];
   newX=TsetX(:,interval);
   [~,~,~,~,betaPLS,~,~,~] = plsregress(newX,TsetY,componentNo);
   y = [ones(size(newX,1),1),newX]*betaPLS;
```

```
    mse=sumsqr(TsetY-y)/size(y,1);
    if mse<localmse
        localmse=mse;
        leaveIndex=j;
    elseif j<size(intervals,1)
        counter=counter+1;
    elseif j==size(intervals,1) && counter==(size(intervals,1)-1)
        flag=1;
    end
end
  counter=0;
if flag==1  %no more intervals should be kicked out at this iteration
    break;
end

intervals(leaveIndex,:)=[];

end
fi=intervals;
finalmse=localmse;
bestintervalNo=intervalsNo;
end

%----------------------Step 3: final results---------------

%extract all variable indexes into one row
finalIntervals=[];
```

```
for k=1:size(fi,1)
   finalIntervals=[finalIntervals,fi(k,:)];
end

sTsetX=TsetX(:,finalIntervals); %selected variables for training set
sVsetX=VsetX(:,finalIntervals); %selected variables for validation set

%draw graph
figure,plot(TsetX(1,:));hold,grid
plot(finalIntervals,TsetX(1,finalIntervals),'s')
legend('Original variables','Selected variables')
xlabel('Variable index')


end
```

FORWARD INTERVAL PARTIAL LEAST SQUARES (FIPLS)

```
function [finalmse,finalIntervals,bestintervalNo,sTsetX,sVsetX] = FiPLS(TsetX,TsetY,VsetX,componentNo,intervalsNo)

%%  Forward Interval Partial Least Squares (FiPLS)


% number of PC should be the same for all local PLS models


%----------------------Input----------------------------

% TsetX: spectral data matrix of training set

%       size:n(samples) x m(varibales),

% TsetY: reference data of training set: n(samples) x p(properties)

% VsetX: spectral data matrix of validation set: n(samples) x m(varibales)

% componentNo: nubmer of PLS components

% intervalsNo: (optional) the desired number of intervals. default = auto-divide intervals


%----------------------Output----------------------------

% finalmse: best mse of cross-validation

% finalIntervals: index set of selected variables whose has best mse of cross-validation

% bestintervalNo: best number of intervals

% sTsetX: variables-selected spectra for training set

% sVsetX: variables-selected spectra for validation set


%%  Last modified by Shupeng Hu. Sept. 15th 2019
```

```
%----------------------Step 1:configure inputs------------

[~,n]=size(TsetX);
min_intervals=5; % at least 3 intervals
max_intervals=floor(n/(componentNo+1)); % maximum number of intervals

if nargin==4
  intervalsNo=-1;
end

%-------------------Step 2:seek the best spectral regions--

finalmse=10;
localmse=10;
flag=0;
counter=0;
if intervalsNo==-1
for i=min_intervals:max_intervals
   disp(['Currently there are ',num2str(i),' intervals']);

      width=floor(n/i); % identical width for each interval
      intervals=IntervalPartition(width,i); %partition intervals
      currentIntervals=[];

%forward stepwise iterations
while size(currentIntervals,2)<width*i
for j=1:size(intervals,1)
```

```
        interval=[currentIntervals,intervals(j,:)];
        newX=TsetX(:,interval);
        [~,~,~,~,betaPLS,~,~,~] = plsregress(newX,TsetY,componentNo);
        y = [ones(size(newX,1),1),newX]*betaPLS;
        mse=sumsqr(TsetY-y)/size(y,1);
       if mse<localmse
         localmse=mse;
         addIndex=j;
       elseif j<size(intervals,1)
          counter=counter+1;
       elseif j==size(intervals,1) && counter==(size(intervals,1)-1)
          flag=1;
       end
end
counter=0;
if flag==1  %no more intervals should be selected at this interation
    flag=0;
    break;
end
currentIntervals=[currentIntervals,intervals(addIndex,:)];
intervals(addIndex,:)=[];
end

    if localmse<finalmse  % compare the mse between iterations
finalIntervals=currentIntervals;
finalmse=localmse;
bestintervalNo=i;
```

```
    end

localmse=10; % reset local mse for next iteration
end

else % given a fixed desired number of intervals

width=floor(n/intervalsNo); % identical width for each interval
intervals=IntervalPartition(width,intervalsNo); %partition intervals
currentIntervals=[];

%forward stepwise iterations
while size(currentIntervals,2)<width*intervalsNo
for j=1:size(intervals,1)
        interval=[currentIntervals,intervals(j,:)];
        newX=TsetX(:,interval);
        [~,~,~,~,betaPLS,~,~,~] = plsregress(newX,TsetY,componentNo);
        y = [ones(size(newX,1),1),newX]*betaPLS;
        mse=sumsqr(TsetY-y)/size(y,1);
       if mse<localmse
          localmse=mse;
          addIndex=j;
       elseif j<size(intervals,1)
           counter=counter+1;
       elseif j==size(intervals,1) && counter==(size(intervals,1)-1)
           flag=1;
       end
```

```
end
counter=0;
if flag==1  %no more intervals should be selected at this interation
   break;
end

currentIntervals=[currentIntervals,intervals(addIndex,:)];
intervals(addIndex,:)=[];
end
finalIntervals=currentIntervals;
finalmse=localmse;
bestintervalNo=intervalsNo;
end

%----------------------Step 3: final results---------------

sTsetX=TsetX(:,finalIntervals); %selected variables for training set
sVsetX=VsetX(:,finalIntervals); %selected variables for validation set

%draw graph
figure,plot(TsetX(1,:));hold,grid
plot(finalIntervals,TsetX(1,finalIntervals),'s')
legend('Original variables','Selected variables')
xlabel('Variable index')

end
```

MULTIPLE LINEAR REGRESSION (MLR)

```
function [stats,outliers,y,RMSEP,R2,RPD] = MLR(TsetX,TsetY,VsetX,VsetY,alpha)
%%              MLR(Multiple Linear Regression)

%-----------------------------Input----------------------
% TsetX: spectral data matrix of training set
%       size:n(samples) x m(varibales),
%       n>m, otherwise, this function has problem
% TsetY: reference data of training set: n(samples) x p(properties),p=1
% VsetX: spectral data matrix of validation set: n(samples) x m(varibales)
% VsetY: reference data of validation set: n(samples) x p(properties),p=1
% alpha: confidence level=(1-alpha)%

%-----------------------------Output----------------------
% stats: statistics of calibration set
%       stats(1)= R2
%       stats(2)= F-statistic,
%       stats(3)= p-value
%       stats(4)= error variacne
% outliers: possible outliers in calibration set
% y: predicted response for validation set
% RMSEP: root mean squared of standard error of prediction
% R2: coefficient of determination
% RPD: ratio of standard error of performance to deviation

%%   Last modified by Shupeng Hu. March 18th 2018
```

```
%b: regression coefficients

%rint: residuals interval

%stats: see it in output

[b,~,~,rint,stats] = regress(TsetY,[ones(size(TsetX,1),1),TsetX],alpha);


%------------------------------Validation------------------%Diagnose outliers by finding the residual intervals that do not contain 0

outliers = find((rint(:,1)<0 & rint(:,2)>0)==false);


y=[ones(size(VsetX,1),1),VsetX]*b; %predicted response


%criteria

RMSEP=sqrt(sumsqr(VsetY-y)/size(y,1));

R2=1-(sumsqr(VsetY-y)/sumsqr(VsetY-mean(VsetY)));

RPD=sqrt(sumsqr(VsetY-mean(VsetY))/(size(y,1)-1))/RMSEP;


end
```

PRINCIPAL COMPONENTS REGRESSION (PCR)

```
function
[eigenPercent,MSE,outliers,stats,y,RMSEP,R2,RPD]=PCR(TsetX,TsetY,VsetX,VsetY,
alpha,componentsNo)

%%              PCR(Principal Components Regression)


%----------------------Input-----------------------------% TsetX: spectral data matrix of
training set

%       size:n(samples) x m(varibales),

% TsetY: reference data of training set: n(samples) x p(properties)

% VsetX: spectral data matrix of validation set: n(samples) x m(varibales)

% VsetY: reference data of validation set: n(samples) x p(properties)

% alpha: confidence level=(1-alpha)%

% componentsNo: the number of seleced principal components


%-----------------------------Output----------------------

% eigenPercent: the percentage of the total variance explained by each principal
component

% MSE: mean squared of standard error in leave-one-out cross validation for
0:ncomponents

% outliers: possible outliers in calibration set

% stats: statistics of calibration set

%       stats(1)= R2

%       stats(2)= F-statistic,

%       stats(3)= p-value

%       stats(4)= error variacne

% y: predicted response for validation set

% RMSEP: root mean squared of standard error of prediction
```

```
% R2: coefficient of determination
% RPD: ratio of standard error of performance to deviation

%%      Last modified by Shupeng Hu. July 10th 2019


%----------------------------use the PCA function----------%loading: principal component coefficients
%score: PCA score matrix: n*k, k is the number of principal components
%eigenPercent: see it in output
[loading,score,~,~,eigenPercent,~] = pca(TsetX);

%----------------------principal components selection------
%calculate the MSE in cross validation
MSE = sum(crossval(@pcrsse,TsetX,TsetY,'kfold',5),1)/size(TsetX,1);
selectedScore=score(:,1:componentsNo); %size: omponentsNo
selectedLoading=loading(:,1:componentsNo); %size: m*componentsNo

%----------------------------Regression----------------
%b: regression coefficients
%rint: residuals interval
%stats: see it in output
[b,~,~,rint,stats] = regress(TsetY,[ones(size(selectedScore,1),1),selectedScore],alpha);

%----------------------------Validation----------------
%Diagnose outliers by finding the residual intervals that do not contain 0
outliers = find((rint(:,1)<0 & rint(:,2)>0)==false);
```

```
%get the regression coefficient vector corresponding to the selected components
B=selectedLoading*b(2:size(b,1),1);

y=[ones(size(VsetX,1),1),VsetX]*[mean(TsetY)-mean(TsetX)*B;B];          %predicted response

%criteria
RMSEP=sqrt(sumsqr(VsetY-y)/size(y,1));
R2=1-(sumsqr(VsetY-y)/sumsqr(VsetY-mean(VsetY)));
RPD=sqrt(sumsqr(VsetY-mean(VsetY))/(size(y,1)-1))/RMSEP;

end
```

PARTIAL LEAST SQUARES REGRESSION (PLSR)

```
function [eigenPercent,MSE,stats,y,RMSEP,R2,RPD] = PLSR(TsetX,TsetY,VsetX,VsetY,componentNo)

%%  PLSR(Partial Least Squares Regression)


%----------------------Input----------------------------

% TsetX: spectral data matrix of training set

%  size:n(samples) x m(varibales),

% TsetY: reference data of training set: n(samples) x p(properties)

% VsetX: spectral data matrix of validation set: n(samples) x m(varibales)

% VsetY: reference data of validation set: n(samples) x p(properties

 total variance explained by each principal component)

% componentNo: the number of selected PLS components


%-----------------------------Output----------------------

% eigenPercent:the percentage of the

%  eigenPercent(1,:)=percentage of variance explained in X by each PLS component

%  eigenPercent(2,:)=percentage of variance explained in Y by each PLS component

% MSE: mean-squared errors for PLS models with 0:ncomp components

%  MSE(1,:)=mean-squared errors for the predictor variables in X

%  MSE(2,:)=mean-squared errors for the response variable(s) in Y

%stats:contains W ¡ª A p-by-ncomp matrix of PLS weights so that XS = X0*W.

%  T2 ¡ª The T2 statistic for each point in XS.

%  Xresiduals ¡ª The predictor residuals, that is, X0-XS*XL'.

%  Yresiduals ¡ª The response residuals, that is, Y0-XS*YL'.

% y: predicted response for the validation set
```

```
% RMSEP: root mean squared of the standard error of prediction
% R2: coefficient of determination
% RPD: the ratio of the standard error of performance to deviation

%%      Last modified by Shupeng Hu. July 11th 2019

%---------------------------use the PLS function----------
%XLoading: predictor loading
%YLoading: response loading
%XSocre: predictor scores
%YScore: response scores
%betaPLS: PLS regression coefficients
%eigenPercent: see in the output
%MSE: see in the output
%stats: see in the output
%Leave-one-out cross validation= full-fold cross validation
%[~,~,~,~,betaPLS,eigenPercent,MSE,stats]                                    =
plsregress(TsetX,TsetY,componentNo,'cv',size(TsetX,1));
[~,~,~,~,betaPLS,eigenPercent,MSE,stats]                                    =
plsregress(TsetX,TsetY,componentNo,'cv',5);


%----------------------------------validaiton-------------
y = [ones(size(VsetX,1),1),VsetX]*betaPLS;

%criteria
RMSEP=sqrt(sumsqr(VsetY-y)/size(y,1));
R2=1-(sumsqr(VsetY-y)/sumsqr(VsetY-mean(VsetY)));
```

```
RPD=sqrt(sumsqr(VsetY-mean(VsetY))/(size(y,1)-1))/RMSEP;


end
```