CRANFIELD UNIVERSITY

Eleni Anthippi Chatzimichali

Development and Optimisation of Chemometric Techniques for the Evaluation of Meat Freshness

Cranfield Health

PhD Thesis

Supervisor: Professor Conrad Bessant

2013

CRANFIELD UNIVERSITY

CRANFIELD HEALTH

PHD THESIS

2013

ELENI ANTHIPPI CHATZIMICHALI

Development and Optimisation of Chemometric Techniques for the Evaluation of Meat Freshness

Supervisor: Professor Conrad Bessant

ABSTRACT

Muscle foods such as meat, fish and poultry are an integral part of human diet. Over time, such food succumbs to spoilage, resulting from various intrinsic and extrinsic factors, the most significant of which is microbial activity. Spoilage changes the organoleptic properties of meat, rendering it unacceptable to the consumer, and may ultimate result in the food becoming toxic. Spoilage is therefore of major commercial and public health interest.

This thesis describes the development and application of a novel suite of software tools designed to support novel instrumental approaches for the accurate, rapid and inexpensive evaluation of meat freshness. A pipeline was built for the analysis of highly heterogeneous data obtained by a diverse range of high-throughput techniques across four three-class case studies. As a first step, PCA was applied for dimensionality reduction, feature extraction and exploratory analysis. PLS-DA and SVMs were employed as classifiers, and classification ensembles implemented as a means of improving classification accuracy. Rigorous validation and evaluation methods based on bootstrapping and permutation testing were applied to ensure that the performance metrics are representative of real-world application, and to ascertain the statistical significance of the results. This was made possible by the development of an advanced optimisation approach, which reduced the computational demands of SVM tuning by up to ~ 90× times. The functionality of the pipeline was further enhanced by exploiting GPA and CPCA as data fusion techniques, to evaluate whether better classification accuracy is achieved when integrated as opposed to standalone datasets are used.

SVM ensembles proved to be the most powerful and accurate classification method since they produced consistently higher prediction rates ($\%CC$) than PLS-DA. Among the analytical techniques, HPLC was established as the most diagnostic method for the assessment of meat freshness, with a $\%CC$ of 80%. Among the two data fusion techniques, CPCA outperformed GPA. However, CPCA only exceeded standalone HPLC in a minority of cases, presenting an overall $\%CC$ of 82%.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my supervisor Professor Conrad Bessant for his overwhelming contribution and support during this project. Your ever-present advice and guidance are worthy of my lasting gratitude. Thank you for the time and dedication you invest in me.

To my friends and colleagues in Cranfield University I declare my deep indebtedness for their wholehearted support. Thank you for providing me with the necessary impetus to better myself.

My deepest gratitude goes above all to my beloved parents. This project would have not been feasible without your ever-lasting support, trust and encouragement. A huge thank you for always being there for me.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

TABLE OF EQUATIONS

ABBREVIATIONS

%CC	Percentages of Correctly Classified Samples
AJAX	Asynchronous JavaScript and XML
ATR	Attenuated Total Reflectance
CPCA	Consensus Principal Component Analysis
CSS	Cascading Style Sheet
CSV	Comma Separated File
DNA	Deoxyribonucleic Acid
DOM	Document Object Model
E-NOSE	Electronic Nose
FTIR	Fourier Transform infrared (spectroscopy)
GPA	Generalized Procrustes Analysis
HCA	Hierarchical Cluster Analysis
HPLC	High Performance Liquid Chromatography
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
LOOCV	Leave-One-Out Cross-Validation
LV	Latent Variable
MAP	Modified Atmosphere Packaging
NIPALS	Non-linear Iterative Partial Least Squares
OPA	Ordinary Procrustes Analysis
PC	Principal Component
PCA	Principal Component Analysis
PCR	Polymerase Chain Reaction
Perl	Practical Extraction and Reporting Language
PLS	Partial Least Squares
PLS-DA	Partial Least Squares Discriminant Analysis
PNG	Portable Network Graphics
RBF	Radial Basis Function
RIA	Rich Internet Application

RMSE	Root Mean Square Error
RMSECV	Root Mean Square Error of Cross-Validation
SSE	Sum of Squared Errors
SVD	Singular Value Decomposition
SVMs	Support Vector Machines
SYMBIOSIS-EU	Scientific sYnerganisM of nano-Bio-Info-cOngi Science for an Integrated system to monitor meat quality and Safety during production, storage, and distribution in the European Union
XHTML	Extensible HyperText Markup Language
XML	Extensible Markup Language
WWW	World Wide Web

1 Introduction and Literature Review

1.1 Introduction

1.1.1 Overview of Systems Biology

Breakthrough biological discoveries over the past decades, such as the revolutionary discovery of the double helix structure of DNA in 1953 by Watson and Crick, catalysed the blossoming of molecular biology (Watson and Crick, 1953). Acquiring information about the structure and properties of DNA and proteins led to outstanding progress in the years that followed as presented in Figure 1-1. Molecular biology has chiefly focused on identifying and investigating individual biological molecules by studying their properties and functions either as isolated entities or as small sets of components in very simple model systems. However, the reductionist approach adapted by molecular biology was not sufficient to interpret the intrinsic complexity of biological systems.

The Human Genome Project has profoundly altered the practice and view of contemporary biology (Hood, 2003; Venter *et al.*, 2001). In the post-genome era, the massive amount of biological data acquired by the advance of high-throughput technologies led to the rapid shift of interest towards systems biology. The marked increase in the amount of genomic, proteomic and metabolomic data due to the constant improvements in high-throughput tools, has granted the scientific community the opportunity to study complex biological systems as an integrated whole. Thus, systems biology emerged as a necessity helping us understand these complex system dynamics, as these are the key to understanding life. Systems analysis has historically been applied in a plethora of scientific fields such as economics, physics, psychology and most recently biology, covering a multitude of different areas such as developmental biology, ecology and immunology (Westerhoff *et al.*, 2004).

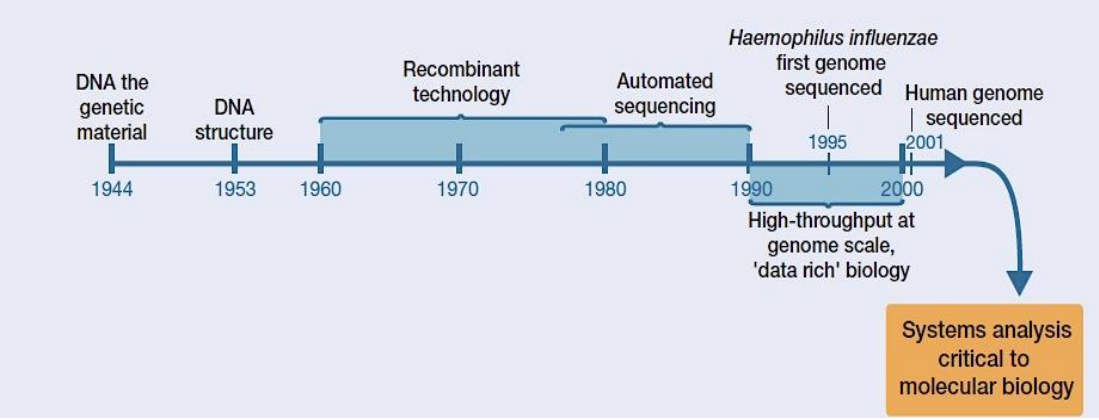


Figure 1-1 Evolution from molecular biology to systems biology

The line of inquiry represents the way mainstream molecular biology, under the pressure for system-level study, started investigating groups of molecules rather than single macromolecules, while simultaneously investigated their interactions. The figure has been adapted from Westhoff *et al.* (2004).

A system-level approach aims to generate novel technologies and effective tools based on the collection, integration, analysis, graphical visualisation, and ultimately modelling of biological information (Ideker *et al.*, 2001; Hood, 2003). Mathematical modelling is the backbone of contemporary systems biology. To this day the term "mathematical" is usually hidden behind a "computational approach" (Mesarovic *et al.*, 2005). A model is an effort to represent all the integrated highly heterogeneous information that derives from multiple experimental sources in an abstract manner. Current advances in systems biology and computing science are prompting scientists to use sophisticated mathematical models and powerful *in silico* simulations. Despite the continuous acquisition of new information as well as the overwhelming progress of computational and experimental methods, the high complexity of biological systems will always constitute an obstacle for the construction of a general, integrated and functionally meaningful model based on complete understanding (Butcher *et al.*, 2004; Filkenstein *et al.*, 2004).

1.1.2 The 'omics' disciplines

Ever since the first automated DNA sequencing machine (See Figure 1-1), there has been a tremendous increase in the development of high-throughput platforms leading to the accumulation of vast amounts of highly heterogeneous biological data. These large-scale sets of data and biological information have inspired several novel fundamental concepts – namely, the 'omics' disciplines. These disciplines propel systems-level understanding, having as a chief aim the simultaneous quantification and identification of the building blocks of a biological system such as genes, proteins or metabolites, as well as the investigation of the interactions among them such as protein-protein.

Three of the most important 'omics' sources are genomics, proteomics and metabolomics. The term genomics was established to denote the analysis of the entire genome – the complete genetic sequence – of an organism. In all cellular organisms, the genome is composed of deoxyribonucleic acid (DNA). Proteomics can be defined as the scientific field that focuses on the study of the proteome. The proteome is the entire collection of proteins that are expressed by a particular genome. However, even though the genome of a cellular organism is static – it alters only when mutations occur – the proteome changes constantly as a result of internal and external factors. Metabolomics is likewise defined as the comprehensive profiling of the metabolome. The metabolome consists of all the biochemicals and metabolites produced by a cellular organism. Metabolites are substances either required for or produced by biochemical reactions of metabolism that occur within the cells of an organism. Metabolomics allows scientists to study and compare the relationships between an organism's genotype and phenotype, as well as the relationships between the genotype and the environment (Hassani *et al.*, 2010).

This project will be focusing on the field of metabolomics, and in particular the study of metabolites responsible for meat spoilage. The analytical techniques that will be used in this project are briefly presented as follows.

1.1.2.1 Fourier Transform Infrared (FTIR) Spectroscopy

Fourier Transform Infrared (FTIR) spectroscopy is a very rapid (running over a few seconds) non-destructive analytical technique used for high-throughput biochemical fingerprinting (Ellis and Goodacre, 2001). In FTIR, a particular bond absorbs light or electromagnetic radiation by an infrared beam at a specific wavelength (Ellis *et al.*, 2004). As a result of FTIR analysis, an infrared absorbance spectrum can be extracted, which may be used as a biochemical or metabolic "fingerprint" of the samples. However, FTIR spectra tend to be quite complex featuring hundreds or thousands of variables, thus necessitating the use of statistical methods for their analysis. FTIR in combination with multivariate statistical techniques has proven to be a very fast and accurate method for food-based analyses and bacterial detection (Nicolaou *et al.*, 2011).

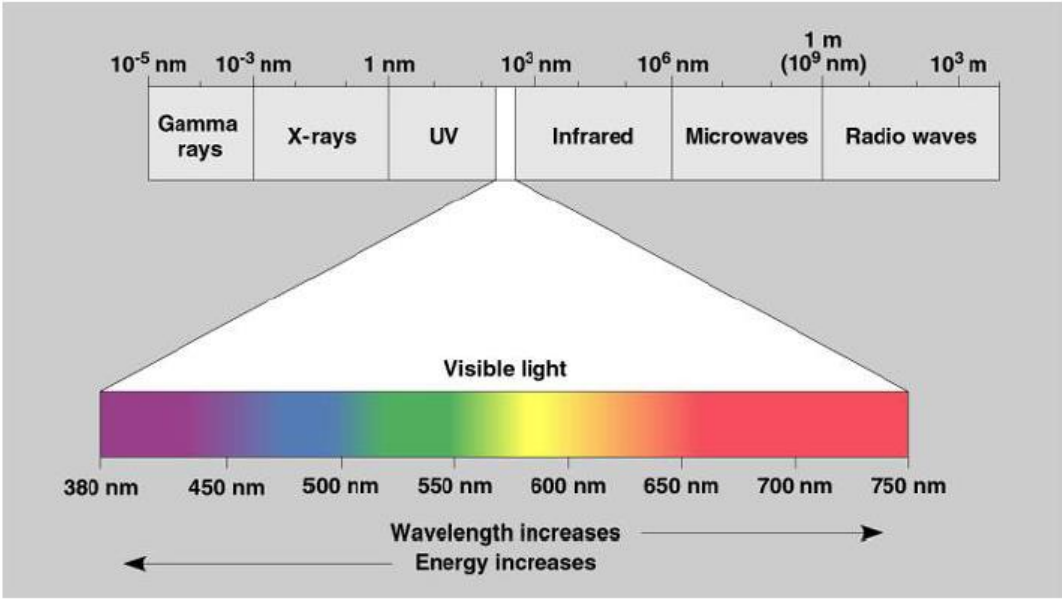


Figure 1-2 Electromagnetic spectrum

The figure has been extracted from Sattlecker (2011).

1.1.2.2 High Throughput Liquid Chromatography (HPLC)

High-performance liquid chromatography (HPLC) is a chromatographic technique used to separate a mixture of chemical compounds. It is mainly used in biochemistry and analytical chemistry to identify, quantify as well as purify the individual components in a mixture. The HPLC instrument consists of a solvent reservoir, transfer line with frit, high-pressure pump, sample injection device, column, detector, and data acquisition, usually together with data evaluation (Meyer, 2013)

1.1.2.3 Electronic Nose (e-nose)

An electronic nose (e-nose) is an instrument applied for the rapid non-destructive detection and analysis of microbial volatile compounds. The electronic noses attempt to mimic the human organoleptic olfactory interpretation (Persuad and Dodd, 1982). The instrument consists of an array of chemical gas sensors, which are capable of detecting and recognising simplex or complex odours.

Electronic noses have shown great promise in the field of food analysis as a means of evaluating freshness and investigating shelf life. E-noses have become increasingly popular due to the fact that they resemble human sensory evaluation, but also since they are rapid, low-cost non-destructive techniques. Even so, the repeatability with electronic noses has been questioned since they present instabilities due to severe instrumental drift (Ellis and Goodacre, 2001).

1.1.2.4 Raman Spectroscopy

Raman spectroscopy is also a non-destructive method that can be considered to be complementary to FTIR spectroscopy. Both Raman and FTIR are powerful metabolic fingerprinting methods as they reflect accurately the phenotype of a sample, including changes to its metabolism (Nicolaou *et al.*, 2011). The major advantage of Raman spectroscopy over FTIR is the fact that the contribution from water is very small and thus can be used directly on food without recourse to ATR (Argyri *et al.*, 2013).

1.1.3 Microbial Spoilage in Meat

Systems biology has gained in importance in food science and the food industry due to an increasing focus on food for better health and the demand for products of consistently high quality (Ellis *et al.*, 2002; Hassani *et al.*, 2010). Out of all foods that are a vital part of human diet, meat has been described as the most perishable of all. Muscle foods such as meat or poultry become unacceptable to the consumer when organoleptic changes occur due to spoilage (Ellis *et al.*, 2002). Spoilage can be defined as "any change in a food product that renders it unacceptable to the consumer from a sensory point of view" (Gram *et al*., 2002; Ercolini *et al.*, 2006)

Meat spoilage may be the result of a plethora of intrinsic and extrinsic factors, the most significant of which is microbial activity (Gram *et al*., 2002). Even though changes of food substances during storage may be the result of endogenous enzymatic processes within muscle tissue post-mortem, it is generally accepted that detectable organoleptic spoilage is a result of decomposition and the formation of metabolites caused by the growth of microorganisms (Ellis and Goodacre, 2001; Ellis *et al.*, 2002). These organoleptic characteristics usually include the development of off-odours and off-flavours, the formation of slime in addition to any changes in the appearance such as discoloration; thus, consumers consider the meat as being undesirable. Due to its moist highly nutritious surface, meat stored at between -1 and 25°C favours the growth of a wide range of spoilage bacteria. Under aerobic conditions, spoilage organisms that belong primarily to the genus *Pseudomonas* attach more rapidly to meat surfaces than other spoilage bacteria (Ellis *et al.,* 2002). The organoleptic changes may vary depending on the microbial association contaminating the meat and the conditions under which it is stored. The development of organoleptic spoilage is related to microbial consumption of meat nutrients such as sugars and free amino acids, and the release of undesired volatile metabolites (Ercolini *et al.*, 2006).

Fears over microbiological food safety issues have led to the requirement for a rapid and accurate detection system for microbiologically spoiled or contaminated meat. To date, various methods have been proposed to measure and detect bacterial spoilage in meat such as enumeration methods based on microscopy, ATP bioluminescence and the measurement of electrical phenomena as well as detection methods based on immunological procedures (Ellis *et al*., 2002). However, these techniques are time-consuming, labour-intensive and generate retrospective information; thus, they cannot be used for on- or at-line monitoring (Ellis *et al*., 2002; Argyri *et al*., 2011). Polymerase chain reaction (PCR) techniques may also be investigated; however, the main limitation of these techniques is the high equipment cost, the demand for highly trained staff as well as the risk of cross-contamination (Nicolaou *et al*., 2011).

The research for reliable meat-quality sensors has led to the development of a plethora of rapid, non-invasive, and relatively inexpensive methods based on analytical instrumentation techniques such as Fourier transform infrared (FTIR) spectroscopy and electronic nose technology (Argyri *et al.*, 2010; Panagou *et al.*, 2010). The present study aims to develop an automated, reproducible and quantitative approach that defines the spoilage state of a product objectively. The application of analytical techniques such as the ones presented in Section 1.1.2 in conjunction with multivariate statistical techniques and machine learning algorithms may prove to be an effective, extremely fast and accurate method, which could have practical applications to ensure the quality and safety of meat and meat products.

1.2 Multivariate Analyses and Chemometrics

The vast amount of biological information generated by the advanced analytical instruments such as the omics fields, demand appropriate multivariate statistical tools for data analysis. Multivariate analysis can be defined as "the simultaneous statistical analysis of a collection of random variables" (Izenman, 2008). According to Vermuza and Filzmoser (2009), chemometrics may be defined as the "extraction of chemically relevant information out of analytical chemical data by mathematical and statistical tools".

The multivariate techniques applied in the field of chemometrics are conventionally divided into two main categories, namely supervised and unsupervised. Unsupervised methods "attempt to disclose naturally occurring groups and structures within the dataset without previous knowledge of any class assignment" (Alvarez-Ordoñez and Prieto, 2012). They chiefly focus on the discovery of patterns, trends, clusters and/or outliers in the data, and they include techniques such as Principal Component Analysis (PCA) and cluster analysis. On the other hand, supervised learning algorithms "make use of a priori knowledge of classes to guide the characterisation or classification process" (Alvarez-Ordoñez and Prieto, 2012); these algorithms generate prediction models for regression, classification, pattern recognition, or machine learning tasks. Characteristic examples of supervised learning involve Partial Least Squares Discriminant Analysis (PLS-DA) and Support Vector Machines (SVMs), among many others.

1.3 Data pre-treatment

Nowadays, the extraction of relevant information from highly heterogeneous datasets constitutes a major challenge (van den Berg *et al.*, 2006). It is well established that prior to the application of any type of data analysis, proper data pre-treatment is crucial for the outcome and the interpretability of the results. Data pre-treatment can make the difference between a useful model and no model at all. Therefore, biological data under investigation are often scaled, centered and/or transformed. The application of pre-treatment techniques may prove to be extremely fruitful, especially under circumstances where the variables span over wide and different ranges. In addition, pre-treatment techniques aim to minimise the influence of disturbing factors such as measurement noise.

The selection of chemometrics method to be applied, strongly influences the selection of the data pre-treatment methods. Different techniques focus on different aspects of the data. For instance, clustering algorithms focus on revealing similarity and dissimilarity patterns, whereas PCA attempts to explain the maximum variation based on a few meaningful components. Thus, a certain pre-treatment method may enhance the results of one technique and obscure the results of another.

1.3.1 Mean-centering

Mean-centering is conducted by subtracting the mean of each variable (column). "Centering converts all the concentrations to fluctuations around zero instead of around the mean" (van den Berg *et al.*, 2006). This is step is usually performed so that all the components found by PCA have as their origin the centre (centroid) of the data (Craig *et al.*, 2006). In general, mean-centering enhances the interpretability of a model; it can be useful when different variables have different means.

$$X_{mc} = X - \overline{X}$$

Equation 1 Mean-centering formula

1.3.2 Auto-scaling

In auto-scaling, also known as unit or unit variance scaling, each variable (column) is scaled to unit variance by using the standard deviation as the scaling factor. As the initial step, mean-centering is performed by subtracting the column mean from every data value. Subsequently, scaling is applied by dividing the centered columns by their standard deviation s. Auto-scaling is crucial if different variables are measured over very different ranges or units such as temperature, pressure and concentration (Brereton, 2009). Once scaled, all the variables will have the same weight and will be equally important in the analysis (Wold *et al.*, 2001; van den Berg *et al.*, 2006). If the data are mean-centered, the weighting reflects the covariance of the variables, while in unit variance scaling the weighting reflects their correlation (Craig *et al.*, 2006). The mathematical equation for auto-scaling is

$$X_{as} = \frac{(X - \overline{X})}{s}$$

Equation 2 Auto-scaling formula

1.4 Multivariate Analysis: Unsupervised Methods

1.4.1 Principal Component Analysis

As thoroughly described in Section 1.1 scientists attempt *via* high-throughput platforms to collect as much information as possible from a single experiment. This may result in the generation of a large number of measurements (variables), a subset of which may not be very informative or even related to the study. Datasets with such great number of variables tend to present high dimensionality, correlations and redundancies. The plots of Figure 1-3 demonstrate three distinct cases, where the data illustrate low, medium and high redundancy respectively. The higher the redundancy, the more difficult it becomes to reveal patterns and trends in the original data.

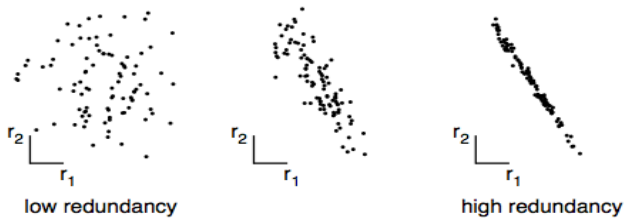


Figure 1-3 Representation of possible correlations and redundancies in high-dimensional data
For two random variables r_1 and r_2, the plot on the left displays no obvious relationship between the variables. However, the variables of the plots in the centre and the right are highly correlated since one can be used to predict the other. The figure has been extracted from Shlens, 2005.

Principal Component Analysis (PCA) (Jackson, 1991; Wold *et al.*, 1987) is the most commonly used technique for dimensionality reduction, data compression and feature extraction. The PCA algorithm reduces the initial number of possibly correlated variables into a new lower number of uncorrelated variables, known as the Principal Components (PCs). Geometrically, we can imagine the input data as a cloud of points in a high-dimensional space. As illustrated in Figure 1-4, this cloud of points is probably longer in a certain direction of the pattern space; in this direction the data appear to be most different and PCA draws the first axis (PC).

The first PC places all points the farthest apart from each other, extracting thus the highest variance. Similarly, a perpendicular to the first PC axis is drawn for the second PC, which accounts for the second highest variance. The process is repeated to get multiple orthogonal principal components. Each successive orthogonal axis displays a decreasing amount of the total variance.

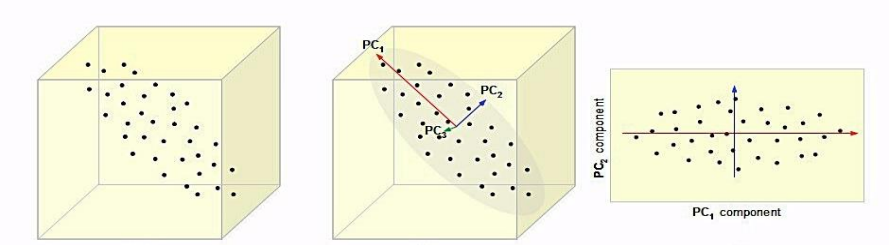


Figure 1-4 Extracting the Principal Components

PCA projects the input data into a subspace of reasonable and meaningful dimension by setting new directions in the pattern space. Thus, the projected cloud of points is as dispersed as possible. The figure has been extracted from Kavraki (2007).

In matrix notation, suppose that m variables have been observed on n instances. The generated multivariate dataset forms an $n \times m$ data matrix X with n rows (observations) and m columns (variables). Thus, a cloud of n points is created in an m-dimensional space, where a new axis is used per variable. The PCA algorithm reduces the size m of possibly correlated variables into d new uncorrelated variables (PCs), where $d \leq m$. Each PC can be expressed mathematically as an orthogonal linear combination of the original variables $x_i \in \mathbb{R}^n$. In PCA, the original matrix X can be decomposed into the scores matrix T $(n \times d)$, loadings matrix P $(d \times m)$ and a residuals matrix E. Several algorithms can be used for data decomposition, the most widely applied of which are Singular Value Decomposition (SVD) and the Nonlinear Iterative Partial Least Squares (NIPALS) (Wold, 1975) algorithm. In general, the mathematical equation for PCA can be described by

$$X = T \cdot P^T + E$$

Equation 3 PCA scores and loadings

1.4.2 Cluster Analysis

Cluster analysis consists of a set of unsupervised methods that are used in numerous data mining tasks. The clustering algorithms attempt to partition a dataset into several subsets – the clusters – so that data belonging to the same cluster are mutually similar, providing a sense of homogeneity.

1.4.2.1 Hierarchical Cluster Analysis (HCA)

Hierarchical clustering (HCA) is based on calculating the distances between n elements found in a given matrix X of size $n \times m$. The distances represent the degree of similarity/dissimilarity between these objects. The shorter the distance, the more similar the objects are with each other. HCA is based on two important categories of algorithms – distance and linkage algorithms.

Distance algorithms determine how the similarity or "distance measure" between two given objects is calculated. The most widely used distance algorithms include Euclidean and Mahalanobis distance, among others. For instance, for two objects p and q in X, the Euclidean distance in m-dimensional space satisfies the equation

$$d(p,q) = \sqrt{\sum_{i=1}^{M} (p_i - q_i)^2}$$

Equation 4 Euclidean distance algorithm

Hierarchical clustering is graphically represented in tree structures, also known as dendrograms. Linkage algorithms determine how the clustering is performed. A bottom-up linkage algorithm includes the following steps:

1. Each object forms and belongs to its own cluster
2. The two closest clusters are linked together
3. The two linked clusters are aggregated into a single new cluster
4. The algorithm keeps iterating from Step 2 until the number of clusters is one

Linkage algorithms consist of various different algorithms such as single linkage, complete linkage and Ward's method. For two given clusters X and Y, a single linkage algorithm calculates the shortest distance between the two clusters as described in Equation 5

$$d(X,Y) = \min_{x \in X, y \in Y} \|x - y\|$$

Equation 5 Single linkage algorithm

Where x and y are elements of the clusters X and Y respectively.

1.4.2.2 k-means Clustering

k-means clustering is an unsupervised clustering technique that attempts to "minimise the sum of point-to-centroid distances, summed over all k clusters" (Arthur and Vassilvitskii, 2007). The objective function E for k-means clustering is

$$E = \sum_{k=1}^{C} \sum_{i=1}^{n_j} \left\| x_i^{(k)} - c_k \right\|^2$$

Equation 6 k-means clustering algorithm

Where, C is the number of clusters, $x_i^{(k)}$ is the i^{th} pattern belonging to the k^{th} cluster, c_k is the centre of cluster k and n is the number of data points. The algorithm's steps can be described as follows:

1. Initially, the number of clusters k is selected, for instance $k = 4$
2. Randomly the items are assigned to the k clusters
3. A new centroid is calculated for each of the k clusters (a distinct set of points belongs to a certain centroid)
4. The distance of each item towards the k centroids is calculated
5. The items are subsequently assigned to the closest centroid
6. The algorithm keeps iterating until the assignments to the clusters are stable

The algorithm's simplicity and speed makes it an appealing technique for cluster analysis. However, a major disadvantage of this algorithm is that it is sensitive to the selection of the initial partitions.

1.5 Multivariate Analysis: Supervised Learning

This research will solely focus on the investigation of multivariate classification techniques. A classifier, also known as predictor, can be defined as "a function that maps an unlabelled instance to a label using internal data structures" (Kohavi, 1995). Supervised classification derives from the concept of learning by experience (Ciosek *et al.*, 2005). A model is trained to distinguish groups of a predefined dataset where the class of each sample is already known. The training dataset is used to establish a mathematical model, which in turn should be capable of predicting the class membership of ideally unseen data (Izenman 2008). Supervised learning algorithms are characterised by a predefined set of parameters, which may have a profound effect on the resulting performance (Chapelle *et al.*, 2002). Therefore, thorough selection of these parameters is a necessity.

1.5.1 Partial Least Squares – Discriminant Analysis

Partial Least Squares-Discriminant Analysis (PLS-DA) (Barker and Rayens, 2003) is a widely used classification technique in the field of chemometrics (Westerhuis *et al.*, 2008). It is a linear model that consists of Partial Least Squares (PLS) (Wold, 1975) dimensionality reduction and Linear Discriminant Analysis (LDA) applied on the PLS components. Unlike PCA, which attempts to capture the maximum variance, PLS-DA aims to maximise the covariance – accomplish both correlation and maximum variance – between the input data and an output class (Wise *et al.*, 2003; Weber *et al.*, 2011).

In matrix notation, suppose that X is a predictor matrix, which corresponds to independent variables, and y is a class affiliation vector that holds the dependent variables. PLS-DA attempts to model the relationship between dependent and independent variables by projecting the data matrices X and y into a new subspace. The orthogonal axes in the PLS subspace are also known as Latent Variables (LVs). The output of PLS-DA is the product of two smaller matrices, the scores matrix (PLS-DA scores) and the predicted affiliation matrix. Thus, it satisfies the mathematical equations

$$X = T \cdot P^T + E$$
$$y = T \cdot q + f$$

Equation 7 Partial Least Squares - Discriminant Analysis

Where, T represents the PLS score matrix, P and q are the PLS loadings, and E and f are the PLS residuals. Although the PLS scores (LVs) are orthogonal as in PCA, the loadings are not (Brereton, 2009). LVs are likely to offer a better separation between different observations (samples) when compared to PCs since they take the class labels into account (Rossini *et al.*, 2012).

1.5.2 Support Vector Machines

Support Vector Machines (SVMs) (Boser *et al.*, 1992; Cortes and Vapnik, 1995) are a powerful state-of-the-art machine learning technique applied in data mining cases such as classification, regression and novelty detection. Initially introduced by Cortes and Vapnik (1995) for binary classification (Hsu and Lin, 2002; Glasmachers, 2008), SVMs became increasingly popular in the scientific community over the past decade.

1.5.2.1 Linear SVM Classifiers: Separable Data

"The simplest type of classifier is a linear classifier" (Bottou *et al.*, 1994; Brereton *et al.*, 2009). In a hypothetical binary classification problem, the SVM model is given as input a training dataset $S = ((x_1, y_1), \dots, (x_n, y_n))$, where $x_i \in \mathbb{R}^n$ is the set of n input instances and $y_i \in \pm 1$ their associated class labels. The chief goal of any SVM algorithm is to determine a classification function that best fits the training dataset. In the case of linearly separable points, the decision function has the form

$$f(\mathbf{x}) = \text{sgn}(\langle \text{w} \cdot \text{x} \rangle + b) = \text{sgn}\left(\sum_{i=1}^{n} w_i \cdot x_i + b\right)$$

Equation 8 Linear SVM classifier as a decision function

Where w is the weight vector, x_i is the i^{th} training example with a corresponding label y_i and b is the bias. According to Boser *et al.* (1992), w and b are the "adjustable parameters" of the SVM decision function.

In any linearly separable binary dataset, there is an infinite number of possible discriminant hyperplanes that can finely separate the two classes (Bennett *et al.*, 2000; Suykens *et al.*, 2002). All generic planes, including the optimal separating hyperplane, satisfy the equation

$$\mathrm{w} \cdot \mathrm{x} + b = 0$$

Equation 9 SVM linear separating hyperplane

Support vector machines attempt to separate the data by fitting a hyperplane that returns a low generalisation error, while simultaneously aim to maximise the distance or 'margin' between the nearest points of the two classes (Bennett *et al.*, 2000; Suykens *et al.*, 2002). Two parallel class hyperplanes define the margin of the SVM classifier. The supporting class planes can be described by

$$\mathrm{w} \cdot \mathrm{x} + b = \pm 1$$

Equation 10 SVM supporting hyperplanes

The margin of the SVMs is expressed by $\frac{2}{\|w\|}$ (Smola, 1998). According to Boser *et al.* (1992), the margin maximises by minimising the norm $\|w\|^2$. This convex optimisation problem satisfies the equation

$$\min \frac{1}{2} \|w\|^2$$
$$\text{subject to:} \quad y_i(w \cdot x_i + b) \geq 1 \quad i = 1, \ldots, n$$

Equation 11 SVM optimisation problem – primal form (hard-margin SVMs)

The training points that are found on the edge of the margins of each class, which achieve the minimum distance from the optimal decision hyperplane, are termed Support Vectors (SVs). An important concept is that SVMs conduct linear classification based on a different approach than most chemometric methods. The SVM boundary depends solely on the selected support vectors, while the remaining samples have no influence over it (Boser *et al.*, 1992). On the contrary, methods such as PLS-DA use all available samples in order to determine the separating planes between classes (Brereton *et al.*, 2009; Xu *et al.*, 2006).

1.5.2.2 Linear SVM Classifiers: non-separable Data

The previous section provided an overview of linear SVM classifiers when applied to perfectly separable data. However, most of the real-life applications are complex and thus, separation between different classes is not as straightforward. In such cases, more expressive hypothesis spaces are required to describe non-separable linear and nonlinear cases (Christianini *et al*., 2000; Suykens *et al.*, 2002).

Cortes and Vapnik (1995) introduced additional slack variables ξ_i in the implementation of the SVMs in order to address the problem of non-separable data. The slack variables "relax" the hard-margin constraints, leading to softer margins that tolerate misclassifications (Cortes and Vapnik, 1995; Smola, 1998; Christianini *et al*., 2000). The regularisation parameter $C > 0$, known as the penalty error, determines the trade-off between training error toleration and margin maximisation (Chapelle *et al.*, 2002; Boardman and Trappenberg, 2006). As the values of C increase, the misclassifications become more significant. Soft-margin SVMs require the solution of the linearly constrained quadratic minimisation problem:

$$\min \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i$$

$$\text{subject to:} \quad y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

Equation 12 SVM optimisation problem – primal form (soft-margin SVMs)

The constrained optimisation problem of Equation 12, which constitutes the primal objective function, can be solved using standard Lagrangian theory (Burges, 1998; Smola, 1998; Shölkopf and Smola, 2001). Thus, the primal optimisation problem can be expressed in the dual form:

$$\max \sum_i a_i - \frac{1}{2}\sum_i\sum_j a_i a_j y_i y_j x_i^T x_j$$

$$\text{subject to:} \quad 0 \leq a_i \leq C \quad \text{and} \quad \sum_i y_i\, a_i = 0$$

Equation 13 SVM optimisation problem – dual form

Where a_i ($a_i \geq 0$) are the Lagrange multipliers and C their upper bound. Only the support vectors satisfy $a_i > 0$, whereas the remaining instances $a_i = 0$ (Liu *et al.*, 2006; Xu *et al.*, 2006). Therefore, omitting all the training instances that do not constitute the support vectors will result in exactly the same decision boundary (Belousov *et al.*, 2002).

1.5.2.3 Nonlinear SVM Classifiers

A hypothetical case of nonlinear class separation is displayed in Figure 1-5. The input space under study is too complex to provide an optimal hyperplane that accurately separates the classes of the widely scattered data. Boser, Guyon and Vapnik (1992) extended once more the functionality of the SVMs with the introduction of the most powerful SVM attribute, the "kernel trick" (Smola and Shölkopf, 2004). Instead of forming a boundary in the non-separable input space, a nonlinear feature (kernel) function ϕ projects the data into a high – possibly infinite – dimensional feature space $\mathcal{H}$ as demonstrated in Figure 1-5, where linear separation is theoretically feasible (Chapelle and Vapnik, 2000; Cristianini and Shawe-Taylor, 2000). The back-projection of the optimal separating hyperplane from the new feature space to the original input space generates the nonlinear boundary of given complexity (Xu *et al.*, 2006; Brereton, 2009).

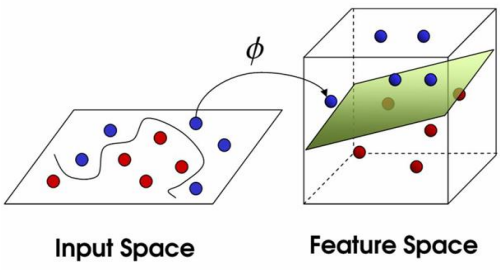


Figure 1-5 Nonlinear SVM classifier

The figure displays an example of feature mapping using the kernel trick. It is obvious that the original data cannot be separated by a linear hyperplane in the two-dimensional input space. The kernel function implicitly maps the data into a new high-dimensional feature space, where linear separation is feasible. The figure has been extracted from Brereton (2009).

Using the feature function $\phi: X \to \mathcal{H}$ for implicit nonlinear mapping from the input space X to a feature space $\mathcal{H}$, the primal optimisation problem can be expressed as:

$$\min \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i$$

$$\text{subject to:} \quad y_i(w \cdot \phi(x_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

Equation 14 SVM optimisation problem – primal form (kernel trick)

Due to the possibly infinite dimensionality of $\mathcal{H}$, the primal optimisation problem of Equation 14 using the feature function $\phi(x)$ may be computationally too hard to solve. Thus, the optimisation problem is usually solved in its dual space, where the dimensionality is much lower than the feature space (Boser *et al.*, 1992). By substituting the kernel trick in the dual form yields:

$$\max \sum_i a_i - \frac{1}{2}\sum_i\sum_j a_i a_j y_i y_j K(x_i, x_j)$$

$$\text{subject to:} \quad 0 \leq a_i \leq C \quad \text{and} \quad \sum_i y_i\, a_i = 0$$

Equation 15 SVM optimisation problem – dual form (kernel trick)

Where $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$ is a predefined kernel function that performs the nonlinear mapping. In addition to the linear kernel $K(x_i, x_j) = x_i^T x_j$, which corresponds to the original linear SVM, the most commonly applied nonlinear kernels include:

Radial Basis Function (RBF): $K(x_i, x_j) = \exp(-\gamma \parallel x_i - x_j \parallel^2) = \exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma^2}\right)$

Polynomial: $K(x_i, x_j) = (\gamma x_i^T x_j + coef_0)^d$

Sigmoid: $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + coef_0)$

Equation 16 Nonlinear SVM kernels

Every kernel is characterised by a set of parameters – the hyperparameters – that have to be optimised for a particular problem (Chapelle and Vapnik, 2000; Xu *et al.*, 2006). The Gaussian Radial Basis Function (RBF) kernel is particularly popular especially in cases where there is little or no knowledge about the data under study. In RBF SVMs, only one kernel parameter has to be optimised – the value of γ or σ – in addition to the regularisation parameter *C*.

The γ value determines the degree of nonlinearity or width of the RBF kernel (Boardman and Trappenberg, 2006; Verplancke *et al.*, 2008), and is inversely related to σ, the spread of the data, where $\gamma = \frac{1}{2\sigma^2}$. Higher values of γ result in greater nonlinearity of the decision boundaries. More specifically, very high values of γ (low values of σ) potentially result in sharp peaks, "spiky" functions and boundaries that surround individual samples as illustrated in Figure 1-6 (Valentini and Dietterich, 2004; Brereton, 2009). As the γ value decreases, the Gaussians become broader with smoother surfaces that fit the data quite well. According to Keerthi and Lin (2003), for small values of γ ($\sigma^2 \to \infty$) the RBF kernel tends towards a linear boundary (Boser *et al.*, 1992; Hsu *et al.*, 2003). Thus, a linear classifier may be considered a special case of the RBF model since "with a suitable combination of hyperparameters (C, γ), the testing accuracy of the RBF kernel is at least as good as the linear kernel" (Boser *et al.*, 1992; Keerthi and Lin, 2003; Hsu *et al.*, 2003; Chang *et al.*, 2010).

In addition, as presented in Section 1.5.2.2, the cost parameter C controls the complexity of the SVM boundaries. More specifically, according to Xu *et al.* (2006), the cost parameter controls the optimal trade-off between the two criteria of Equation 14, maximising the margin and minimising the training error. As $C \to \infty$, the hard margin case is obtained, and thus, lower tolerance of misclassification is allowed (Brereton, 2009). The high values of C will force the creation of extremely complex boundaries that misclassify as few training samples as possible. Large values of C may often lead to instances of overfitting (Foody and Mathur, 2004). On the contrary, a lower value of C creates wider margins, which allows instances close to the boundary to be ignored (Ben-Hur *et al.*, 2010). For very low values of C, independent of the γ value, the SVM models are unable to learn, causing a problem of underfitting (Valentini and Dietterich, 2004).

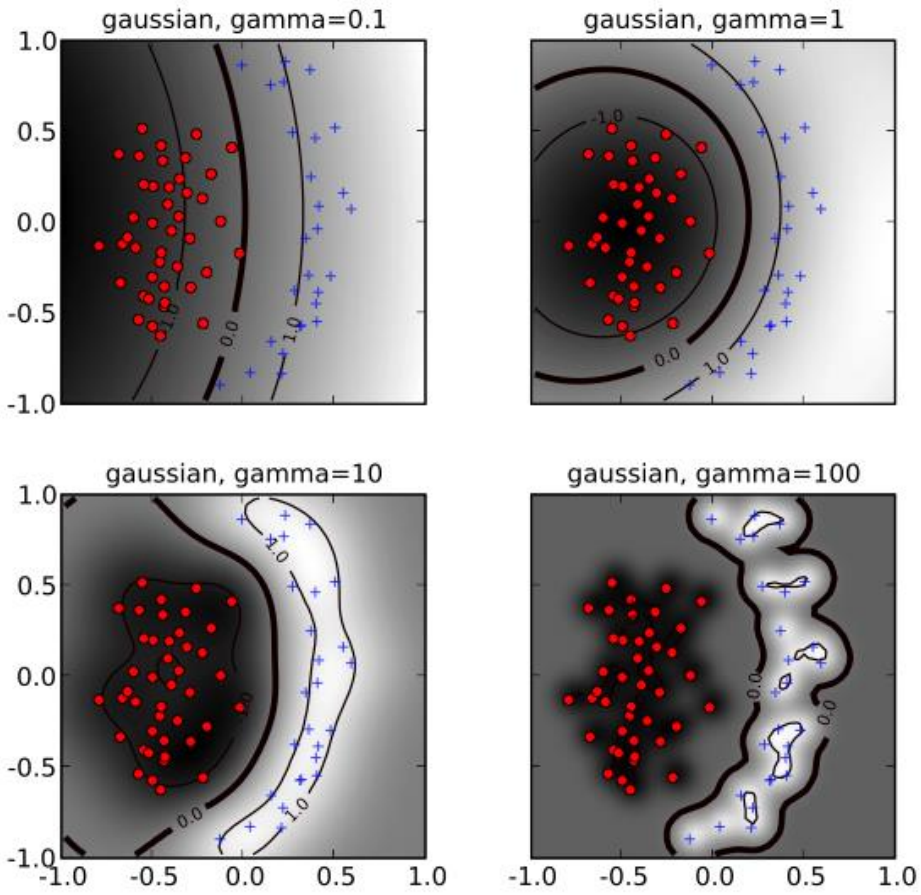


Figure 1-6 The effect of the hyperparameter γ on the SVM boundaries

The figure demonstrates the effect of varying the hyperpameter γ as the cost parameter C is kept constant. For small values of γ, the SVM boundary tends towards linearity. As γ increases, the flexibility and curvature of the decision boundaries increase. For large values of γ, the "spiky" functions and the plethora of narrow Gaussian "bumps" may result to a high training accuracy but low generalisation ability – a case of overfitting. The figure has been extracted from Ben-Hur *et al*. (2010).

1.5.2.4 Multi-class SVMs

SVMs were initially introduced for binary classification problems. Over the years, the functionality of SVMs was extended to allow multi-class cases. Several methodologies have been proposed, the most popular of which are "one-against-all" and "one-against-one". Both methods divide the multi-class problem in a series of binary problems (Duan and Keerthi, 2005).

The "one-against-all" approach (Bottou *et al*., 1994) is the earliest and simplest method proposed, which involves determining how well a sample is modelled by each class individually, and subsequently selecting the class it is modelled-by at its best (Foody and Mathur, 2004; Brereton and Lloyd, 2009). Thus, for a N class problem, N binary classifiers are created and trained, one for each given class (Karatzoglou *et al*., 2006). The "one-against-all" approach is based on a "winner-takes-all" strategy (Duan and Keerthi, 2005). On the contrary, the most recent "one-against-one" (Kressel, 1999) approach constructs several binary SVM classifiers for each available pairwise combination of classes (Hsu and Lin, 2002). Subsequently, the results of all individual classifiers are aggregated using a voting mechanism such as "majority vote" (Duan and Keerthi, 2005). In this case, $N(N-1)/2$ SVM models are created, one for each pairwise combination of classes. According to Hsu and Lin (2002), this approach verily generates robust outcome when employed with SVMs.

1.5.3 Ensemble Models

A major problem in multivariate classification is that often standalone classifiers may achieve very high classification accuracies in the training process, however, their generalisation performance (test performance) when applied to new unseen data may greatly vary. Therefore, instead of using only a single final model, the concept of a classification ensemble is based on the fusion of many diverse yet accurate models to obit a range of predictions (Dietterich, 2000; Westerhuis *et al*., 2008). Thus, this approach aims to improve the overall classification accuracy, and provide more stable and accurate results. An ensemble can be constructed using any type of classifier such as PLS-DA and SVMs.

1.6 Validation

The most crucial step in supervised learning is the assessment of the performance of a classifier on future unseen data (Wold *et al.*, 2001; Izenman, 2008); this is commonly referred to as the "generalisation performance" of the classifier (German *et al.*, 1992). Two equally important performance metrics may be used to estimate the overall predictive power of a pattern recognition system. The first indicator frequently used in chemometrics is the percentage of correctly classified samples ($\%CC$):

$$\%CC = \frac{N_c}{N_c + N_{nc}} \times 100\%$$

Equation 17 Percentage of correctly classified samples (%CC)

Where N_c and N_{nc} are the number of correct and incorrect classifications respectively (Ciosek *et al.*, 2005). The sum of N_c and N_{nc} is equal to the total number of instances n in the dataset. The model with the maximum number of correctly classified samples is considered optimal.

As an alternative, the optimal classifier may be selected based on the prediction error (generalisation error). The best classification model attempts to minimise the prediction error, which is equal to the mean of squared prediction errors as provided by

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{n}}$$

Equation 18 Root Mean Square Error

Where $\hat{y}$ are the predicted values, y are the initially observed values (the real classes) and n the total number of objects in a dataset.

Furthermore, metrics such as the bias and the variance are also very powerful tools for the assessment of a machine learning model. The bias of a method can be defined as "the difference between the expected and the estimated value" (Kohavi, 1995). In addition, the variance indicates the variability of a classifier's predictive power across the different training sets (Bauer and Kohavi, 1999). Ideally, a good classifier presents both low bias and low variance. According to Burges (1998), the generalisation ability of a classifier is highly dependent on the "bias-variance trade-off" (Germal *et al*., 1992).

1.6.1 The holdout method

The holdout method randomly partitions the entire input dataset into two mutually exclusive subsets (Suykens *et al.*, 2002). The two newly created sets are commonly termed as the training and the test set, or holdout set. A common approach is to randomly designate 1/3 of the initial data as the test set, whereas the remaining 2/3 of the data are used to train the model (Kohavi, 1995; Brereton, 2009). The test set is kept aside during the training process and is only used to evaluate the accuracy or the error rate of the trained classifier. In order to assure strong classifier and optimal prediction rates, there should be exactly a third of the instances for each available class label included in the test set (Kohavi, 1995; Brereton, 2009); this approach is often referred to as the stratified holdout method.

The main drawback of this method is the demand for an adequate amount of samples in the test set. The prediction rate tends to increase as more instances are provided. The more instances included in the test set, the higher the bias of the estimate. However, for datasets that the initial number of samples is quite small, the results tend to present high variance. Thus, alternative algorithms such as cross-validation and bootstrapping are applied.

1.6.2 k-fold Cross-Validation

Cross-validation (Wold, 1978) is the most popular validation technique. In k-fold cross-validation, the entire dataset is randomly split into k mutually exclusive subsets – the folds – of approximately equal size (Kohavi, 1995; Duan *et al.*, 2003; Izenman, 2008). The algorithm of k-fold cross-validation performs k iterations in total. As demonstrated in Figure 1-7, in each iteration, one subset is considered to be the test set, while the remaining $k - 1$ folds are used to train the classifier. Therefore, each fold will be used exactly once for testing. As thoroughly described in the previous section, each test set is kept aside and should in no way be used during the development of the model (Brereton, 2006; Westerhuis *et al.*, 2008). The total prediction rate of the classifier is calculated by averaging the individual test results over the k iterations.

A great advantage of k-fold cross-validation is that all examples in the dataset are eventually used for both training and testing. Thus, the bias of this estimate is reduced compared to the holdout method. In general, the outcome of k-fold cross-validation depends highly on the split of the initial dataset into folds. Since this partition is not canonical, it may often lead to instances of high variance, especially for large values of k (Glasmachers, 2008); thus, $k = 5$ or $k = 10$ are most commonly applied for cross-validation (Clarke *et al*. 2009).

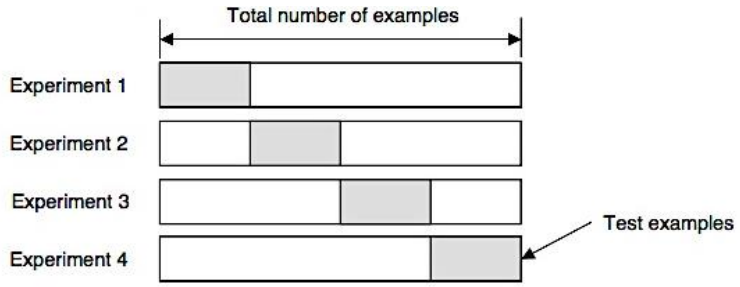


Figure 1-7 *k*-fold Cross-Validation

In k-fold cross-validation, the initial dataset is partitioned into $k = 4$ mutually exclusive folds of approximately equal size. In every run, a single fold is omitted and the remaining $k - 1$ sets are used in the model's training process. One can conclude that the split and the allocation of samples into the folds may easily influence the prediction rates. The figure has been extracted from http://research.cs.tamu.edu/prism/lectures/iss/iss_l13.pdf

1.6.3 Leave-One-Out Cross-Validation

Leave-One-Out Cross-Validation (LOOCV) is the extreme version of k-fold cross-validation. In this case, k is equal to N, which is the total number of samples in the dataset (Duan *et al.*, 2003). Thus, training and testing are repeated N times. During each run, a single sample is used as the test set, while all the remaining $N-1$ samples are used in in the model's training process as illustrated in Figure 1-8.

Even though the LOOCV algorithm produces an almost unbiased estimate of the expected test error, due to its high variance it may often be leading to unreliable estimates (Efron, 1983; Kohavi, 1995; Chapelle and Vapnik, 2000; Duan *et al.*, 2003; Glasmachers, 2008; Clarke *et al*. 2009). Furhermore, LOOCV is a computationally expensive and time-consuming validation method; thus, it is mainly used in cases where the input data are extremely scarce such that the computational expense is no longer a discouraging factor (Cawley *et al.*, 2007).

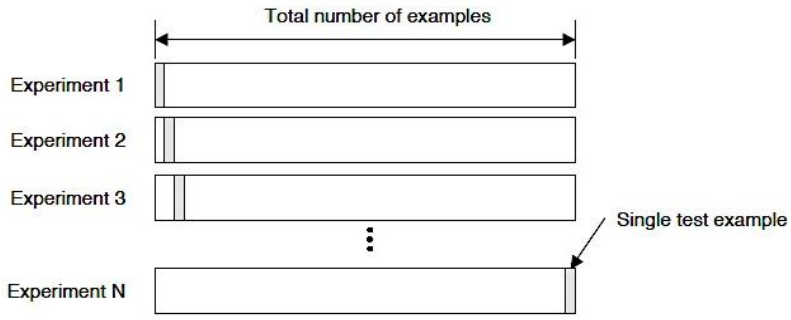


Figure 1-8 Leave-One-Out Cross Validation (LOOCV)

The figure graphically represents the steps of leave-one-out cross-validation. In this method, the number of folds is equal to the number of initial observations. Thus, in every run, all samples but one are used for training, whereas the single sample is kept aside for testing. The figure has been extracted from http://research.cs.tamu.edu/prism/lectures/iss/iss_l13.pdf

1.6.4 Bootstrapping

Bootstrapping is a validation technique initially introduced by Efron *et al.* (1993). Thorough information about the methodology can be found in Efron and Tibshirani (1993). Bootstrapping has proven to be a powerful technique, especially when dealing with relatively small datasets.

Given an initial dataset with n samples, a bootstrap training dataset is created by sampling n instances from the original data uniformly with replacement; based on this approach, any given sample could be present multiple times within the same bootstrap training set. The probability for any given instance not being present in the bootstrap training set after n selections is approximately 36.8% (Kohavi, 1995; Bauer and Kohavi, 1999). These instances constitute the bootstrap test set. A common approach is to repeat bootstrapping a great number of times in order to construct, for example, 100 or even up to 1000 news bootstraps of the same size. The total number of bootstraps strongly depends on the number of samples in the initial dataset. Bootstrapping generates instances of lower variance and relatively moderate bias compared to the previous techniques. Even though bootstrapping is a fairly straightforward method, it consists a computationally demanding statistical procedure that may lead to extremely long execution times.

1.6.5 Model Selection, complexity and the bias-variance trade-off

It is a common approach to use validation techniques such as cross-validation or bootstrapping as a means of optimising the adjustable parameters of a classifier. In order to maximise the performance of a classification model, it is often tempting to repeatedly train the model until a minimum training prediction error (or maximum training accuracy) is achieved (Suykens *et al.*, 2002; Brereton, 2006; Izenman, 2008).

In such cases, the model becomes quite complex. After a certain number of repetitions, this complex model is able to predict very accurately, with almost no errors, the given training dataset. However, when tested on unseen data, the generalisation performance appears to be relatively poor (Boser *et al.*, 1992). The effect where the validation error increases while the training error steadily decreases is known as overfitting (Burges, 1998).

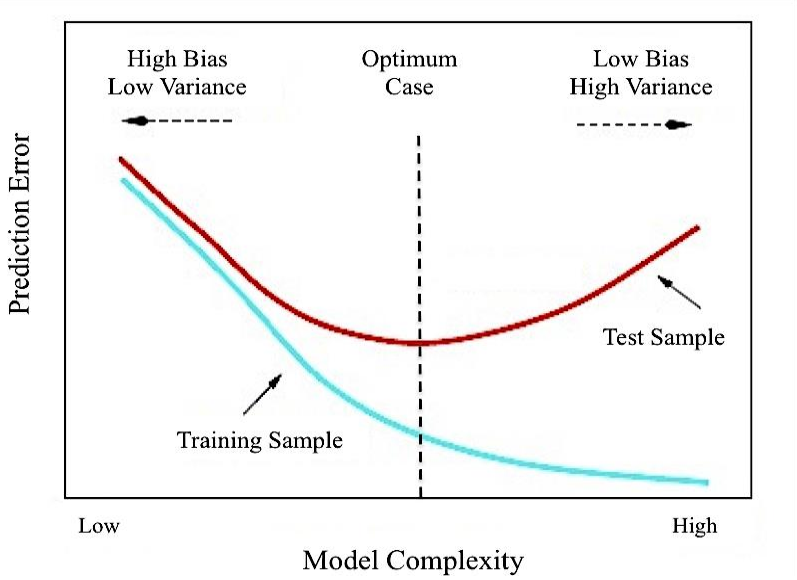


Figure 1-9 Model complexity and overfitting; the bias-variance trade-off

The figure has been adapted from Hastie *et al.* (2009).

In order to avoid the possibility of overfitting, an optimal stopping point has to be determined during the optimisation process as illustrated in Figure 1-9. To address this issue, it is now becoming a common practice to apply the "three-way split" rule. As a first step, the initial dataset is divided into two disjoint subsets, the training and test set, where the test set plays the role of unseen data and is kept aside during the training process (Westerhuis *et al.*, 2008; Smolinska *et al.*, 2012). Subsequently, the training set is further split into training and validation sets using either cross-validation or bootstrapping. The validation (often referred to as optimisation) dataset is used as a pseudo-test set in order to optimise the classifier and stop the training process before overfitting (Westerhuis *et al.*, 2008). Once the optimal parameters have been identified, the independent test set is used to determine the performance of the predictor.

1.7 Permutation Tests

Permutation tests (Good, 2006) are used to assess the statistical significance of the prediction results in addition to providing an objective estimation of the performance and stability of a model. Permutation testing makes use of non-parametric tests on an initial (null) hypothesis, H_0. In a classification problem, permutation tests attempt to prove whether the relationship between observed data and sample classes is really significant or whether a model could have been built to group samples into any arbitrary class.

In each permutation iteration, the input data matrix remains unaltered while the associated class vector is randomly shuffled; thus, the class distribution in the dataset remains unaltered, however, the samples correspond to randomly assigned classes. This procedure randomises the association between the two sets of variables. Permutation testing is repeated in total a minimum of 100 times until a clear stable distribution of results is obtained. At the end of permutation testing, and assuming that the initial hypothesis is true, we can determine the frequency of models that presented accuracies equal or higher than the original model. This frequency metric, as illustrated in Figure 1-10, is commonly referred to as the p-value (Hubert and Schultz, 1976).

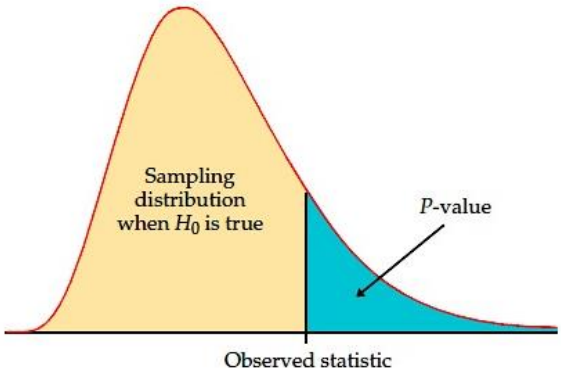


Figure 1-10 Permutation tests and the *P*-value

The figure displays generated permutation distribution under the null hypothesis, and highlights the p-value for the observed statistic. The figure has been extracted from Hesterberg *et al.* (2003).

1.8 Aims and objectives

The overall aim of this thesis was the construction of a suite of software tools designed to support novel instrumental approaches for the accurate, rapid and inexpensive evaluation of meat freshness. This work was carried out as part of the SYMBIOSIS-EU project, funded by European Commission Framework 7. The overall aim was addressed by the thesis' objectives, which are presented as follows.

Initially, a multivariate analysis pipeline was constructed as the first working prototype of the suite of tools. The statistical pipeline was developed for the analysis of the heterogeneous standalone data obtained by a single case study ("Shelf life beef fillets stored in air at 0, 5, 10, 15 and 20°C"). Unsupervised techniques were applied for dimensionality reduction, feature extraction and the investigation of underlying patterns in the data. In addition, classification models were built using PLS-DA and SVMs, whereas ensembles of individual classifiers were implemented as a means of increasing the generalisation performance. Various validation and evaluation methods were meticulously examined in order to ensure that the performance metrics are representative of real-world application, and to provide an indication of the statistical significance of the classification results.

A high-level heuristic approach for the optimisation of the computationally intensive SVM tuning was implemented, which in addition to parallel programming, is applied in order to significantly reduce the overall execution times, minimise the computational complexity and satisfy the demand for computational power.

Furthermore, various data integration algorithms for the fusion of heterogeneous data into a "global" consensus were examined as a means of determining whether better generalisation performance is achieved when integrated as opposed to standalone datasets are used.

In addition, in order to ensure the generic nature and applicability of the implemented software to real-world problems, the pipeline was thoroughly tested on three further independent case studies ("Shelf life of minced beef stored in air, MAP, and in active packaging at 0, 5, 10 and 15°C", "Survey of minced beef" and "Pork stored in air and MAP").

Finally, the latest visualisation techniques, graphics libraries and web technologies were investigated in order to develop a wide range of graphs, dynamically generated reports and interactive visualisation tools that enhance the interpretability of the analysis results.

2 Development of the multivariate analysis pipeline for the detection of meat spoilage

2.1 Introduction

This chapter introduces a first prototype of the constructed multivariate analysis pipeline for the analysis of standalone heterogeneous data obtained by various analytical techniques. The pipeline was designed and implemented upon a single case study using samples of shelf life beef fillets stored in air at 0, 5, 10, 15 and 20°C. The first step of the analysis includes the application of unsupervised methods for the extraction of prominent features, dimensionality reduction and the investigation of underlying patterns in the data. The datasets are subsequently imported into multi-class machine learning models, which include PLS-DA and SVMs. Classification ensembles were implemented as a means of enhancing the generalisation performance of the individual models. Finally, thorough model validation and evaluation methods were applied to ensure that the performance metrics are representative of real-world application, as well as to provide an indication of the statistical significance of the results.

2.2 Materials and Methods

2.2.1 Case study 1: "Shelf life beef fillets stored in air at 0, 5, 10, 15 and 20°C"

2.2.1.1 Sample Preparation

The first case study of this thesis investigates shelf life beef fillets stored in air at 0, 5, 10, 15 and 20°C (Argyri, 2010). A detailed explanation of the experimental techniques and the methodology used in order to obtain the data for this case study can be found in Argyri (2010), Argyri *et al.* (2010) and Panagou *et al.* (2010).

In brief, fresh deboned beef fillets were purchased from a meat market in Athens (Greece) and transported under refrigeration to the laboratory within 30 minutes. Upon arrival, the samples were prepared by cutting the beef fillets into pieces with dimensions of 40mm wide, 50mm long and 10mm thick. After maintaining them for an hour at 4°C, the samples were subsequently placed in 90mm Petri dishes and stored in high-precision incubation chambers (±0.5°C) where they were left to spoil at 0, 5, 10, 15 and 20°C (Argyri, 2010). In case study 1, three experimental techniques have been employed; namely, FTIR spectroscopy, HPLC and e-nose.

2.2.1.2 Sensory Analysis

Sensory evaluation was performed during storage according to Gill and Jeremiah (1991) by a sensory panel of five trained staff members (staff from the laboratory) (Argyri, 2010). The assessment process was conducted under controlled conditions of light, temperature and humidity. Sensory assessment was based on the perception of colour and odour prior to and after cooking for 20 minutes at 180°C in a preheated oven, while taste was described solely/only after cooking (Argyri *et al.*, 2010; Panagou *et al.*, 2010; Argyri *et al*., 2013). A meat sample, stored at -20°C, freshly thawed and cooked, was presented to the panel as a reference sample.

Each sensory attribute was scored using a three-point hedonic scale. The samples were classified into three distinct categories: fresh, semi-fresh and spoiled samples. Fresh samples were characterised by bright colours, typical of fresh oxygenated meat, and the lack of any off-flavours (Papadopoulou *et al.*, 2011). For the semi-fresh samples, the formation of off-flavours was perceptible, but the samples were still considered of acceptable quality. Finally, a persistent dull or unusual colour, in addition to the presence of unacceptable off-flavours and putrid, sweet, sour or cheesy odours were considered indicative of microbial spoilage and the samples were classified as spoiled (Argyri *et al*., 2013). The integer values 1, 2 and 3 were used to describe fresh, marginal (semi-fresh) and unacceptable (spoiled) samples respectively. Score 1.5 was later introduced to indicate the first sign of meat spoilage.

Sensory panel scores were of the highest importance since they were used for the purposes of pattern recognition. However, a major drawback was the lack of homogeneity among the original sensory scores across the different instruments. For instance, the sensory scores provided along with the FTIR dataset consisted of three distinct classes; the dummy variables 1, 2 and 3 were used to represent fresh, semi-fresh and spoiled samples respectively. However, for other datasets such as HPLC and e-nose, five different classes of sensory scores were present. In this case, the number 1 was used for fresh, 1.5 for semi-fresh, in addition to 2, 2.5 and 3 that were used for spoiled samples. Therefore, the standardisation of the sensory scores prior to analysis was a necessity. More specifically, all 1.5 values were assigned to class 2 (semi-fresh), and values in the range between 2 and 3 to class 3 (spoiled). Finally, only three distinct levels of classification were retained for the analysis – namely 1, 2 and 3.

2.2.1.3 Fourier Transform Infrared (FTIR) Spectroscopy

FTIR analysis was undertaken using a ZnSe 45° ATR (Attenuated Total Reflectance) crystal on a Nicolet 6700 FTIR spectrometer equipped with a DLaTGS detector with KBr Window (Argyri, 2010). Measurements were conducted on a thin slice of the aerobic upper surface (8 x 1 x 0.5 cm) of the beef fillets that was excised and placed in intimate contact with the crystal (Argyri *et al.*, 2010; Panagou *et al.*, 2010).

The spectrometer was programmed to collect spectra in the mid-IR range between 4000 and 400cm^{-1}. Spectra over the wavenumber range 1500–1000cm^{-1}, which reveal the metabolic fingerprint of spoilage (Ellis and Goodacre, 2001), were extracted and subjected to smoothing using the Savitzky-Golay algorithm (Argyri, 2010). The final FTIR dataset comprises 76 samples and 255 spectra in total. Based on the provided sensory scores, these 76 samples are classified into 26 fresh (F), 16 semi-fresh (SF) and 34 spoiled (S) samples. Figure 2-1 shows the mean FTIR spectra per each distinct class within the fingerprint region.

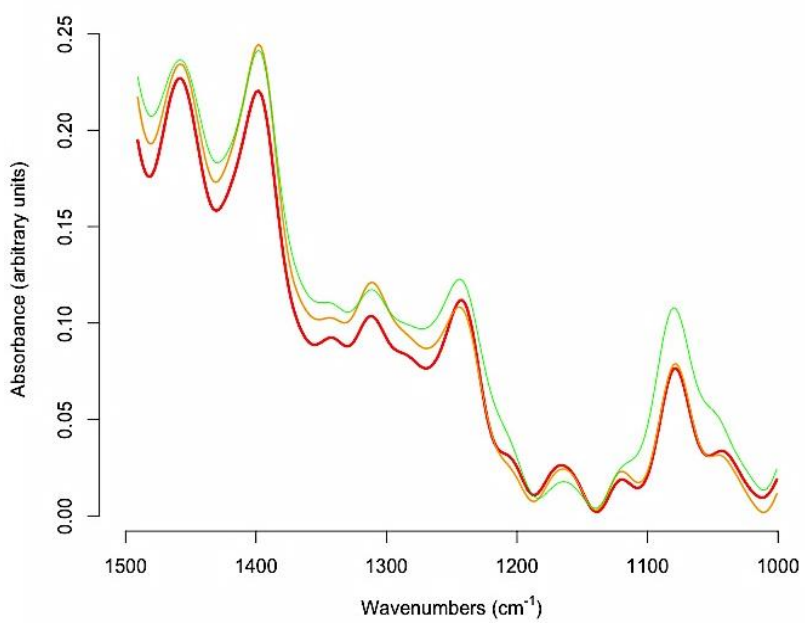


Figure 2-1 Mean FTIR spectra for case study 1 in the fingerprint region (1500-1000 cm^{-1})

The plot depicts the mean FTIR spectra for each distinct class of shelf life beef fillets (stored in air at 0, 5, 10, 15 and 20°C). The spectral region between 1500 and 1000cm^{-1} reveals the metabolic fingerprint of spoilage. Colour representation is used to identify the three classes as determined by the relevant sensory scores: fresh (red colour), semi-fresh (orange colour) and spoiled (green colour).

2.2.1.4 High Throughput Liquid Chromatography (HPLC)

The analysis was performed using a Jasco (Japan) HPLC equipped with a Model PU-980 Intelligent pump, a Model LG-980-02 ternary gradient unit pump and a MD-910 multi-wavelength detector (Argyri, 2010). The software used for the collection and the processing of the spectra in order to give concentrations of key metabolites was the Jasco Chrompass Chromatography Data system v1.7.403.1. Spectral data were collected from 200 to 600nm; however, chromatogram integration was performed at 210nm and the purity of the peaks was examined through the software using all spectral ranges. Solutions of oxalic, citric, malic, lactic, acetic, formic, tartaric, succinic and propionic acids were used as reference substances, analysed using the same programme, and their spectra were compared with the samples for the identification of the peaks (Argyri, 2010).

In total, 52 samples were analysed in duplicate. Based on the provided sensory scores, the HPLC samples consist of 20 fresh (F), 14 semi-fresh (SF) and 18 spoiled (S) samples.

2.2.1.5 Electronic Nose (e-nose)

The volatile profile of beef samples was determined using an electronic nose – the "Libra Nose" – by Technobiochip (Napoli, Italy) (http://www.technobiochip.com/). The instrument consists of an array of eight electronic chemical sensors (EU Patent [EP1505095]), as is depicted in Figure 2-2. In brief, 5g of sample were placed inside a 100 ml volume glass jar; the samples were left at room temperature (20°C) to enhance desorption of volatile and semi-volatile compounds from the meat into the gas phase. The headspace was pumped over the sensors of the electronic nose and the generated signal was continuously and in real time recorded to a personal computer.

In total, measurements of 36 samples with their replicates using eight sensors were provided; however, due to instability issues, one of the eight sensors had to be removed hence leading to a total of seven sensors. Based on the provided sensory scores, the 36 samples of e-nose consist of 10 fresh (F), 7 semi-fresh (SF) and 19 spoiled (S) samples.

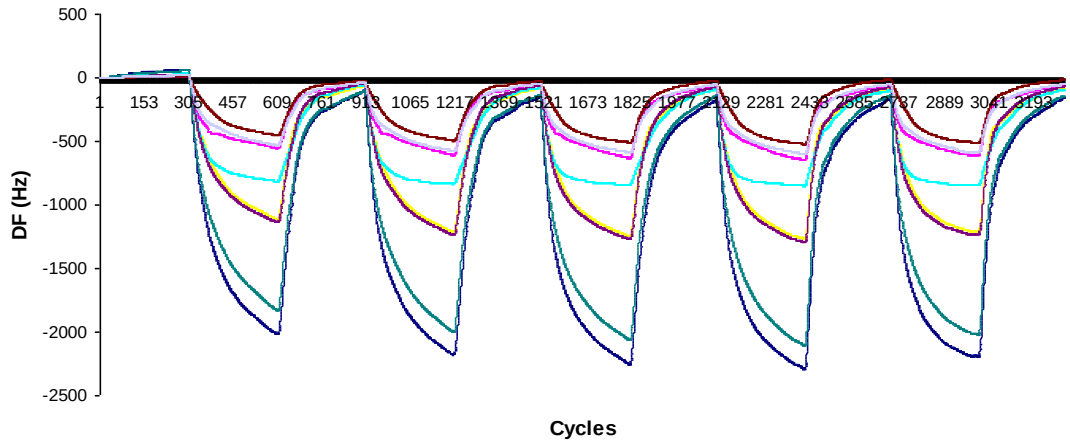


Figure 2-2 Sampling with Libra e-nose

The picture illustrates the responses of the eight sensors of the Libra electronic nose during sampling. The figure was provided by Argyri *et al.* (personal contact).

2.2.1.6 Data Overview

For each experimental technique, the total number of samples and variables as well as their data composition as described in the previous sections is summarised in Table 1.

Datasets	FTIR	HPLC	e-nose
Fresh (F)	26	20	10
Semi-Fresh (SF)	16	14	7
Spoiled (S)	34	18	19
Total #Samples	76	52	36
Total #Variables	255	18	7

Table 1 The sizes and data composition of standalone datasets from case study 1 prior to analysis

2.2.2 Data pre-Processing and Dimensionality Reduction

The first impediment that had to be overcome towards the construction of the multivariate analysis pipeline was the pronounced divergence of the datasets' dimensions. Based on the entries of Table 1, not all samples were analysed by every analytical technique. In order to acquire directly comparable results, the rows of each dataset were filtered according to the samples' names and sensory scores; thus, only the common samples were extracted and used throughout the pipeline.

In the case of shelf life beef fillets, the intersection process when all three experimental datasets are considered simultaneously is illustrated in the Venn diagram of Figure 2-3. By the end of this step, all the samples of the intersected datasets refer to corresponding objects. Based on this data intersection, the common samples for case study 1 consist of 10 fresh, 6 semi-fresh and 16 spoiled samples.

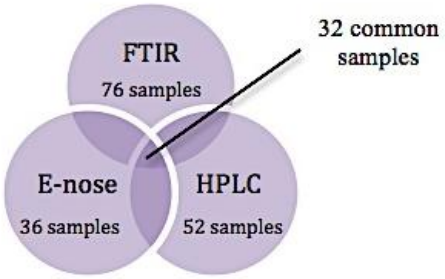


Figure 2-3 Data intersection

The figure demonstrates the samples' intersection based on the data integration of FTIR, HPLC and electronic nose. The samples' names and sensory values are cross-matched, and only the 32 common samples are extracted.

Once filtered, each standalone dataset is mean-centered, using Equation 1. Mean-centering merely aids the interpretation of the results; however, when applied prior to PCA, it ensures that the first Principal Component captures the maximum variance instead of describing the mean of the data. Due to substantial differences in their numerical ranges, the datasets were also standardised (auto-scaled) prior to analysis using Equation 2. After PCA, a predefined number of Principal Components is extracted from the PC scores of each dataset, which can be subsequently imported into a classification model or ensemble. HCA and k-means clustering were also investigated on the standalone pre-processed data in order to highlight any underlying patterns.

2.2.3 Standalone Classifiers: PLS-DA models with LOOCV

The first and simplest classification model to be investigated on standalone datasets was PLS-DA. The optimisation process in PLS-DA involves the identification of the optimum number of latent variables (LVs), for which the highest percentage of correctly classified samples ($\%CC$) is accomplished. As an initial approach, and in the presence of an insufficient number of samples (see Table 1) leave-one-out cross-validation (LOOCV) was applied directly on the entire dataset. The number of LVs was determined using a stepwise addition method, by gradually increasing in each iteration the number of LVs by one. Once the validation loop comes to an end, the average values of the predicted accuracies are calculated and the optimal number of components (LVs) is identified.

2.2.4 Ensemble of Classifiers

In an effort to enhance the overall accuracy ($\%CC$) of the classifiers, while simultaneously control the bias-variance trade-off and minimise the instances of overfitting (see Section 1.6.5), the use of ensembles of classifiers was also evaluated.

1. **Selection of the Classification Model**

First and foremost, the selection of the classification model to be applied had to be decided upon. There is no straightforward way of determining *a priori* which classification algorithm is the best; the selection of a classification model or kernel function highly depends on the problem under investigation. In cases where there is very little or no knowledge about the data under study, often more than one type of classifier may need to be tested. The choice of the classifier determines the hyperparameters to be optimised. In the case of PLS-DA, we are looking for the optimum number of latent variables (LVs), whereas in the case of RBF SVMs, the hyperparameters C and γ have to be optimised.

2. **Random split**

For a given input dataset *D*, a random fraction of samples is removed and kept aside as an independent test set during the training process of the model. This selection of samples forms the dataset D_{test}. This test set typically comprises a third of the original samples. Using a stratified holdout approach as described in Section 1.6.1, the test set consists of the same balance of sample classes as the initial dataset *D*. The remaining samples that are not selected, form the training set D_{train}. Since the test set is kept aside during the whole training process, the risk of overfitting is minimised (Ramadan *et al.*, 2006).

3. Validation Techniques

In the case of bootstrapping, a bootstrap training set $D_{bootTrain}$ is created by randomly picking n samples with replacement from the training dataset D_{train}. The total size of $D_{bootTrain}$ is equal to the size of D_{train}. Since bootstrapping is based on sampling with replacement, any given sample could be present multiple times within the same bootstrap training set. The remaining samples not found in the bootstrap training set make up the bootstrap test set $D_{bootTest}$. Similarly, for k-fold cross-validation, the initial dataset D is partitioned into k mutually exclusive folds; $k = 10$ (*10*-fold cross-validation) was employed according to Section 1.6.2. In each iteration, a single fold will be used to form the test set $D_{kfoldTest}$, while the remaining samples constitute the $D_{kfoldTrain}$. In the ultimate case of LOOCV, $D_{loocvTest}$ consists of a single sample, while the remaining samples form $D_{loocvTrain}$.

4. Hyperparameter optimisation

According to Section 1.5.2.3, nonlinear SVMs are usually considered a reasonable first choice. In the case of RBF models with bootstrapping, the SVMs are built and optimised using $D_{bootTrain}$ and $D_{bootTest}$ for different hyperparameter settings. More specifically, for each given combination of the hyperparameters C and γ, a new SVM model is trained with $D_{bootTrain}$ and tested with $D_{bootTest}$.

The most intuitive and fairly naïve approach for parameter selection involves an exhaustive grid-search over an extensive range of hyperparameters. However, this is an extremely time-consuming and computationally intensive procedure, even if there is more than adequate processor power. Therefore, in this work, the parameter search was implemented based on the approach suggested by Hsu *et al.* (2003), also described in Meyer *et al.* (2003), in a two-step approach using a combination of a coarse and fine grid-search. Initially, the values of C and γ increase exponentially with ranges equal to $C = [2^{-5}, 2^{-3}, \dots, 2^{15}]$ and $\gamma = [2^{-15}, 2^{-13}, \dots, 2^{5}]$ respectively. The combination of hyperparameters that gives the highest overall classification accuracy is recorded as optimal. Once an optimal region is located on the grid, a finer grid-search is conducted in the "neighbourhood" of good parameters.

Linear SVMs were also investigated since they generally provide accurate results, and are relatively easy and fast to train. For the linear kernel, only the regularisation parameter C needs to be optimised; similar to RBF SVMs, the values $C = [2^{-5}, 2^{-3}, \dots, 2^{15}]$ were investigated using a combination of loose and fine tuning.

To avoid reliance on one specific bootstrapping split, bootstrapping is repeated at least 100 times until a clear winning parameter combination emerges. Several methods can be used to determine the winning parameter; most commonly, the statistical average or the parameter that has most frequently been recorded as optimal is used. A similar approach is applied for the optimisation of the number of LVs in the case of PLS-DA. Since the datasets of case study 1 are classified into three distinct classes (fresh, semi-fresh and spoiled samples), the implemented models will constitute multi-class models.

5. Construction of the classification ensemble

Ultimately, the optimal parameters are used to train a new classifier with the full D_{train} dataset and test it on the independent test set D_{test}, which has been left aside during the entire optimisation process.

Even though the approach described thus far generates an excellent classifier, the random selection of test samples in the initial split may have been fortunate. For a more accurate and reliable overview, the whole process is repeated a minimum of 100 times, as illustrated in Figure 2-4, until a stable average classification rate emerges. The output of this repetition consists of at least 100 individual classification models built using the optimum parameter settings. Rather than isolating a single classifier, all individual classification models are fused into a classification ensemble.

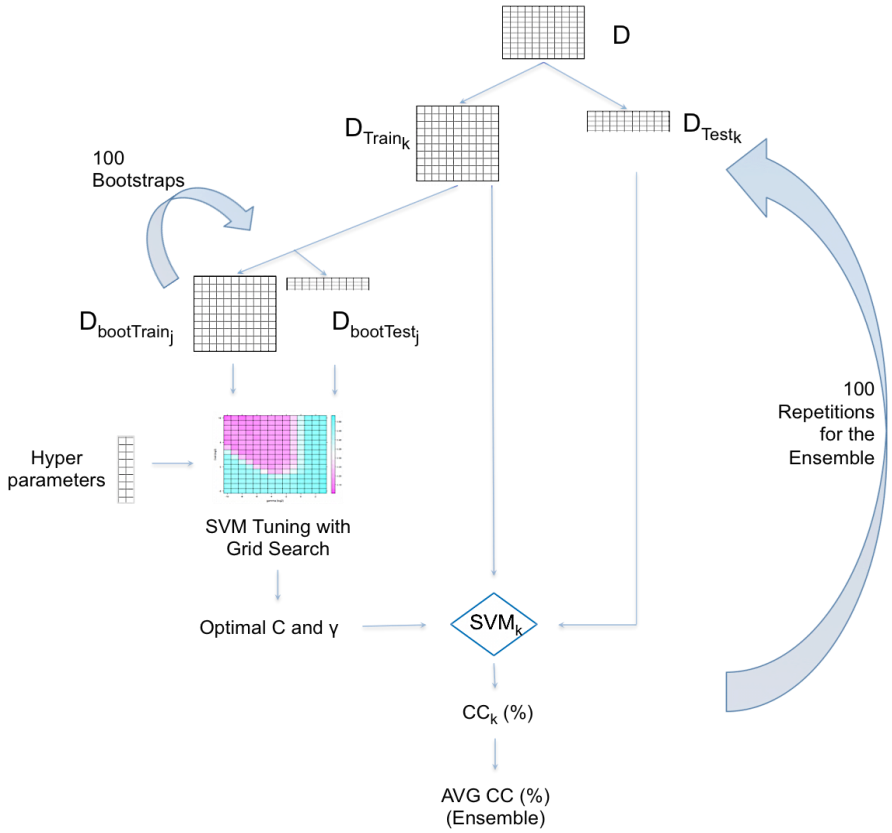


Figure 2-4 The process of constructing an ensemble of RBF SVMs optimised *via* bootstrapping

6. Permutation Tests

Permutation testing is a widely-applied process used in order to provide an indication of the statistical significance of the classification results. In a permutation test, the entries of the original class vector are randomly shuffled, while the class distribution is preserved. This approach destroys all the sample membership information since the samples of a permuted dataset correspond to randomly assigned classes (Westerhuis *et al*., 2008). The whole model building process as described in steps 2-5 is once more repeated for the "false" (permuted) classes. In general, permutation testing is performed at least 100 times until a stable distribution of results is obtained.

2.2.5 The Architecture

The application was developed on an Apple iMac under the operating system Mac OS X version 10.6.8, running on a 2.66 GHz quad-core Intel Core i5 processor and 4 GB memory.

2.2.6 Implementation in R

The steps in the multivariate analysis pipeline were implemented using the **R** platform (**R** Development Core Team, 2012; http://www.r-project.org/), a free open-source software environment for statistical computing and graphics. **R** offers its users the means of loading optional code, data and documentation; these optional sources that include a set of functions, libraries and integrated programming languages, are commonly referred to as packages. The packages can be easily installed and distributed, thus **R** provides great extensibility to its users. The methodology presented thus far was implemented based on the following packages.

The implementation of PCA is offered by a plethora of built-in and add-on **R** packages; among these, the **stats** package (**R** Core Development Team, 2012) provides the most commonly applied functions, namely **princomp()** and **prcomp()** for the implementation of PCA based on the NIPALS algorithm. Even so, a built-in script was produced that conducted PCA *via* SVD (see Section 1.4.1)

The **e1071** (Dimitriadou *et al*., 2010) package offers an **R** interface to the **LIBSVM** (Chang *et al.*, 2011) **C++** library. This package was not only used to build the SVMs, but also to perform optimisation of the hyperparameters based on the different types of kernels (linear, radial) using the simplistic approach of a grid search (Hsu *et al.*, 2010). The optimisation step may be conducted using the built-in function **tune()**, which applies a selected type of a validation algorithm such as cross-validation or bootstrapping as the optimisation method. In multi-class cases, the **e1071** package applies a 'one-against-one' approach (see Section 1.5.2.4).

The **plsgenomics** package (Boulesteix *et al.*, 2011) performs binary or multi-class classification *via* the **pls.lda()** function. The implementation is based on the algorithm described in Boulesteix (2004).

The **boot** package (Davison and Hinkley, 1997; Canty and Ripley, 2012) provides an interface to the original **S** library (http://statwww.epfl.ch/davison/BMA/library.html) for parametric and non-parametric bootstrapping and permutation testing.

2.3 Results and Discussion

2.3.1 Principal Component Analysis

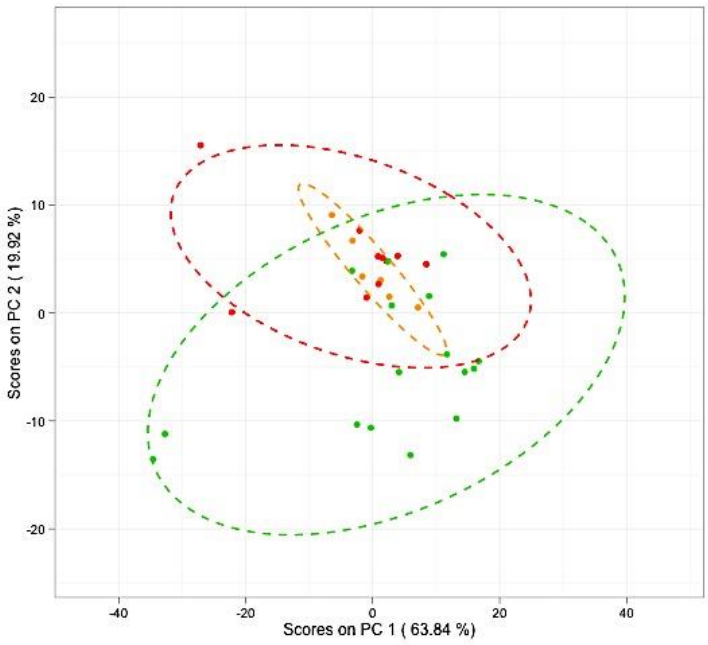
As an initial approach, PCA was employed on the standalone pre-processed data of case study 1. The percentage variance and cumulative variance for each successive PC is presented in Table 2. Successive PCs correspond to smaller percentage variance. In case study 1, it is noteworthy that the first three PCs of the FTIR and e-nose data account for at least 90% of the variance. In the case of e-nose the cumulative variance for the first three PCs even approximates 98%; thus, in this case we can graphically represent the former high-dimensional data using only two or three PCs without losing any valuable information. On the contrary, in the case of HPLC, the cumulative variance is significantly lower for the first PCs. The two-dimensional scores plots of Figure 2-5 are a powerful visual aid to assess the results of PCA and reveal any underlying patterns in the data. In addition, the scatterplots were enhanced with dynamically generated 95% confidence ellipses for each distinct class. The ellipses aimed to illustrate the density of the samples, highlight the formation of any clusters of samples deriving from the same class as well as identify any outliers.

	FTIR		**HPLC**		**e-nose**	
PCs	%Var	%Cum Var	%Var	%Cum Var	%Var	%Cum Var
PC1	63.84	63.84	35.96	35.96	80.26	80.26
PC2	19.92	83.76	19.04	55.00	15.14	95.40
PC3	5.70	89.46	9.12	64.12	2.19	97.59
PC4	4.40	93.86	7.36	71.48	1.34	98.93
PC5	2.73	96.59	6.66	78.14	0.64	99.57

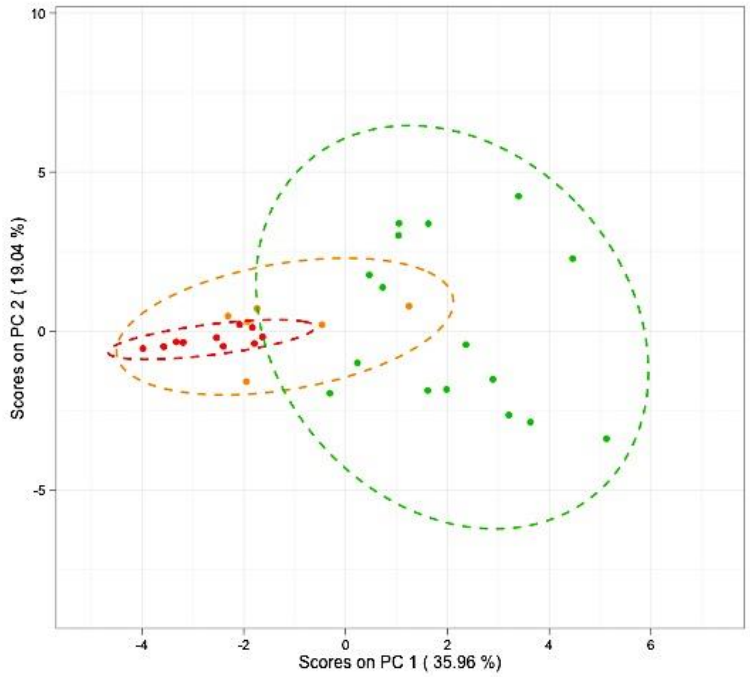
Table 2 PCA proportion and cumulative variance captured for the datasets of case study 1

It is noteworthy that even though HPLC presented the lowest variance for the first two PCs compared to the other two techniques, the PCA scores of Figure 2-5 demonstrate two well-defined clusters for the fresh and spoiled samples; the samples are linearly separable with the semi-fresh samples overlapping in between the other two classes. On the contrary, for the FTIR and e-nose datasets no well-defined clusters or linear separation between the samples of the different classes are obvious. The ellipses also enhance the detection of any outlying samples (outliers), such as the ones present in FTIR. Based solely on the outcome of this unsupervised method, HPLC appears to be the most discriminative technique. HCA and *k*-means clustering were also applied on the standalone pre-processed data in addition to PCA, however their results did not demonstrate any notable clustering and thus are not included.

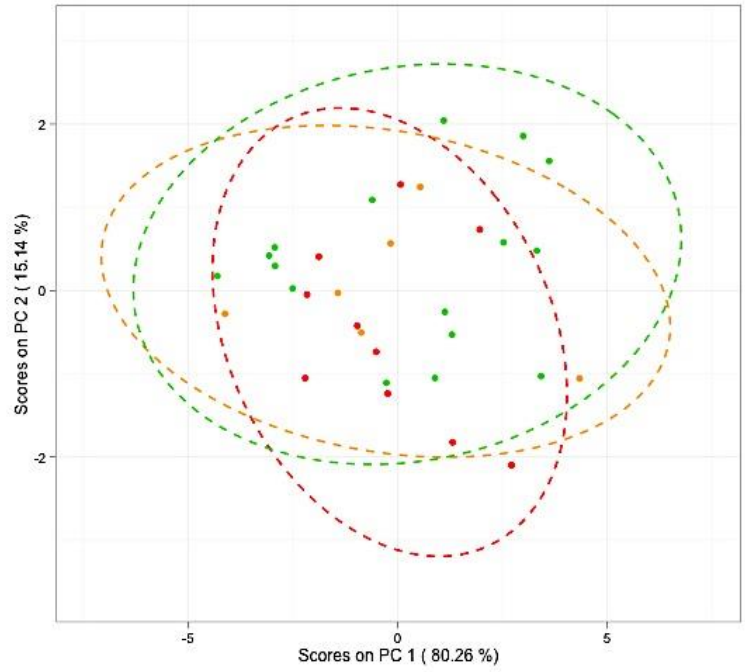
However, it is worth mentioning that the first few PCs do not necessarily contain the most discriminative information of the data (Schmid *et al.*, 2009). This is due to the fact that PCA is an unsupervised method that calculates the principal components based on the accounted variance, while it completely disregards any correlation with the classes. The outcome of PCA is hence prone to subjective interpretation. Thus, supervised learning techniques such as PLS-DA and SVMs are more likely to offer a better discrimination of the samples compared to PCA (Rossini *et al*., 2012)



(a) FTIR data



(b) HPLC data



(c) e-nose data

Figure 2-5 PCA score plots with 95% confidence ellipses for case study 1

The two-dimensional scatterplots illustrate the scores of the first two PCs. Dynamically generated 95% confidence ellipses per each class were added in the plots in order to highlight the presence of any clusters and/or outliers. Colour representation was used to identify the three classes as determined by the relevant sensory scores: fresh (red colour), semi-fresh (orange colour) and spoiled (green colour). For comparison purposes, only the 32 common samples (10 fresh, 6 semi-fresh and 16 spoiled samples) are depicted in each plot.

2.3.2 Classification Results

2.3.2.1 Overall Accuracies (*%CC*)

The prediction results of all implemented classification models are illustrated in the bar charts of Figure 2-6 as percentages of correctly classified samples ($\%CC$). The first model to be tested on raw standalone data was PLS-DA with LOOCV, which constitutes a single classifier, and not an ensemble of classifiers. According to Section 1.6.3, LOOCV is a nearly unbiased, albeit with a high variance, technique that may often lead to misleading results (Efron, 1983; Kohavi, 1995; Duan *et al.*, 2003; Glasmachers, 2008); in this instance, since the test set is used for both model construction and optimisation purposes, it may often lead to over-optimistic results. For case study 1, the overall accuracies of FTIR, HPLC and e-nose are equal to 63%, 84% and 59% respectively. Indeed, when the afore-mentioned percentages are compared to the accuracies of the other classification models of Figure 2-6, they appear to be overly optimistic.

In the case of the classification ensembles, HPLC clearly demonstrates the highest overall accuracies among the three instrumental techniques. The HPLC data appear to generate higher $\%CC$ in the case of SVMs, especially for linear SVMs, compared to PLS-DA. Even though PLS-DA and linear SVMs both construct linear decision boundaries, the difference in accuracies can be possibly explained by the fact that support vector machines conduct linear classification based on a different approach than PLS-DA. As stated in Brereton *et al.* (2009), the SVM boundary depends solely on the selected support vectors, while the remaining samples have no influence over it. On the contrary, methods such as PLS-DA use all available samples in order to determine the separating planes between the classes (Xu *et al.*, 2006). Finally, since the results of both linear and nonlinear (RBF) SVM ensembles approximate 80%, it can be concluded that the boundaries of the RBF SVMs may have been nearly linear, but still retain a wide margin (Brereton *et al.*, 2009); as stated in Section 1.5.2.3, "with a suitable combination of hyperparameters (C, γ), the testing accuracy of the RBF kernel is at least as good as the linear kernel" (Boser *et al.*, 1992; Keerthi and Lin, 2003; Hsu *et al.*, 2003; Chang *et al.*, 2010).

In the case of FTIR, linear classifiers (PLS-DA and linear SVMs) demonstrate higher overall predictions ($\%CC$) compared to nonlinear SVMs; in this instance, the overall accuracy of both PLS-DA and linear SVMs is equal to 59%, whereas the accuracy of RBF SVMs is notably lower. Since linear separation enhances the percentages of correctly classified samples, the application of nonlinear (RBF) SVMs was found to be unsuitable in this instance. Therefore, we can only assume that the data are relatively easy to separate in the input space using only linear models, and hence there is no necessity for a nonlinear projection into a high-dimensional feature space. Indeed, according to Xu *et al*. (2006), the relatively complex boundaries and formulation of kernel-based SVMs may not appeal very much in cases where the classes are nearly or completely linearly separable. Furthermore, according to Belousov *et al.* (2002), simplistic linear classification models such as PLS-DA may frequently outperform newer, more powerful classifiers.

Finally, the e-nose dataset returns poor results for every type of classifier. Based on the PCA scores plot of Figure 2-5, one can only assume that the widely scattered and overlapping e-nose data may request extremely complex boundaries to successfully discriminate the different classes; therefore, nonlinear SVMs is expected to outperform the remaining classifiers. Indeed, it is interesting to note that the ensemble of RBF SVMs performs significantly better than the ensembles of linear classifiers, obtaining an accuracy of 47%. In this case, the ensemble of PLS-DA demonstrated the lowest overall accuracy across all implemented models.

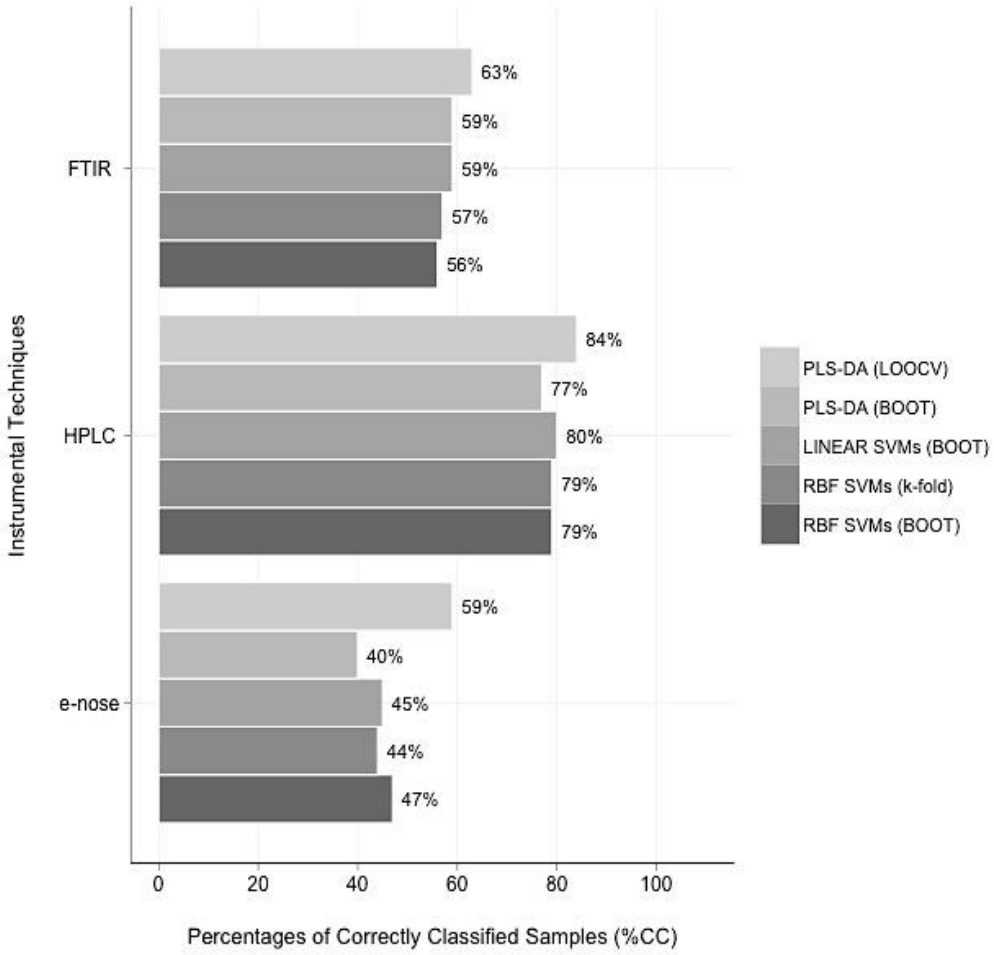


Figure 2-6 Overall accuracies (%CC) for the standalone datasets of case study 1

The figure illustrates the overall performance of all implemented classification ensembles on the standardised standalone datasets of case study 1. The bars represent the percentages of correctly classified samples (%CC) and are coloured according to the combination of classification model (PLS-DA, linear and nonlinear SVMs) and optimisation technique (LOOCV, *10*-fold cross-validation and bootstrapping). PLS-DA with LOOCV constitutes a single classifier, whereas the remaining models compose ensembles of classifiers. The overall accuracies have been rounded towards the nearest integer.

2.3.2.2 Class Prediction Accuracies

Even though the overall accuracy ($\%CC$) of a classification model is of the utmost importance, the success of a classifier may be also assessed by a plethora of other performance metrics. In a multi-class case, it is interesting to investigate class predictions, which determine how well a sample belonging to a specific class has been predicted. The percentages of correctly classified samples per class and instrument are summarised in Figure 2-7. The comparison of the class predictions presented in the figure confirms that the overall accuracies of a classifier may be occasionally misleading. For instance, even though the overall accuracies of a linear and nonlinear classifier may appear to be similar, a closer inspection of the class predictions may reveal significant differences among the different classifiers.

As presented in Section 2.2.2, the data intersection approach extracted for case study 1 a total of 32 common samples along with their respective sensory scores; these samples consist of 10 fresh (F), 6 semi-fresh (SF) and 16 spoiled (S) samples. In this instance, the spoiled samples constitute the majority class, whereas semi-fresh samples the minority class.

Even though HPLC generated the highest overall accuracies ($\%CC$) among the datasets of case study 1, FTIR presented the strongest and equally stable class predictions among all classification models, with noteworthy better rates for semi-fresh samples than the remaining techniques. The FTIR data presented better overall as well as per-class prediction rates for the linear PLS-DA classifiers *via* bootstrapping or LOOCV; this is verified by both the overall accuracies of Figure 2-6 and the class predictions of Figure 2-7. As thoroughly discussed in the previous section, it may be the case that the FTIR samples are linearly separable, thus, the complex boundaries of an SVM may not be able to separate them as accurately. Furthermore, it is notable that the classifiers, and especially the nonlinear SVMs, are relatively biased towards the majority class resulting in higher percentages of correctly classified spoiled samples.

In the case of HPLC, fresh and spoiled data are easily separable since their class prediction rates reach up to a maximum of 94% and 99% respectively. Also, the HPLC data present higher classification accuracies in the case of SVMs since the class predictions of fresh and spoiled samples exceed even those of the overoptimistic PLS-DA with LOOCV. Furthermore, semi-fresh samples are consistently difficult to predict; however in the case of HPLC they present higher class accuracies than the other two analytical techniques, presenting a maximum of 26%.

Finally, it is obvious that the classifiers in the case of e-nose hardly ever manage to correctly predict the semi-fresh (SF) samples since the SF class accuracies return a maximum of 1% for the complex nonlinear RBF boundaries. On the contrary, the classifiers mainly identify spoiled samples leading to class accuracies consistently over 60%; in the case of SVMs, the e-nose data present better results since the class prediction of spoiled samples approximates 90%. Thus, it appears that spoiled samples dominate all e-nose models to the point that all fresh and semi-fresh samples were misclassified as spoiled samples in the majority of cases. The fresh samples do present a better performance for PLS-DA models than SVMs but not nearly as good as PLS-DA with LOOCV.

Based on the documented class predictions obtained thus far, we can conclude that the majority of all misclassifications derive from the classifiers' inadequacy to correctly classify semi-fresh samples. In addition, as expected, all implemented classification models, but especially SVMs, are biased towards the majority class (spoiled samples), while they present high misclassification rates for the minority class (semi-fresh samples). The documented confusion matrices verify this hypothesis since the majority of misclassified samples were falsely predicted as spoiled.

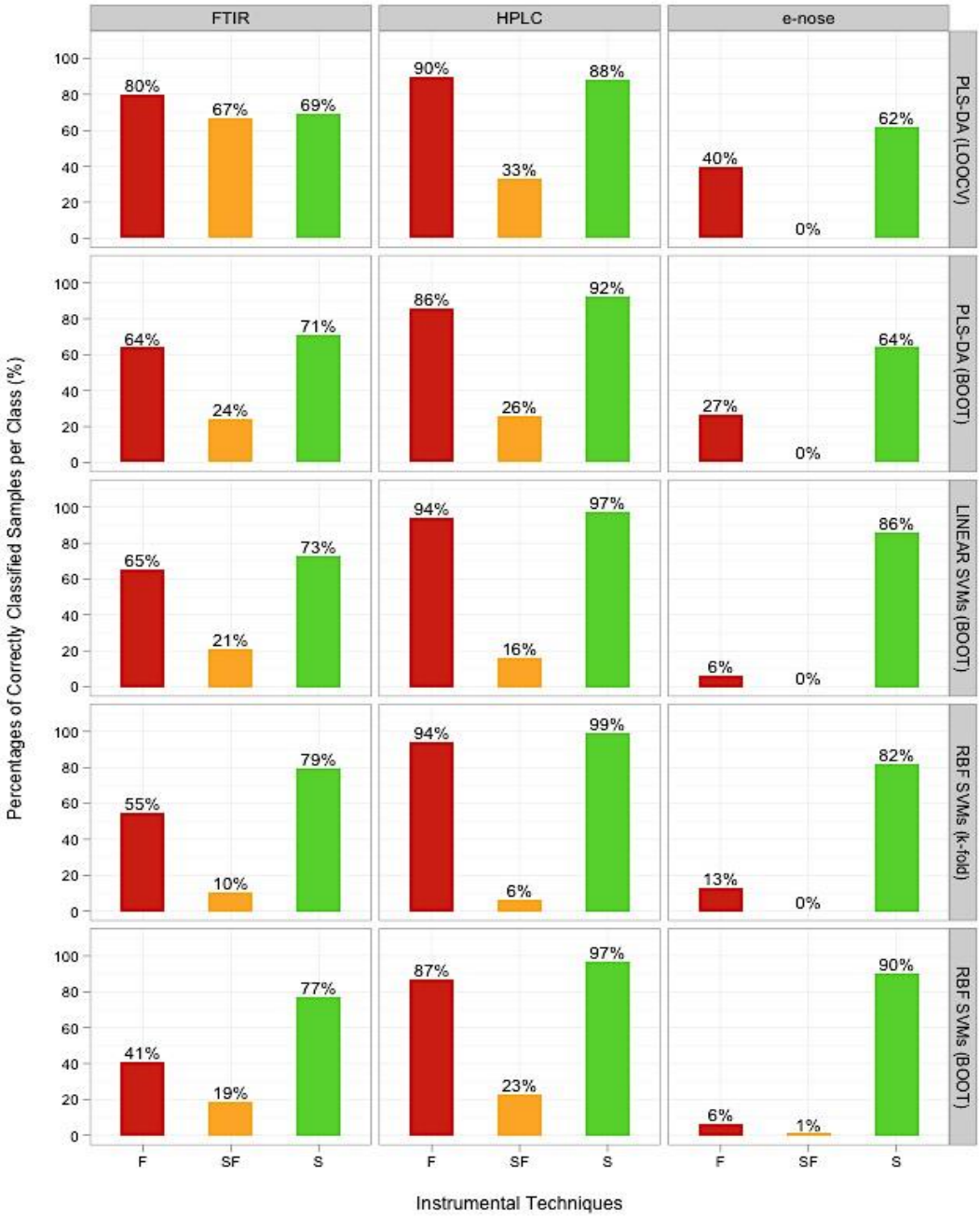


Figure 2-7 Class prediction rates of the standalone (prior to PCA) datasets for case study 1

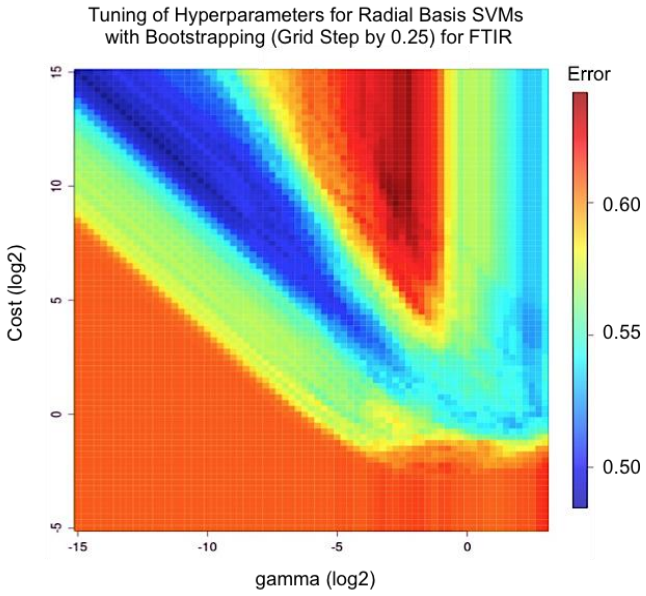
The figure illustrates the percentages of correctly classified samples per each individual class, when the auto-scaled standalone datasets of case study 1 are imported (prior to PCA) in the analysis pipeline. The class predictions are compared based on the instrumental techniques and classification models. Colour representation was used once more to identify the three classes as determined by the relevant sensory scores: fresh (red colour), semi-fresh (orange colour) and spoiled (green colour). All the class predictions presented in the figure are based on testing, preceded by thorough model training, using the optimal hyperparameters. The percentages have been rounded up to the nearest integer.

2.3.2.3 Comparison of validation techniques for the optimisation of the (RBF) SVM hyperparameters

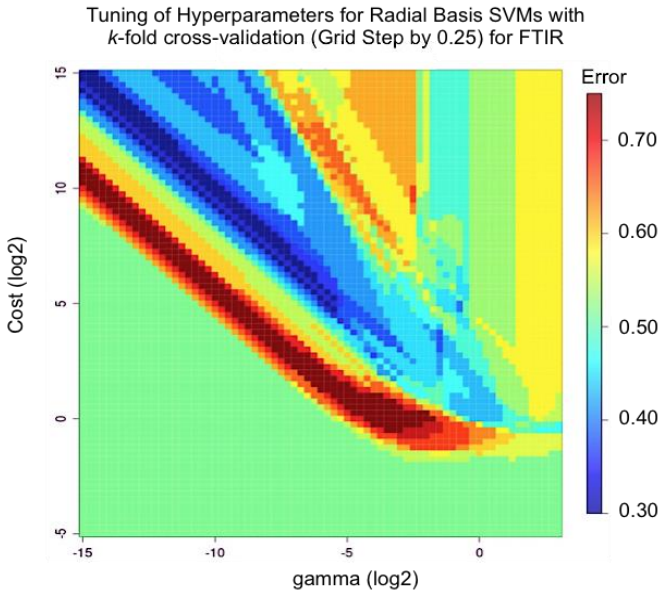
As thoroughly explained thus far, the prediction power and accuracy of a classifier is chiefly dependent on the optimisation (tuning) of its hyperparameters. The grid and surface plots of Figure 2-8 and Figure 2-9 aim at assessing the outcome of the optimisation process and defining the best technique. Simulations were initiated using a coarse grid-search; the grid resolution was gradually refined to a grid-step of $\log_2 0.25$, forming a final grid space of 6561 points as presented in Figure 2-8. For each combination of hyperparameters in the grid, bootstrapping, 10-fold cross-validation and LOOCV were applied on the training set.

According to Section 1.6.3, even though LOOCV is a nearly unbiased technique, it often presents cases of unacceptable high variance, especially when applied to relatively small datasets such as the ones of case study 1. In addition, LOOCV is a computationally expensive validation technique that may lead to long execution times and computationally prohibitive solutions for relatively large datasets (Boardman and Trappenberg, 2006). On the contrary, *10*-fold cross-validation, proved to be the fastest technique among the three. A great advantage of *10*-fold cross-validation is that all instances within a dataset are eventually used for both training and testing. However, since the outcome highly depends on the random split into folds, this approach may lead to training instabilities and relatively high variance (see Section 1.6.2). In addition, cross-validation proved to be extremely prone to overfitting, especially during the fine-tuning process. Therefore, this technique proved to be unreliable. Ensembles of SVMs were also optimised using bootstrapping. Fine grid-search proved to be extremely fruitful since all overall prediction accuracies increased by a minimum of 2%. As presented in Section 1.6.4, the probability for any given instance not being selected in a bootstrap set is approximately 36.8%; thus, bootstrapping minimises the chances of overfitting (Kohavi, 1995). Indeed, overfitting was no exception in this case either, however, it was only present in a miniscule number of instances. Even though bootstrapping is a fairly straightforward method, it constitutes a computationally demanding statistical procedure that leads to extremely long runs. The execution times may increase exponentially for larger datasets.

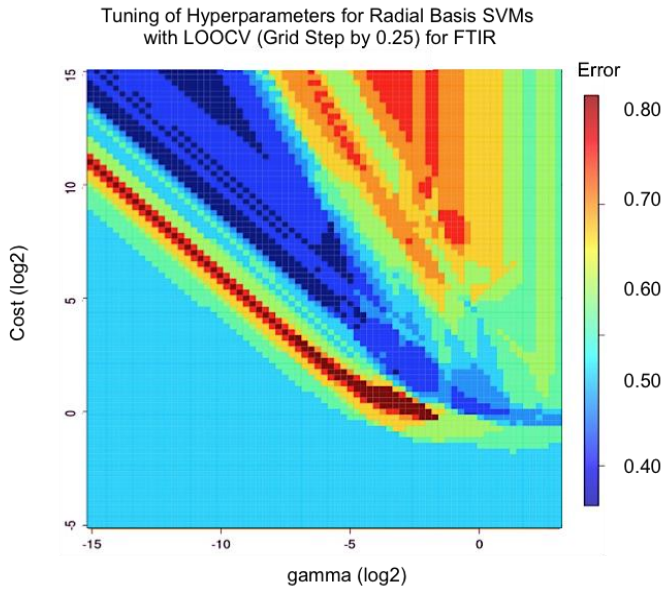
The topology of the three-dimensional surface plots of Figure 2-9 consists mainly of sharp peaks and flat plateaus. The flat plateaus correspond to regions of robust parameters that generate good results, whereas the sharp peaks are usually formed due to noise. In the case of both LOOCV and *10*-fold cross-validation, the surface plots of Figure 2-9 clearly demonstrate a great number of sharp peaks, which can be interpreted as local instabilities, possibly due to high variance. Furthermore, it is noteworthy that the rough surfaces of cross-validation, present more than one optimal solution, leading to many local optima. In the case of *10*-fold cross-validation, local extrema (minima or maxima) may by the effect of random partitioning of the training data since the performance strongly depends on the particular data split. LOOCV minimises the effects of local extrema *via* the complete evaluation of all permutations of the training set at each point in the parameter search (Boardman and Trappenberg, 2006). On the contrary, the surface plot of bootstrapping depicts a plethora of flat plateaus with stable and smooth transitions. Bootstrapping appears to minimise the variance that is obvious in the cases of cross-validation. Based on all documented results, bootstrapping can be considered the most accurate and stable optimisation method among the three, especially when dealing with relatively small datasets such as the ones case study 1.



a)



b)



c)

Figure 2-8 Comparison of the optimisation of the hyperparameters of RBF SVMs *via* bootstrapping, 10-fold cross-validation and LOOCV respectively

The grid plots demonstrate the dependence of the overall prediction error on the RBF hyperparameters C and γ for different optimisation (validation) techniques. The FTIR data were used for demonstrative purposes. The total number of hyperparameter combinations forms a grid of 6561 points. A colour scheme is used to emphasise the divergence in performance. Grid points with dark blue colour are considered optimal since they correspond to robust combinations of hyperparameters with low error; on the contrary, areas with dark red colour result in high error (rates) and hence are considered unacceptable.

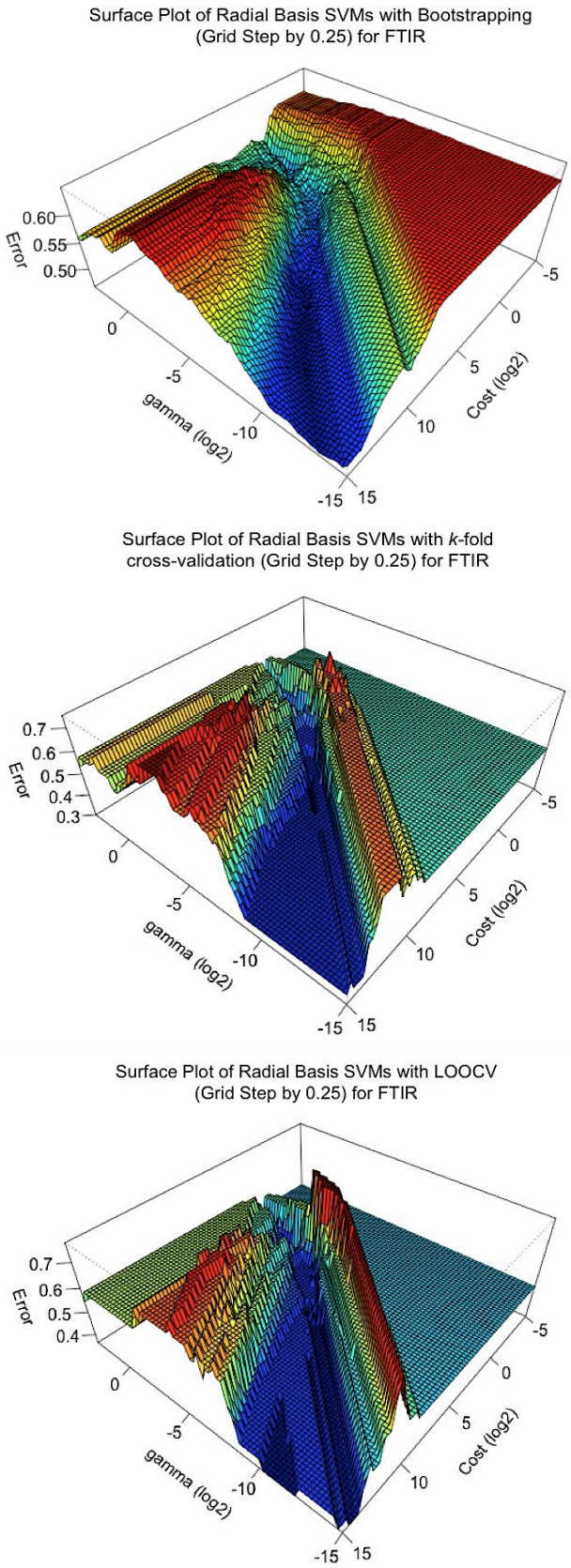


Figure 2-9 Three-dimensional error surface plots for the optimisation of the RBF parameters

The three-dimensional surface plots of this figure correspond to the two-dimensional grid plots of Figure 2-8. This visual aid makes it easy and straightforward to identify sharp peaks, flat plateau regions and local extrema.

2.4 Conclusion

This chapter presented the construction of a prototype multivariate analysis pipeline for the evaluation of meat freshness. The pipeline was developed on a single case study (case study 1) using the samples of shelf life beef fillets, stored in air at 0, 5, 10, 15 and 20°C. In this case study, data have been acquired from three analytical techniques: FTIR spectroscopy, HPLC and e-nose. In addition, the provided sensory scores classified the samples in three distinct classes: fresh, semi-fresh and spoiled samples.

As an initial step, unsupervised methods were applied on the standalone data for dimensionality reduction and the extraction of prominent features. The scores of the first two PCs from each standalone dataset were graphically represented in a two-dimensional scatterplot, which was further enhanced with 95% confidence ellipses for each class as a means of identifying any clusters and/or outliers in the data. However, only in the case of HPLC there was an obvious linear separation between fresh and spoiled samples, whereas the remaining analytical techniques presented highly overlapping samples.

In addition, the machine learning techniques that were employed include linear and nonlinear SVMs in addition to PLS-DA. Ensembles of individual classifiers were implemented to stimulate the prediction ability of single classifiers. Based on the obtained classification accuracies, the standalone classifier PLS-DA when optimised with LOOCV tends to exaggerate the classification performance and lead to misleading results. Among all implemented models, the ensembles of SVMs produced higher classification accuracies ($\%CC$) than PLS-DA. More specifically, the highest $\%CC$ was generated by standalone HPLC in the case of linear SVMs, and was equal to 80%. On the contrary, e-nose proved to be the technique with the least discriminative power resulting in poor results for all different classifiers. As far as the per-class accuracies are concerned, the semi-fresh samples were consistently difficult to correctly classify, whereas the classifiers, and especially SVMs, appeared to be biased towards the majority class (spoiled samples).

Furthermore, a thorough comparison between various validation techniques verified that LOOCV and k-fold cross-validation are prone to instances of high variance and overfitting. On the contrary, bootstrapping proved to be the most accurate and thorough technique; however, it often leads to extremely long execution times. Furthermore, in the case of nonlinear (RBF) SVMs, a two-step grid-search was employed as a means of identifying the optimal hyperparameters. The combination of bootstrapping with the computationally intensive grid-search does improve the overall classification accuracies ($\%CC$), however, it results in enormous time and computational costs. As a result, permutation tests were not feasible thus far due to the computationally expensive tuning approach. The optimisation of the SVM tuning process is therefore a necessecity.

3 Optimisation of the RBF SVM tuning process *via* bootstrapping

3.1 Introduction

This chapter introduces a new heuristic methodology for speeding up the optimisation process of the SVM hyperparameters *via* bootstrapping, chiefly focusing on the cases of SVMs with the RBF kernel. In this novel approach, a fast and robust approximation algorithm for constrained nonlinear optimisation, the Box complex algorithm, was used to replace the widely applied yet computationally intensive grid-search. The Box complex algorithm in addition to parallel programming was incorporated in the multivariate analysis pipeline as a means of significantly minimising the computational complexity and execution times as well as improving the overall performance of the classifiers.

3.2 Materials and Methods

The optimisation techniques presented in this chapter were implemented based on the datasets of case study 1 (see Sections 2.2.1 and 2.2.2).

3.2.1 Parallel Computing

With the exponential growth of computational power over the past decades, parallel programming is currently a fact. The fundamental idea of parallel computing is that p processors should be p times faster than a single processor. Among the plethora of available parallel architectures, the master/slave model is particularly popular and straightforward in its implementation (Hong *et al.*, 2005).

In the master/slave architecture, a single processor is randomly assigned to be the master, while the remaining processors constitute the slaves. Every parallel job in this model consists of a pre-processing, a slave and a post-processing task, which must be executed in this order (Sahni and Vairaktaris, 1996); all pre- and post-processing tasks are performed by the master, whereas the calculation tasks are carried out by the slave processors (Figure 3-1). More specifically, the master applies a "divide-and-conquer" approach by splitting any complex problems – commonly referred to as "embarrassingly parallel" problems – into smaller tasks, which are subsequently allocated to the slave processors. Since all the individual subcalculations are independent, no communication is needed between the slave processors. Once the execution of the subtasks has been completed, the master collects any responses and partial results back from the slave processors in order to produce the final output.

The number of parallel tasks is usually equal to the number of available processors. Even though one would expect that p available slaves would result to a p times speedup, this is not attainable in practice since computational tasks vary greatly in complexity and size as demonstrated in Figure 3-1. Furthermore, the communication and exchange of data between the master and the slaves in addition to the aggregation of results by the master, adds significant delays and overhead (Vera *et al.*, 2008). Even so, the appropriate distribution of tasks to the slaves may result in substantial gains towards efficiency, performance and scalability.

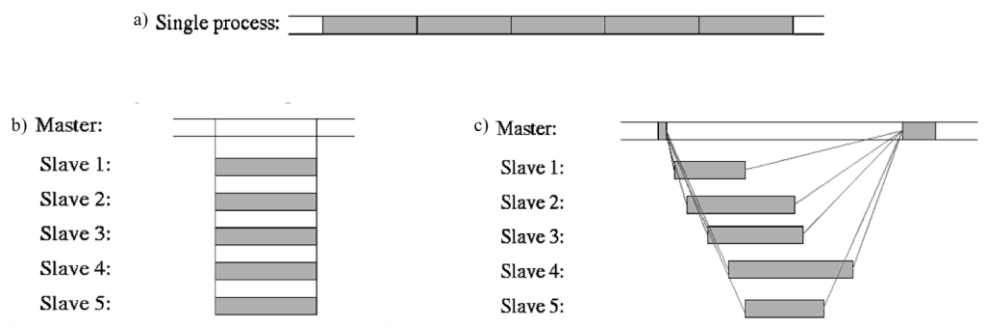


Figure 3-1 Master/Slave architecture

The figure illustrates three distinct cases: a) an "embarrassingly parallel" problem executed in sequence using a single processor, b) the ideal case, where the problem is executed p times faster with p slaves, c) a real situation, where the tasks vary both in size and execution time. The figures have been adapted from Tierney (2003), available at http://homepage.stat.uiowa.edu/~luke/talks/uiowa03.pdf.

In this research, computationally demanding tasks such as model evaluation and optimisation, also eminently time-consuming techniques such as bootstrapping, would greatly benefit from the application of parallel programming. The analysis pipeline that was presented in Section 2.2.4 involves several "embarrassingly parallel" iterations as summarised in Figure 3-2. Since these tasks run independently of each other, a divide-and-conquer parallelisation has been indeed feasible. There are several communication mechanisms that allow computational tasks to run cooperatively in parallel across a single multicore machine. In this work, the Message Passing Interface (MPI) (Gabriel *et al.*, 2004) *via* the **Rmpi** (Yu, 2011) interface is employed.

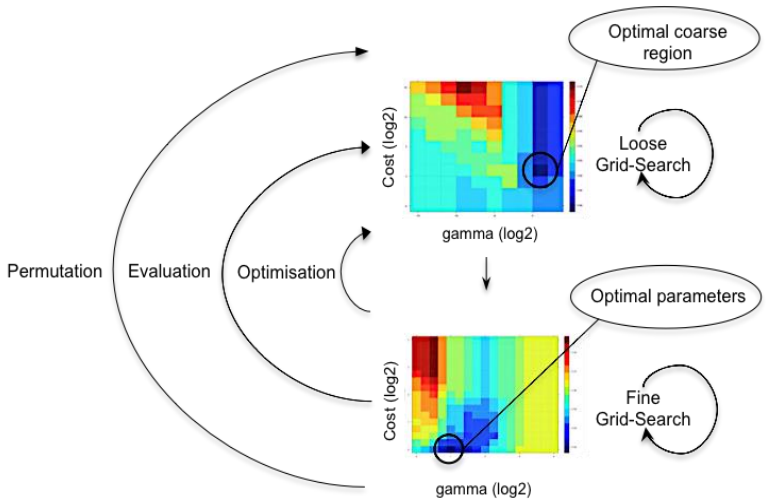


Figure 3-2 Embarrassingly parallel problems in the analysis pipeline

The figure demonstrates all the iterations running in sequence within the analysis pipeline. Each of these problems runs independently, thus it can be easily parallelised.

3.2.2 Approximation Algorithms

Based on the overall accuracies of Figure 2-6, the ensembles of RBF SVMs were noticeably the most promising classification method. The generalisation performance of a nonlinear (RBF) SVM is greatly dependent on a suitable selection of C and γ. Therefore, a two-step grid-search approach was initially applied as a means of hyperparameter optimisation (see Section 2.2.4). A grid-search over an arbitrary range of parameter values is an intuitive but computationally intensive technique, to the point that even parallel programming proved to be inadequate. In addition, the

precision of the classification accuracy is subject to the predefined grid resolution (Boardman and Trappenberg, 2006). Furthermore, the applicability of grid-searches is limited since they allow the simultaneous assessment of only two hyperparameters; therefore, they cannot be employed in cases where several hyperparameters need to be optimised simultaneously. Thus, the use of high-level approximation algorithms may help significantly minimise the execution times and the computational complexity. In general, there are two main categories of constrained nonlinear optimisation algorithms; namely, gradient-based algorithms and direct search methods. The function $f: \mathbb{R}^n \rightarrow \mathbb{R}$ to be optimised, whether constrained or unconstrained, is commonly referred to as the objective or cost function.

A reasonable first approach would be to employ the widely popular gradient descent methods for the optimisation of nonlinear SVMs. Gradient descent methods use either the first derivative (gradient) or the second derivative (Hessian) of a given cost function in order to identify the optimal point. However, such algorithms misbehave in the presence of local extrema. The local optima (minima or maxima) and the occasional isolated peaks that were presented in the surface plots of Figure 2-10 can capture or deceive the gradient-based search algorithms (Staelin, 2003; Boardman and Trappenberg, 2006). In addition, the extremely flat plateau regions near the periphery of the plots may also confuse gradient-based algorithms since the search is more vulnerable to small errors or noise in the gradient estimation (Staelin, 2003). Thus, gradient descent methods are found to be impractical in this research.

Another category of widely applied optimisation algorithms is the direct search optimisation methods (Kolda *et al.*, 2003). The main attribute of direct search methods is that they seek out the minimum of a given objective function using only function values at given points of the function, and do not attempt to form an approximate gradient at any of these points. No derivatives of the cost function are required, which makes the algorithm very tolerant to noisy problems. Thus, these robust algorithms may prove to be best-suited in the tuning process of complex SVMs.

3.2.3 Nelder-Mead Simplex Algorithm

The Nelder-Mead algorithm (Nelder and Mead, 1965) is the most widely employed direct search method for unconstrained nonlinear optimisation (Lagarias *et al.*, 1998; Kolda *et al.*, 2003). The algorithm is commonly referred to as the Nelder-Mead simplex; however, it greatly varies from the popular simplex algorithm by Dantzig (Dantzig, 1987), which is solely used for linear programming. The Nelder-Mead method constitutes an extension of the simplex-based algorithm initially proposed by Spendley, Hext and Himsworth (Spendley *et al.*, 1962).

The Nelder-Mead algorithm approximates the optimal point by assessing the values of a given nonlinear cost function $f: \mathbb{R}^n \rightarrow \mathbb{R}$, without using any implicit or explicit derivative information (Kolda *et al.*, 2003). In comparison to the fixed shape simplex of Spendley *et al.* (1962), the Nelder-Mead algorithm is based on a variable shape simplex, which allows adaptions both in size and shape. "A simplex S in $\mathbb{R}^n$ is a geometric figure (polytope) in n dimensions of nonzero volume that is the convex hull of $n+1$ vertices" where n is the total number of variables (Lagarias *et al.*, 1998); for example, a simplex in $\mathbb{R}^2$ has the form of a triangle (Nelder and Singer, 2009). In order to identify the optimal point (the optimal solution), the Nelder-Mead algorithm constructs a series of new simplices in an iterative manner. In each iteration, the vertices of the simplex are sorted according to their cost function values. The algorithm attempts to replace the worst vertex with a new point, which depends on the worst point and the centre of the best vertices using four possible steps: reflection, expansion, (outside and inside) contraction and shrink (Lagarias *et al.*, 1998) as illustrated in Figure 3-3. A thorough review and step-by-step explanation of the Nelder-Mead simplex methodology can be found in Lagarias *et al.* (1998) and Nelder *et al.* (2009).

The greatest appeal of the Nelder-Mead direct search algorithm according to Nelder and Singer (2009) is that it easily "adapts itself to the local landscape" such as the three-dimensional surface plots of Figure 2-9; the simplex is elongating itself down long slopes, it alters direction when encountering a valley at an angle, and it contacts as it approximates the minimum (Nelder and Singer, 2009).

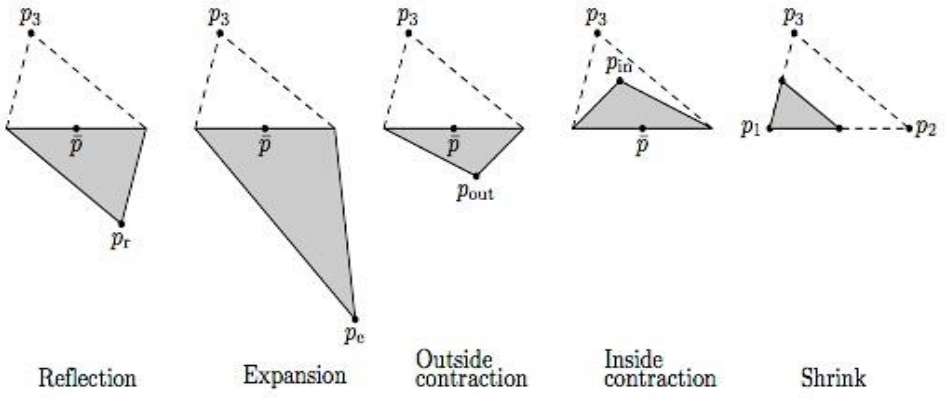


Figure 3-3 The steps of the Nelder-Mead algorithm

The Nelder-Mead steps include: reflection, expansion, (inside and outside) contraction and shrink. The figure was extracted from http://www.math.uiuc.edu/documenta/vol-ismp/42_wright-margaret.pdf

3.2.4 Box Constrained Simplex Algorithm

The Box complex algorithm (Box, 1965) is a derivative-free method used for nonlinear constrained optimisation. Unlike the other direct search methods of Section 3.2.3, which are solely applied to unconstrained optimisation problems, Box extended the functionality of the algorithm by Spendley *et al.* (1962) by explicitly incorporating bounds and/or nonlinear inequality constraints into the search space via a constrained simplex, namely the "complex". Similar to the Nelder-Mead simplex, a complex S in $\mathbb{R}^n$ is a flexible mathematical figure in n dimensions made up of at least $m \geq n+1$ vertices, where n is the total number of variables; the use of $m = 2n$ vertices is recommended by Box. Each point's coordinates in the complex correspond to individual variables of the objective function. The complex moves around the solution space by expanding or contracting in any direction as long as it is feasible. The Box complex algorithm solves the following constrained minimisation problem

$$\begin{aligned} &\min f(x) \quad \text{subject to:} \\ &\ell_i \leq x_i \leq u_i, \quad i = 1, n \\ &g_j(x) \geq 0, \quad j = 1, m \end{aligned}$$

Equation 19 Box complex algorithm

Where $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is the cost function, x is the vector of parameter estimates, $\ell_i, u_i \in \mathbb{R}^n$ are the lower and upper parameter bounds (explicit constraints) respectively, n is the total number of parameters and m the total number of positive nonlinear implicit constraints $g(x)$. The approximation algorithm of Box complex performs four main steps towards identifying the optimum solution: reflection, expansion, contraction and shrinkage. A detailed description of the Box complex steps is provided in Box (1965).

3.2.5 Implementation in R

The **Rmpi** (Yu, 2011) package, which provides the MPI interface for **R**, was used as a means of implementing parallel programming. The package allows the creation of **R** scripts that run cooperatively in parallel across multiple machines, or multiple CPUs on one machine. The **Rmpi** backbone is based on the master/slave model that was presented in Section 3.2.1.

In addition, the **neldermead** (Bihorel and Baudin, 2012) package was used for the implementation of the Box complex algorithm. The package comprises an **R** port to the **neldermead** module originally developed for **Scilab** (Chancelier *et al*., 1990; http://www.scilab.org), which provides a set of constrained and unconstrained direct search optimisation algorithms based on the simplex methodology.

3.3 Results and Discussion

3.3.1 Linear Models

As an initial step towards optimising the analysis pipeline, parallel programming was applied to all implemented classification models. Even though the speedup of fairly simple optimisation techniques such as cross-validation is of interest, the chief aim is to examine the speedup of computationally tedious techniques such as bootstrapping. Parallel programming reduced the execution times to at least a third of the initial times as demonstrated in Figure 3-4.

According to the figure, the relationship between the execution times and the number of slave processors is deviating from an ideal linear speedup. More specifically, there is a noteworthy improvement, represented by a big slope in the plot, from sequential (one processor) to parallel programming with two processors. Thus, the "divide-and-conquer" approach indeed provides a significant speedup. For the HPLC and e-nose data, the recorded speedup is minimal after a certain point. Occasionally, for short calculations and/or small datasets such as the case of e-nose in Figure 3-4, the communication between the master and the slaves becomes so computationally expensive that the overhead forces the increase rather than the decrease of execution times after a certain (achieved) speedup; in this case, no further parallelisation is feasible. Thus, it is interesting to note that relatively large datasets such as FTIR, which demand more complex and time-consuming calculations, benefit to a greater extent from parallel programming than small datasets.

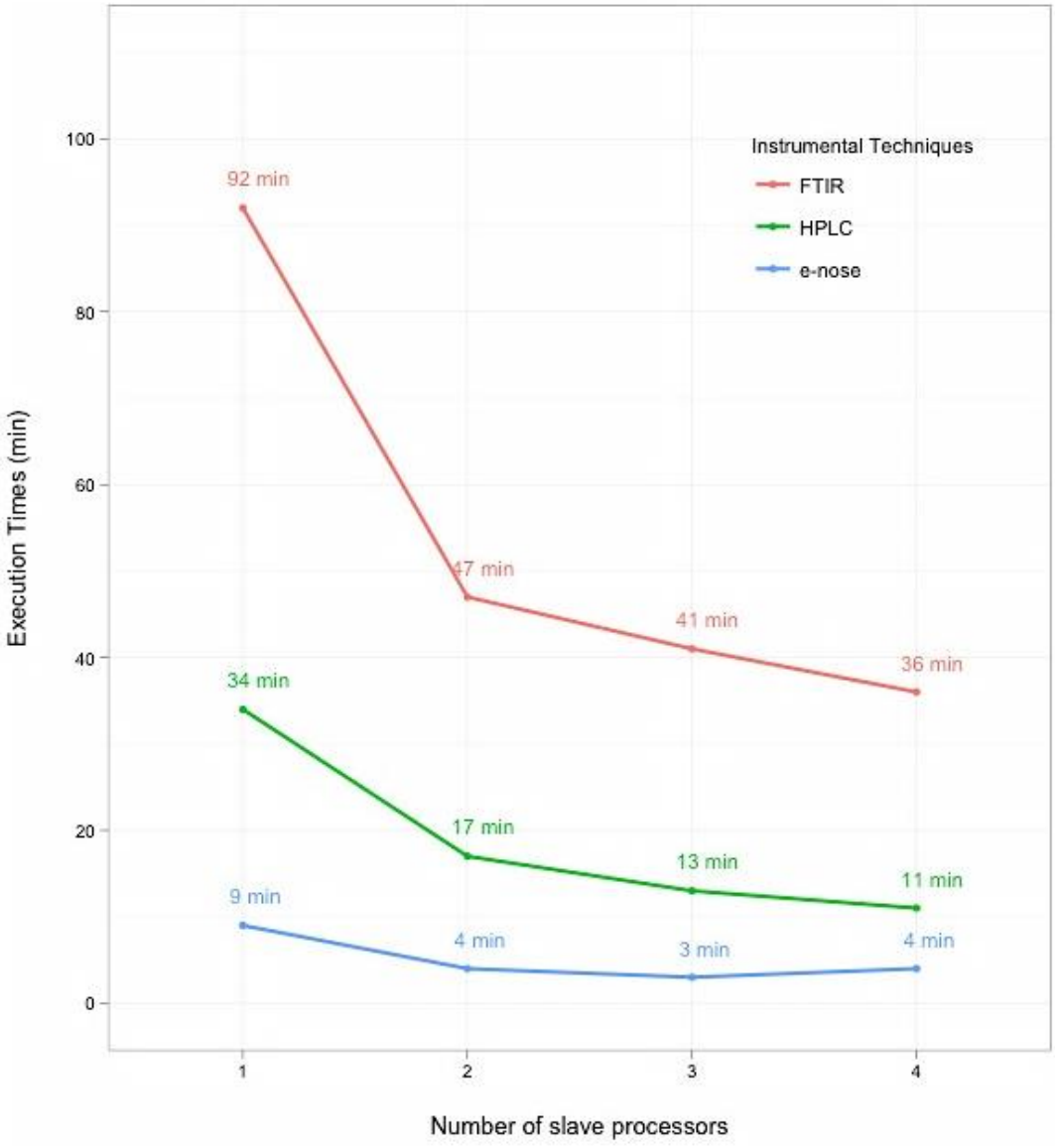


Figure 3-4 The relationship between the number of slave processors (master/slave model) and the execution times of an ensemble of PLS-DA with bootstrapping

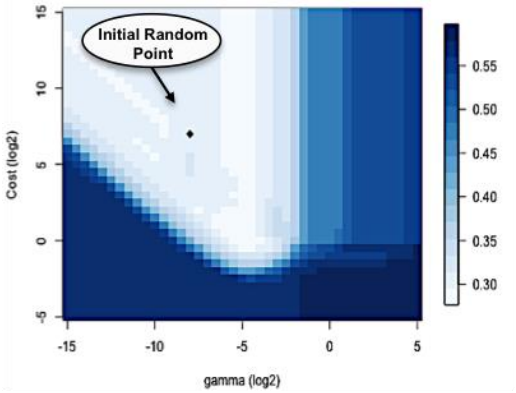
The figure illustrates the decrease in the execution times based on the number of slave processors. In this case, the master/slave architecture is employed to parallelise the bootstrapping iterations. It is obvious that the overall execution times have significantly improved with the application of parallel programming.

3.3.2 Nonlinear Models

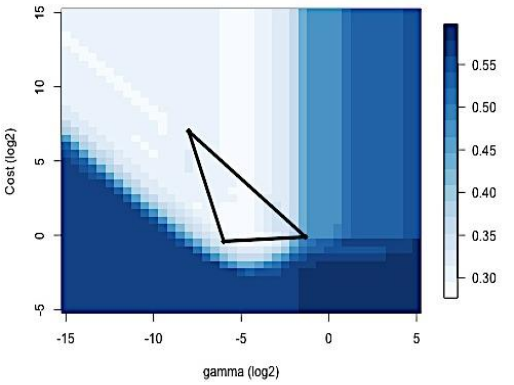
In order to test the performance and the applicability of the new heuristic for real-world cases, a simulation based on the methodology presented thus far was conducted using the HPLC dataset. The simulation aims towards the comparison between the grid-search and the Box complex algorithm for the optimisation of ensembles of nonlinear SVMs (RBF) *via* bootstrapping. In order for all results to be directly comparable, the exact same train, test and validation datasets were used for both algorithms. The algorithms are assessed based on their average train and test accuracies as well as the execution times.

In Section 2.2.4 a combination of a coarse grid-search followed by a finer grid-search was proposed as a means of optimising the RBF hyperparameters (C, γ). The percentages of correctly classified samples ($\%CC$) of these models were presented in Figure 2-6. In addition, a new ensemble of SVM classifiers was optimised and tested using an exhaustive grid-based search with a refined resolution of $\log_2 0.5$, which allowed greater grid granularity. The Box complex algorithm was also employed for the minimisation of the average bootstrapping test error during the training process of the SVMs. In this case, the inequality constraints correspond to the minimum and maximum predefined value boundaries that were set for the RBF hyperparameters (C, γ) in Section 2.2.4, where $\log_2 \gamma \in \{-15,\ 5\}$ and $\log_2 C \in \{-5, 15\}$. The formation of the initial complex begins with the selection of a random feasible point that must satisfy the minimum and maximum hyperparameter constraints as presented in Figure 3-5.

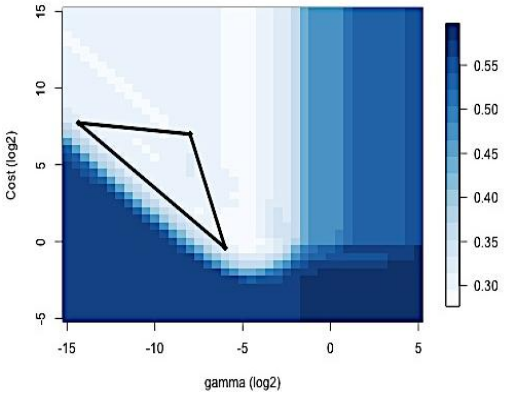
A classification model has been selected at random out of the ensemble of classifiers (100 independent classifiers) for demonstrative purposes as a means of visually assessing the optimisation outcome of the two techniques. Figure 3-5 illustrates step-by-step the Box complex simplices towards finding the optimal combination of hyperparameters. In addition, the high-resolution grid plot that constitutes the background of each figure, derives from the grid-search optimisation with a resolution equal to $\log_2 0.5$; each grid point corresponds to an average training error of 100 independent bootstrap iterations for a predefined combination of hyperparameters.


Initial Random Point
Cost (log2)
gamma (log2)

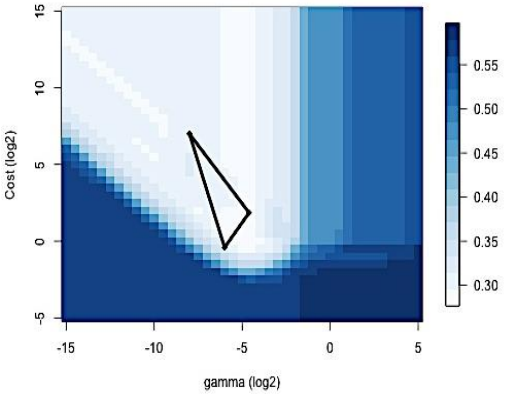
1)


Cost (log2)
gamma (log2)

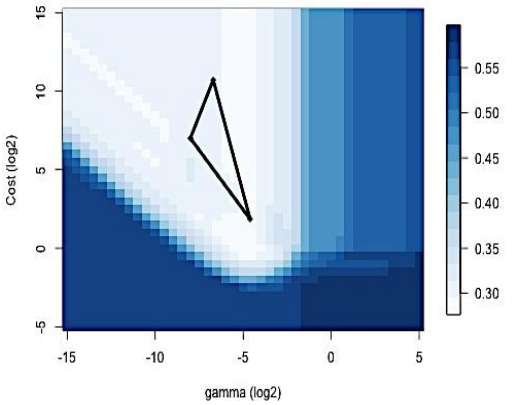
2)


Cost (log2)
gamma (log2)

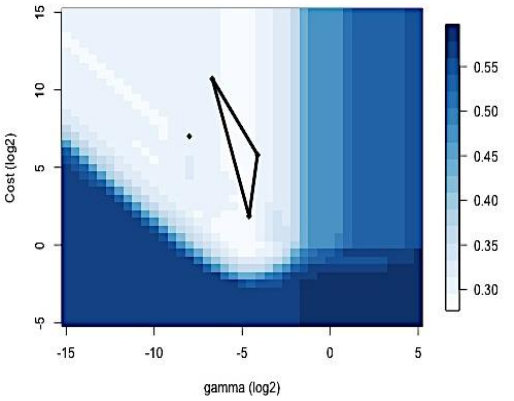
3)


Cost (log2)
gamma (log2)

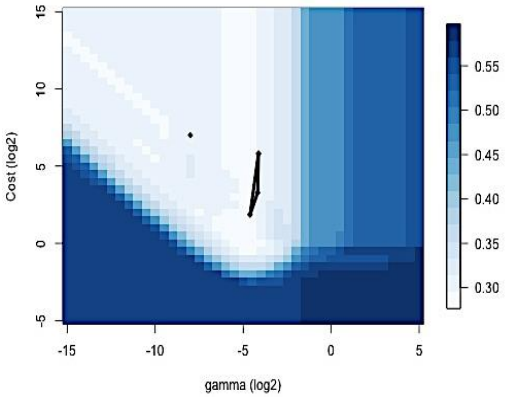
4)


Cost (log2)
gamma (log2)

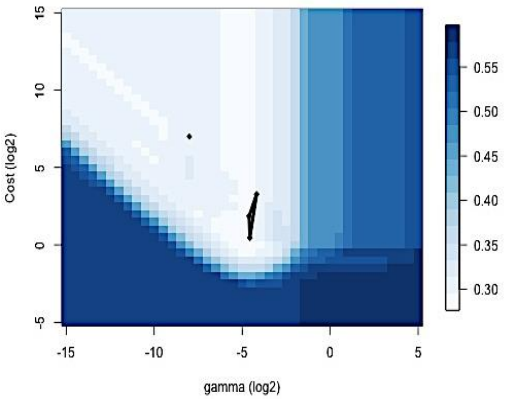
5)


Cost (log2)
gamma (log2)

6)


Cost (log2)
gamma (log2)

7)


Cost (log2)
gamma (log2)

8)

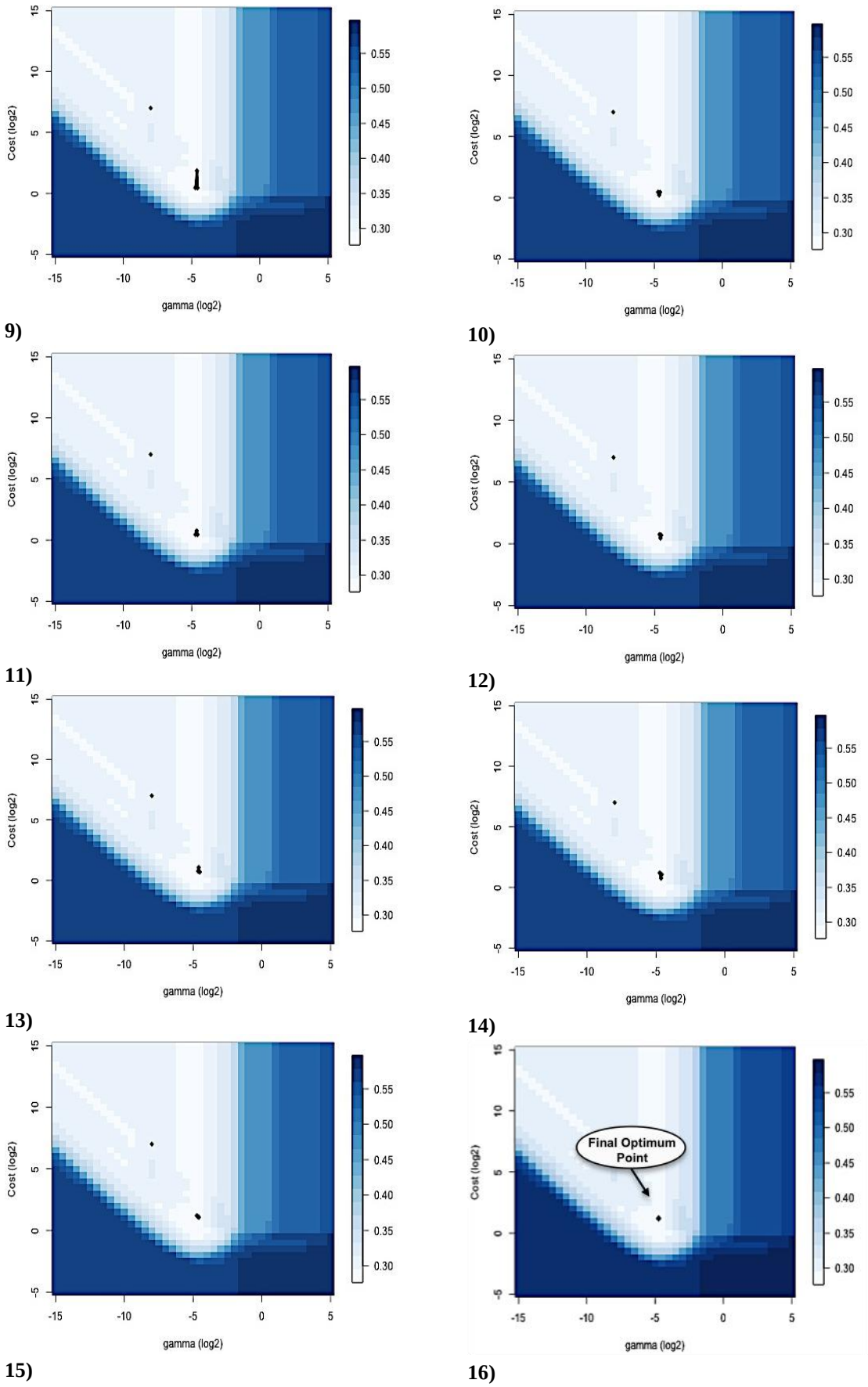


Figure 3-5 Step-by-step representation of the Box complex algorithm towards identifying the optimal hyperparameters and the minimum bootstrapping error (HPLC data)

In the example of Figure 3-5, the Box complex algorithm performed 14 iterations and 54 function evaluations in total before successfully identifying the optimal combination of hyperparameters. The initial combination of C and γ produced an average bootstrapping test error equal to 0.31. After the application of the Box complex algorithm, the bootstrapping error decreased to 0.28. Based on the plots, the simplices become extremely small as they contract towards the minimum. In the final plot, no further improvement can be performed. Based on the graphs of Figure 3-5, we can conclude that the optimal combination of hyperparameters is indeed identified within robust areas of the grid.

The optimisation results of Figure 3-5 derive from a single classifier. For the entire classification ensemble (100 individual classifiers), the optimal hyperparameters as selected by the Box complex algorithm are illustrated in Figure 3-6. In order to highlight any underlying patterns, the plots include contours of density estimations. It is interesting to note that only three out of 100 points are further apart from the rest and may be located in unacceptable regions with high prediction error.

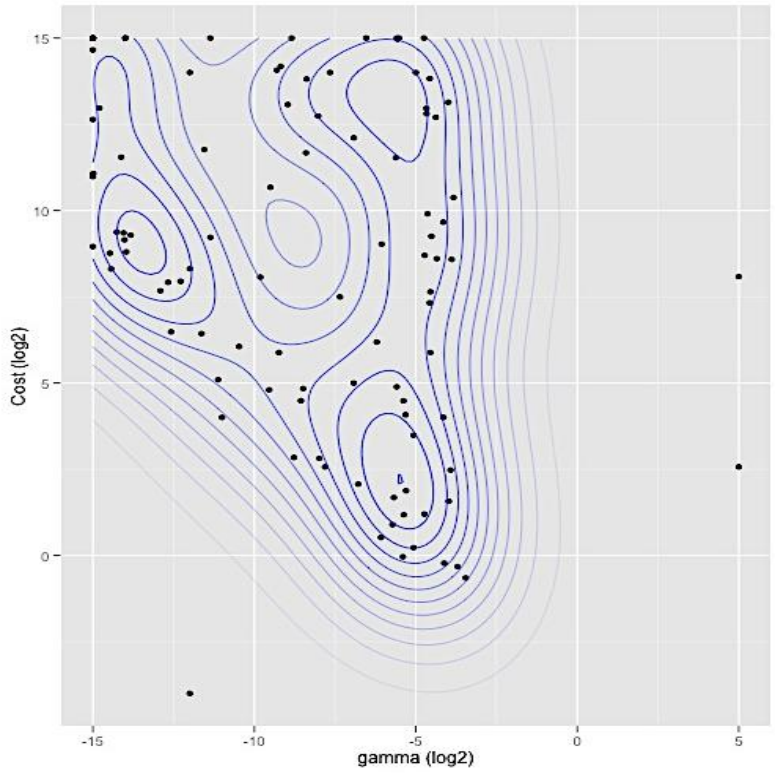


Figure 3-6 Contour plots of the density estimation of the optimal hyperparameters as defined by the Box complex algorithm

As an additional visual aid, these optimal points are plotted once more in Figure 3-7 over a grid plot of the average performances of all 100 optimised classifiers as produced by the exhaustive grid-search. Once again, it is obvious that, barring three instances, all selected hyperparameters by the Box algorithm are located in robust regions on the grid. Based on these rare instance, we can conclude that even though the algorithm is robust since it is tolerant to noisy problems, it does suffer from one limitation; it is highly dependent on the randomly selected initial point, upon which the first complex (constrained simplex) is constructed. Therefore, if the initial point is chosen at random within an unacceptable region, it may fail to converge and terminate its functionality.

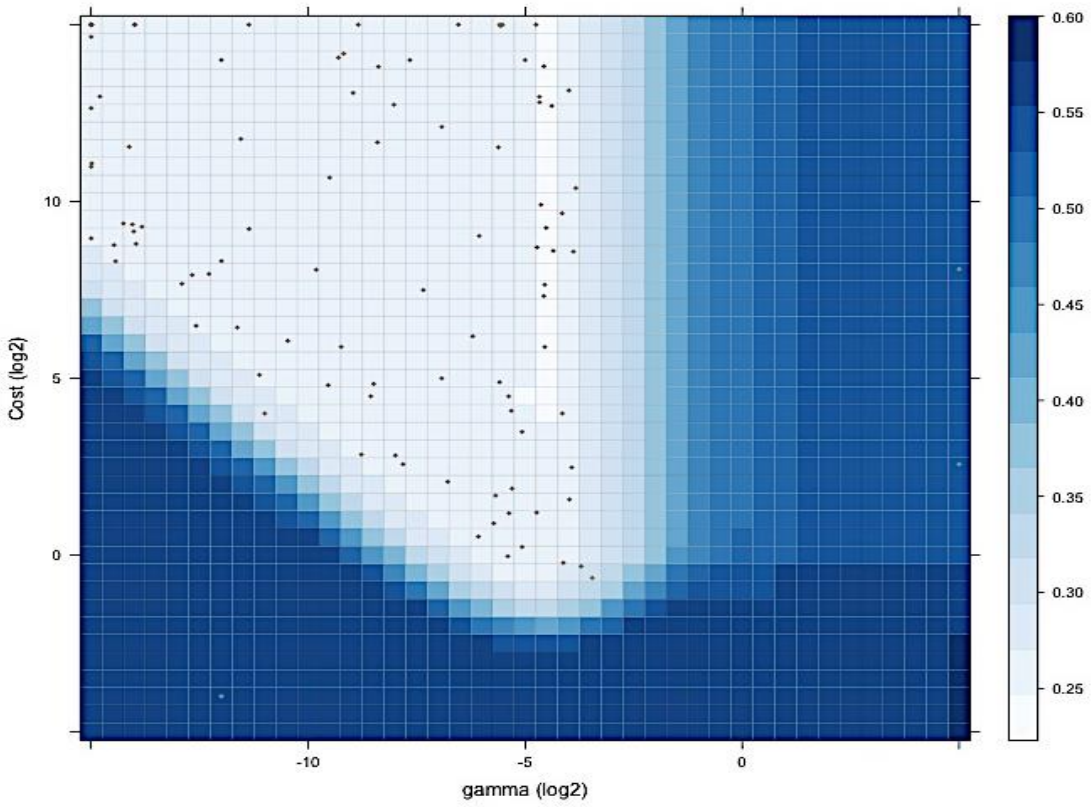


Figure 3-7 Density, filled-contour, grid and contour plots for the HPLC optimisation

The optimal hyperparameters of the Box complex algorithm are plotted on a grid plot formed by the average training performances of 100 classifiers as obtained from the exhaustive grid-search. Each classifier has been optimised using 100 bootstrap iterations.

Thus far, only the optimisation (bootstrapping) predictions of the classifiers have been investigated. Nevertheless, a successful and powerful machine learning model is established by examining its generalisation performance. Figure 3-8 provides a direct comparison of the percentages of correctly classified samples ($\%CC$) between the two optimisation techniques. Both ensembles achieve an overall test accuracy ($\%CC$) of 79%. However, the Box complex algorithm utilised on average only 13 step iterations and 48 function evaluations towards the successful identification of the optimal hyperparameters as presented in Figure 3-9. On the contrary, the exhaustive grid-search required on average 1681 evaluations, which amount to 168100 evaluations in total, thus leading to an enormous computational cost. Finally, Figure 3-8 verifies that at least 100 standalone classifiers are required in an ensemble optimised using the Box complex algorithm in order to generate an accurate and stable result.

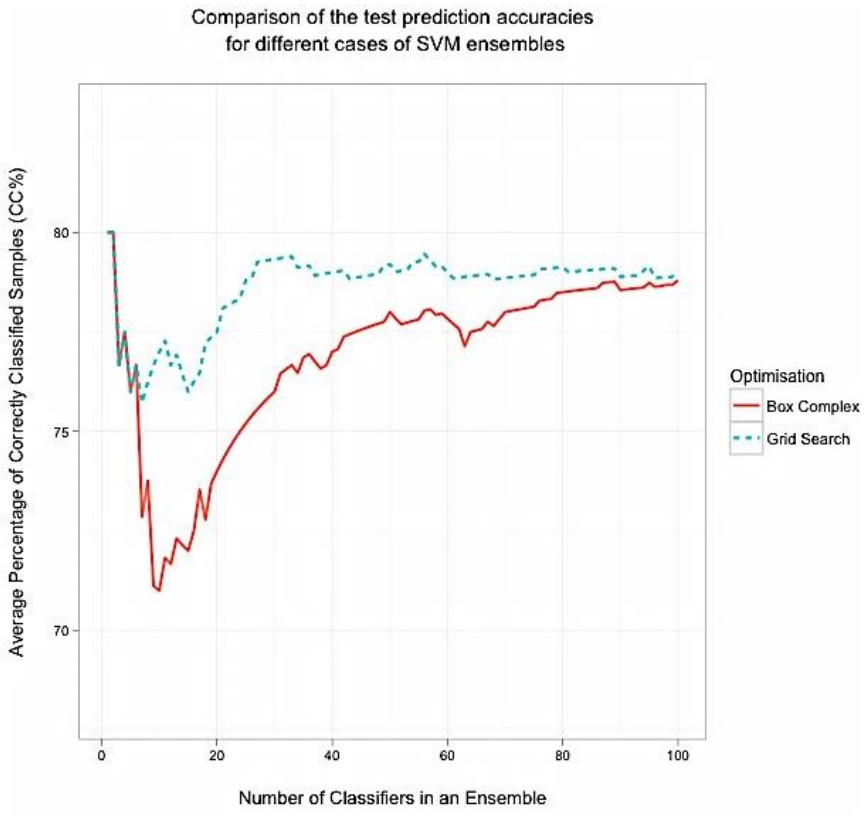
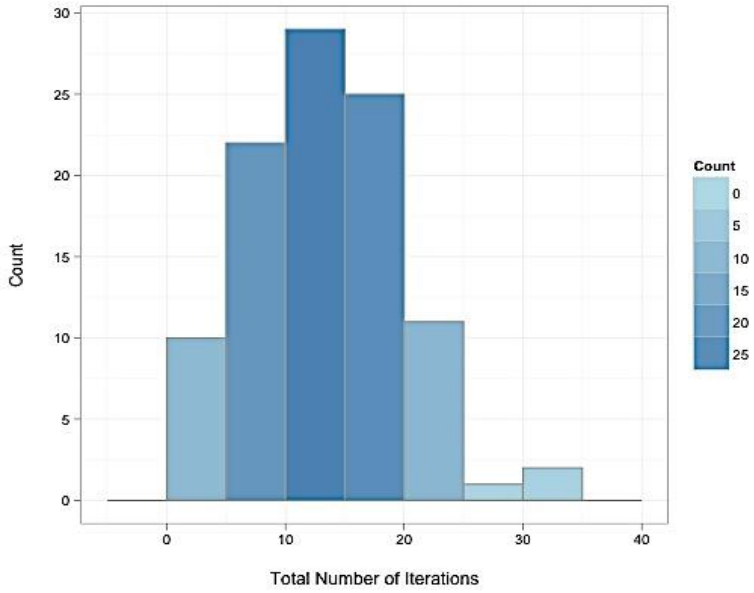
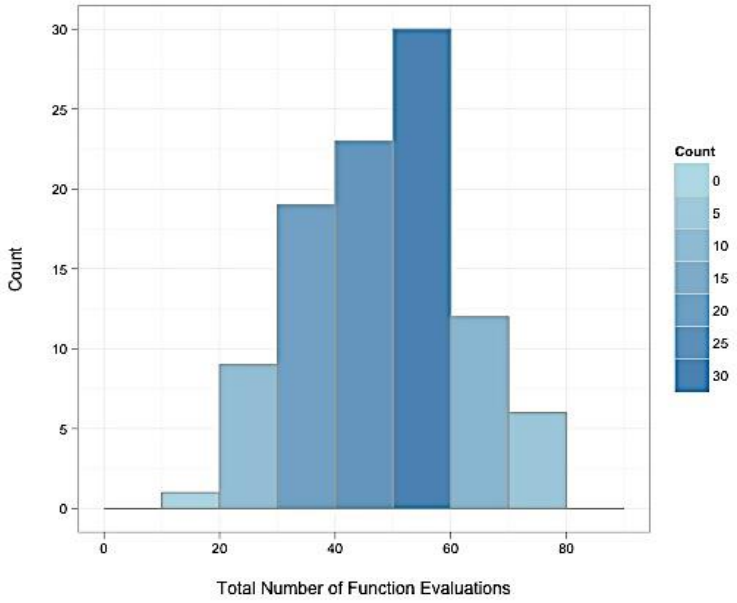


Figure 3-8 Comparison of the prediction accuracies between the grid-Search and the Box complex algorithm (HPLC data)

The plot demonstrates the average test accuracies for the HPLC data against the number of classifiers in an ensemble. Even though the two techniques present different trends to start with, the final obtained overall accuracy was equal to 79% for the grid-search and 78.8% for the Box complex. When rounded to the nearest integer, both techniques account for 79% overall accuracy.



a) Number of iterations



b) Number of function evaluations

Figure 3-9 Histograms of the number of iterations and function evaluations respectively for an ensemble of nonlinear (RBF) SVMs optimised using the Box complex algorithm.

The figure demonstrates the distribution of the total number of iterations and total number of function evaluations respectively as obtained by the application of the Box complex algorithm; the algorithm utilised on average a total of 13 steps and 48 function evaluations towards the successful identification of the optimal hyperparameters *via* bootstrapping. On the contrary, the exhaustive grid-search required on average 1681 function evaluations, which result in 168100 function evaluations in total, thus leading to an enormous computational cost. These values appear to be extremely burdensome when compared to the number of evaluations of the Box complex algorithm.

In a similar manner, the Box complex algorithm was applied to the remaining standalone datasets (FTIR and e-nose) to verify that the obtained results do not deviate from the ones obtained by the grid-search approach that were presented in Figure 2-6. The overall execution time for an exhaustive grid search as presented thus far was approximately 27 hours; thus, the two-step grid proposed in Section 2.2.4 is indeed a big improvement. The execution times for the construction of a full ensemble of RBF SVMs when optimised with different techniques *via* bootstrapping are available in Figure 3-10.

Permutation tests were not feasible prior to the introduction of the new heuristic methodology. An estimation of the run times for the execution of permutation tests using the proposed loose-tuning/fine-tuning approach is illustrated in Figure 3-11; the execution times of 100 permutations tests for all different datasets and optimisation techniques are also displayed in Figure 3-11. It is truly noteworthy that the Box complex algorithm in combination with parallel programming accomplished a speedup up to ~ 91× times as illustrated in Figure 3-12.

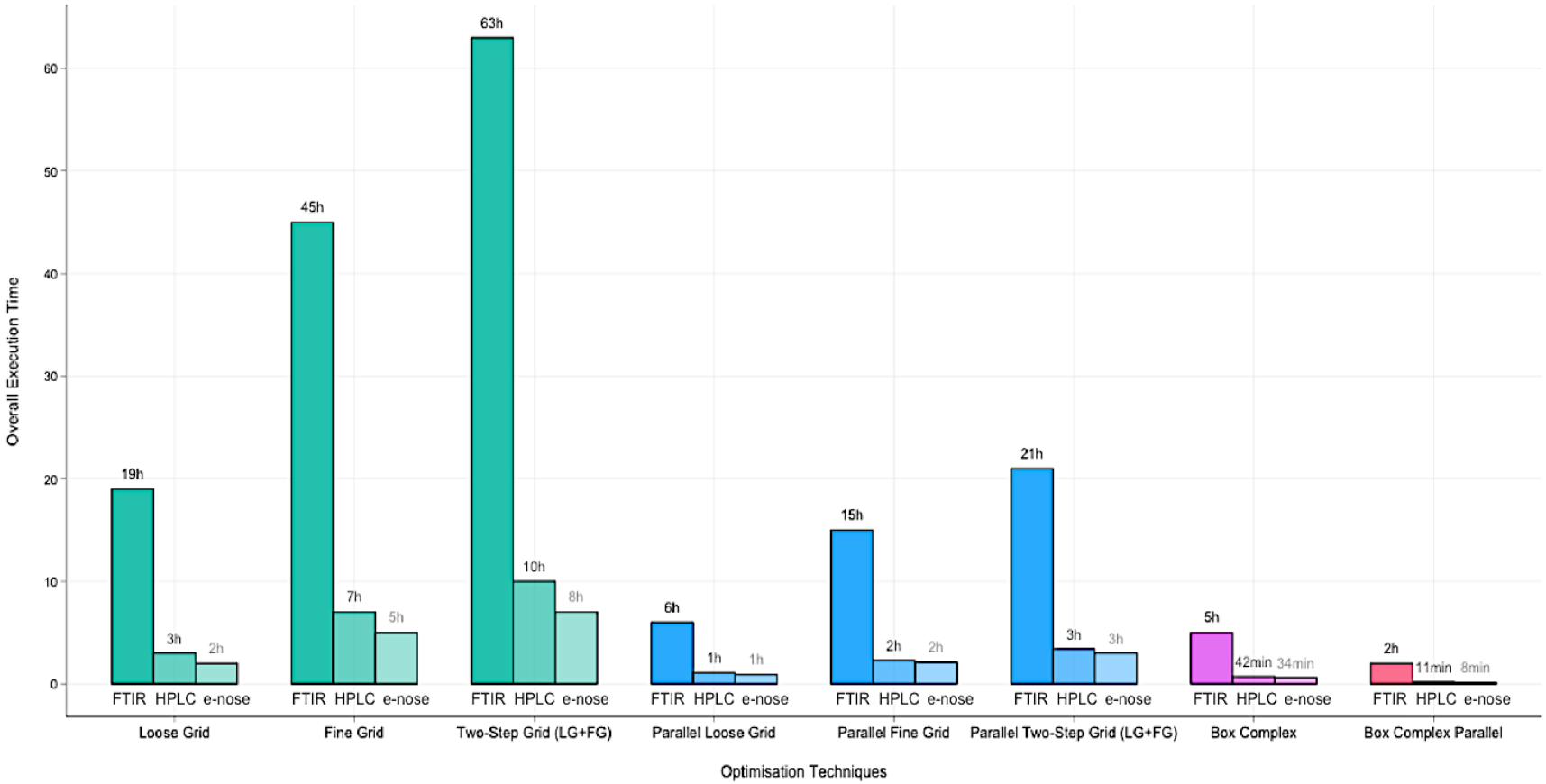


Figure 3-10 Comparison of the execution times for the tuning of a single RBF SVM ensemble, when optimised with different techniques, *via* bootstrapping

The graph provides a direct comparison between various implemented techniques for the optimisation of the SVM (RBF) hyperparameters in combination with bootstrapping. The execution times are based on the architecture described in Section 2.2.5. The execution times were rounded up to the nearest integer.

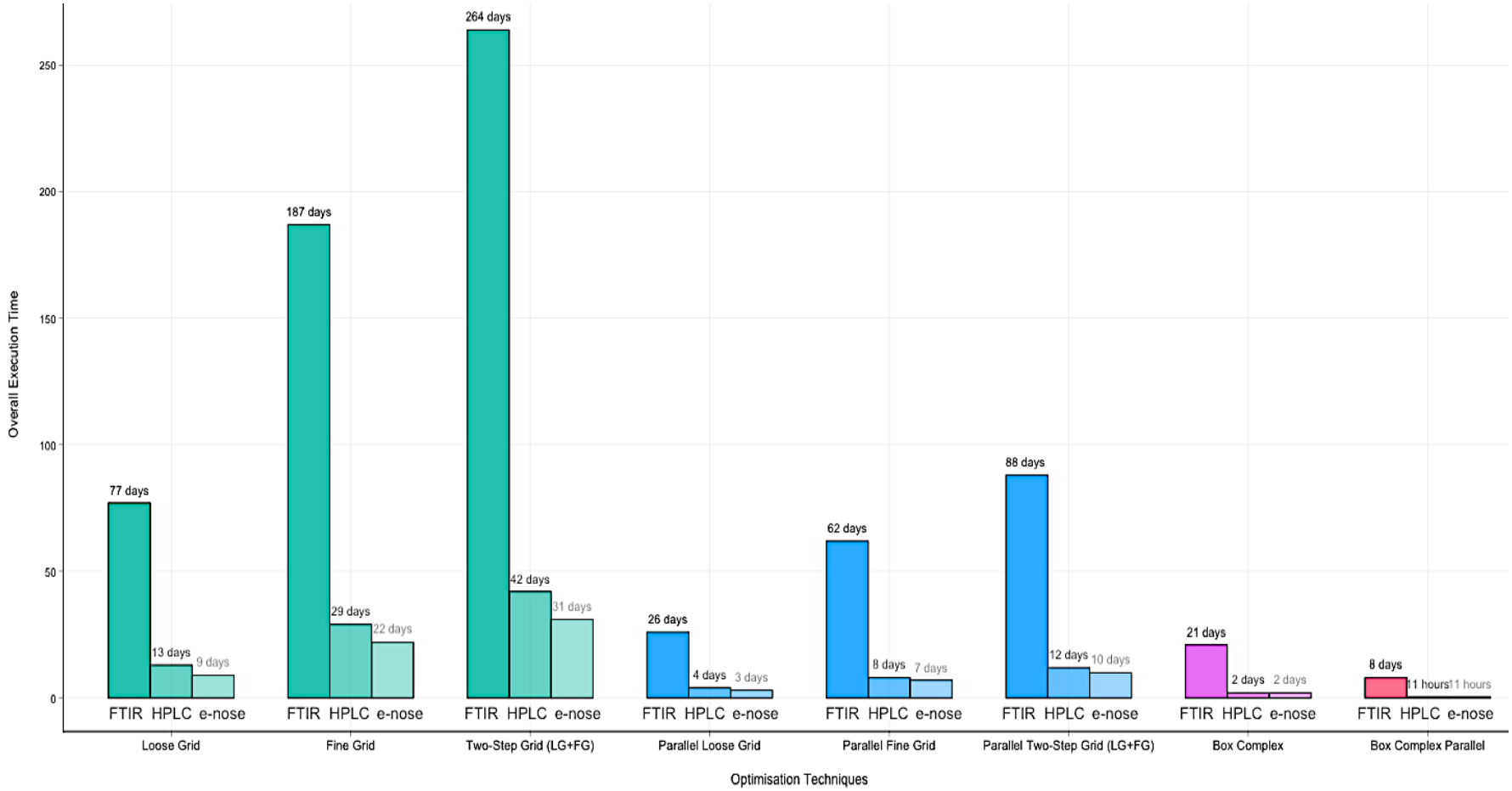


Figure 3-11 Comparison of the execution times of 100 permutation tests for the (RBF) SVMs when optimised with different techniques, *via* bootstrapping

The graph provides a direct comparison of the execution times for 100 permutation tests when different optimisation techniques are used for the tuning process in combination with bootstrapping. The execution times are based on the architecture described in Section 2.2.5. The execution times were rounded up to the nearest integer.

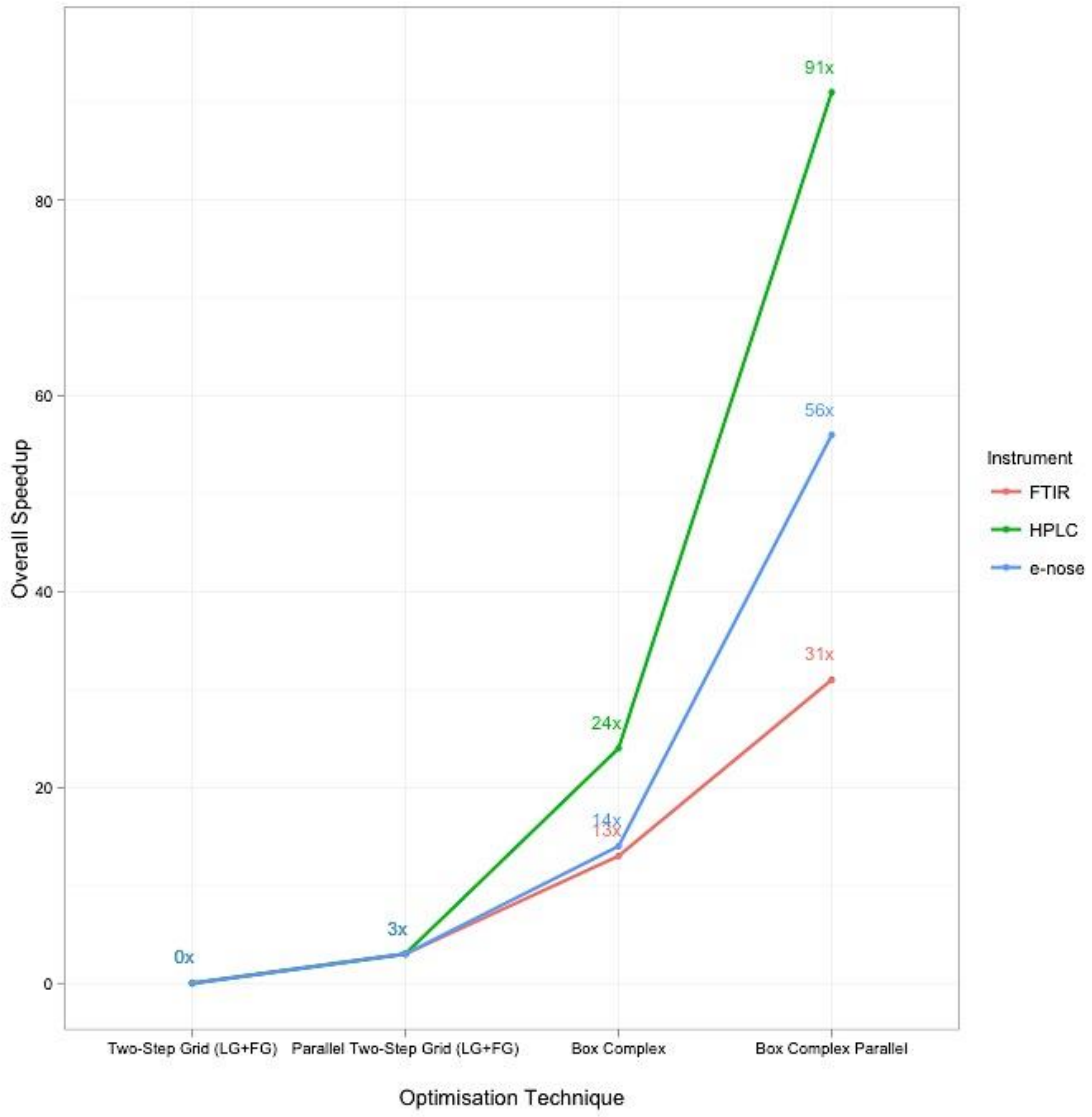


Figure 3-12 Speedup produced by the different optimisation techniques

Let us assume that we start with a two-step grid-search (loose-tuning plus fine-tuning). At first, the application of parallel programming provides a speedup around ~ 3× of the initial time. The application of sequential Box complex increased the speedup up to ~ 24×, which is 8× faster than the previous approach. Finally, the combination of Box complex and parallel programming result in up to ~ 91×, which is notably 30× faster than the first approach and approximately 4× faster than the second one. The differences in speedup may be justified by the number of iterations and function evaluations required by the Box complex to identify the optimum hyperparameters. In addition, the size of the dataset and any potential overhead play also a crucial role.

3.4 Conclusion

In this chapter, a new heuristic methodology has been presented that can reliably identify optimal parameter settings for nonlinear SVMs (RBF kernel) with relatively small computational effort. The Box complex algorithm has significantly minimised the computational complexity and execution times of the hyperparameter optimisation and model construction. Furthermore, the addition of parallel programming to the implemented analysis pipeline, simplified “embarrassingly parallel” problems such as permutation tests into smaller faster tasks, and has proven to be extremely fruitful when applied to relatively large datasets.

4 Integration of Heterogeneous Data

4.1 Introduction

This chapter provides a thorough investigation of multi-block and morphometric data fusion techniques in order to determine whether better classification performance is achieved when integrated as opposed to standalone datasets are used. The hypothesis is that fusion of heterogeneous experimental data that derive from diverse sources may enable a more reliable method for spoilage detection as opposed to single instruments, as different instruments may provide complementary information about the biochemical state of a sample. In this context, we seek to improve the overall classification accuracy. However, the inherent complexity and heterogeneous nature of these data pose a significant challenge to the fusion process. All integrated datasets were analysed alongside the standalone data using the implemented statistical pipeline, while rigorous permutation testing indicated the statistical significance of the results.

4.2 Materials and Methods

4.2.1 Data Integration

In recent years, there has been an increasing necessity for integrative analysis of the enormous amounts of data deriving from the 'omics' fields. Data fusion can be defined as "the integration of data and knowledge collected from disparate sources by different methods into a consistent, accurate, and useful whole" (Synnergren *et al.*, 2009; Wolpert, 1992; Roussel, 2003; Liu, 2004). According to Steinmetz *et al.* (1999), also described in Roussel *et al.* (2003), Smilde *et al.* (2005) and Smolinska *et al.* (2012), data integration techniques are organised in three distinct levels: low-level, mid-level and high-level fusion.

In low-level fusion, raw standalone data from different sources are concatenated at data level prior to any pre-processing. In this model, the columns (variables) of different data blocks are positioned next to each other, while the rows refer to corresponding entities. In theory, data integration should be of the highest efficiency when implemented at a low level. However, according to Smilde *et al.* (2005) this fusion approach is not considered optimal as it suffers from several drawbacks; first and foremost, when data are integrated at the lowest level, the possibility of dealing with noisy and highly redundant data is relatively high. Furthermore, the simplistic approach of concatenating all the raw data together may lead to extremely large datasets with disproportional ratios between observations and variables. Finally, the widely differing numerical ranges of the highly heterogeneous data as well as the lack of appropriate pre-processing may have a dramatic impact on the classification results.

In the mid-level or intermediate level fusion, data integration is performed after the application of dimensionality reduction and/or feature extraction techniques, usually at the principal component or latent variables level (Smolinska *et al.*, 2012). As the data from various sources are probably not homogeneous, they are initially subjected to suitable pre-processing methods (see Section 1.3) in order to standardise them. Since raw data often contain redundant information, dimensionality reduction and/or variable selection techniques are commonly applied on individual standardised data for the extraction of prominent features. The extracted features from each dataset are used to form a concatenated matrix or average consensus. The fused matrix is subsequently imported into a classification model. For both low-level and intermediate level fusion, classification is performed once the data integration step has been completed.

On the contrary, high-level fusion involves the integration of the responses from all individual classification models that have been built on standalone pre-processed data, commonly after the application of decomposition methods. In this approach, the results of several models are fused using statistical averaging or majority voting in order to produce a single final output. However, high-level integration has to face the limitation that the models do not consider the correlations between the different data sources.

4.2.2 Procrustes Analysis

Procrustes rotation or (ordinary) Procrustes Analysis (Gower, 1975; Gower, 2010) is a morphometric technique used for statistical shape analysis that "facilitates comparison between two matrices by rotating, scaling and reflecting one matrix so that it fits as closely as possible to the other" (Gower, 2010). The geometrical interpretation of a matrix in Procrustes Analysis is a shape of landmark configurations. Therefore, transformations are applied on one of the two shapes so that it matches as best as possible the other shape, usually referred to as the "target" shape. As illustrated in Figure 4-1, the three main transformations in Procrustes Analysis include translation (mean-centering), rotation and isotropic scaling. Translation is the centering process of a landmark configuration. Rotation/Reflection involves the rotation of points in order to minimise the difference between them. Scaling standardises the size of a landmark configuration against the size of the centroid.

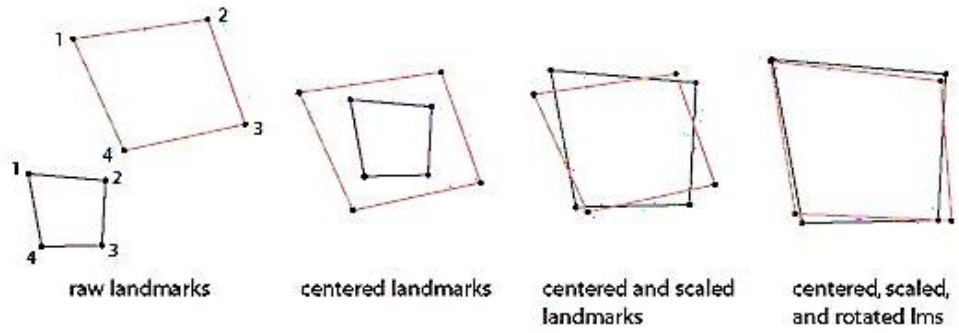


Figure 4-1 Procrustes Analysis superimposition

The picture displays the geometric transformations performed by Procrustes Analysis, which include translation, rotation/reflection and scaling. The figure has been extracted from http://www.virtual-anthropology.com/virtual-anthropology/geometric-morphometrics/procrustes-superimposition

Given two initial matrices X_1 and X_2, where X_1 is of size $n \times p$ and X_2 of size $n \times d$, matrix X_1 is transformed into X_1T in order to match target matrix X_2, where T is a transformation matrix. Subsequently, the algorithm attempts to match points X_{1i} to X_{2i}. In Procrustes Analysis, it is essential that both matrices X_1 and X_2 have the same number of points (rows) n, referring to the same entities (Dijksterhuis and Gower, 1992). On the other hand, the equality in the number of columns is not of great importance. Thus, points in X_2 may have a smaller number of columns than those in X_1. In this case, the matrices may be artificially equalised by adding columns

of zeros (Wu *et al.*, 2002). The "goodness-of-fit" criterion for matrix X_1 to match X_2 is provided by the sum of squared errors as presented in Equation 20. The value of $d(X_1, X_2)$ measures the degree of dissimilarity between the two matrices after rotational and scaling effects have been applied. In this state, the shapes have become as similar as possible.

$$d(X_1, X_2) = \min \|X_1 T - X_2\|^2$$

Equation 20 Procrustes Analysis rotation criterion

Procrustes Analysis is an asymmetric method and its outcome depends solely on the choice of the reference object. Generally

$$d(X_1, X_2) \neq d(X_2, X_1)$$

Equation 21 Procrustes Analysis asymmetric dissimilarities

4.2.3 Generalised Procrustes Analysis

Generalised Procrustes Analysis (GPA) (Gower, 1975; Gower, 2010) is performed in cases where k matrices $X_1, X_2, \cdots, X_k$, with $k \geq 2$, need to be simultaneously transformed and matched against each other without any specific order. An intuitive initial approach would be to examine and evaluate all pair combinations of the initial k shapes by calculating the dissimilarity measures using the mathematical type of Equation 20. In this case, the sum of all dissimilarity measures could be presented as

$$S = \min \sum_{h<k}^{K} \|X_h T_h - X_k T_k\|^2$$

Equation 22 Procrustes rotation criterion in GPA

However, this approach results once more to asymmetry based on Equation 21. Thus, the idea of a single target matrix requires reconsideration (Gower, 2010). GPA was introduced in order to find a group average configuration (Dijksterhuis and Gower, 1992; Dijksterhuis, 1994) or consensus (Gower, 1975), to which the individual subspaces are compared simultaneously (Figure 4-2). This new approach provides greater homogeneity and, in contrast to ordinary Procrustes Analysis, GPA is indeed a

symmetric method since the ordering of the objects does not affect the result. As illustrated Figure 4-2, the GPA algorithm is commonly performed after the application of PCA, in the subspace created by the Principal Components (Bessant *et al.*, 1999; Smilde *et al.*, 2003; Andrade *et al.*, 2004).

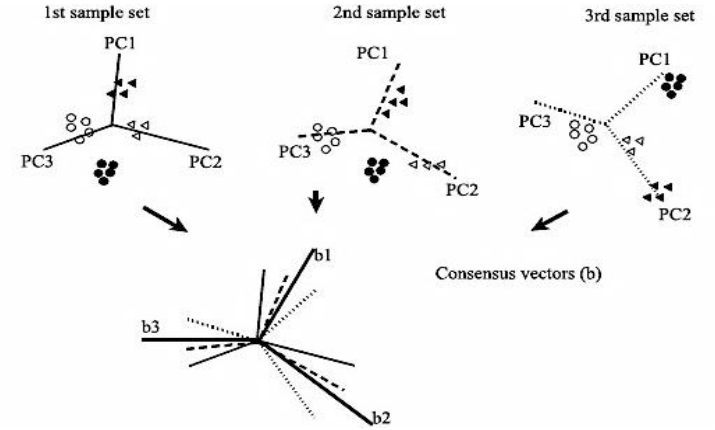


Figure 4-2 Generalised Procrustes Analysis

The figure graphically represents the construction of the GPA consensus shape, based on which, the individual subspaces are compared simultaneously. In this case, each individual subspace is created by the Principal Components. The figure has been extracted from Andrade *et al.*, 2004.

The GPA algorithm is based on computing an iteratively updated average, the consensus G, which is playing the role of the target matrix (Gower, 1975; Dahl *et al.*, 2004). Along with the consensus, a set of the rotated matrices is provided, as similar to G as possible, deriving from the initial shapes (Gower, 1975). The GPA criterion can be mathematically defined as

$$g(X_1, X_2, \dots, X_k) = K \sum_{k=1}^{K} \|X_k T_k - G\|^2$$

Equation 23 Generalised Procrustes Analysis criterion using a consensus

Where k is the number of matrices, T_i is the optimal transformation of the i^{th} profile X_i and G is the average shape or consensus, which can be described as

$$G = \frac{1}{K} \sum_{k=1}^{K} (X_k T_k)$$

Equation 24 GPA Consensus

4.2.4 Multi-block Principal Component Analysis

Multi-block PCA (Westerhuis *et al.*, 1998; Smilde *et al.*, 2003) is an extension of the widely used Principal Component Analysis, applied in cases where multiple blocks of data are present. A block of data or "data block" constitutes a logical entity, which commonly represents the data obtained from a single source. Multi-block methods have been developed to detect underlying relationships and common patterns between several individual blocks of data (Brereton, 2006; Xu and Goodacre, 2012). Furthermore, multi-block techniques have been applied extensively as a means of integrating heterogeneous data from multiple analytical origins into a unified "consensus" view (Xu and Goodacre, 2012).

Several multi-block PCA algorithms have been introduced to date, among which, consensus PCA (CPCA) is probably the most commonly applied multi-block technique for classification purposes. The CPCA algorithm was initially introduced by Wold *et al.* (1987) and further investigated by Westerhuis *et al.* (1998). In addition, a detailed review of the technique can be also found in Qin *et al*. (2001) and Smilde *et al.* (2003).

Let us consider B standalone matrices or blocks, each deriving from a single source or instrument as presented in Figure 4-3. A block matrix of size $n \times m_b$ may be denoted as X_b, where $1 \leq b \leq B$. Commonly in multi-block cases, as with most data fusion techniques, the samples (rows) of different blocks refer to corresponding entities. Prior to the application of CPCA, the data have to be subjected to suitable pre-processing. In order for all variables to have equal weights in the analysis, auto-scaling is applied according to the methodology described in Section 1.3.2. In addition, since the individual blocks may differ significantly with respect to their number of variables (Figure 4-3), the datasets are also subjected to "block-scaling" (Smilde *et al.*, 2003; Brereton, 2009); the block-scaling factor is equal to the inverse of the square root of the number of variables within a specific block (Hassani *et al*., 2010). This scaling prevents the dominant influence of a single block since each block will contribute the same amount of variance to the consensus of CPCA method. After data pre-processing, all individual blocks are concatenated into a single super-matrix X of size $n \times m$, where $X = [X_1\ X_2\ \ldots\ X_b]$ and $m = m_1 + \cdots + m_b$.

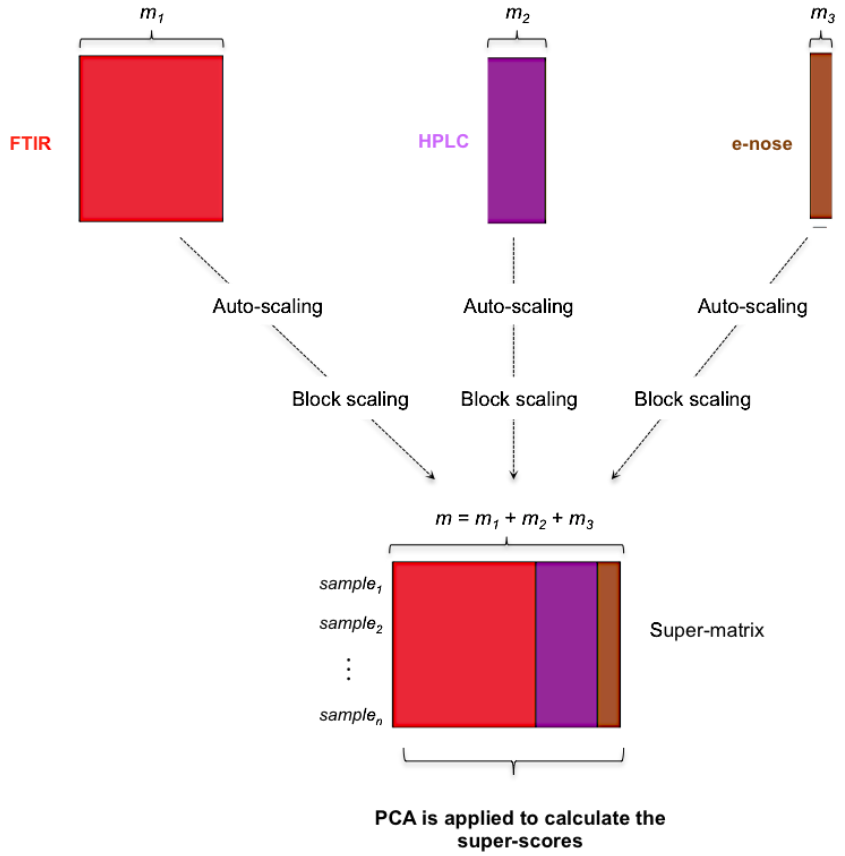


Figure 4-3 Steps of CPCA for the datasets of case study 1

The heterogeneous FTIR, HPLC and e-nose data constitute three standalone data blocks. The figure highlights the need for the application of suitable pre-processing, and specifically for block-scaling since the number of variables among the datasets varies greatly. All pre-processed data are concatenated into a single super-matrix, upon which CPCA is performed.

The output of CPCA consists of the super-scores matrix, the blocks' scores and loadings matrices, and the block weights. The super-scores matrix demonstrates the "global" variation trend across the heterogeneous blocks and constitutes the consensus object at the "super-level" (Hassani *et al.*, 2010). In addition, each block's scores and loading matrices provide a unique pattern of each block under the global consensus (super scores). Finally, the block weights demonstrate the contribution of each block to the super scores matrix. The CPCA algorithm "normalises the block and super loadings, and deflates the residual matrices based on super scores" (Qin *et al.*, 2001). In brief, the CPCA algorithm according to Qin *et al.* (2001) is described as follows:

1. Select vector $t_{T,i}$ as a starting point
2. Compute the block loadings
   $p_{b,i} = X_{b,i}^T t_{T,i} / \left\| X_{b,i}^T t_{T,i} \right\|$
3. Compute the block scores
   $t_{b,i} = X_{b,i} p_{b,i}$
4. Consider the matrix
   $T_i = [t_{b,i} \quad \dots \quad t_{B,i}]$
5. Compute the global loadings
   $p_{T,i} = T_i^T t_{T,i} / \left\| X_i^T t_{T,i} \right\|$
6. Update the global scores
   $t_{T,i} = T_i p_{T,i}$
7. Iterate steps 2 - 6 until the convergence of $t_{T,i}$
8. Deflate the residuals $X_{b,i+1} = \left(I - t_{T,i} t_{T,i}^T \,/\, t_{T,i}^T t_{T,i}\right) X_{b,i}$

The most notable breakthrough contribution pertaining to the field of multi-block algorithms was made by Westerhuis *et al.* (1998), whereby it was proved that the super-scores of CPCA are identical to the scores of normal PCA when applied on the concatenated set of blocks (the super-matrix) (Qin *et al*., 2001; Smilde *et al*. 2003) (see Figure 4-3). In addition, according to Westerhuis *et al.* (1998) regular PCA can be also used to calculate the individual block scores and loadings. Normal PCA may be applied using a plethora of algorithms, the most popular of which is the NIPALS algorithm (see Section 1.4.1).

In the context of this project, consensus PCA was primarily investigated as a data integration technique rather than for the purpose of detecting an underlying common pattern between the different blocks. The super-scores of CPCA, which constitute the consensus matrix, are used as input into the implemented classifiers.

4.2.5 Data Integration and Analysis Pipeline

The functionality of the multivariate analysis pipeline developed in the previous chapters was further extended to incorporate data fusion techniques. As presented in Section 2.2.1, the datasets of case study 1 have been acquired from beef samples using three main experimental techniques: spectroscopy (FTIR), high performance liquid chromatography (HPLC) and electronic nose. Initially, the datasets are cross-referenced based on the samples' names and sensory values (see Section 2.2.2). Based on this filtering, a total of 32 common samples are inserted in the data integration pipeline along with the respective sensory scores as depicted in Figure 4-4.

As PCA is the first step towards data integration, the principal components of each experimental technique were also tested using the multivariate analysis pipeline. More specifically, the PCA data were analysed in a repetitive manner by increasing the number of principal components each time by one until the optimal number of PCs, responsible for the highest classification accuracy, was identified. Since PLS-DA employs the PLS algorithm for data decomposition and dimensionality reduction (Section 1.5.1), the application of PCA with this type of classifier is considered redundant. However, in order to obtain directly comparable classification results, this approach was also investigated and the outcome is presented as follows.

In this work, data integration is employed using a "mid-level fusion" approach based on the methodology presented in Section 4.2.1. As a first step towards analysing the integrated datasets, unsupervised methods for dimensionality reduction and the extraction of prominent features are applied on pre-processed data. In the case of GPA, prior pre-treatment such as mean-centering and/or scaling is not a necessity since these steps are included in the algorithm. Thus, raw data are subjected to PCA, and a predefined number of Principal Components (PCs) is extracted from each dataset. The PCA scores of each instrumental technique are concatenated in a three-way data matrix, which is subjected to geometric transformations. The output of GPA consists of the average (consensus) matrix in addition to all individual transformed matrices for the given combination of experimental techniques.

In the case of CPCA, standalone pre-processed (auto-scaled and block-scaled) data are concatenated into a single matrix – the super matrix. Subsequently, the super matrix is subjected to normal PCA. The output of PCA on all the concatenated blocks together (super matrix) constitutes the super scores, which will be used as the data integration consensus provided by CPCA.

In both cases, the consensus, as produced by the data fusion techniques, is subsequently used as input into an ensemble of classifiers. Classification techniques that have been implemented include PLS-DA in addition to linear and nonlinear SVMs. According to Section 2.3.2.3, bootstrapping proved to be the most accurate of all validation techniques; therefore, it has been employed for hyperparameter optimisation. Finally, the classifiers are subjected to rigorous permutation testing, which is currently feasible due to the novel optimisation approach presented in Chapter 3. The permutation results provide an indication of the statistical significance of the results.

4.2.6 Implementation in R

The **shapes** package (Dryden, 2012) consists of a set of programming tools in **R** for the statistical analysis of shapes. The package was used for the implementation of the GPA using the function **procGPA()**. Along with the transformed shapes and the consensus, the algorithm returns among others, the measures of dissimilarities between the input matrices.

Consensus PCA (CPCA) can be implemented by applying normal PCA as provided by a wide range of built-in and add-on **R** functions (see Section 2.2.8). Even so, in this work, a function was produced that conducted PCA *via* SVD.

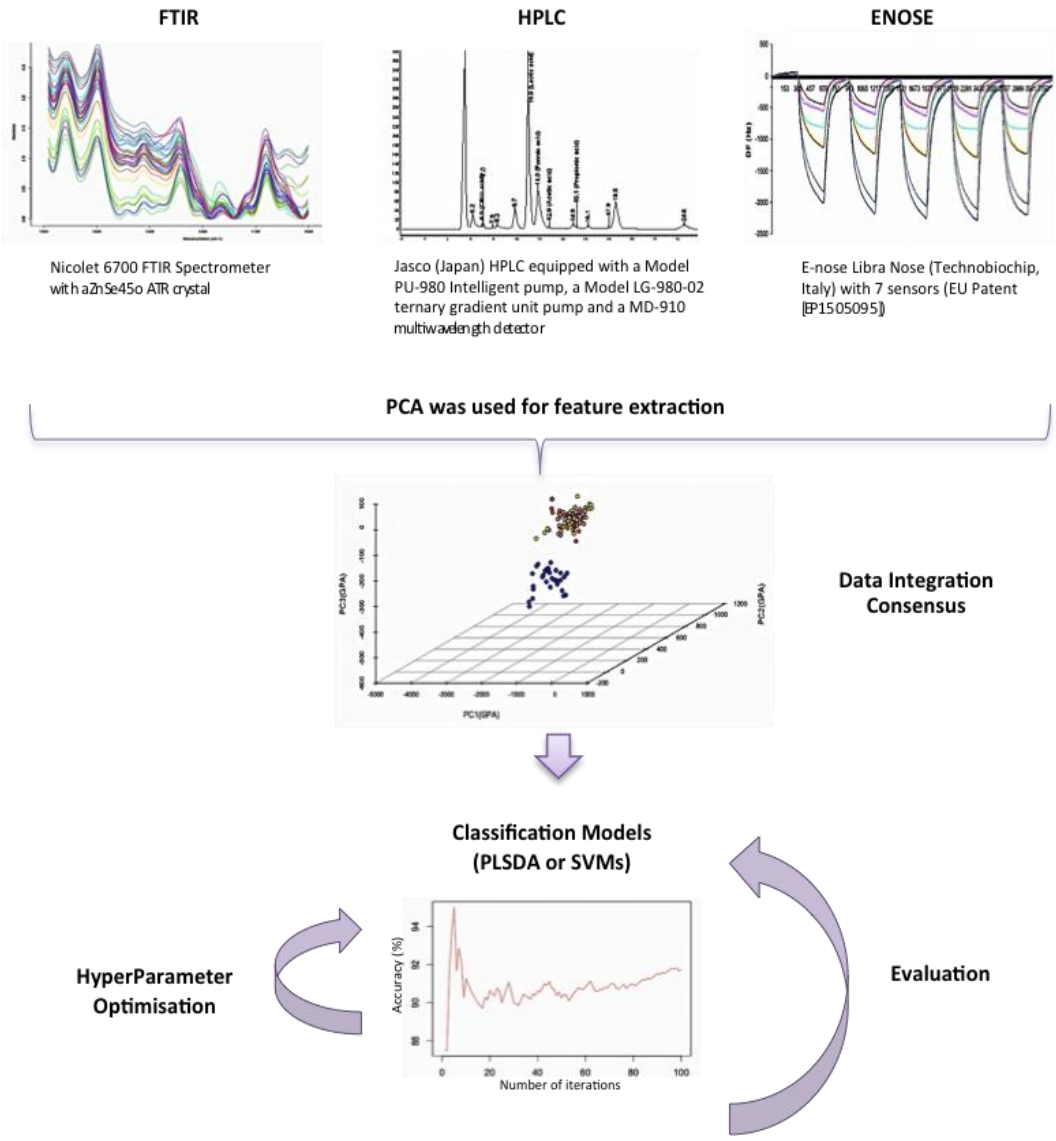


Figure 4-4 Data integration workflow

The figure represents the steps of the data integration and analysis process when all the datasets of case study 1 are used as input in the pipeline. FTIR, HPLC and e-nose constitute three individual blocks of data. In both data fusion techniques, the consensus is formed in the subspace created by the Principal Components at either the local or global (super) level.

4.3 Results and Discussion

4.3.1 Exploratory Data Analysis

The first implemented data fusion technique involved the GPA algorithm. As illustrated in Figure 4-5, the PCA scores from each experimental technique of case study 1 expand over wide and different numerical ranges. The GPA algorithm uses the output of PCA to perform statistical shape analysis in order to minimise the dissimilarity between the initial datasets. Figure 4-5 demonstrates step-by-step the geometric transformations (translation, rotation and isotropic scaling) performed by the GPA algorithm on all three standalone datasets simultaneously. Once the shapes are fitted as closely as possible to each other, the algorithm makes use of the individual transformed matrices in order to construct the consensus. The consensus matrix, as presented in Figure 4-6, is further used as the input into a classification ensemble.

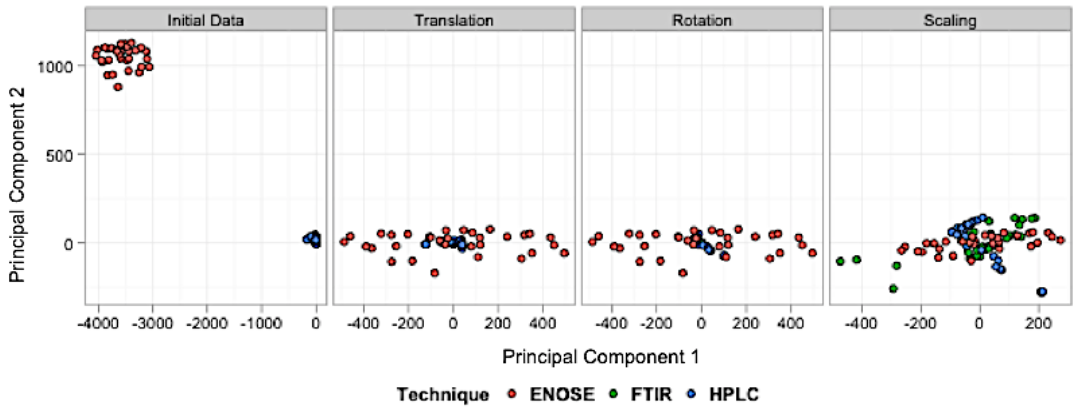
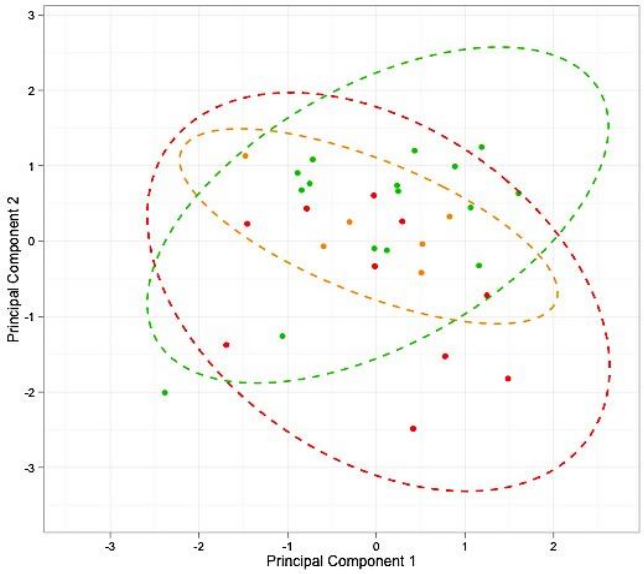
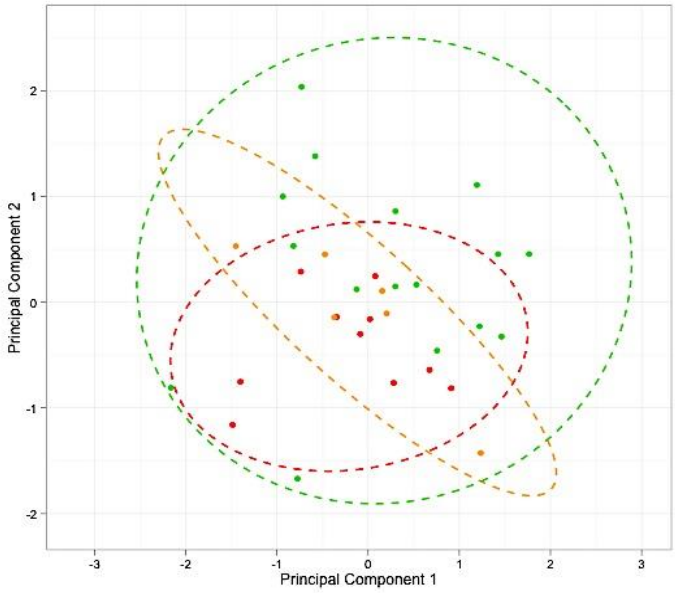


Figure 4-5 The steps of GPA (shown in order from left to right) when applied on the datasets of case study 1

The two-dimensional scatterplots display the standalone datasets prior and after the application of GPA when all three experimental techniques of case study 1 are investigated simultaneously. In this case, the GPA algorithm is applied in the subspace created by the first two principal components. In order to demonstrate the differences in range of the initial datasets under study, PCA was applied on raw standalone data with no prior pre-processing. The GPA algorithm subjects all individual datasets to translation (mean-centering), rotation and isotropic scaling. In the final plot, the shapes are as similar to each other as possible.



a) GPA



b) CPCA

Figure 4-6 The consensus of the first two Principal Components based on the fusion of all three experimental techniques of case study 1 using GPA and CPCA respectively

The consensus output of GPA is compared against the super-scores obtained by CPCA; in both cases, all three experimental techniques of case study 1 (FTIR, HPLC, e-nose) were used simultaneously for the data fusion. Dynamically generated 95% confidence ellipses per each class were added in the plots in order to highlight the presence of any clusters and/or outliers. Colour representation was used to identify the three classes: fresh (red colour), semi-fresh (orange colour) and spoiled (green colour). Based on the graphs, the consensus shapes do not provide any obvious discrimination between the three classification groups since the distinct classes are highly overlapping. It is apparent that the good clustering between fresh and spoiled samples as observed by standalone HPLC in Figure 2-5 was destroyed during the fusion process with the other two analytical techniques.

4.3.2 Classification Results

4.3.2.1 Overall Accuracies (%CC)

The classification results of all the standalone datasets for case study 1 are illustrated in the bar charts of Figure 4-7 as percentages of correctly classified samples ($\%CC$). The overall accuracies of the standalone datasets prior to PCA have been thoroughly described in Section 2.3.2. As PCA is the first step towards data integration, the principal components of each experimental technique were also tested using the multivariate analysis pipeline according to Section 4.2.5

Based on the bar charts of Figure 4-7, the overall accuracy of FTIR is clearly enhanced when the raw data are subjected to PCA; in particular, the results of both PLS-DA and SVMs increase by approximately 5%. Even so, as in the case of raw FTIR, the ensemble of nonlinear (RBF) SVMs performs relatively worse than the linear classifiers. In order to ensure that the low accuracy of the RBF models is not due to overfitting and/or the newly incorporated Box complex approximation algorithm, the result was verified by executing once more the grid search approach of Section 2.2.4. Indeed, the nonlinear boundaries of the RBF kernel proved to be too complex to correctly classify the simple FTIR data; linear separation clearly gives better results for this. Xu *et al.* (2006) report that chemometric algorithms such as PLS-DA are more efficient when applied on traditional analytical techniques such as spectroscopy, where the data are linear and well understood. In addition, according to Smoliska *et al*. (2012), a major drawback among kernel-based methods, also commonly encountered in the auto-scaling process, is that useful information on the importance of the variables is permanently lost. This impediment is crucial in the case of FTIR, where the spectral data may contain prominent peaks in significant biochemical absorption regions that may reveal differences between the samples of different classes. Furthermore, according to Section 1.5.2.3, an RBF SVM should be able to perform at least as well as a linear SVM (Boser *et al.*, 1992; Keerthi and Lin, 2003; Chang *et al.*, 2010); thus, we can only assume that the optimisation process was unable to identify suitable combinations of hyperparameters that result in the same

accuracies as the linear models; possibly, the cost and/or gamma values that generate nearly linear boundaries did not satisfy the provided constraints.

In the case of HPLC, PCA boosts the overall accuracy of the simplistic PLS-DA ensembles, while it decreases the results of both linear and nonlinear SVMs by approximately 5%. Since both types of SVMs produce a lower accuracy comparing to PLS-DA, we can only assume that the background of SVMs is the underlying cause for this result; as presented in Section 1.5.2.1, PLS-DA constructs the decision boundaries based on all available samples as a whole, whereas SVMs are solely based on the selection of support vectors. A thorough investigation of the class predictions may help towards justifying this hypothesis.

Finally, the PCA scores from the e-nose dataset perform significantly better for all implemented classification models when compared to the accuracies of raw data. In particular, the SVM ensembles reach a maximum overall accuracy of 50%, the highest recorded result for this dataset. Despite the improvement in the results, as with the raw standalone data, the generalisation performance of e-nose is relatively poor since it generates the lowest accuracies among the three experimental techniques.

The classification results of the integrated datasets using GPA and CPCA are displayed in Figure 4-8. For the GPA algorithm, SVMs produce at least as good results as PLS-DA in the majority of cases, with the linear SVMs taking the lead among the three classification ensembles. In addition, in the pairwise combination of FTIR with HPLC, as well as the simultaneous fusion of all three experimental techniques, linear SVMs have produced somewhat higher results when compared to those by the standalone principal components. On the contrary, the pairwise integration of either FTIR or HPLC with e-nose decreases the overall accuracy. It can be therefore concluded that e-nose clearly dominates the outcome of data integration and classification in this instance.

Furthermore, in the majority of integrated datasets, the GPA algorithm produces higher overall accuracies compared to CPCA when linear and nonlinear SVM classifiers are employed. For the same instances, when PLS-DA is applied as the classification technique, the results between the two data integration techniques

appear to be similar (the differences are less than 1%). However, CPCA clearly improves the outcome of the integration between HPLC and e-nose, since the overall accuracies of all classifiers have increased by approximately 10% compared to GPA. Based on all documented classification results, the highest overall accuracy, equal to 80%, was obtained for standalone HPLC prior and after the application of PCA. Even though the analysis of integrated datasets did demonstrate relatively good performance, the results were not as great as standalone HPLC.

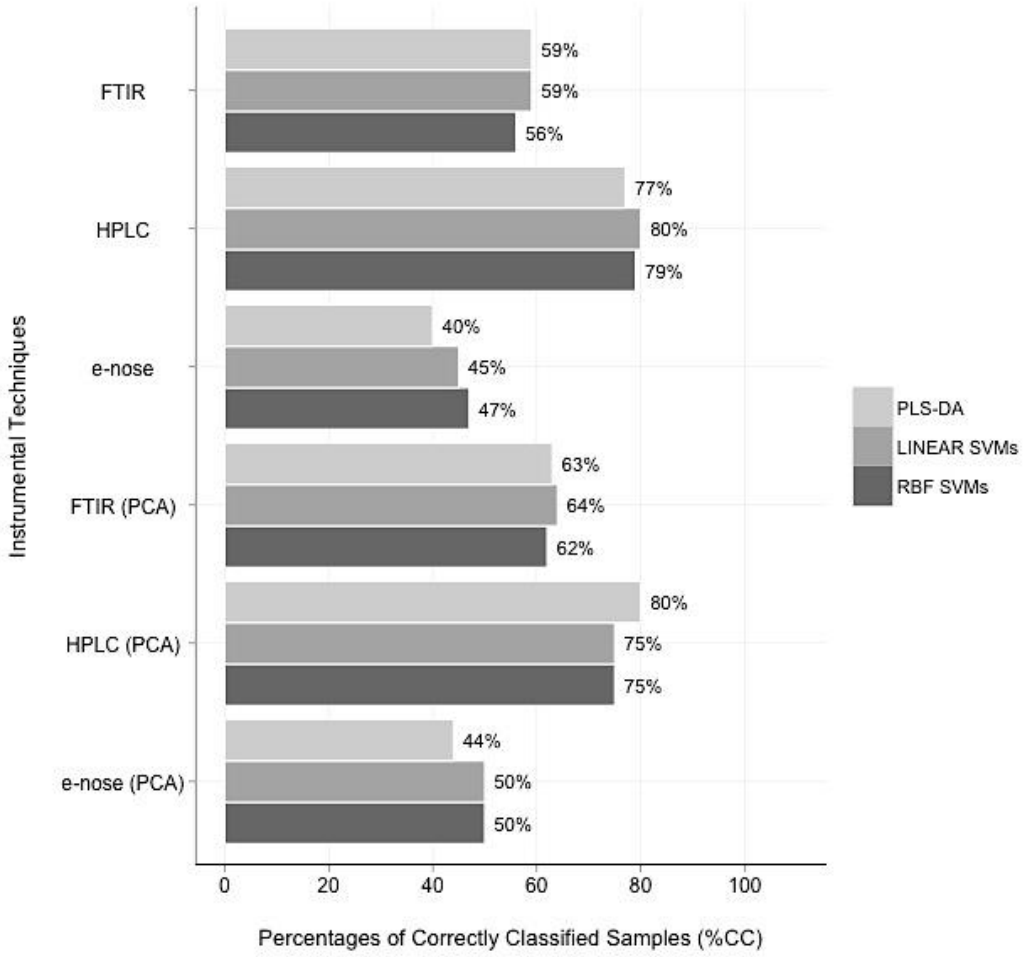


Figure 4-7 Overall accuracies (%CC) for the standalone datasets of case study 1

The figure illustrates the overall performance of all implemented classification ensembles on the standalone datasets of case study 1. The bars represent the percentages of correctly classified samples (%CC) and are coloured according to the classification model under study (PLS-DA, linear and RBF SVMs). Analyses have been conducted both prior (raw data) and after PCA. In all implemented classifiers, bootstrapping was applied for hyperparameter optimisation. The overall accuracies have been rounded towards the nearest integer.

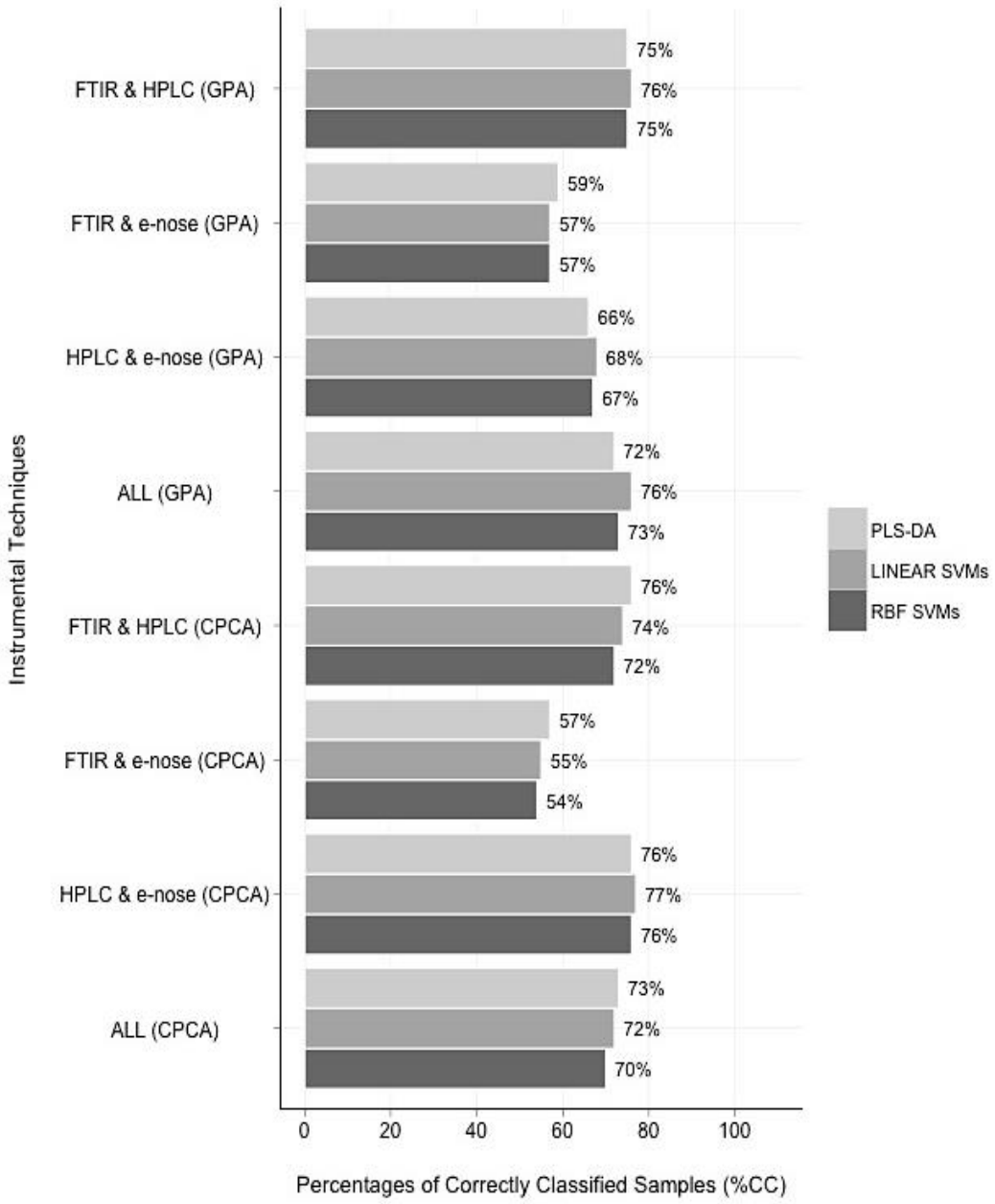


Figure 4-8 Classification Results for the integrated datasets of case study 1

The figure illustrates the overall performance of all implemented classification ensembles on the integrated datasets of case study 1. The bars represent the percentages of correctly classified samples (%CC) and are coloured according to the classification model under study (PLS-DA, linear and RBF SVMs). Data integration has been performed using both GPA and CPCA. In all implemented classifiers, bootstrapping was applied for hyperparameter optimisation. The overall accuracies have been rounded towards the nearest integer.

4.3.2.2 Class Prediction Accuracies

In addition to the overall accuracies, the per-class percentages of correctly classified samples for the standalone and integrated datasets of case study 4 are depicted in Figure 4-9 and Figure 4-10 respectively. In this case study, spoiled samples constitute the majority class; thus, as expected, spoiled samples obtain outstanding class accuracies throughout all classification models and experimental techniques; in the majority of cases, these class percentages were well above 90%.

The class predictions of the raw standalone datasets were determined earlier, in Section 2.3.2.2. Since PCA is the first step towards data integration, the per-class accuracies of the individual PCA scores for each experimental technique are examined as follows. Based on Figure 4-9, the PCA scores of FTIR in the case of the PLS-DA ensembles increase the class accuracies of both fresh and spoiled samples, while the semi-fresh accuracies are equal to 0%. Since case study 1 is a multi-class problem, semi-fresh samples, which constitute the minority class, are extremely difficult to predict. Furthermore, in the case of linear SVMs, the application of PCA has proven to be extremely fruitful since the fresh and semi-fresh class accuracies increase by 11% and 25% respectively. On the contrary, the class predictions of spoiled samples decrease by 10%. Finally, for nonlinear (RBF) SVMs, the FTIR accuracies for all three classes increase substantially, but most importantly the class prediction of SF samples reaches the notable percentage of 31%.

For the HPLC data, PLS-DA provides a strong nearly linear separation between fresh and spoiled samples with prediction rates above 90%, while the semi-fresh samples present a noteworthy accuracy of 17%. In the case of linear SVMs, the fresh and spoiled class predictions drop compared to raw HPLC, while the SF prediction rate increases by 32%; this fact justifies the decreased overall accuracy of HPLC in Figure 4-7. As far as the RBF SVMs are concerned, the class predictions follow the trend of linear SVMs.

Similar to FTIR, the PLS-DA ensembles for the e-nose data increase the rates of fresh and spoiled samples, while the semi-fresh accuracies are equal to 0%. Even though linear and nonlinear SVMs generate higher percentages of correctly classified samples ($\%CC$), PLS-DA demonstrates better classification accuracies for the fresh samples compared to the SVMs. Finally, in the case of linear and nonlinear (RBF) SVMs, both fresh and semi-fresh accuracies significantly decrease, while the prediction rates of the majority class approximate 100%.

To summarise the previous observations, as with raw data, the highest per-class accuracies for the semi-fresh samples are recorded in the case of FTIR data when SVMs are applied. In addition, the high overall accuracies of HPLC are justified by the nearly perfect class predictions of fresh and spoiled samples, especially for the linear classifiers (PLS-DA and SVMs). Finally, based on the e-nose class predictions it is obvious that the implemented classifiers have no discriminative power to correctly classify the semi-fresh samples; it appears that the boundaries of both the linear and nonlinear classifiers are dominated by the majority class, thus resulting in outstanding class predictions for the spoiled samples.

According to Section 4.3.2.1, it is obvious that e-nose is a dominant technique that strongly influences the outcome of the integration and analysis process. In the case of GPA, the class accuracies from the pairwise fusion of either FTIR or HPLC with e-nose verify this hypothesis. Based on Figure 4-10, the high fresh and semi-fresh accuracies obtained by standalone FTIR and HPLC decrease once the integration is performed, for all the different types of classifiers. Furthermore, the integrated datasets present at least 80% in the class predictions of spoiled samples. This is to be expected, not only because spoiled samples constitute the majority class, but also because the standalone e-nose data most often resulted in high percentages of correctly classified spoiled samples. For the integrated dataset of FTIR and HPLC, the overall accuracies and class predictions of both linear and nonlinear models demonstrate similar results. After the integration, the spoiled accuracies increase for all three classifiers, while the performance of fresh samples drops for the linear models (PLS-DA and linear SVMs). The most noteworthy improvement from standalone to integrated datasets for all individual classes (sensory scores) was documented for the RBF SVMs. Finally, in the integration of all three datasets, it

appears that the different experimental techniques verily provide valuable complementary information in the consensus. The class predictions of spoiled samples demonstrate an increase to at least 91%, with better performances accomplished by the SVMs; this accuracy may be justified once more due to the presence of the e-nose dataset in the consensus. In addition, FTIR and HPLC contribute to the notably good predictions of the semi-fresh samples, whereas the performance of fresh samples decreases for all three classifiers.

The CPCA per-class accuracies follow the same pattern as GPA in all cases besides the integration of HPLC and e-nose. It is noteworthy to mention that in the majority of cases, CPCA produces significantly higher classification rates for the fresh samples than GPA; on the contrary, GPA clearly favours both semi-fresh and spoiled samples, thus leading to higher class predictions compared to CPCA. Even though the application of multi-block PCA did overcome the limitations noted by GPA, it still did not produce as good or even greater results than standalone HPLC.

As a general observation, all standalone and integrated datasets produce high rates for spoiled samples when tested with RBF SVMs. On the contrary, fresh samples demonstrate the best class rates for linear models, and especially in the case of PLS-DA. Semi-fresh samples are particularly favoured by linear SVMs, with noteworthy predictions for FTIR and HPLC. It is therefore apparent that the classifiers vary in their ability to distinguish between the different classes and the accuracy with which individual classes were classified differed markedly. Since the three classifiers operate in very different ways, they may be viewed as complementary sources of information rather than competing options.

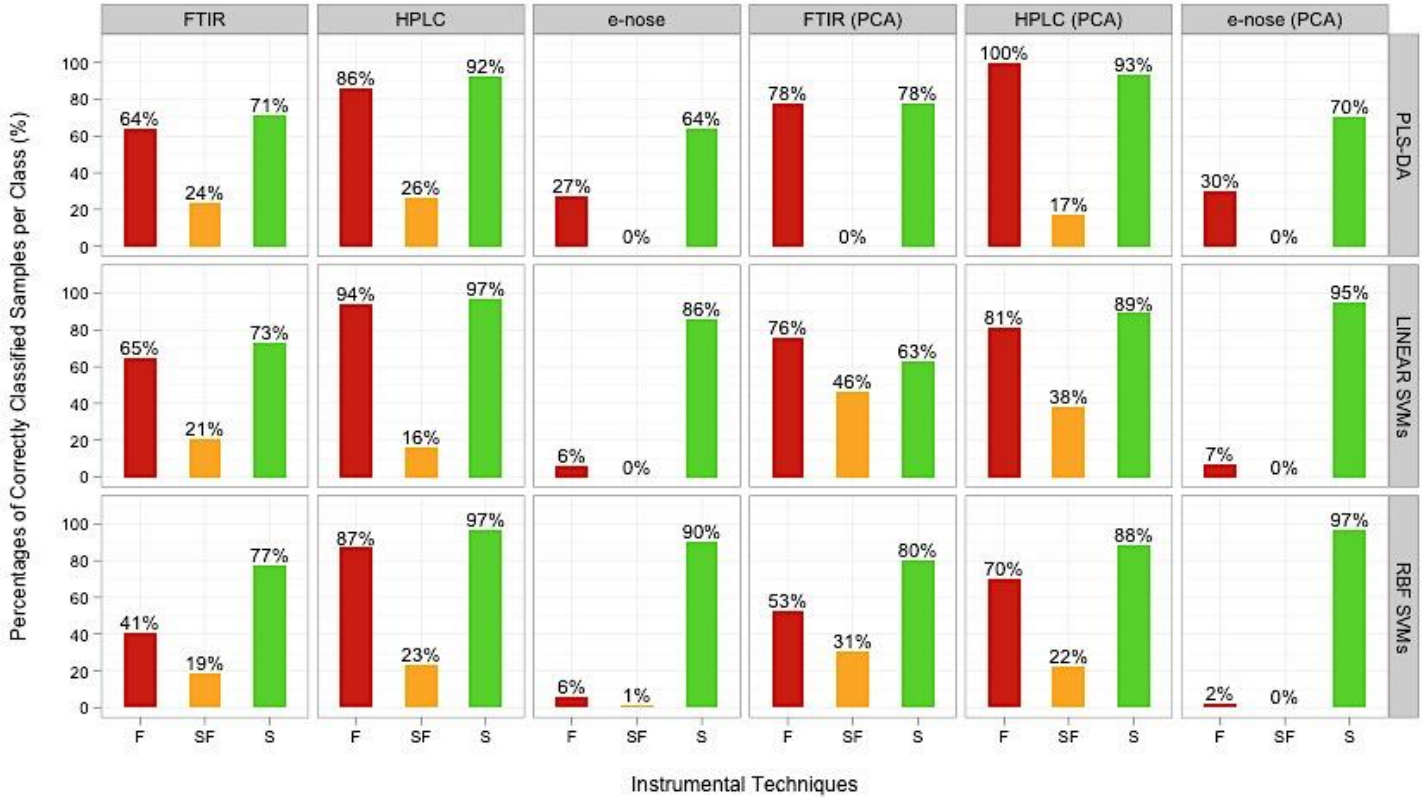


Figure 4-9 Class prediction rates of the standalone (prior and after PCA) datasets for case study 1

The figure illustrates the percentages of correctly classified samples per each distinct class, when the standalone datasets (prior and after PCA) of case study 1 are imported in the analysis pipeline. The class predictions are compared based on the instrumental techniques and classification models. Colour representation is used to identify the three classes as determined by the relevant sensory scores: fresh (red colour), semi-fresh (orange colour) and spoiled (green colour).

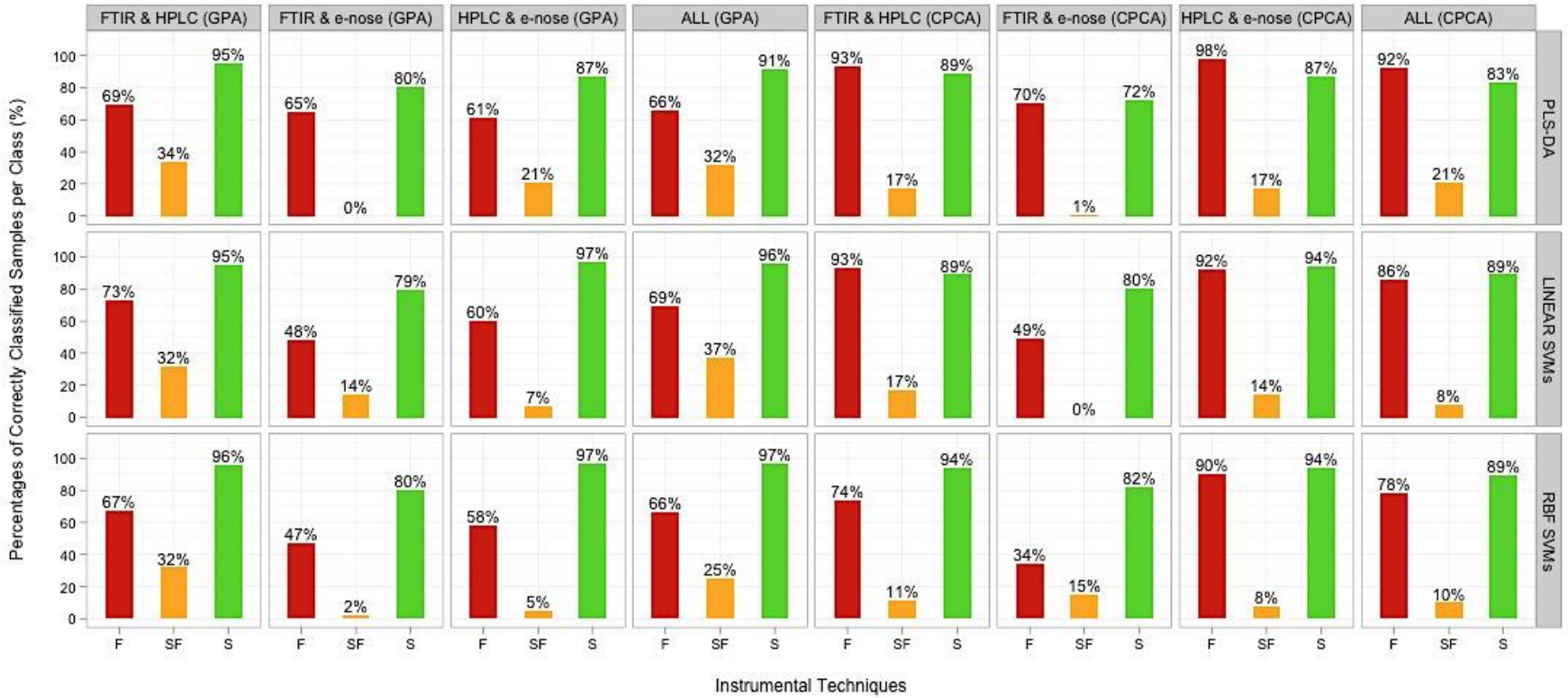


Figure 4-10 Class prediction rates of the integrated datasets for case study 1

The figure illustrates the percentages of correctly classified samples per each distinct class, when the integrated datasets of case study 1 are imported in the analysis pipeline. Data fusion was performed using GPA and CPCA. It is noteworthy that CPCA produces significantly higher percentages of correctly classified fresh samples than GPA; however, the prediction rates of semi-fresh samples (minority class) are better when the GPA algorithm is applied.

4.3.3 Permutation Tests

Even though thorough model validation and evaluation methods have been applied to ensure that the performance metrics are representative of real world application, "accuracy estimates are usually meaningless without a confidence interval" (Kohavi, 1995; Brereton, 2006; Harrington, 2006).

As a means of providing an indication of the statistical significance of the results, permutation tests were applied. The background of permutation testing was demonstrated in Sections 1.7 and 2.2.4. By randomising the data with respect to the sensory scores (classes), any prior association between the initial data and the classes is destroyed, while their initial distributional properties are preserved (Wu *et al.*, 2002; Westerhuis *et al.*, 2008). As permutation testing is performed repeatedly a large number of times, a reference distribution for the null hypothesis is obtained. The 95% confidence interval (C.I.), which is equal to two standard deviations from the mean, is calculated based on the distribution of permuted classification results. If the observed non-permuted value is higher than both 95% confidence bounds, then the initial result is indeed significant. Metrics such as the p-value are also frequently reported in permutation testing; the p-value is equal to the proportion of permuted values that are at least as good as the observed statistic (Hubert and Schultz, 1976).

In the context of this work, each permutation constitutes a single classification ensemble, which consists of 100 individual classifiers; each of these classifiers includes 100 bootstrapping iterations for the purposes of hyperparameter optimisation. The permutation tests were executed a total of 100 times for each dataset under study, which results to a total of one million iterations per dataset. Under the null hypothesis, the original non-permuted value is considered another random case. Thus, only 99 actual permutations are indeed required, in addition to the observed value, leading to 100 permutations in total; for the specific number of iterations, the lowest possible p-value will be equal to $(0 + 1)/(99 + 1) = 0.01$. Finally, all the permuted samples were drawn as an individual step prior to analysis to assure that the outcome of randomisation is not biased in any way.

Initially, permutation testing was applied solely on nonlinear (RBF) SVMs since the overall performance of these models was considered to be promising. The permutation results for the datasets of case study 1 are illustrated in the histograms of Figure 4-11. For all experimental data under study besides standalone e-nose, the non-permuted overall accuracies are found well above the 95% confidence values; in particular, they are even greater than the 99% confidence intervals. For instance, the non-permuted $\%CC$ for the HPLC data is equal to 79%, which is significantly larger than 51%, the value corresponding to the 95% confidence interval of the permuted distribution. In the same instance, the best performance of the permuted data was noted at 59%, whereas the minimum at 45% as presented in Table 3. On the contrary, the low overall accuracy of the e-nose dataset, equal to 49%, was found below the upper bound of the 95% confidence interval; in this case, the result is considered non-significant since it can be ascribed purely to chance. It can be concluded that e-nose does not have any discriminant power since its prediction ability is not better than that of a random classifier. Finally, these hypotheses are verified by the calculated p-values. The majority of datasets produced p-values equal to 0.01; however, the e-nose dataset returned a larger p-value equal to 0.08. While a p-value of 0.08 is not large, it is rarely regarded as statistically significant.

Kernel-based SVMs verily produced high classification accuracies; however, simplistic linear classifiers such as PLS-DA also demonstrated notable performance. Thus, permutation testing was conducted once more on the datasets of case study 1 using PLS-DA classifiers in order to statistically assess their results. Figure 4-12 depicts the relevant graphs. The PLS-DA permutation distributions demonstrate a similar trend to the SVM classifiers; once more, the outcome of e-nose is consistent with chance, whereas the remaining datasets render statistically significant results. As with RBF SVMs, the majority of datasets produced p-values equal to 0.01 besides the e-nose dataset, which returned a p-value of 0.57.

Even though the interpretation of the individual results from permutation testing is of the utmost importance, a direct comparison of the distributions of different classifiers may help us determine which classification technique is the best. Figure 4-13 and Figure 4-14 attempt to highlight any similarities and/or differences between the various permutation distributions. In addition, Table 3 and Table 4 summarise the most important descriptive statistics of these distributions. Based on the superimposed density curves of Figure 4-13 and the boxplots of Figure 4-14, it is noteworthy that the density estimations of the two classifiers obtain completely different spreads; all PLS-DA distributions cover wider ranges of values hence presenting greater variability and larger spreads, whereas the SVM distributions are clustered much tighter. Therefore, SVMs appear to be more consistent than PLS-DA since their results do not vary as much with each permutation. In addition, the minimum and maximum values of the PLS-DA distribution appear to be relatively extreme comparing to SVMs; in this case, the PLS-DA models achieve both the highest and lowest recorded values. Furthermore, the majority of the distributions are skewed right with the SVMs providing greater symmetry around the median compared to PLS-DA; based on the graphs and the entries of Table 3, a subset of the SVM distributions approximate a nearly Normal distribution. In addition, according to the median values of the density estimates (Table 3 and Table 4), it is obvious that the SVM distributions present higher mean and median than PLS-DA; therefore, the most likely observations ($\%CC$) of the permuted SVMs are considerably higher than the ones achieved by PLS-DA. Thus, we can markedly conclude that the RBF SVMs constitute stronger classifiers in comparison to the PLS-DA models since they generate consistently better results.

Permutation tests on linear SVMs were not implemented due to the long execution times of this process. Since linear and nonlinear SVMs presented similar overall classification accuracies ($\%CC$), RBF SVMs were used for permutation testing for the new case studies.

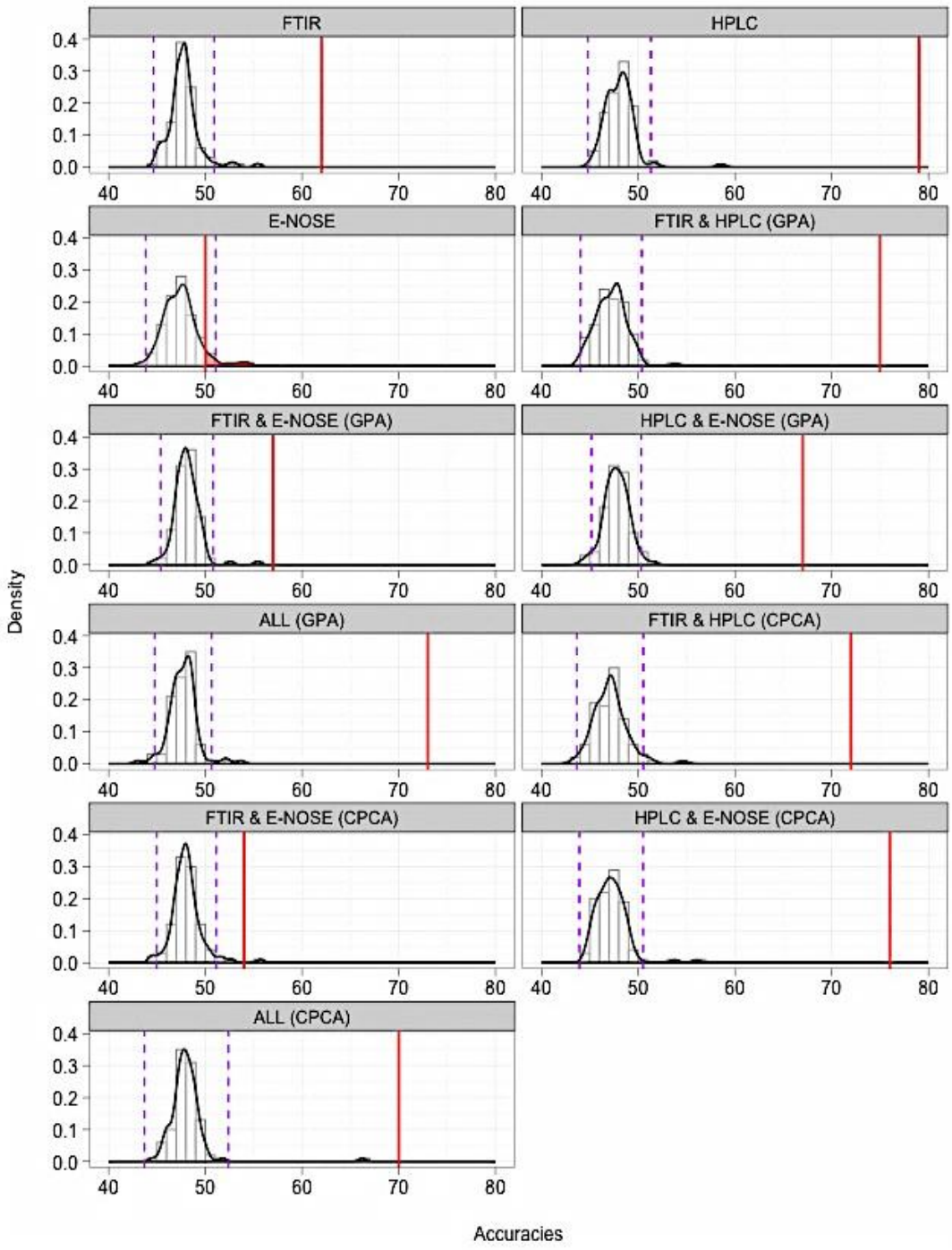


Figure 4-11 Distribution plots of the permutation tests on the datasets of case study 1 using nonlinear (RBF) SVMs

The figure depicts the histograms and density curves of the permuted results for the RBF SVMs, when applied on the datasets of case study 1. The red vertical lines highlight the original non-permuted overall accuracies (%CC) per each standalone or integrated dataset. The dashed purple lines indicate the 95% confidence intervals (two standard deviations from the mean). For the e-nose dataset, the red highlighted area represents the proportion of the distribution that is equal or greater than the observed non-significant value. The permutation distributions of the RBF SVMs approximate a nearly Normal distribution.

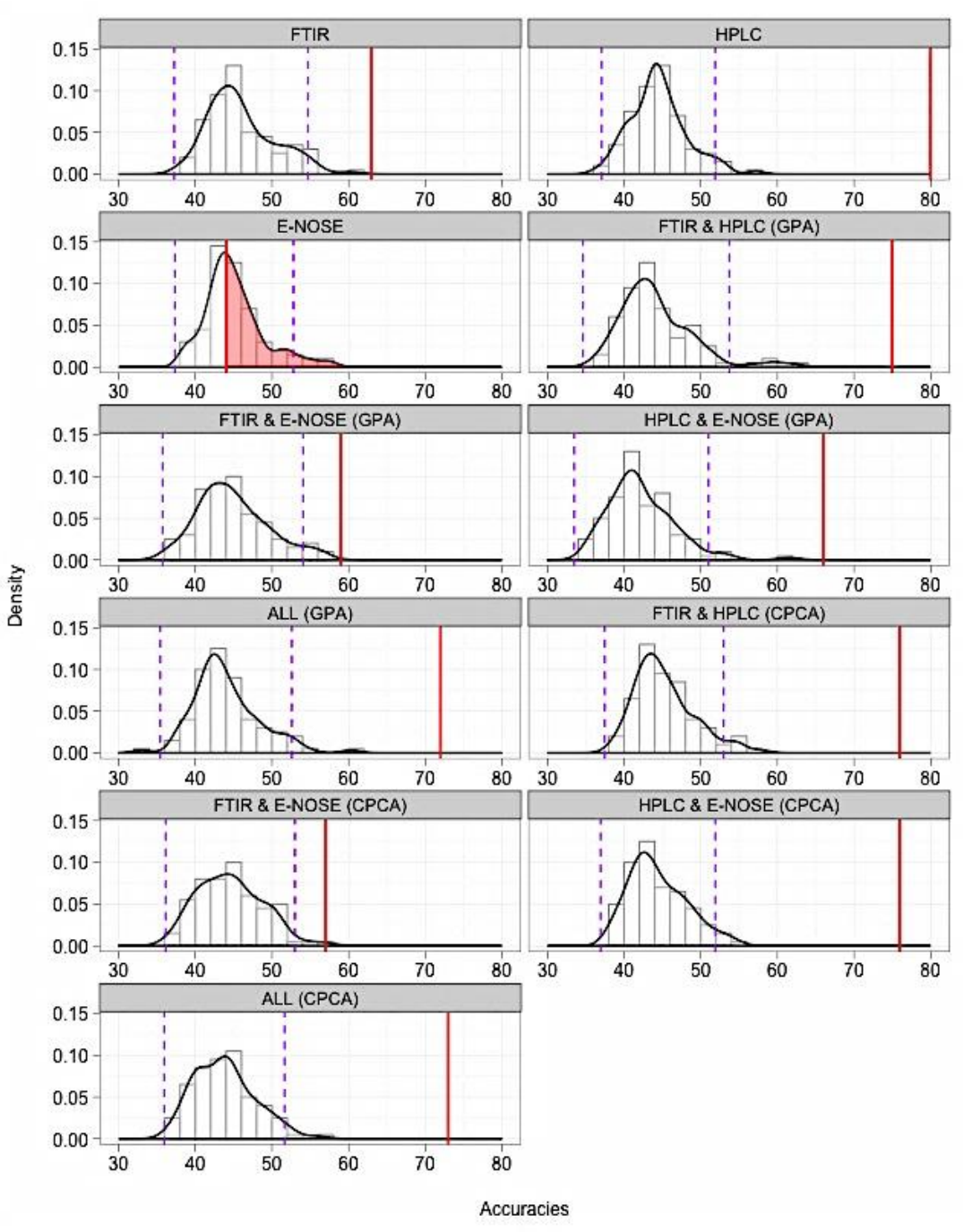


Figure 4-12 Distribution plots of the permutation tests on the data of case study 1 using PLS-DA

The figure depicts the histograms and density curves of the permuted results for the PLS-DA ensembles, when applied on the datasets of case study 1. Similar to the SVM permutation plots, the red vertical lines highlight the initial non-permuted overall accuracies (%CC) per each standalone or integrated dataset. In addition, the dashed purple lines indicate the 95% confidence intervals (two standard deviations from the mean). For the e-nose dataset, the red highlighted area represents the proportion of the distribution that is equal or greater than the observed non-significant value. As with kernel-based SVMs, the result obtained by the analysis of the e-nose dataset is considered non-significant. Finally, based on the graphs, the PLS-DA distributions appear to be skewed right.

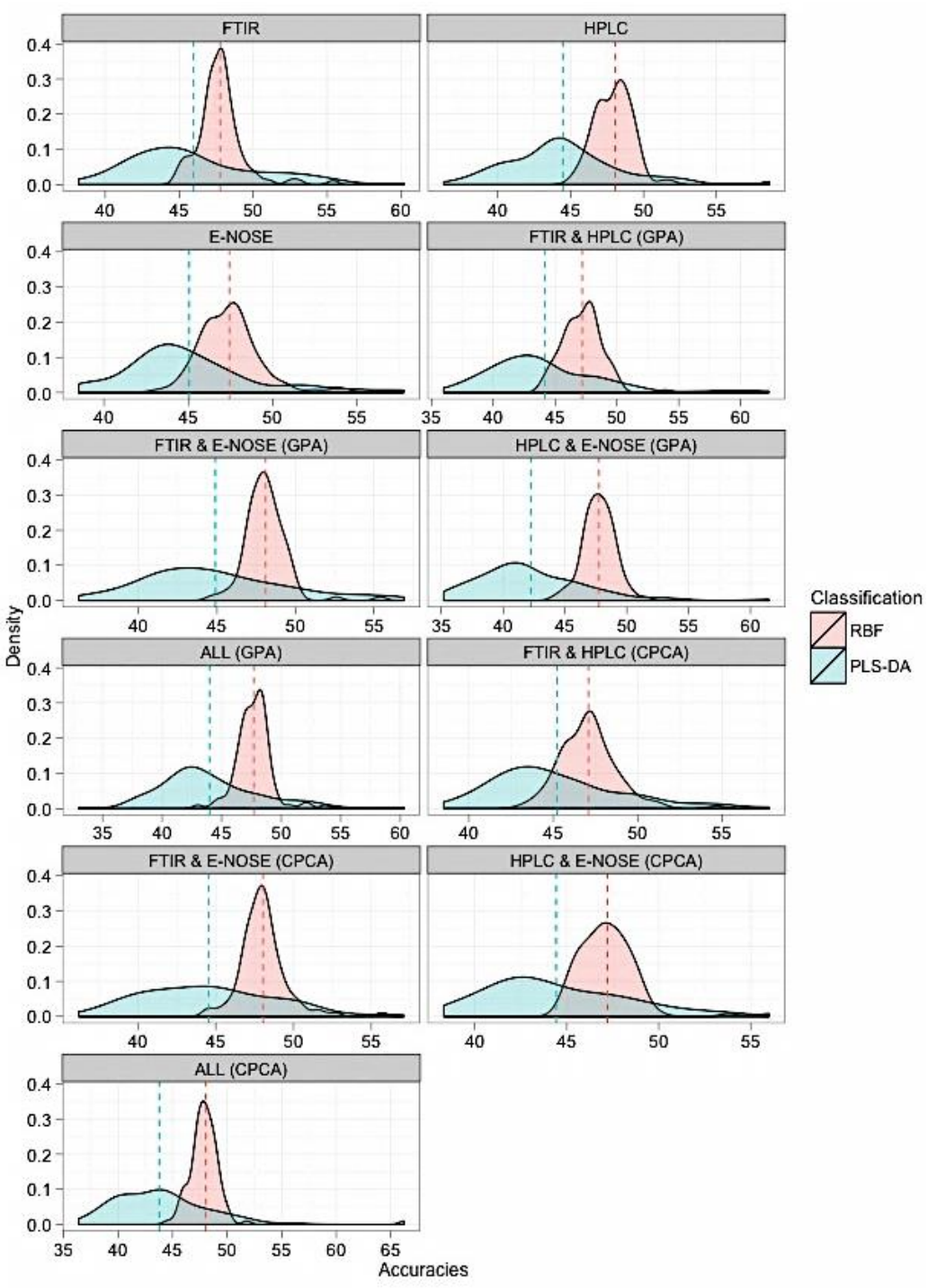


Figure 4-13 Superimposed density plots of the permutation tests on the datasets of case study 1 using PLS-DA and nonlinear (RBF) SVMs

The figure provides a visual comparison of the permutation distributions when different classification models are applied on the datasets of case study 1; the distributions for PLS-DA and SVMs are depicted in a semi-transparent blue and red colour respectively. In these plots, the dashed lines represent the mean values of each density curve and are coloured accordingly. By superimposing the density plots, major differences in the shape, spread and location of the distributions can be identified.

	RBF SVMs					
Datasets	**Original %CC**	**Mean Value**	**Median Value**	**Min Value**	**Max Value**	**Upper 95%C.I.**
FTIR	62%	48 %	48%	45%	55%	51%
HPLC	79%	48%	48%	45%	59%	51%
E-NOSE	50%	47%	48%	43%	54%	51%
FTIR & HPLC (GPA)	75%	47%	47%	44%	54%	50%
FTIR & E-NOSE (GPA)	57%	48%	48%	45%	55%	51%
HPLC & E-NOSE (GPA)	67%	48%	48%	44%	52%	50%
ALL (GPA)	73%	48%	48%	43%	54%	51%
FTIR & HPLC (CPCA)	72%	47%	47%	43%	55%	51%
FTIR & E-NOSE (CPCA)	54%	48%	48%	44%	56%	51%
HPLC&E-NOSE (CPCA)	76%	47%	47%	45%	56%	51%
ALL (CPCA)	70%	48%	48%	44%	66%	52%

Table 3 Descriptive statistics of the permutation distributions obtained by RBF SVMs (case study 1)

The results presented in Table 1 have been rounded towards the nearest integer.

	PLS-DA					
Datasets	**Original %CC**	**Mean Value**	**Median Value**	**Min Value**	**Max Value**	**Upper 95%C.I.**
FTIR	63%	46%	45%	38%	60%	55%
HPLC	80%	44%	44%	36%	57%	52%
E-NOSE	44%	45%	44%	38%	58%	53%
FTIR & HPLC (GPA)	75%	44%	43%	36%	62%	54%
FTIR & E-NOSE (GPA)	59%	45%	44%	36%	57%	54%
HPLC & E-NOSE (GPA)	66%	42%	42%	35%	62%	51%
ALL (GPA)	72%	44%	43%	33%	60%	53%
FTIR & HPLC (CPCA)	76%	45%	44%	38%	58%	53%
FTIR & E-NOSE (CPCA)	57%	45%	44%	36%	57%	53%
HPLC&E-NOSE (CPCA)	76%	44%	44%	38%	55%	52%
ALL (CPCA)	73%	44%	44%	36%	56%	52%

Table 4 Descriptive statistics of the permutation distributions obtained by PLS-DA (case study 1)

The results presented in Table 2 have been rounded towards the nearest integer.

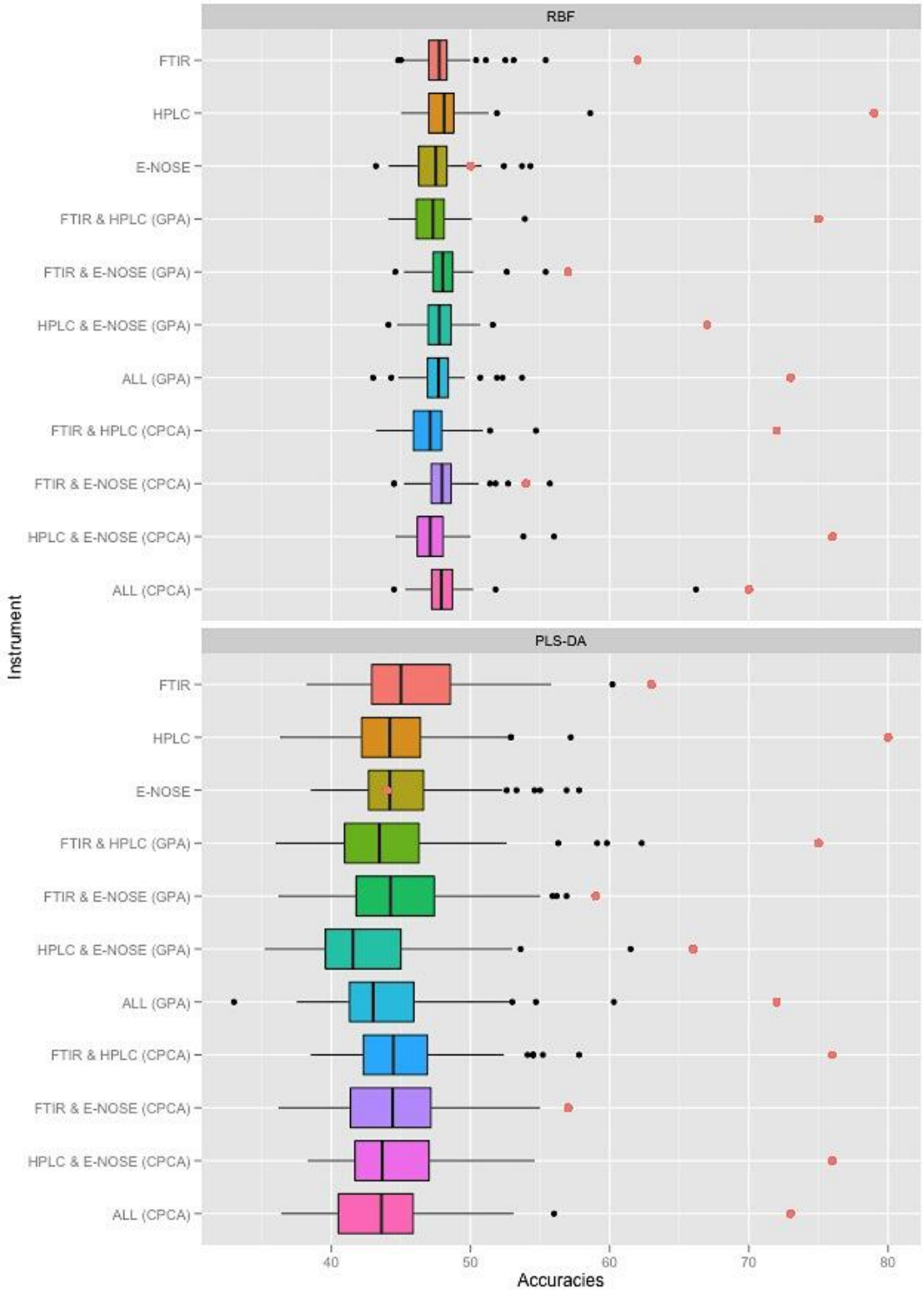


Figure 4-14 Boxplots representing the outcome of permutation testing when PLS-DA and RBF SVMs are applied on the datasets of case study 1

The boxplots provide a powerful visual aid for a straightforward comparison of the descriptive statistics of a given permutation distribution. Each boxplot illustrates the "five-number summary": namely, the minimum, first (lower) quartile, median, third (upper) quartile and maximum value. In addition, the observed non-permuted values are highlighted in a red colour.

4.4 Conclusion

In this chapter, the functionality of the constructed multivariate analysis pipeline was once more extended to incorporate data integration techniques. Various approaches for the fusion of data from the analytical instruments have been evaluated in order to determine whether different instruments provide complementary information, which when brought together in an integrated analysis can provide a more reliable method for spoilage than single instruments. Generalised Procrustes Analysis (GPA) was the first data fusion technique to be investigated. The algorithm attempts to minimise the dissimilarities of the heterogeneous data by applying geometric transformations and simultaneous shape superimposition towards building a single consensus configuration. The alternative technique of consensus PCA (CPCA) was also implemented within the pipeline. The algorithm generates as an output a consensus scores matrix on the "super level".

Prior to permutation testing, no optimal classification method could be determined since the classification results of all different types of classifiers appeared to be equally good. However, in addition to verifying the statistical significance of the obtained results, the outcome of permutation testing clearly established SVMs as more powerful and robust techniques than PLS-DA since they consistently produced higher generalisation accuracies.

The results obtained by GPA and CPCA were found to be greatly similar to each other, with the latter taking precedence in the weak cases presented by Procrustes. The $\%CC$ values obtained by the fused models were compared to the analysis results of the standalone datasets as presented in Chapter 2. For case study 1, HPLC as a standalone technique produced the best overall $\%CC$, equal to 80%; in this instance, the results of both data integration techniques did not accomplish any improvement in the overall classification accuracy since they did not exceed the accuracy of standalone HPLC. However, these findings may only hold true for this particular case study, and thus the pipeline will be further applied on new real-world case studies as a means of establishing its generalisability.

5 Application of the multivariate analysis pipeline on new case studies

5.1 Introduction

Thus far, a novel suite of chemometric tools for the integration and analysis of highly heterogeneous analytical datasets has been designed, implemented and tested upon a single case study ("Shelf life beef fillets stored in air at 0, 5, 10, 15 and 20°C"). This chapter aims to verify the applicability of the constructed multivariate analysis pipeline on several independent case studies, and thus go some way to establishing the generic applicability of the developed tool suite. By reproducing the analyses and making the findings available through comprehensive comparison, we ensure that the implemented tool is representative of real-world application, and that it may be further employed by other scientists to studies and areas of inquiry far wider than the present study.

5.2 Materials and Methods

Thorough information about the experimental protocols of the new case studies can be found in Argyri (2010), Argyri *et al.* (2010), Argyri *et al.* (2011), Argyri *et al*. (2013) and Papadopoulou *et al.* (2011). The analytical techniques used in all case studies are the same as the ones of case study 1; besides Raman spectroscopy, the instruments' technical specifications can be found in Section 2.2.1. In addition, sensory evaluation of the meat samples was performed according to Gill and Jeremiah (1991) as described in Section 2.2.2. Each designated class represents a distinct status of spoilage (fresh, semi-fresh and spoiled samples). The standalone and fused datasets of each case study under investigation were inserted in the implemented analysis pipeline according to Section 4.2.5.

5.2.1 Case study 2: *"Shelf life of minced beef stored in air, MAP, and in active packaging at 0, 5, 10 and 15°C"*

5.2.1.1 Sample Preparation

A detailed explanation of the experimental techniques and the methodology of case study 2 can be found in Argyri (2010) and Argyri *et al*. (2011). In brief, fresh minced beef (pH 5.5) was purchased from a meat market in Athens (Greece) and transported under refrigeration to the laboratory within 30 minutes, where it was stored at 1°C for 1-2 hours. Two portions of 75 g were placed onto Styrofoam trays, where one was used for chemical and microbiological analysis, and the other for sensory assessment. Three packaging conditions were applied on the samples of minced beef: air, modified atmospheres (MAP) (40% CO_2 / 30% O_2 / 30% N_2), and modified atmospheres with the presence of the volatile compounds of oregano essential oil (active packaging). Subsequently, the samples were stored in high precision incubation chambers under 0, 5, 10 and 15°C until spoilage was pronounced (intense discoloration and presence of off-odours). In case study 2, two analytical techniques were employed; namely, FTIR spectroscopy and HPLC.

5.2.1.2 Fourier Transform Infrared Spectroscopy (FTIR)

The spectrometer was programmed to collect spectra in the mid-IR range between 4000 and 400cm^{-1}. Out of the initial 1,869 variables, only the spectra that reveal the metabolic fingerprint of spoilage between the ranges of 1500 to 1000cm^{-1} were extracted and used for classification purposes. Thus, the final FTIR dataset consists of 187 samples (with their replicates) and 259 variables. Based on the provided sensory scores, the 187 samples of FTIR consist of 66 fresh (F), 26 semi-fresh (SF) and 95 spoiled (S) samples. Figure 5-1 shows the mean FTIR spectra of the samples of minced beef per each distinct class in the fingerprint region between 1500 and 1000 cm^{-1}.

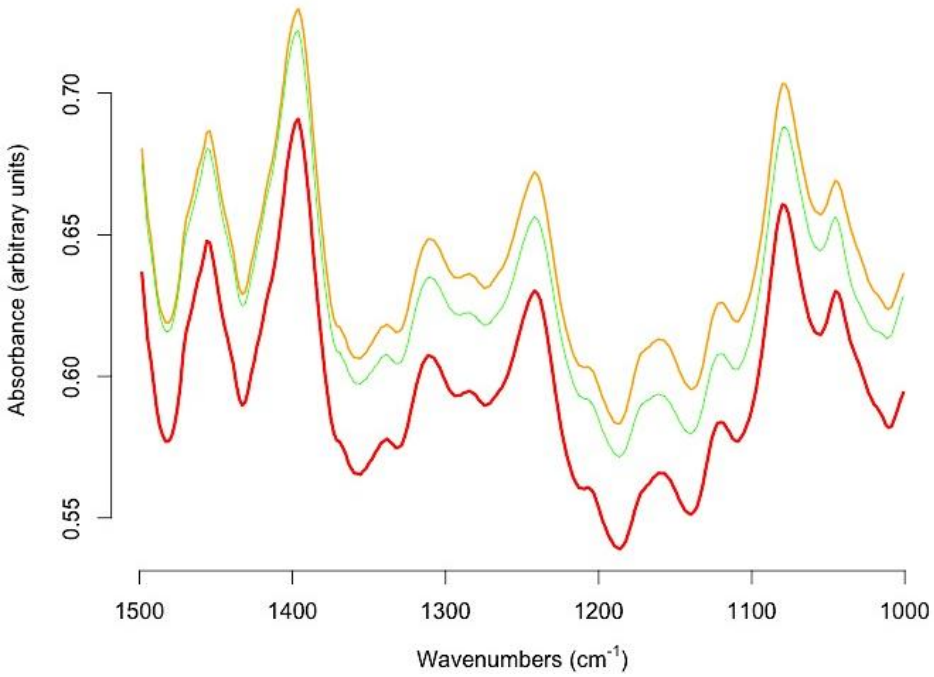


Figure 5-1 Mean FTIR spectra for case study 2 in the fingerprint region (1500-1000 cm^{-1})

The plot depicts the mean FTIR spectra for the samples of shelf life minced beef (stored in air, MAP, and in active packaging at 0, 5, 10 and 15^{o}C) per each distinct class. The spectral region between 1500 and 1000cm^{-1} reveals the metabolic fingerprint of spoilage. Colour representation is used to identify the three classes as determined by the relevant sensory scores: fresh (red colour), semi-fresh (orange colour) and spoiled (green colour).

5.2.1.3 High Throughput Liquid Chromatography (HPLC)

In the case of HPLC, a total of 76 samples were analysed in duplicate. Based on the provided sensory scores, the 76 samples of HPLC consist of 26 fresh (F), 12 semi-fresh (SF) and 38 spoiled (S) samples.

5.2.1.4 Data Overview

For each experimental technique, the total number of samples and variables as well as their data composition as described in the previous sections is summarised in Table 5.

Datasets	FTIR	HPLC
Fresh (F)	66	26
Semi-Fresh (SF)	26	12
Spoiled (S)	95	38
Total #Samples	187	76
Total #Variables	1869	16
Total #Variables (fingerprint region)	259	–

Table 5 The sizes and data composition of standalone datasets from case study 2 prior to analysis

5.2.2 Case study 3: "Survey of minced beef"

5.2.2.1 Sample Preparation

A detailed explanation of the experimental techniques and the methodology of case study 3 can be found in Argyri (2010, 2011, 2013). In brief, fresh minced beef (pH 5.5) was purchased from a meat market in Athens (Greece) and transported under refrigeration to the laboratory within 30 minutes, where it was stored at 1°C for 1-2 hours. Two portions of 75 g were placed onto Styrofoam trays, where one was used for chemical and microbiological analysis, and the other for sensory assessment. Two packaging conditions were applied on the meat samples of case study 3: air and modified atmospheres (MAP) (40% CO_2 / 30% O_2 / 30% N_2). Subsequently, the samples were stored aerobically and under MAP at 5°C until spoilage was pronounced (intense discoloration and presence of off-odours). In case study 3, two analytical techniques were employed; namely, FTIR and Raman spectroscopy.

5.2.2.2 Fourier Transform Infrared Spectroscopy (FTIR)

Similar to case study 1, the FTIR dataset was generated from measurements based on a thin slice of the aerobic upper surface of the beef fillets that was excised and placed in intimate contact with the crystal (Argyri, 2010; Panagou *et al.*, 2010; Argyri *et al.*, 2013). The spectrometer was programmed to collect spectra in the mid-IR range between 4000 and 400cm^{-1}. Out of the initial 1,739 variables, only the spectra that reveal the metabolic fingerprint of spoilage, between the ranges of 1500 and 1000cm^{-1}, were extracted and used for classification purposes. Thus, the final FTIR dataset consists of 150 samples of minced beef with their replicates and 259 variables. Based on the provided sensory scores, the 150 samples of FTIR consist of 28 fresh (F), 52 semi-fresh (SF) and 70 spoiled (S) samples. Figure 5-2 shows the mean FTIR spectra of the samples of minced beef per each distinct class in the fingerprint region between 1500 and 1000 cm^{-1}.

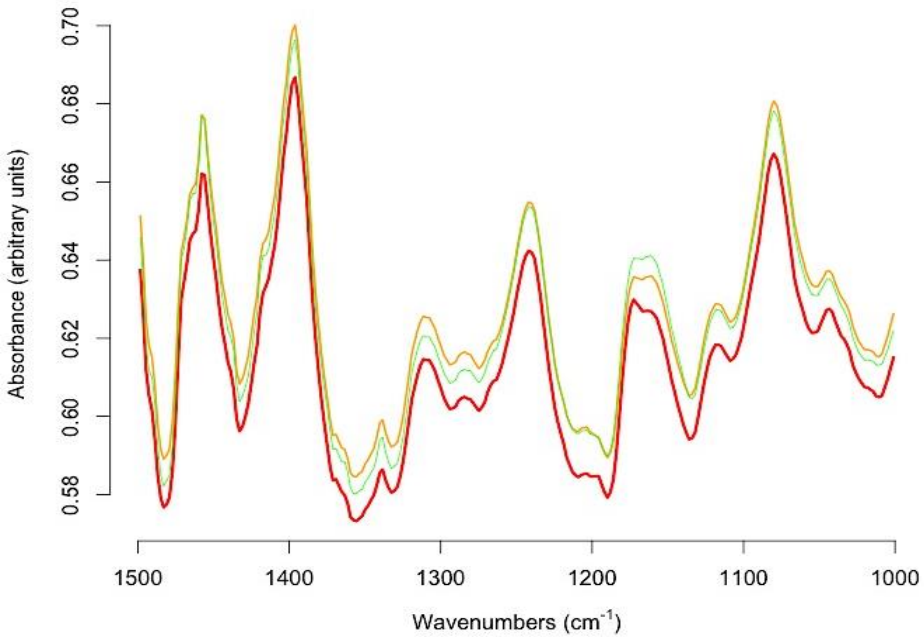


Figure 5-2 Mean FTIR spectra for case study 3 in the fingerprint region (1500-1000 cm^{-1})

The plot depicts the mean FTIR spectra of minced beef samples (stored aerobically and under MAP at 5°C) per each distinct class. The spectral region between 1500 and 1000cm^{-1} reveals the metabolic fingerprint of spoilage. Colour representation is used to identify the three classes as determined by the relevant sensory scores: fresh (red colour), semi-fresh (orange colour) and spoiled (green colour).

5.2.2.3 Raman Spectroscopy

Raman analysis was performed using a 633nm DeltaNu Advantage probe with a right-angled sampling attachment with the aperture positioned 16 mm above the surface of the meat sample delivering ~ 6 mW laser power (Argyri, 2010, 2013). The spectra were acquired over a Stokes Raman shift range between 200 and 3400 cm^{-1} at medium resolution (6 cm^{-1}). The initial Raman dataset comprised 147 samples with their replicates and 1024 variables in total. Out of the initial spectral range, the spectra between the ranges of 400 to 1800cm^{-1} were extracted and used for classification purposes. Thus, the final Raman dataset consists of 147 samples and 450 variables in total. According to the provided sensory scores, the 147 samples of Raman consist of 28 fresh (F), 49 semi-fresh (SF) and 70 spoiled (S) samples. Figure 5-3 shows the mean Raman spectra of the samples of minced beef per each distinct class in the region between 200 and 3400 cm^{-1}.

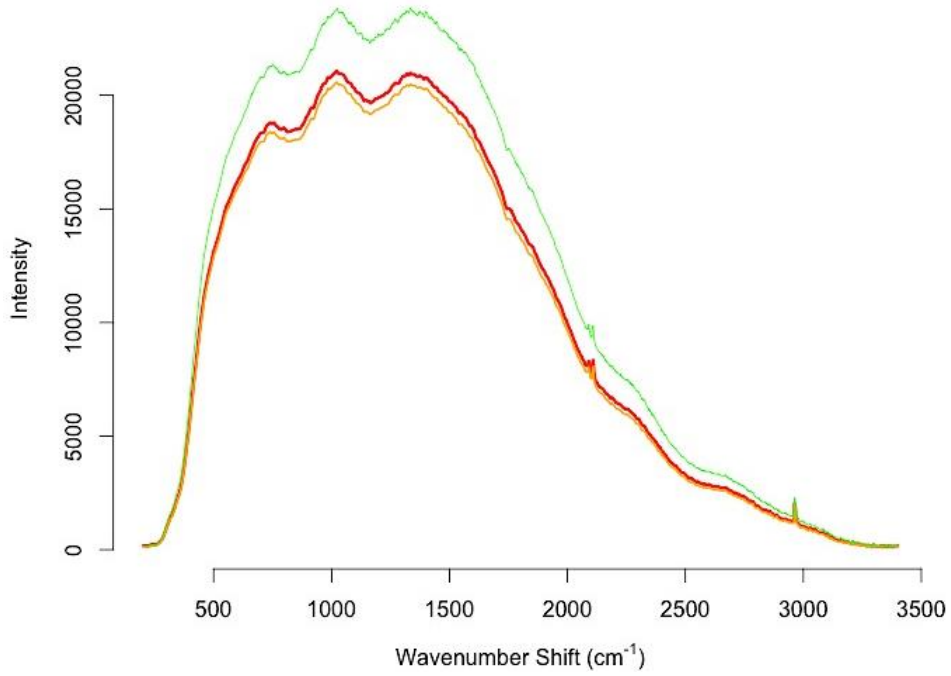


Figure 5-3 Mean Raman spectra for case study 3 in the range 200-3400 cm^{-1}

The plot depicts the mean spectra of minced beef samples (stored aerobically and under MAP at 5°C) per each distinct class. The spectral region between 400 and 1800cm^{-1} reveals the metabolic fingerprint of spoilage. Colour representation was used to identify the three classes as determined by the relevant sensory scores: fresh (red colour), semi-fresh (orange colour) and spoiled (green colour).

5.2.2.4 Data Overview

For each experimental technique, the total number of samples and variables as well as their data composition as described in the previous sections is summarised in Table 6.

Datasets	FTIR	Raman
Fresh (F)	28	28
Semi-Fresh (SF)	52	49
Spoiled (S)	70	70
Total #Samples	150	147
Total #Variables	1739	1024
Total #Variables (fingerprint region)	259	450

Table 6 The sizes and data composition of standalone datasets from case study 3 prior to analysis

5.2.3 Case study 4: "Pork stored in air and MAP"

5.2.3.1 Sample Preparation

A detailed explanation of the experimental techniques and the methodology of case study 4 can be found in Papadopoulou *et al.* (2011). In brief, fresh minced pork (pH 5.6 – 5.8) was purchased right after grinding from a meat market in Athens (Greece) and transported under refrigeration to the laboratory within 30 minutes. Two portions of 50 g were placed onto Styrofoam trays, where one was used for chemical and microbiological analysis, and the other for sensory assessment. The samples were packaged and stored aerobically and under MAP in high precision incubators at 0, 5, 10, and 15°C for up to 340 hours, until spoilage was pronounced (intense discoloration and presence of off-odours). In case study 4, three analytical techniques were employed; namely, FTIR spectroscopy, HPLC and e-nose.

5.2.3.2 Fourier Transform Infrared Spectroscopy (FTIR)

Similar to the previous case studies, the spectrometer was programmed to collect spectra in the mid-IR range between 4000 and 400cm^{-1}. Out of the initial 1,739 variables, only the spectra that reveal the metabolic fingerprint of spoilage, between the ranges of 1500 and 1000cm^{-1}, were extracted and used for classification purposes. The final FTIR dataset consists of 150 samples of minced beef with their replicates and 259 variables. Based on the provided sensory scores, the 150 samples of FTIR are classified into 18 fresh (F), 53 semi-fresh (SF) and 62 spoiled (S) samples. Figure 5-2 shows the mean FTIR spectra of the pork samples per each distinct class in the fingerprint region between 1500 and 1000 cm^{-1}.

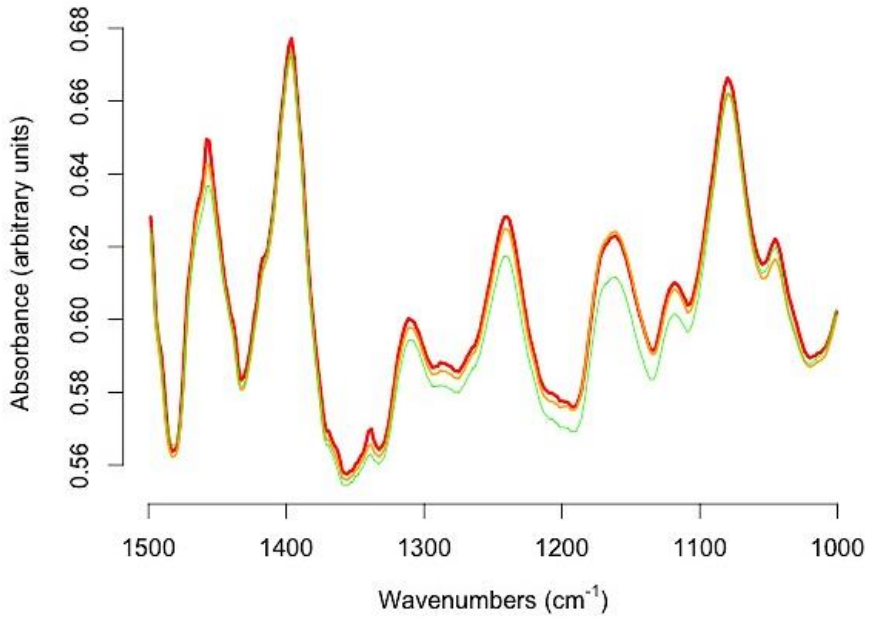


Figure 5-4 Mean FTIR spectra for case study 4 in the fingerprint region (1500-1000 cm^{-1})

The plot depicts the mean FTIR spectra of pork samples (stored aerobically and under MAP at 0, 5, 10, and 15°C) per each distinct class. The spectral region between 1500 and 1000cm^{-1} reveals the metabolic fingerprint of spoilage. Colour representation is used to identify the three classes as determined by the relevant sensory scores: fresh (red colour), semi-fresh (orange colour) and spoiled (green colour).

5.2.3.3 High Throughput Liquid Chromatography (HPLC)

In the case of HPLC, a total of 174 samples were analysed in duplicate. Based on the provided sensory scores, the 174 samples of HPLC consist of 26 fresh (F), 65 semi-fresh (SF) and 83 spoiled (S) samples.

5.2.3.4 Electronic nose (e-nose)

In the case of e-nose, measurements of 90 samples (with their four replicates) using eight sensors were provided; based on the provided sensory scores, the 90 samples of e-nose consist of 18 fresh (F), 40 semi-fresh (SF) and 32 spoiled (S) samples.

5.2.3.5 Data Overview

For each experimental technique, the total number of samples and variables as well as their data composition as described in the previous sections is summarised in Table 7.

Datasets	FTIR	HPLC	e-nose
Fresh (F)	18	26	18
Semi-Fresh (SF)	53	65	40
Spoiled (S)	62	83	32
Total #Samples	133	174	90
Total #Variables	468	17	8
Total #Variables (fingerprint region)	259	–	–

Table 7 The sizes and data composition of standalone datasets from case study 4 prior to analysis

5.2.4 The architecture

The implemented multivariate analysis pipeline was tested on the new case studies on an Apple Mac Mini under the operating system Mac OS X version 10.8.2, running on a 2.3 GHz quad-core Intel Core i7 processor and 4 GB memory.

5.3 Results and Discussion

5.3.1 Case study 2

In case study 2 ("Shelf life of minced beef stored in air, MAP, and in active packaging at 0, 5, 10 and 15°C"), data have been acquired from two main experimental techniques: FTIR spectroscopy and high performance liquid chromatography (HPLC). The data intersection approach presented in Section 2.3.1 extracted a total of 75 common samples along with their respective sensory scores, which were inserted in the analysis pipeline; these samples consist of 25 fresh (F), 12 semi-fresh (SF) and 38 spoiled (S) samples. In this instance, the spoiled samples constitute the majority class, whereas the semi-fresh samples the minority class.

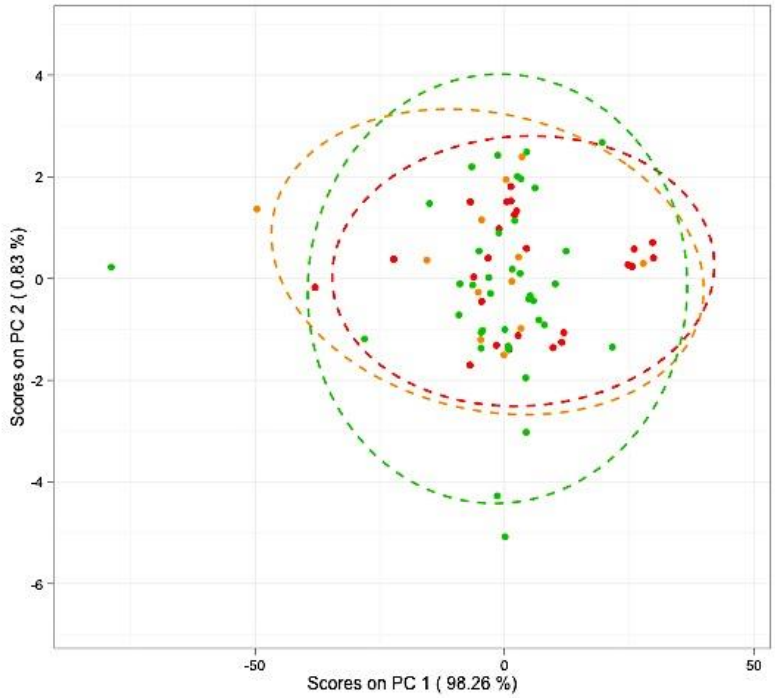
Initially, PCA was applied on standalone pre-processed data for dimensionality reduction purposes in addition to investigating any underlying patterns in the data. The percentages of variance and cumulative variance for each experimental technique are presented in Table 8. In the case of FTIR, the first two PCs account for 99% of the variance. On the contrary, the HPLC data accumulate only 53.23% of the variance for the first three PCs, while at least five PCs are required to capture approximately 70% of the total variance. It is apparent that the percentages of variance and cumulative variance present a similar trend to the datasets of case study 1 (See Table 2)

	FTIR		**HPLC**	
PCs	%Var	%Cum Var	%Var	%Cum Var
PC1	98.26	98.26	25.89	25.89
PC2	0.83	99.09	14.12	40.01
PC3	0.33	99.42	13.22	53.23
PC4	0.21	99.63	9.22	62.45
PC5	0.16	99.79	7.87	70.32

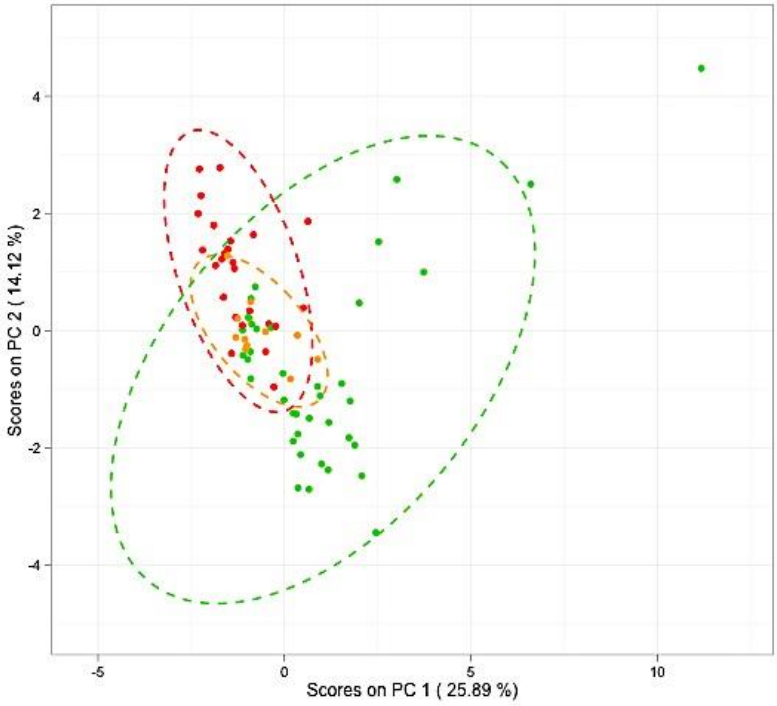
Table 8 PCA proportion and cumulative variance captured for the datasets of case study 2

Figure 5-5 displays the PCA scores plots for the two datasets under study. Similar to Section 2.3.1, 95% confidence ellipses for each distinct class were added to the plot in order to highlight the density of the samples within a single class in addition to the formation of any clusters and/or outliers. In the case of FTIR, no distinct clusters can be identified since the different types of samples are notably overlapping; in addition, several outlying samples for each distinct class, which are located outside the confidence ellipses, can be identified in the plot. In addition, the first two PCs for HPLC do not demonstrate any well-defined clusters, however, they do provide a better separation between fresh and spoiled samples; in this instance, the number of overlapping samples is smaller than FTIR.

Data integration of the datasets originating from FTIR and HPLC was performed using GPA and CPCA as described in Section 4.2.5. The geometrically transformed consensus of GPA is depicted in Figure 5-6. The outcome of CPCA – the super scores – is plotted along with the output of GPA. As can be concluded from the figure, both integrated datasets provide a relatively good separation between the fresh and spoiled samples with the exception of a miniscule of outliers. However, semi-fresh samples are once more overlapping in-between the other two classes.



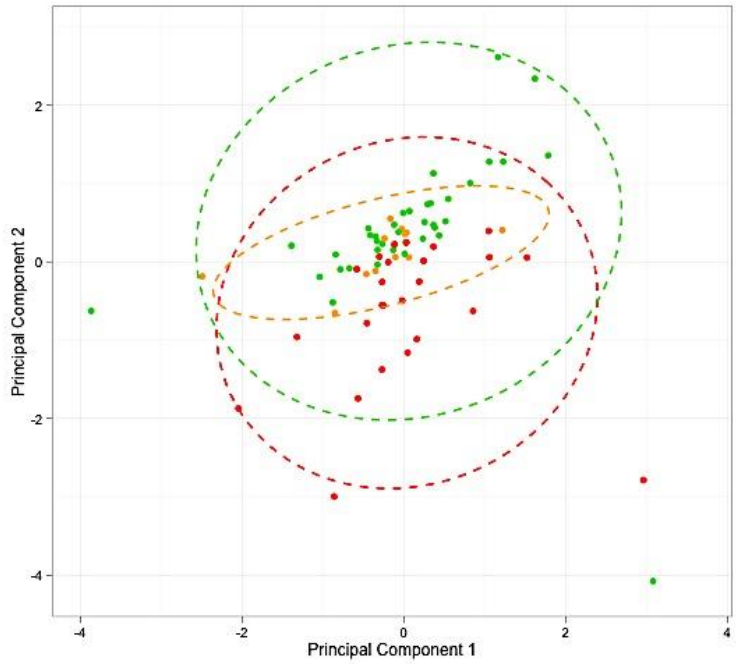
(a) FTIR data



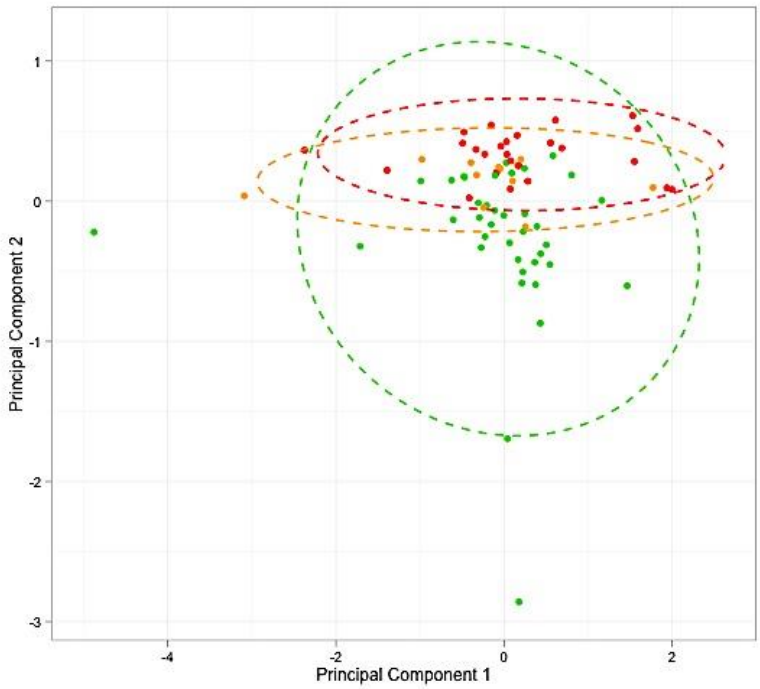
(b) HPLC data

Figure 5-5 PCA scores plots with 95% confidence ellipses for case study 2

The two-dimensional scatterplots illustrate the scores of the first two PCs. Dynamically generated 95% confidence ellipses per each class were added in the plots in order to highlight the presence of any clusters and/or outliers. The colour representation used in the plot is similar to Figure 2-5. For comparison purposes, only the 75 common samples of shelf life minced beef (25 fresh, 12 semi-fresh and 38 spoiled samples) are depicted in each plot.



(a) GPA



(b) CPCA

Figure 5-6 The consensus of the first two Principal Components based on the fusion of the two experimental techniques from case study 2 using GPA and CPCA respectively

The consensus of GPA is compared against the super-scores of CPCA in two-dimensional space (scores of the first two PCs). It is obvious that the most discriminative experimental technique (HPLC) has influenced the outcome of the data fusion by providing good separation between fresh and spoiled samples.

The classification results of the standalone and integrated datasets of case study 2 are presented in Figure 5-7 as percentages of correctly classified samples ($\%CC$). In the case of FTIR, the overall accuracy obtained by the RBF SVMs is extremely low compared to the other two linear techniques. It can be therefore concluded that the projection of the data into a high dimensional space by the kernel-based SVM is unsuitable in this instance. On the contrary, PLS-DA achieves 66%, the highest overall accuracy among the three techniques; as stated in Section 1.5.2, this result is justified by the fact that PLS-DA uses all available samples instead of a limited number of samples (the support vectors) to create the separation boundaries (Boser *et al.*, 1992; Brereton *et al.*, 2009; Smolinska *et al.*, 2012); thus, it is less prone to be dominated by the majority class and the highly imbalanced classes. Furthermore, when the FTIR data were subjected to PCA, the overall accuracy of PLS-DA increased by 3% and by at least 10% in the case of SVMs. Even though PCA notably enhances the overall performance of all implemented classifiers, linear SVMs once more take the lead among the three classification ensembles.

Both linear and RBF SVMs produce an overall accuracy equal to 71% for the HPLC data, which is significantly higher than the overall accuracy of 67% obtained by PLS-DA. According to Section 1.5.2.3, the performance of a nonlinear SVM can be at least as good as the linear SVM (Boser *et al.*, 1992; Keerthi and Lin, 2003; Hsu *et al.*, 2003; Chang *et al.*, 2010); thus, suitable combinations of the RBF hyperparameters (C and γ) have formed boundaries that tend towards linearity and hence result in equal overall accuracies between the linear and nonlinear SVMs (Figure 5-7). In addition, the application of PCA has proven extremely fruitful in the case of nonlinear SVMs, where the highest recorded accuracy, equal to 73%, is observed.

For case study 2, data integration was not as fruitful as expected. The documented $\%CC$ achieved by both GPA and CPCA do not exceed the highest recorded accuracy of 73% by standalone HPLC. However, in the case of CPCA and the linear classifiers (PLS-DA and linear SVMs), the overall accuracies of the integrated datasets are at least as good as those acquired by the standalone datasets. In the case of GPA, all results for the integrated dataset are lower than those of the individual datasets. This

observation confirms that indeed CPCA is a better data fusion technique compared to GPA.

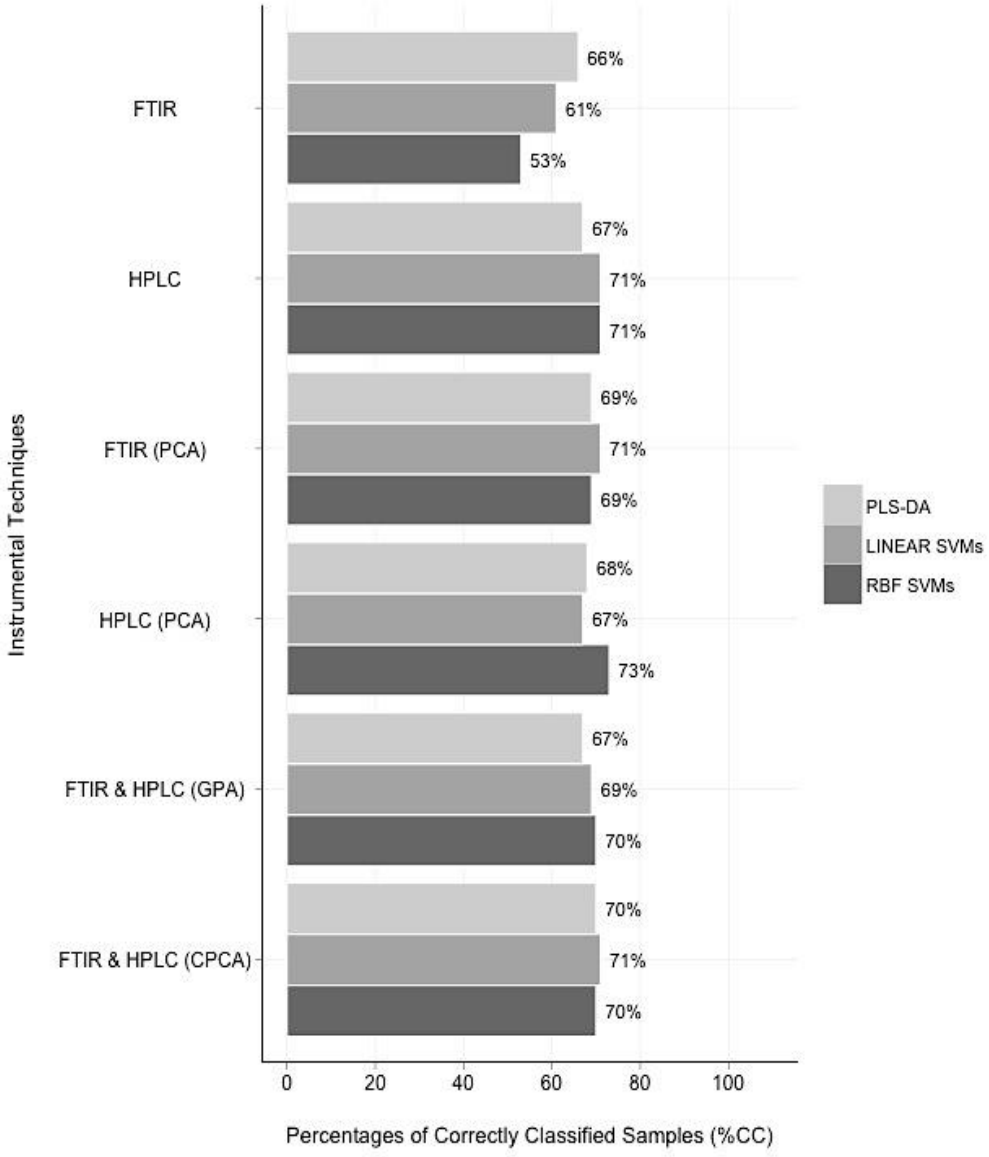


Figure 5-7 Overall accuracies (%CC) for the standalone and integrated datasets of case study 2

The figure illustrates the overall performance of all implemented classification ensembles on the standalone and integrated datasets of case study 2. The bars represent the percentages of correctly classified samples (%CC) and are coloured according to the classification model under study (PLS-DA, linear and RBF SVMs). In the case of standalone datasets, analyses have been conducted both prior (raw data) and after PCA. Data integration has been performed using both Generalized Procrustes Analysis (GPA) and Consensus PCA (CPCA). In all implemented classifiers, bootstrapping was applied for hyperparameter optimisation. The overall accuracies have been rounded towards the nearest integer.

As established thus far, assessing the performance of a classifier solely based on the overall accuracies may be occasionally misleading. A closer inspection of the class predictions may reveal significant differences in the classifiers' ability to discriminate the individual classes. The class rates for the datasets of case study 2 are depicted in Figure 5-8. Based on the graphs, the class predictions of fresh and semi-fresh samples in the case of PLS-DA and linear SVMs verify that the FTIR present higher accuracies for linear classifiers; on the contrary, as expected, the nonlinear SVMs strongly favour the majority class (spoiled samples), while the predictions of the minority class (semi-fresh samples) are equal to 0%. It is noteworthy that the application of PCA on the FTIR dataset provides a better separation of the data since the accuracies of fresh samples improve by at least 5% for the linear classifiers and by 50% for the nonlinear classifiers. Even so, the percentages of correctly classified semi-fresh samples remain relatively low. Prior to PCA, the HPLC data generate considerably higher class accuracies than FTIR, with equally good predictions for fresh and spoiled samples. Both linear and nonlinear SVMs demonstrate higher class rates for the majority class compared to PLS-DA; this suggests that the SVM boundaries are strongly skewed towards the minority class (semi-fresh samples) due to the disproportionate class ratios. Once the data are subjected to PCA, the class accuracies of spoiled samples increase by at least 10% leading to an excessively high rate of 95%. However, it is interesting to note that the percentages of fresh and semi-fresh samples decrease markedly, with only the exception of PLS-DA that retains good class predictions.

In the majority of cases, the integrated dataset obtained by GPA produces lower class accuracies than those by the standalone techniques. In this instance, PLS-DA produces equally good accuracies for fresh and spoiled samples, while the SVMs are clearly dominated by the majority class. On the contrary, the CPCA algorithm achieved notable predictions for the fresh samples, which are significantly higher than those of the standalone techniques. In addition, the low accuracies of the semi-fresh samples improved, while the prediction rates of spoiled samples remained stable. Thus, we can conclude that CPCA minimises the negative impact of highly imbalanced datasets by combining the strongest assets of the individual origins, such as HPLC.

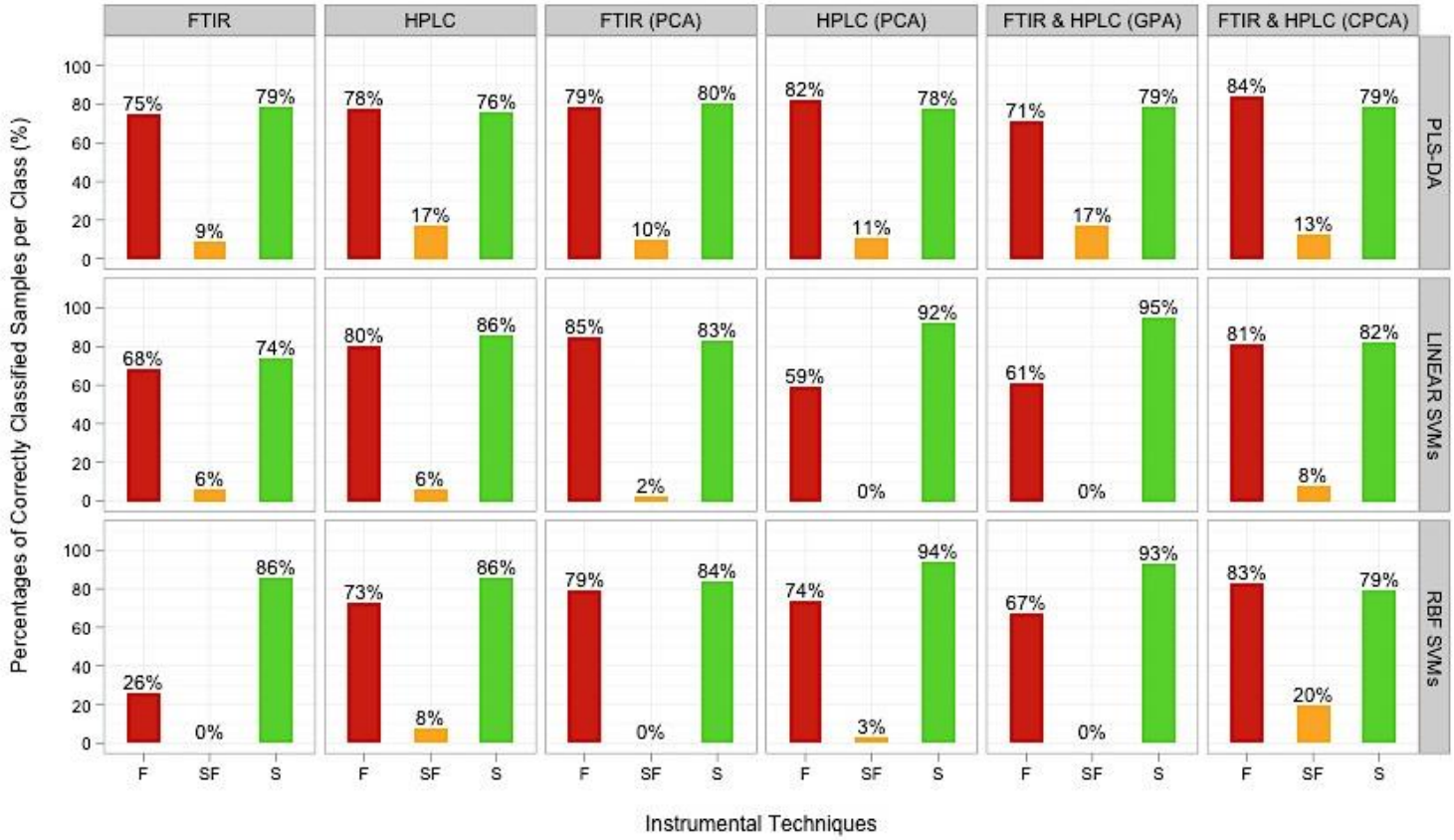
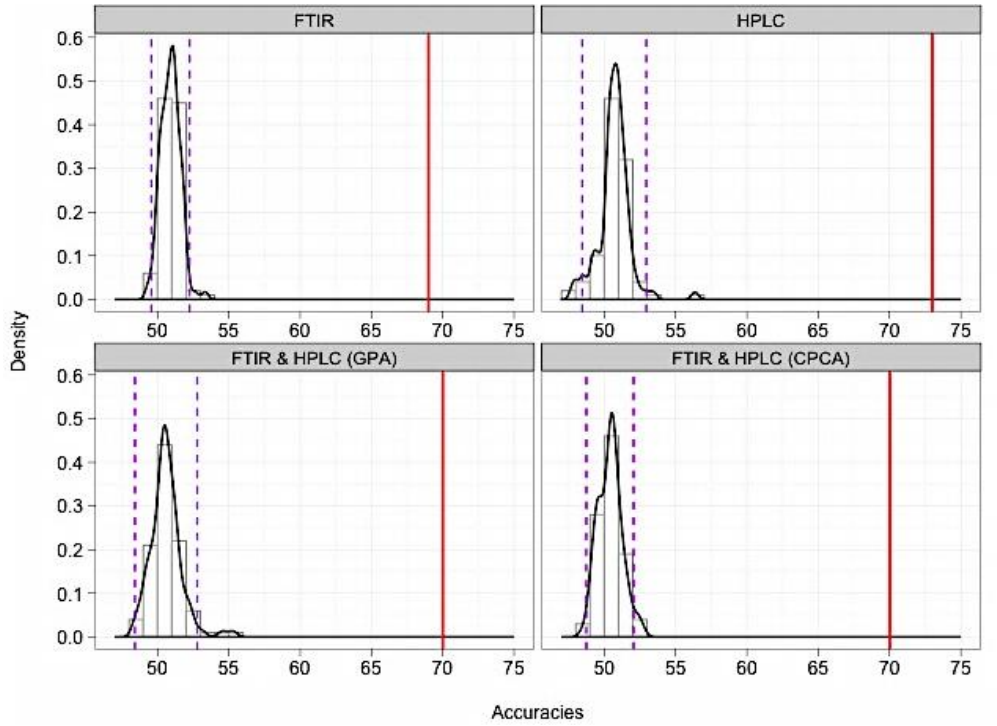


Figure 5-8 Class prediction rates of the standalone (prior and after PCA) and integrated datasets for case study 2

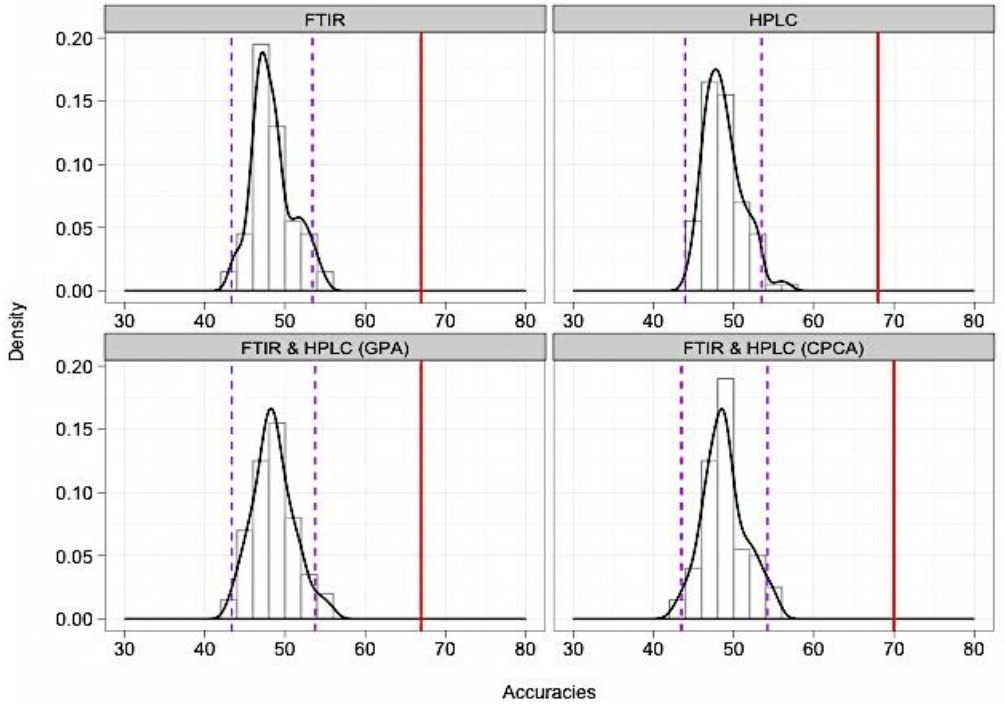
The figure illustrates the percentages of correctly classified samples per each distinct class, when the standalone and integrated datasets of case study 2 are imported into the analysis pipeline. The class predictions are compared in respect to the standalone and fused techniques, and classification models. Based on the graph, it is apparent that semi-fresh samples are consistently difficult to predict.

In order to acquire statistical confidence in the obtained results, permutation testing was applied on the standalone and integrated datasets of case study 2 using RBF SVMs and PLS-DA. According to Figure 5-9, for all experimental data under study, when either linear or nonlinear classifiers are employed, the initial non-permuted overall accuracies ($\%CC$) are found well above the 95% confidence values; more specifically, the percentages of correctly classified samples ($\%CC$) are even greater than the 99% confidence intervals. For all instances, the p-values are equal to 0.01.

Similar to the permutation results of case study 1, RBF SVMs and PLS-DA demonstrate great differences in their permutation distributions. Based on Figure 5-10 and Figure 5-11, PLS-DA covers wider ranges (larger spread) and greater variability than SVMs; once more, the lowest and highest permuted values are recorded for PLS-DA. On the contrary, the SVM distributions demonstrate a smaller spread and hence greater consistency in the results. In addition, based on the entries of Table 9 and Table 10, the permutations of the nonlinear SVMs present a higher mean and median (higher centre) than PLS-DA. Thus, we can conclude that the RBF SVMs constitute more powerful classifiers in comparison to the PLS-DA models since they generate consistently higher classification accuracies. Currently, permutation testing (100 independent permutation tests) for a single dataset is completed within a few hours, as illustrated in Figure 5-12.



(a) RBF SVMs



(b) PLS-DA

Figure 5-9 Distribution plots of the permutation tests on the datasets of case study 2 using RBF SVMs and PLS-DA respectively

The figure depicts the histograms and density curves of the permuted results for the RBF SVMs and PLS-DA ensembles respectively, when applied on the datasets of case study 2. The red vertical lines highlight the original non-permuted overall accuracies (%CC). In addition, the dashed purple lines indicate the 95% confidence intervals. In all instances, the non-permuted results are observed well above the 95% confidence intervals, thus the original results are confirmed as statistically significant.

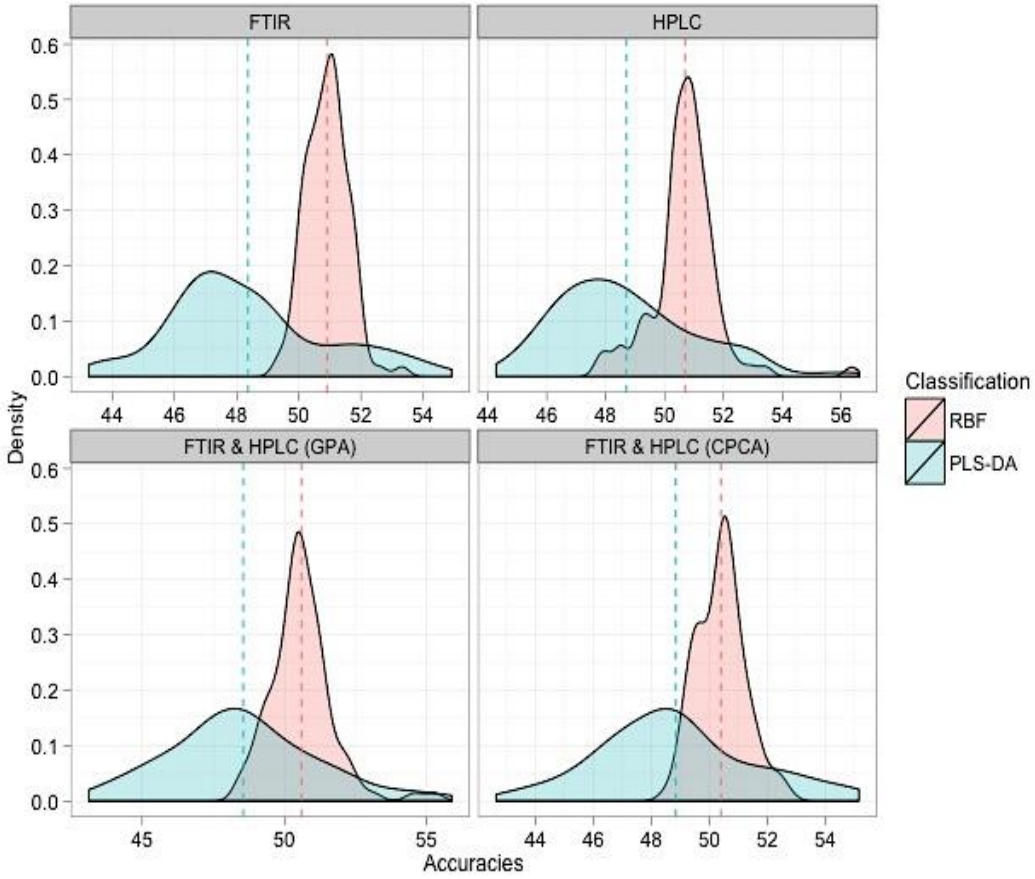


Figure 5-10 Superimposed density plots of the permutation tests on the datasets of case study 2 using PLS-DA and nonlinear (RBF) SVMs

The figure provides a visual comparison of the permutation distributions when different classification models are applied on the datasets of case study 2; the distributions for PLS-DA and SVMs are depicted in a semi-transparent blue and red colour respectively. In these plots, the dashed lines represent the mean values of each density curve and are coloured accordingly. By superimposing the density plots, major differences in the shape, spread and location of the distributions can be identified.

	RBF SVMs					
Datasets	**Original %CC**	**Mean Value**	**Median Value**	**Min Value**	**Max Value**	**Upper 95%C.I.**
FTIR	69%	51%	51%	49%	53%	52%
HPLC	73%	51%	51%	48%	56%	53%
FTIR & HPLC (GPA)	70%	51%	51%	48%	55%	53%
FTIR & HPLC (CPCA)	70%	50%	50%	49%	53%	52%

Table 9 Descriptive statistics of the permutation distributions obtained by RBF SVMs (case study 2)

The results presented in Table 5 have been rounded towards the nearest integer.

	PLS-DA					
Datasets	**Original %CC**	**Mean Value**	**Median Value**	**Min Value**	**Max Value**	**Upper 95%C.I.**
FTIR	67%	48%	48%	43%	55%	53%
HPLC	68%	49%	48%	44%	57%	54%
FTIR & HPLC (GPA)	67%	49%	48%	43%	56%	54%
FTIR & HPLC (CPCA)	70%	49%	49%	43%	55%	54%

Table 10 Descriptive statistics of the permutation distributions obtained by PLS-DA (case study 2)

The results presented in Table 6 have been rounded towards the nearest integer.

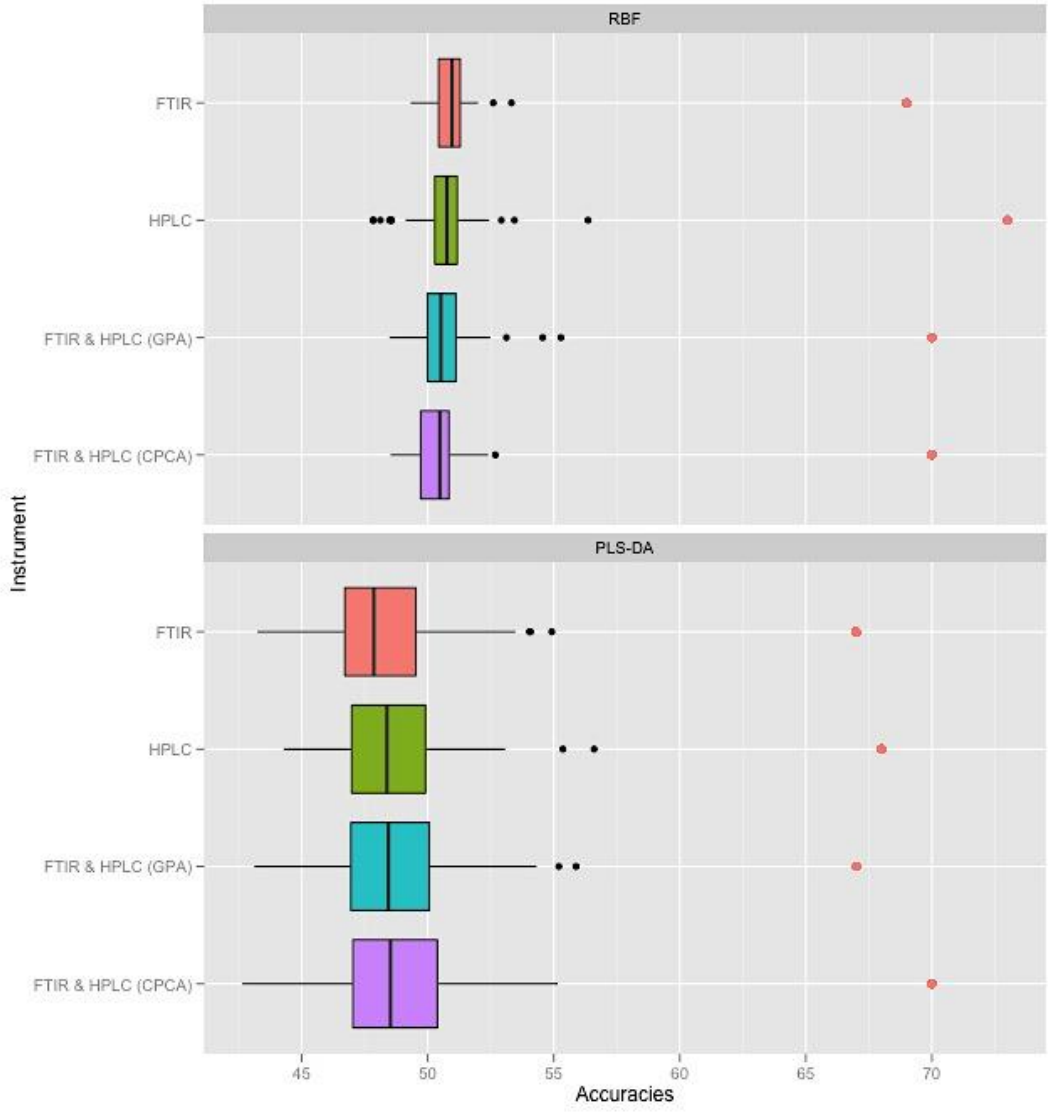


Figure 5-11 Boxplots representing the outcome of permutation testing when RBF SVMs and PLS-DA are applied on the datasets of case study 2

The boxplots provide a powerful visual aid for a straightforward comparison of the descriptive statistics of a given permutation distribution. Each boxplot illustrates the "five-number summary": namely, the minimum, first (lower) quartile, median, third (upper) quartile and maximum value. In addition, the observed non-permuted values are highlighted in a red colour.

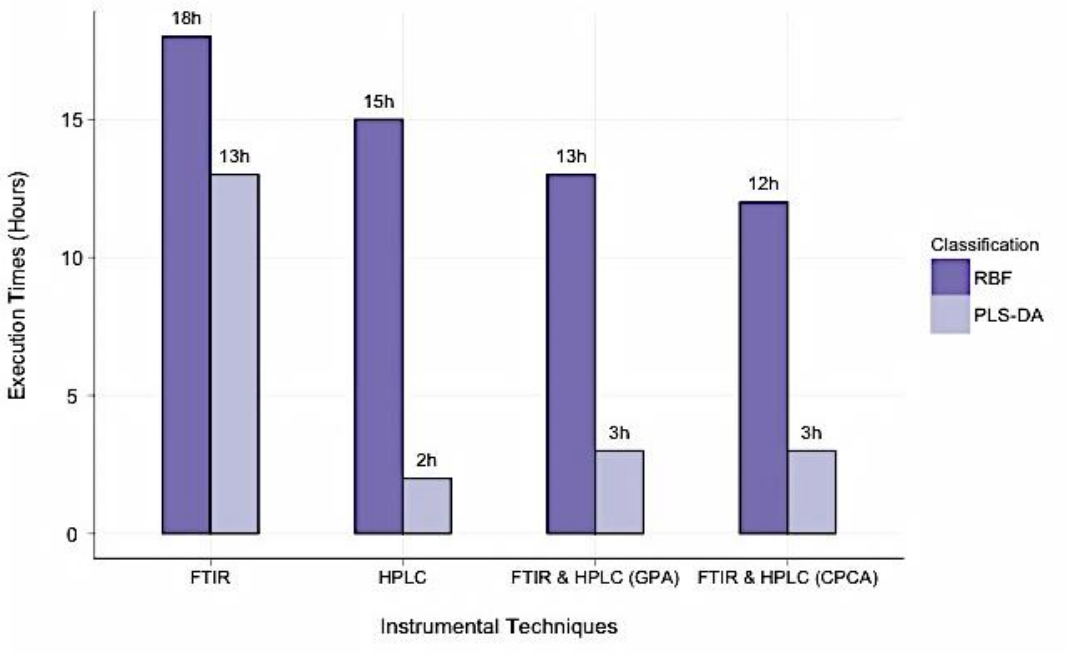


Figure 5-12 Execution times of the permutation tests on the datasets of case study 2

The figure displays the execution times of 100 permutation tests using PLS-DA and RBF SVMs for each standalone and integrated dataset of case study 2. The execution times are based on a fully optimised analysis pipeline featuring parallel programming (master/slave architecture) over eight processors (see Section 5.2.4) as well as fast approximation algorithms for the optimisation of the classifiers' hyperparameters via bootstrapping. The execution times have been rounded towards the nearest integer.

5.3.2 Case study 3

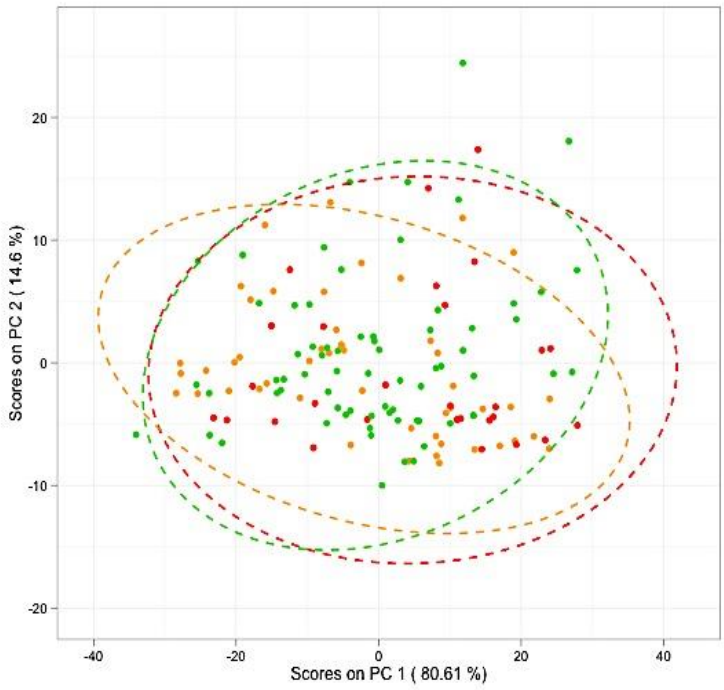
In case study 3 ("Survey of minced beef"), data have been acquired from two analytical techniques: FTIR and Raman spectroscopy. The data intersection approach presented in Section 2.2.2 extracted a total of 147 common samples along with their respective sensory scores, which were imported into the analysis pipeline; these samples consist of 28 fresh (F), 49 semi-fresh (SF) and 70 spoiled (S) samples. In this instance, the spoiled samples constitute the majority class, whereas fresh samples the minority class. It is crucial to note that this dataset is highly imbalanced with ratios that strongly favour the spoiled samples. As a result, these disproportionate class distributions may profoundly influence the performance of the classifiers.

PCA was applied for dimensionality reduction and feature extraction purposes on each of the standalone pre-processed data. The percentages of variance and cumulative variance are presented in Table 11. In the case of Raman, the first two PCs account for nearly 100% of the variance. Similarly, the FTIR data explain 97% of the total variance for the first three PCs. Thus, the data can be graphically represented using only the first two or three PCs without loosing any valuable information. The scores plots of PCA for the two analytical techniques are depicted in Figure 5-13. In both cases, no obvious clusters or any type of visual discrimination is noted between the different classes. The samples of both FTIR and Raman are widely scattered and highly overlapping. Since both datasets require only the first three PCs to reach at least 97% of the variance, the following PCs will not have much discriminative information to add.

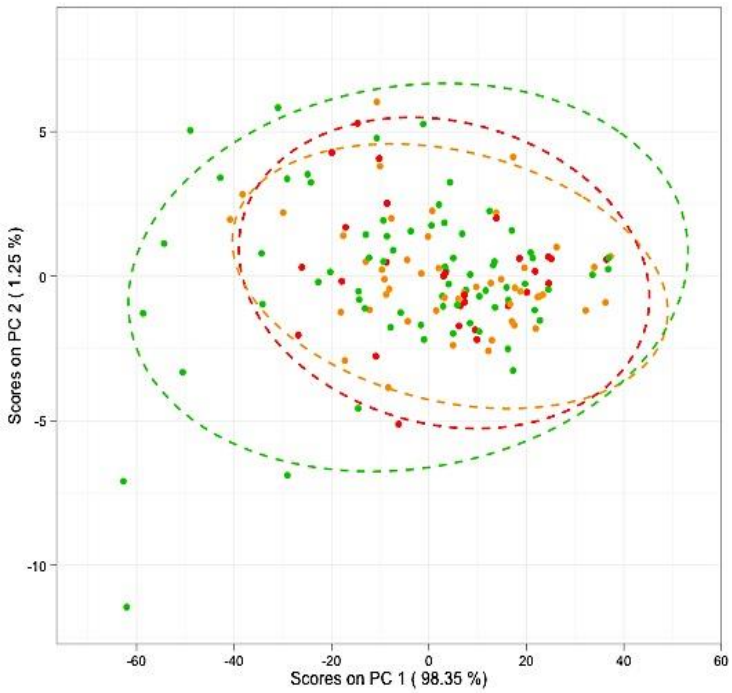
In addition to the standalone datasets, data integration was also performed on the experimental techniques of case study 3 according to Section 4.2.5 and Section 5.3.1. A graphical comparison of the consensus as generated by GPA and CPCA respectively is displayed in Figure 5-14. As with standalone data, the fused datasets do not demonstrate any obvious separation between the fresh, semi-fresh and spoiled samples. On the contrary, the samples overlap, while several outliers are present. Thus, we can assume that linear supervised learning techniques may fail at discriminating the different classes, while nonlinear methods may prove to be more suitable in this instance.

	FTIR		**Raman**	
PCs	%Var	%Cum Var	%Var	%Cum Var
PC1	80.61	80.61	98.35	98.35
PC2	14.60	95.21	1.25	99.60
PC3	2.08	97.29	0.32	99.92
PC4	0.86	98.15	0.07	99.99
PC5	0.61	98.76	0.01	100

Table 11 PCA proportion and cumulative variance captured for the datasets of case study 3



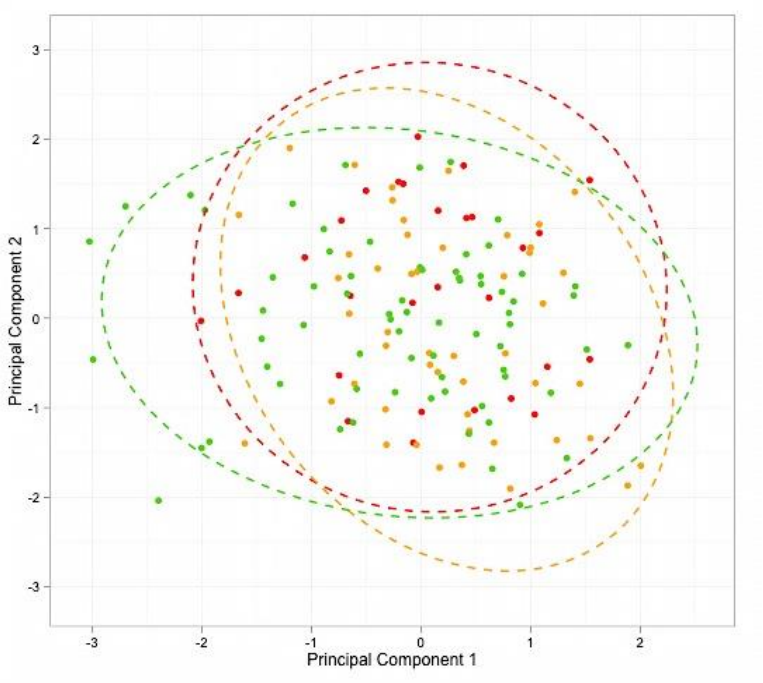
(a) FTIR data



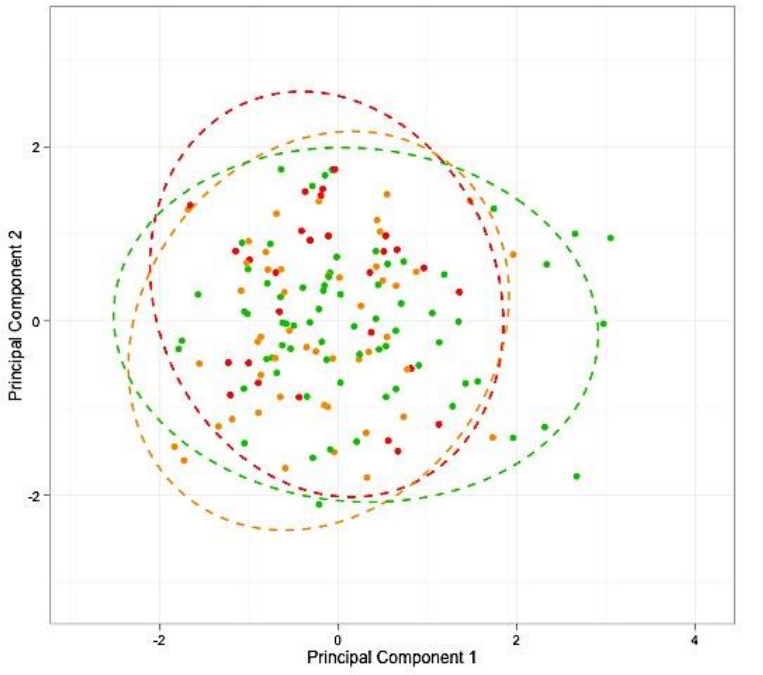
(b) Raman data

Figure 5-13 PCA scores plots with 95% confidence ellipses for case study 3

The two-dimensional scatterplots illustrate the scores of the first two PCs. Dynamically generated 95% confidence ellipses per each class were added in the plots in order to highlight the presence of any clusters and/or outliers. The colour representation used in the plot is similar to Figure 2-5. For comparison purposes, only the 147 common samples of minced beef (28 fresh, 49 semi-fresh and 70 spoiled samples) are depicted in each plot.



(a) GPA



(b) CPCA

Figure 5-14 The consensus of the first two Principal Components based on the fusion of the two experimental techniques from case study 3 using GPA and CPCA respectively

The consensus of GPA is compared against the super-scores of CPCA in two-dimensional space (the scores of the first two PCs are used). In both cases, the data integration does not improve the separation between the distinct classes. Similar to the standalone datasets, the three classes are highly overlapping.

The classification results for the datasets of case study 3 are presented in Figure 5-15 as percentages of correctly classified samples ($\%CC$). The overall accuracies of standalone and integrated datasets for both linear and nonlinear classifiers range from a minimum of 46% to a maximum of 48%. Therefore, we can conclude that the performance of all implemented classification models is equally poor.

This is a case where the highly imbalanced data have profoundly affected the decision boundaries of both linear and nonlinear models. Thorough empirical testing by Chawla *et al*. (2002) and Liu *et al.* (2006) has established that the SVM boundaries become biased as the imbalance ratios increase. In case study 3, the disproportionate data composition has had a profound effect on the classifiers' decision boundaries – even the boundaries by PLS-DA – hence it has dramatically influenced the overall performances. The class predictions of Figure 5-16 verify that the decision boundaries are indeed biased towards the majority class, whereas the ratios of misclassifications for the other two classes are significantly high. Only PLS-DA for the FTIR data presents low, but notably better class predictions for the fresh and spoiled samples compared to SVMs. Since the majority of fresh and semi-fresh samples are misclassified as spoiled, we can firmly conclude that the classification models have no discriminating power between the different classes.

As an attempt to minimise the dominating behaviour of the majority class, the classification models were re-built using different weights for each designated class during the training and testing process; however, this approach did not notably improve the classification results. Therefore, several other approaches such as under-sampling of the majority class and/or over-sampling of the minority class may help to overcome this major impediment of the machine learning algorithms.

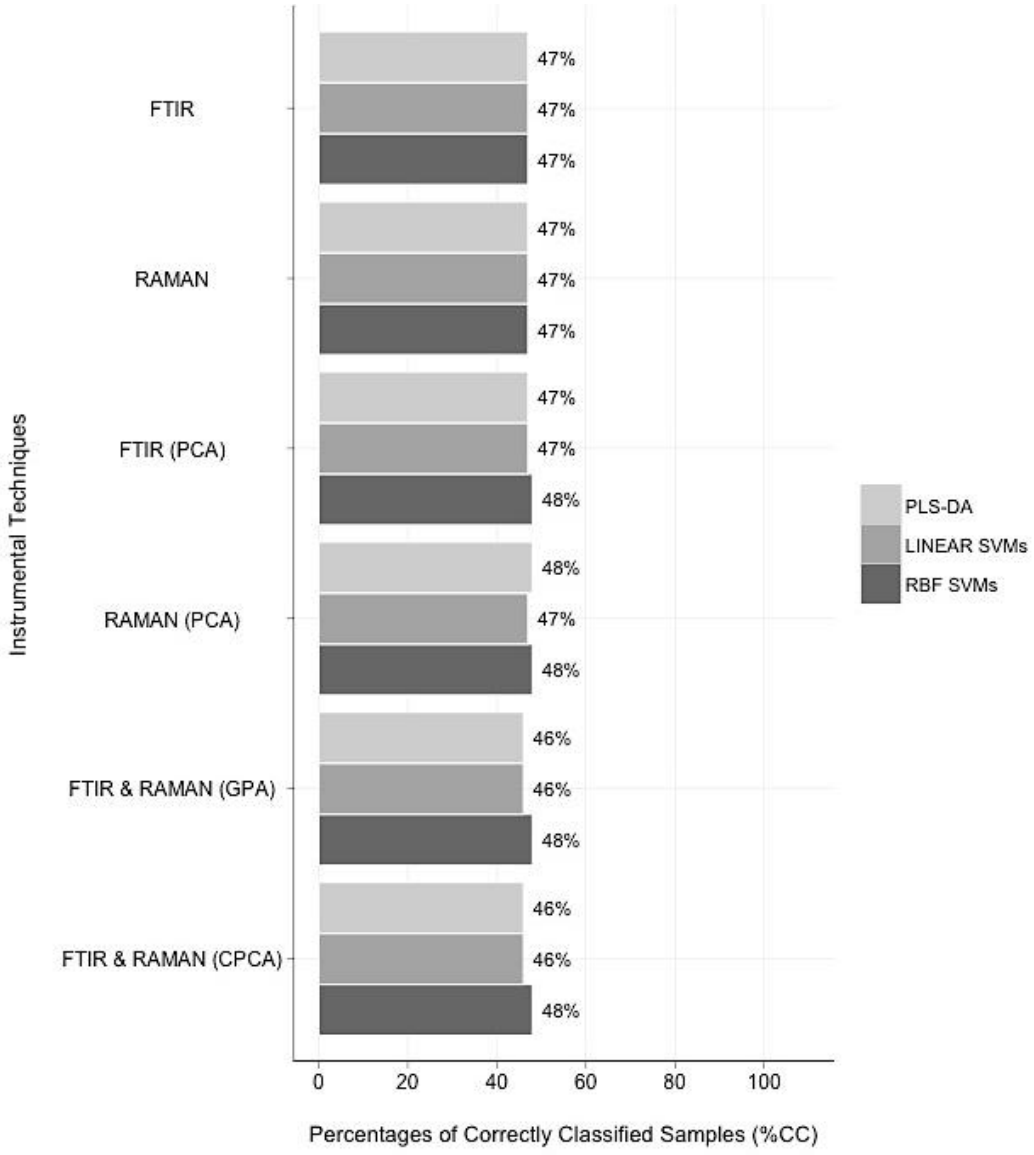


Figure 5-15 Overall accuracies (%CC) for the standalone and integrated datasets of case study 3

The figure illustrates the overall performance of all implemented classification ensembles on the standalone and integrated datasets of case study 3. The bars represent the percentages of correctly classified samples (%CC) and are coloured according to the classification model under study (PLS-DA, linear and RBF SVMs). In the case of standalone datasets, analyses have been conducted both prior (raw data) and after PCA. Data integration has been performed using both GPA and CPCA. In all implemented classifiers, bootstrapping was applied for hyperparameter optimisation. The overall accuracies have been rounded towards the nearest integer.

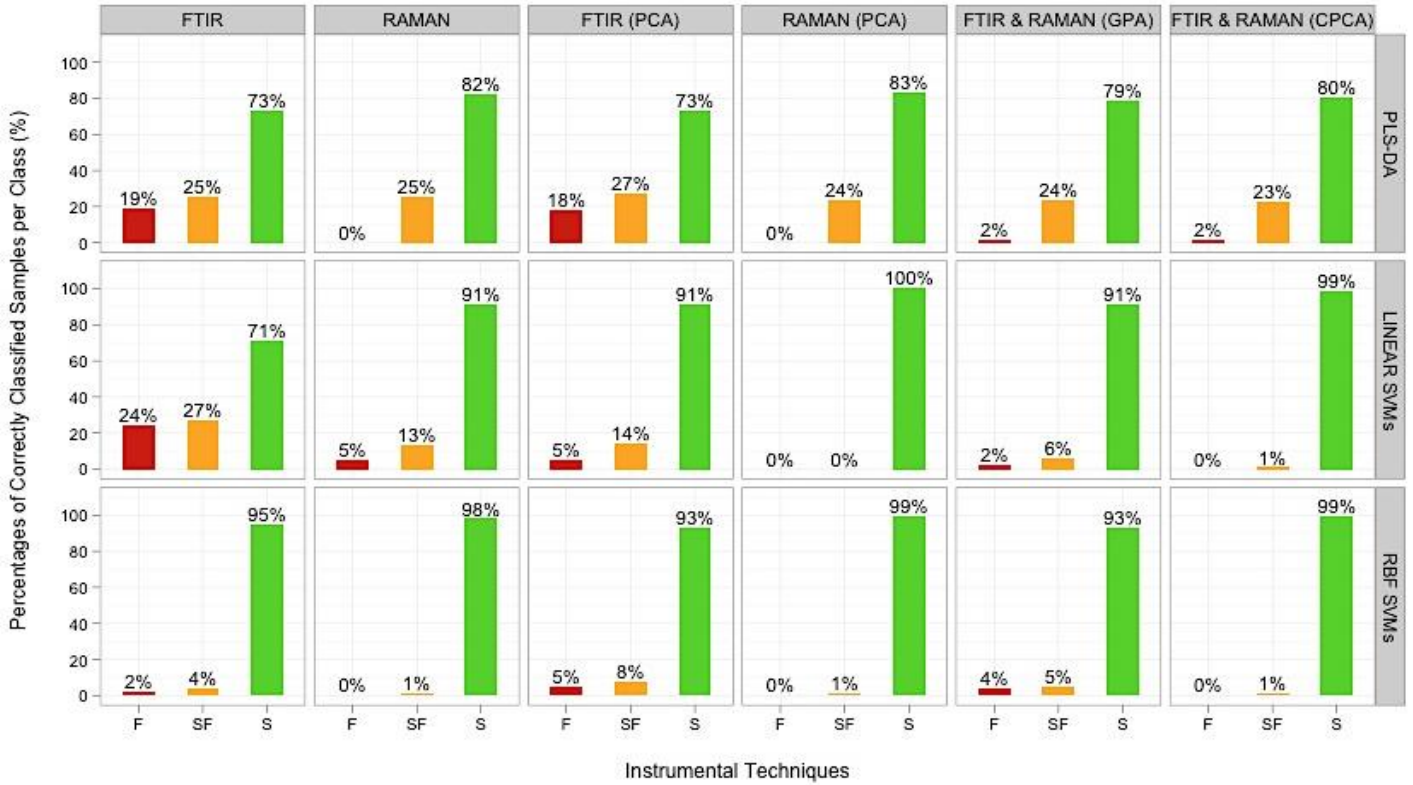


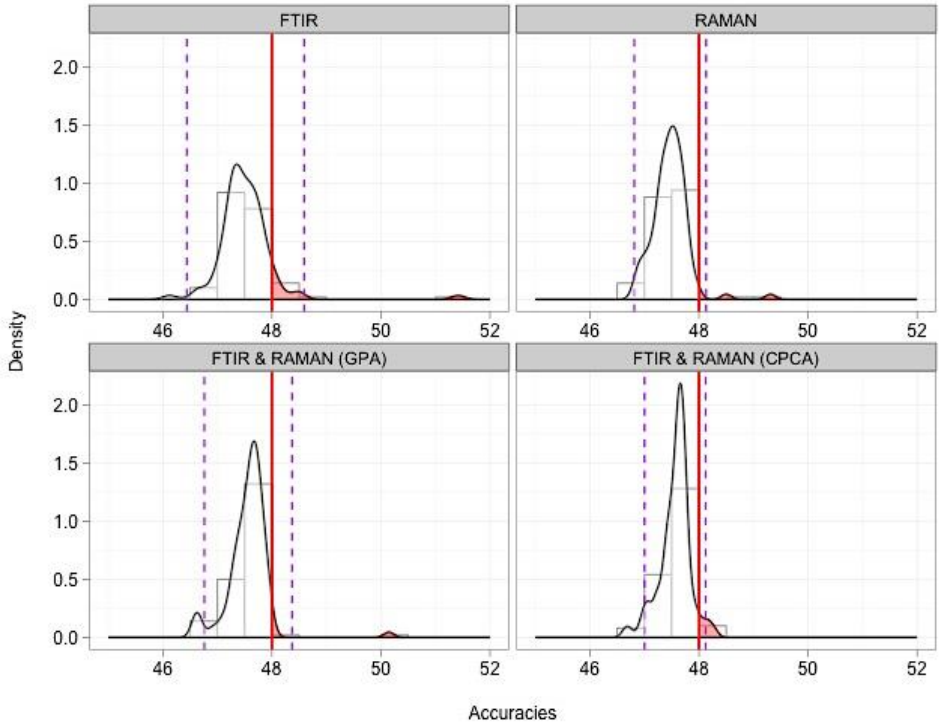
Figure 5-16 Class prediction rates of the standalone (prior and after PCA) and integrated datasets for case study 3

The figure illustrates the percentages of correctly classified samples per each distinct class, when the standalone and integrated dataset of case study 3 are imported in the analysis pipeline. Based on the results of the graphs, it is obvious that the decision boundaries are biased towards the majority class (spoiled samples), resulting in high prediction rates for this class and very low per-class accuracies for the other two classes (fresh and semi-fresh samples).

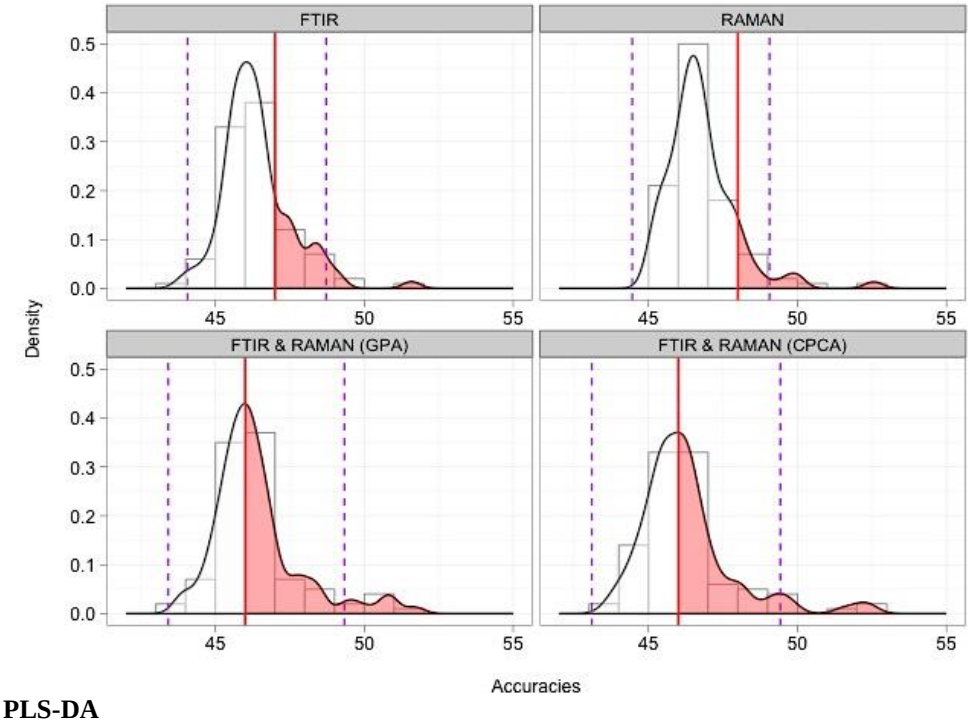
Finally, the outcome of rigorous permutation testing on the datasets of case study 3, as presented in Figure 5-17, has confirmed that all the obtained classification results are statistically non-significant. The results do not inspire any confidence since all the original non-permuted values are found below the 95% confidence intervals. Since the initial overall accuracies are equal to the results of random (permuted) classifiers, we can conclude that the models have no discriminating power between the different classes, and hence the null hypothesis cannot be rejected. The p-values range between 0.03 and 0.1 for the nonlinear SVMs, and between 0.23 and 0.57 for the PLS-DA models.

Even though the permutation tests established the performance of the classifiers as non-significant, RBF SVMs did once more demonstrate higher results than PLS-DA. This is obvious by a closer inspection of the superimposed density estimations of Figure 5-18 in addition to the "five-number summary" of Figure 5-19, while Table 12 and Table 13 provide the supporting descriptive statistics. In addition, the afore-mentioned p-values also verify this hypothesis. It is important to note that the observed values in the case of SVMs are found closer to the upper 95% confidence bounds than the respective values in PLS-DA. In certain cases such as the Raman dataset for the RBF SVMs, the non-permuted $\%CC$ is so close to the upper 95% confidence interval that is it only rejected due to some minor decimal differences.

Finally, the execution times for all permutation tests of case study 3 are illustrated in Figure 5-20. Even though, the run times for the permutations of nonlinear SVMs are approximately 16 times slower than those of PLS-DA, we have proved that SVMs produce consistently better classification results than PLS-DA, even if they appear to be non-significant.



(a) RBF SVMs



(b) PLS-DA

Figure 5-17 Distribution plots of the permutation tests on the datasets of case study 3 using RBF SVMs and PLS-DA respectively

The figure depicts the histograms and density curves of the permuted results for the RBF SVMs and PLS-DA ensembles respectively, when applied on the datasets of case study 3. In this case, all non-permuted results are found lower than the upper 95% confidence bounds; thus, they are considered statistically non-significant. For all the datasets under study, the red highlighted area represents the proportion of the distribution that is equal or greater than the observed non-significant value.

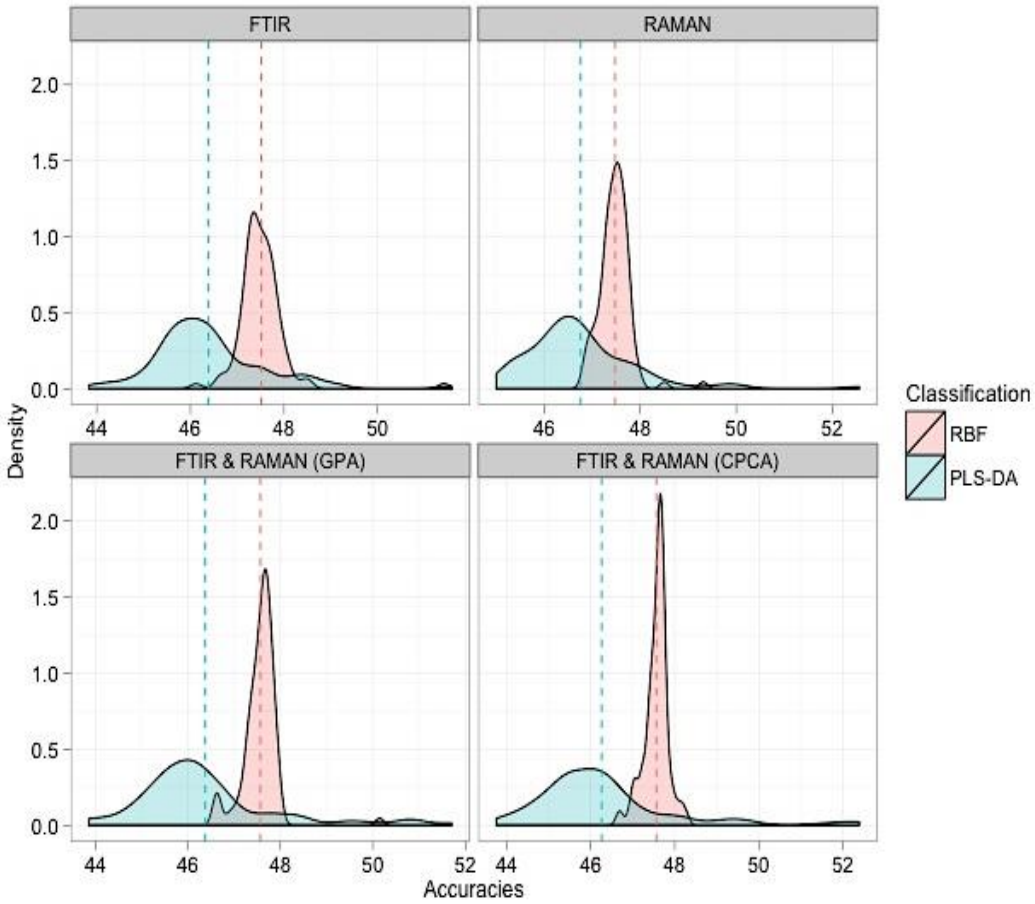


Figure 5-18 Superimposed density plots of the permutation tests on the datasets of case study 3 using PLS-DA and nonlinear (RBF) SVMs

The figure provides a visual comparison of the permutation distributions when different classification models are applied on the datasets of case study 3; the distributions for PLS-DA and SVMs are depicted in a semi-transparent blue and red colour respectively. In these plots, the dashed lines represent the mean values of each density curve and are coloured accordingly. By superimposing the density plots, major differences in the shape, spread and location of the distributions can be identified.

	RBF SVMs					
Datasets	**Original %CC**	**Mean Value**	**Median Value**	**Min Value**	**Max Value**	**Upper 95%C.I.**
FTIR	48%	48%	47%	46%	51%	49%
RAMAN	48%	48%	47%	47%	49%	48%
FTIR & RAMAN (GPA)	48%	48%	48%	47%	50%	48%
FTIR & RAMAN (CPCA)	48%	48%	48%	47%	48%	48%

Table 12 Descriptive statistics of the permutation distributions obtained by RBF SVMs (case study 3)

The results presented in Table 8 have been rounded towards the nearest integer.

	PLS-DA					
Datasets	**Original %CC**	**Mean Value**	**Median Value**	**Min Value**	**Max Value**	**Upper 95%C.I.**
FTIR	47%	46%	46%	44%	52%	49%
RAMAN	48%	47%	47%	45%	53%	49%
FTIR & RAMAN (GPA)	46%	46%	46%	44%	52%	49%
FTIR & RAMAN (CPCA)	46%	46%	46%	44%	52%	49%

Table 13 Descriptive statistics of the permutation distributions obtained by PLS-DA (case study 3)

The results presented in Table 9 have been rounded towards the nearest integer.

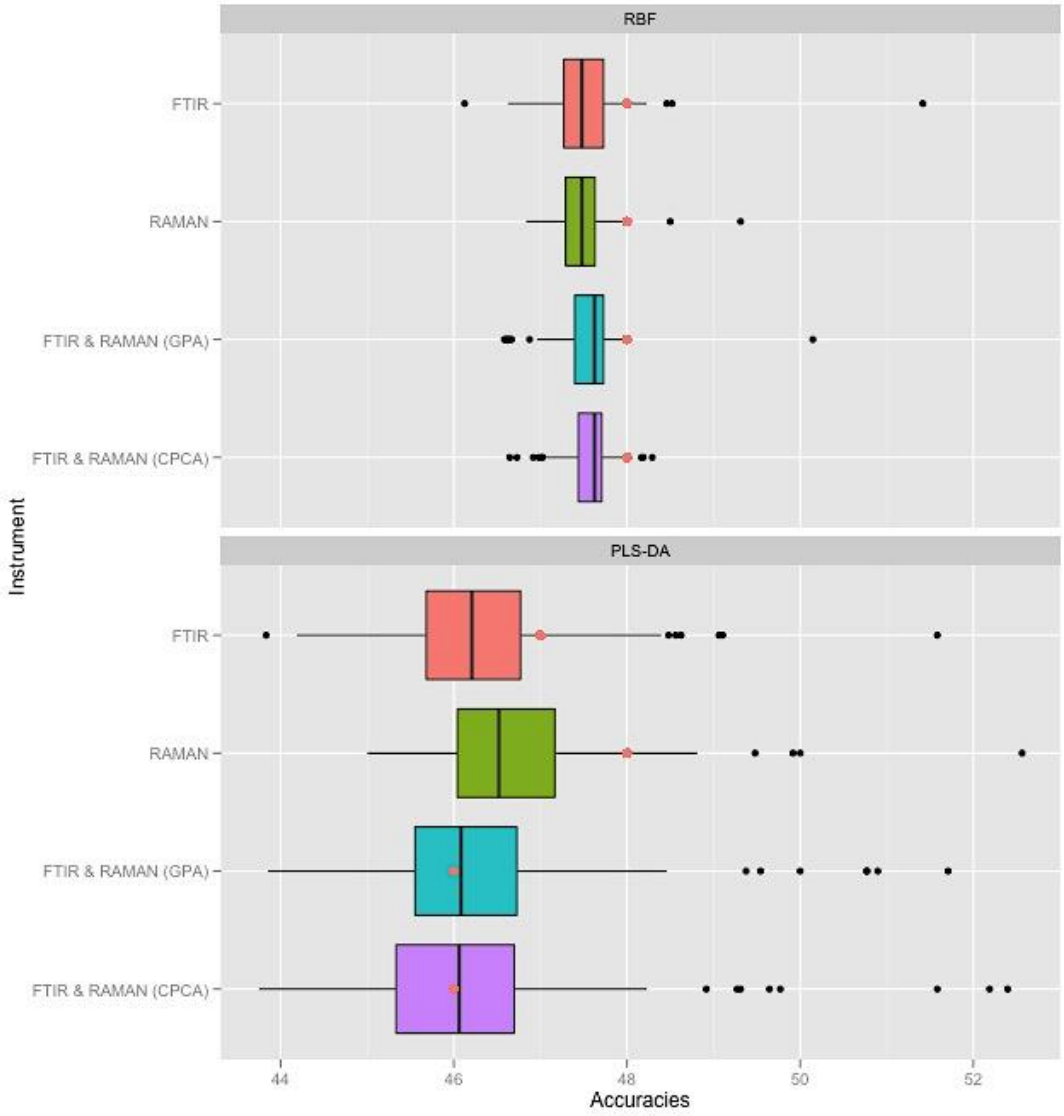


Figure 5-19 Boxplots representing the outcome of permutation testing when RBF SVMs and PLS-DA are applied on the datasets of case study 3

The boxplots provide a powerful visual aid for a straightforward comparison of the descriptive statistics of a given permutation distribution. Each boxplot illustrates the "five-number summary": namely, the minimum, first (lower) quartile, median, third (upper) quartile and maximum value. In addition, the observed non-permuted values are highlighted in a red colour.

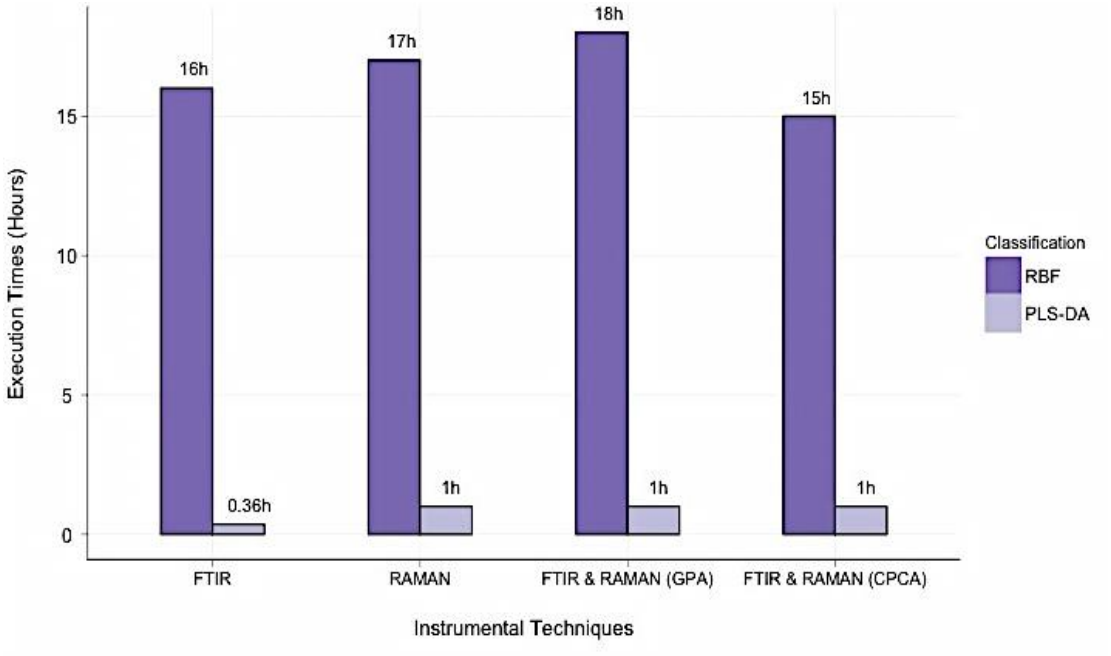


Figure 5-20 Execution times of the permutation tests on the datasets of case study 3

The figure displays the execution times of 100 permutation tests using PLS-DA and RBF SVMs for each standalone and integrated dataset of case study 3. The execution times are based on a fully optimised analysis pipeline featuring parallel programming (master/slave architecture) over eight processors (see Section 5.2.4) as well as fast approximation algorithms for the optimisation of the classifiers' hyperparameters via bootstrapping. The execution times have been rounded towards the nearest integer.

5.3.3 Case study 4

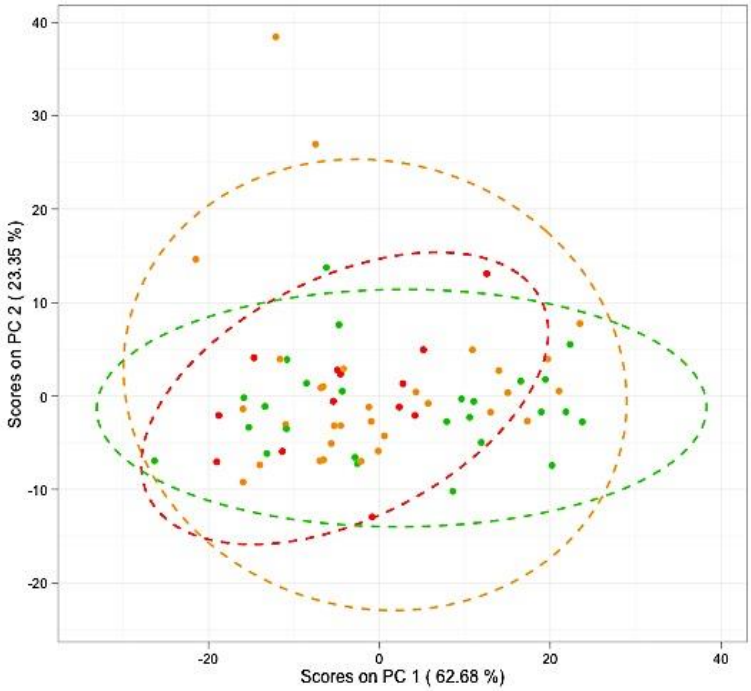
In case study 4 ("Pork stored in air and MAP"), data have been acquired from three main experimental techniques: FTIR, HPLC and e-nose. The data intersection approach presented in Section 2.2.2 extracted a total of 70 common samples along with their respective sensory scores, which were inserted in the analysis pipeline; these samples consist of 13 fresh (F), 31 semi-fresh (SF) and 26 spoiled (S) samples; therefore, unlike the afore-mentioned case studies, the semi-fresh – and not the spoiled – samples constitute the majority class, whereas the fresh samples represent the minority class.

PCA was once more employed as the initial step on pre-processed standalone datasets. The percentages of variance and cumulative variance for each experimental technique of case study 4 are presented in Table 14. The entries of Table 14 demonstrate a similar trend to case study 1 (see Table 2). Figure 5-21 illustrates the PCA scores plots for case study 4. Both FTIR and e-nose data do not demonstrate any clustering or visible separation between the different classes. However, HPLC demonstrates two well-defined clusters for the fresh and spoiled samples respectively, which are linearly separable in two-dimensional space. In this instance, a subset of the semi-fresh samples is starting to form a distinct cluster with the remaining samples overlapping with the other two classes.

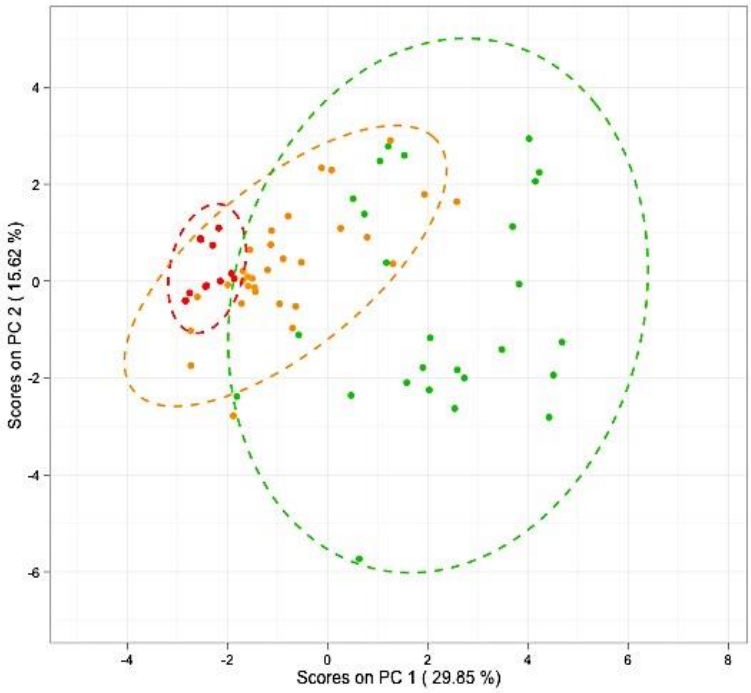
The outcome of the implemented data integration techniques is depicted in Figure 5-22. The figure is indicative of the data fusion for the datasets of case study 4, and is based on the simultaneous integration of all three experimental techniques. In this case, the consensus by GPA provides a nearly linear discrimination between fresh and spoiled samples, while the semi-fresh samples that constitute the majority class are scattered in-between the two classes. On the contrary, the super scores of consensus PCA do not reveal any patterns or trends in the data; the fresh, semi-fresh and spoiled samples are highly overlapping. In this instance, the projection of the data into a high-dimensional space by kernel-based SVMs may prove to be fruitful.

	FTIR		**HPLC**		**e-nose**	
PCs	%Var	%Cum Var	%Var	%Cum Var	%Var	%Cum Var
PC1	62.68	62.68	29.85	29.85	86.95	86.95
PC2	23.35	86.03	15.62	45.47	8.56	95.51
PC3	8.07	94.10	11.60	57.07	1.46	96.97
PC4	2.82	96.92	8.61	65.68	1.04	98.01
PC5	1.01	97.93	6.90	72.58	0.93	98.94

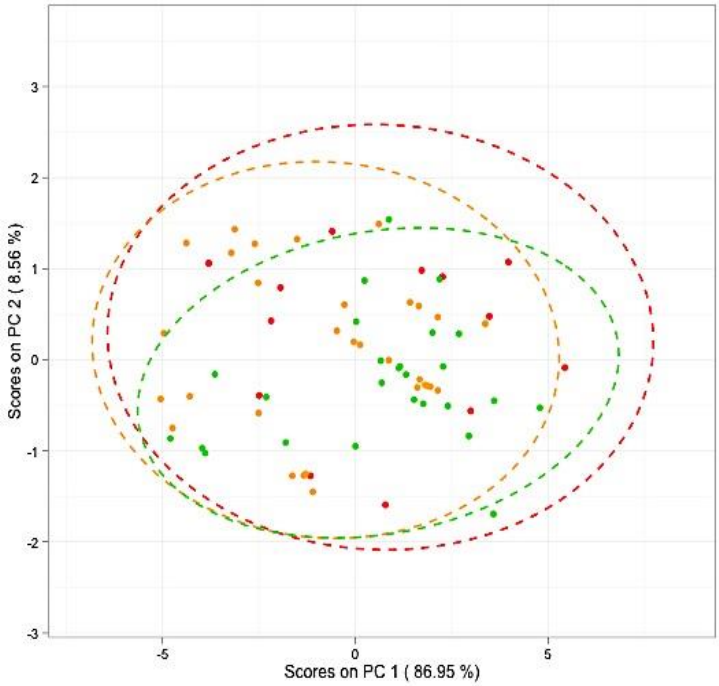
Table 14 PCA proportion and cumulative variance captured for the datasets of case study 4



(a) FTIR data



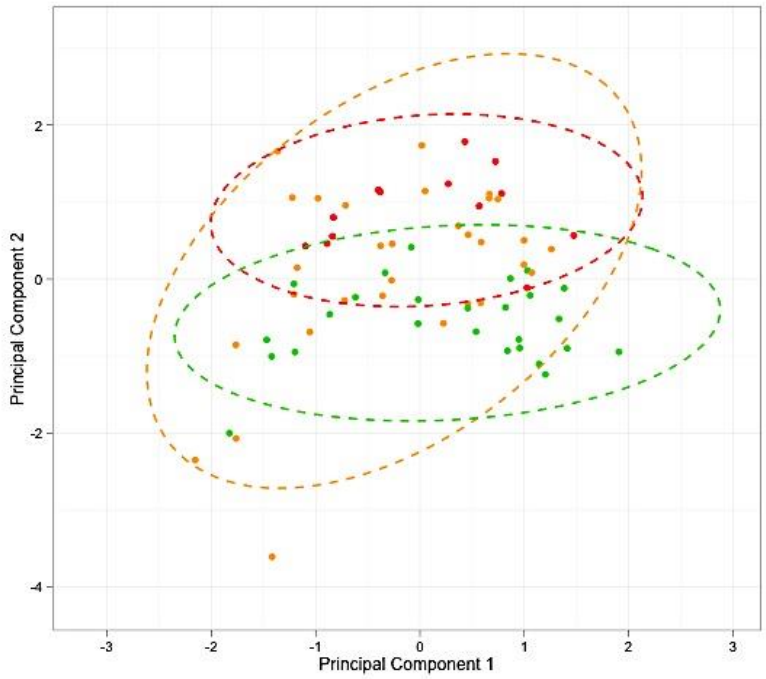
(b) HPLC data



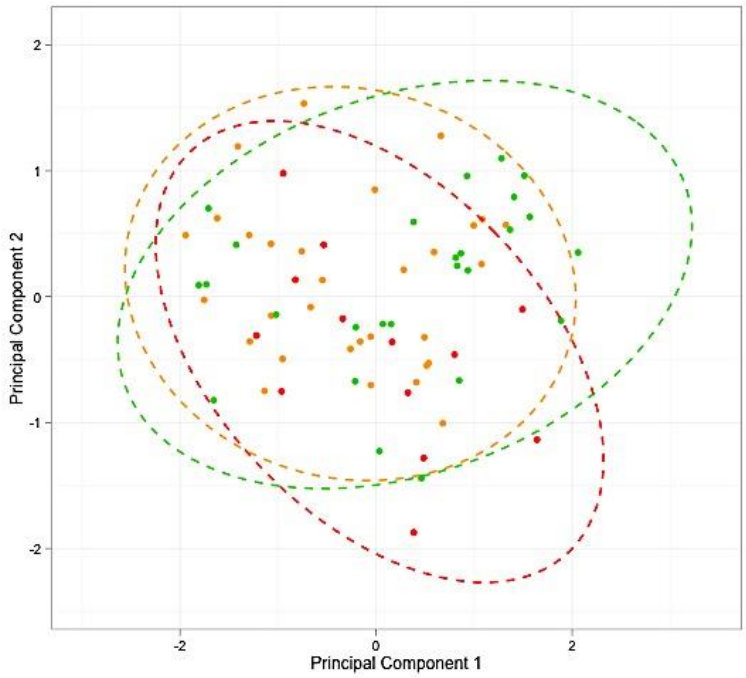
(c) e-nose data

Figure 5-21 PCA scores plots with 95% confidence ellipses for case study 4

The two-dimensional scatterplots illustrate the scores of the first two PCs for the auto-scaled data of case study 4. Dynamically generated 95% confidence ellipses per each class were added in the plots in order to highlight the presence of any clusters and/or outliers. The colour representation used in the plot is similar to Figure 2-5. For comparison purposes, only the 70 common samples of pork (13 fresh, 31 semi-fresh and 26 spoiled samples) are depicted in each plot.



(a) GPA



(b) CPCA

Figure 5-22 The consensus of the first two Principal Components based on the fusion of the two experimental techniques of case study 4 using GPA and CPCA respectively

The consensus of GPA is compared against the super-scores of CPCA in two-dimensional space (the first two PCs are used). In both cases, the data integration does not improve the separation between the distinct classes.

The classification results of the standalone and integrated datasets of case study 4 are displayed in Figure 5-23 and Figure 5-24 respectively as percentages of correctly classified samples ($\%CC$). Once more, the HPLC data (prior and after PCA) clearly demonstrate the highest overall accuracies among the three experimental techniques. In addition, the data appear to be favoured by nonlinear SVMs, for which the highest overall $\%CC$, equal to 78%, is recorded. On the contrary, the results of the linear classifiers, and in particular those of PLS-DA, are significantly lower. Obviously, the nonlinear mapping into a high dimensional space by the kernel-based SVMs has been extremely fruitful in this instance.

In the case of FTIR, similar to the previous case studies, linear classifiers (PLS-DA and linear SVMs) demonstrate higher overall accuracies ($\%CC$) compared to nonlinear SVMs both prior and after the application of PCA. Indeed, the spectral data appear to be extremely easy to separate by linear models and especially by traditional chemometric techniques such as PLS-DA; detailed information to support this theory can be found in Section 4.3.2.1. Therefore, the nonlinear projection of the FTIR data into a high-dimensional space by the RBF kernel has been found unsuitable.

Finally, the e-nose data that have been subjected to PCA, result in significantly higher $\%CC$ according to Figure 5-23. However, the PLS-DA models demonstrate better overall performance compared to the remaining classifiers. Since both types of SVMs produce a lower accuracy comparing to PLS-DA, we can only assume that the background of SVMs is the underlying cause for this result; as presented in Section 1.5.2.1, PLS-DA constructs the decision boundaries using all available samples as a whole, whereas the decision boundaries of the SVMs are solely based on the selection of support vectors. Thus, in the case of imbalanced datasets, the decision boundaries may favour the majority classes, while overlooking the minority classes. A thorough investigation of the class predictions may help towards justifying this hypothesis. Even so, the generalisation performance of e-nose is quite poor as it generates the lowest accuracies among the three experimental techniques.

The classification results of the fused datasets as obtained by GPA and CPCA are illustrated in Figure 5-24. Based on the plots, it is noteworthy that in all cases, the linear classifiers produce higher overall accuracies than nonlinear SVMs. Since the standalone datasets were mostly favoured by linear classification models, it was expected that the integrated datasets would demonstrate a similar pattern.

For the GPA algorithm, PLS-DA and linear SVMs produce exactly the same results in the majority of cases, while the nonlinear SVMs demonstrate significantly lower accuracies. Even so, since all the $\%CC$ values of the fused datasets are lower than those presented by the standalone datasets, the application of GPA as a data integration technique has been found unfruitful for case study 4

However, the application of CPCA has produced the best results observed thus far; in this instance, the maximum noted overall accuracy is equal to 82%, which is significantly higher than the maximum accuracy provided by standalone HPLC, equal to 78%. In general, linear classifiers demonstrate greater performance, with the linear SVMs taking the lead in the majority of cases. In addition, the nonlinear SVMs produce at least as good results as one of the other two classification techniques. It appears that CPCA clearly improves the outcome of data integration by combining the most discriminatory features of the individual experimental techniques, and hence overcomes the limitations of GPA.

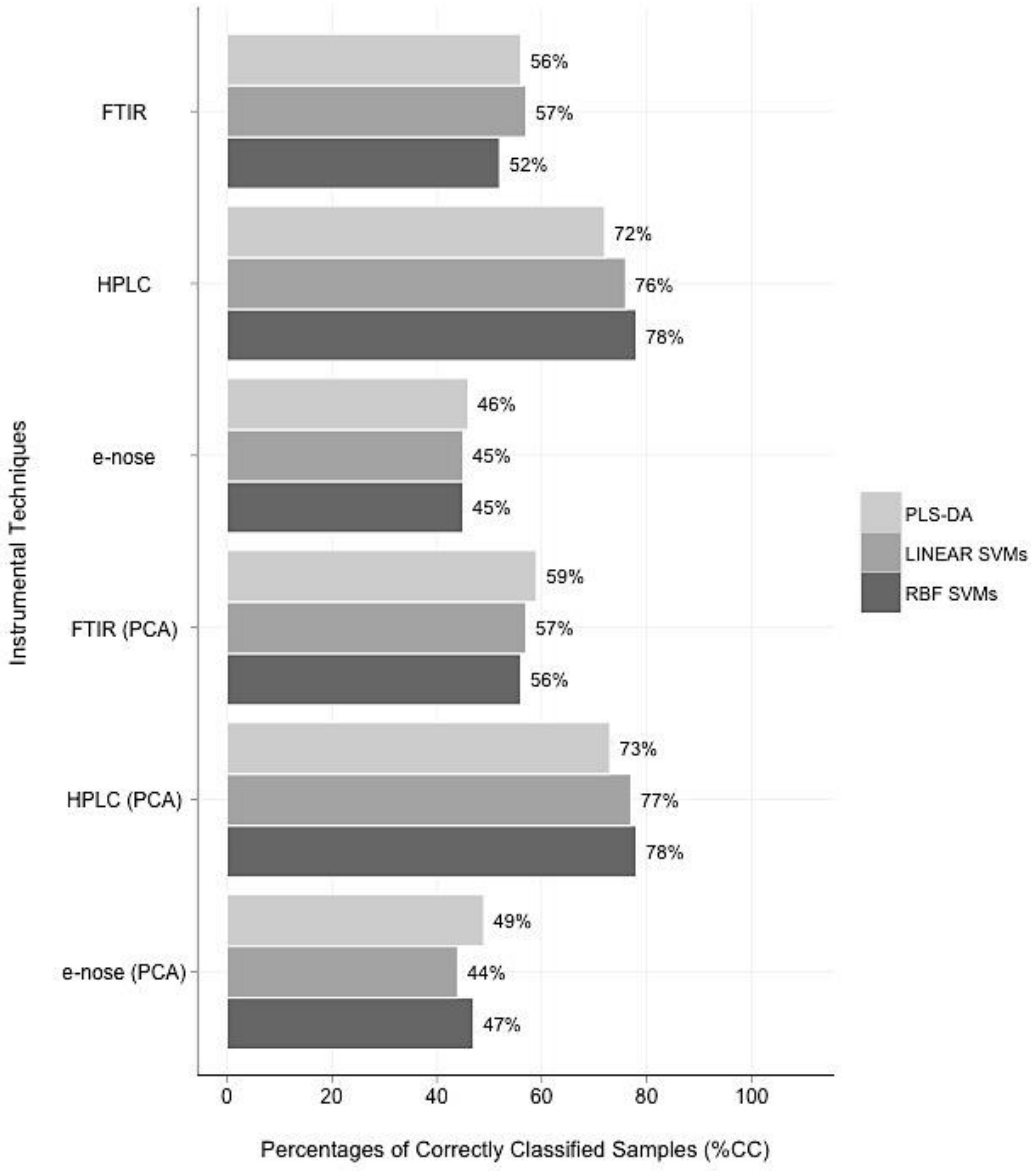


Figure 5-23 Overall accuracies (%CC) for the standalone datasets of case study 4

The figure illustrates the overall performance of all implemented classification ensembles on the standalone datasets of case study 4. The bars represent the percentages of correctly classified samples (%CC) and are coloured according to the classification model under study (PLS-DA, linear and RBF SVMs). Analyses have been conducted both prior (raw data) and after PCA. In all implemented classifiers, bootstrapping was applied for hyperparameter optimisation. The overall accuracies have been rounded towards the nearest integer.

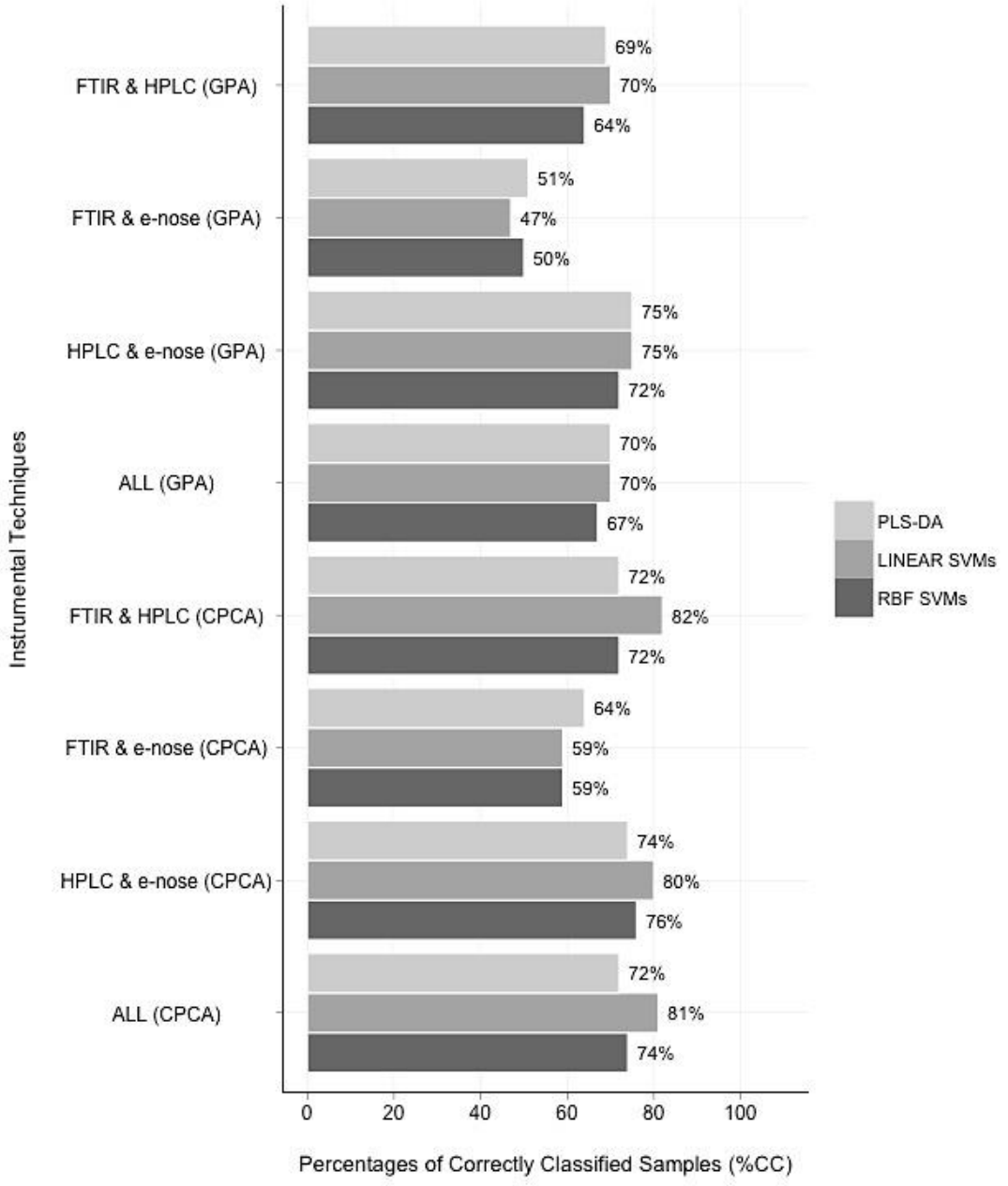


Figure 5-24 Classification Results for the integrated datasets of case study 4

The figure illustrates the overall performance of all implemented classification ensembles on the integrated datasets of case study 4. The bars represent the percentages of correctly classified samples (%CC) and are coloured according to the classification model under study (PLS-DA, linear and RBF SVMs). Data integration has been performed using both GPA and CPCA. In all implemented classifiers, bootstrapping was applied for hyperparameter optimisation. The overall accuracies have been rounded towards the nearest integer.

In addition to the overall accuracies, the per-class percentages of correctly classified samples for the standalone and integrated datasets of case study 4 are depicted in Figure 5-25 and Figure 5-26 respectively. As expected, since semi-fresh (SF) samples constitute the majority class in this case study, they obtain noteworthy class accuracies – all well above 50% throughout all experimental techniques and classification models – in comparison to the SF prediction rates of the previous case studies. The HPLC data prior to PCA generate the best prediction rates among fresh, semi-fresh and spoiled samples, with exceptional percentages for the fresh samples, which represent the minority class. However, once the data are subjected to PCA, the outstanding accuracies of correctly classified fresh samples decrease, while the accuracies of spoiled samples increase. In addition, the class accuracies for the majority class (semi-fresh samples) increase for the PLS-DA classifiers while they decrease for the SVMs.

In addition, the FTIR and e-nose data demonstrate a similar pattern; the classifiers produce equally good class accuracies for the semi-fresh and spoiled samples – the two larger sets of classes in this case study. However, the decision boundaries are unable to correctly discriminate the samples of the minority class (fresh samples), thus resulting in poor prediction rates. In the majority of cases, PLS-DA illustrated higher prediction rates per class than SVMs.

Furthermore, for all fused datasets in this case study, the class accuracies as obtained by GPA underperform when compared to the results by CPCA. It is important to note that the data fusion by GPA strongly favours the spoiled samples even though the semi-fresh samples constitute the majority class in this case study. We can thus conclude that the data integration based on GPA does not produce as fruitful results as expected, and hence is found unfit. On the contrary, the CPCA algorithm achieved outstanding classification rates for the fresh samples, which are higher than those of the standalone datasets. In addition, the accuracies of the semi-fresh and spoiled samples are also enhanced, especially in the case of SVMs, with linear SVMs taking once more the lead among the other classifiers. Thus, we can conclude that CPCA minimises the dominance of weak instrumental techniques, while it enhances the integration outcome by combining the strongest assets of the individual origins.

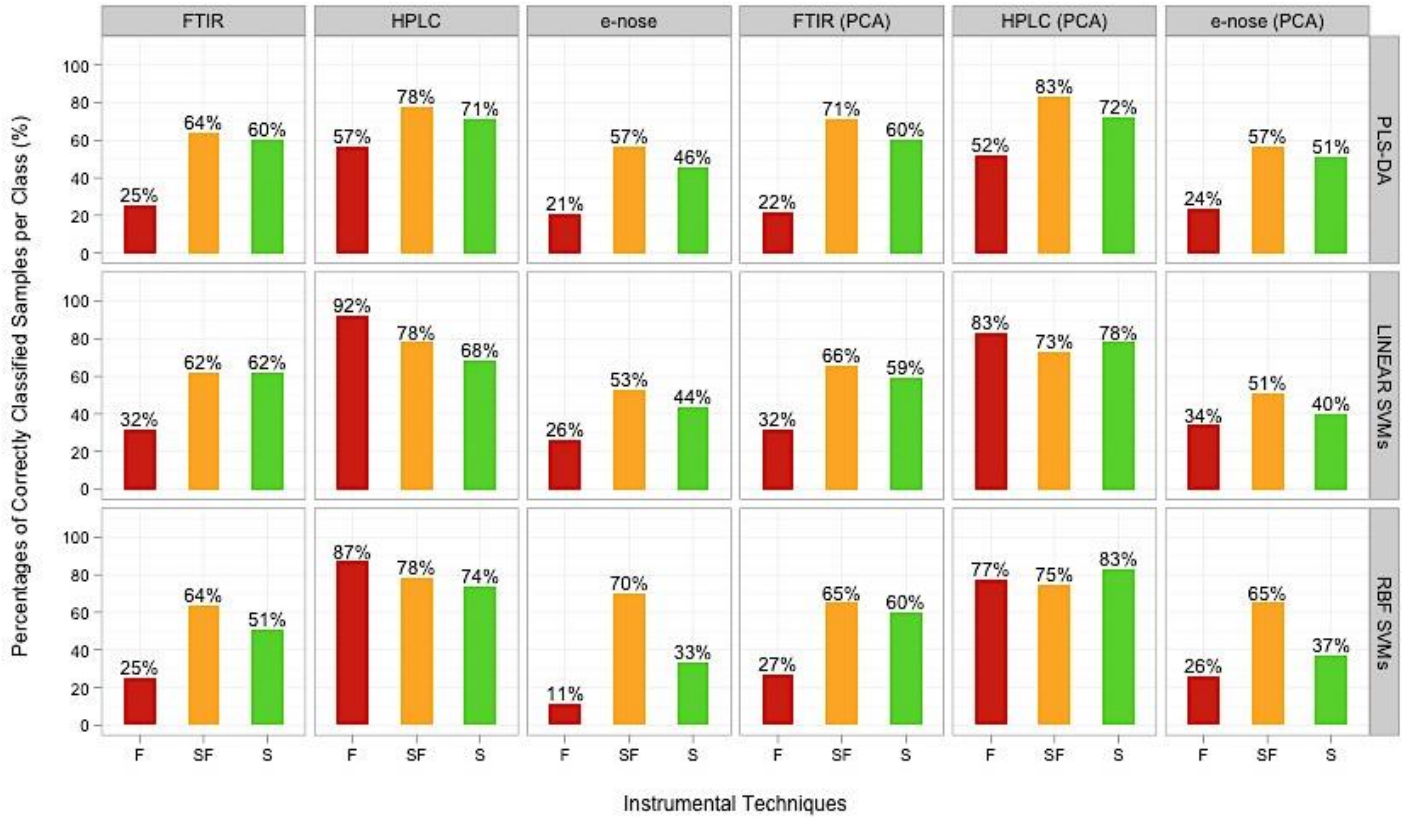


Figure 5-25 Class prediction rates of the standalone (prior and after PCA) datasets for case study 4

The figure illustrates the percentages of correctly classified samples per each distinct class, when the standalone datasets of case study 4 are imported in the analysis pipeline. It is noteworthy that even though the semi-fresh samples constitute the majority class in this instance, the class prediction rates of spoiled samples appear to be equally good.

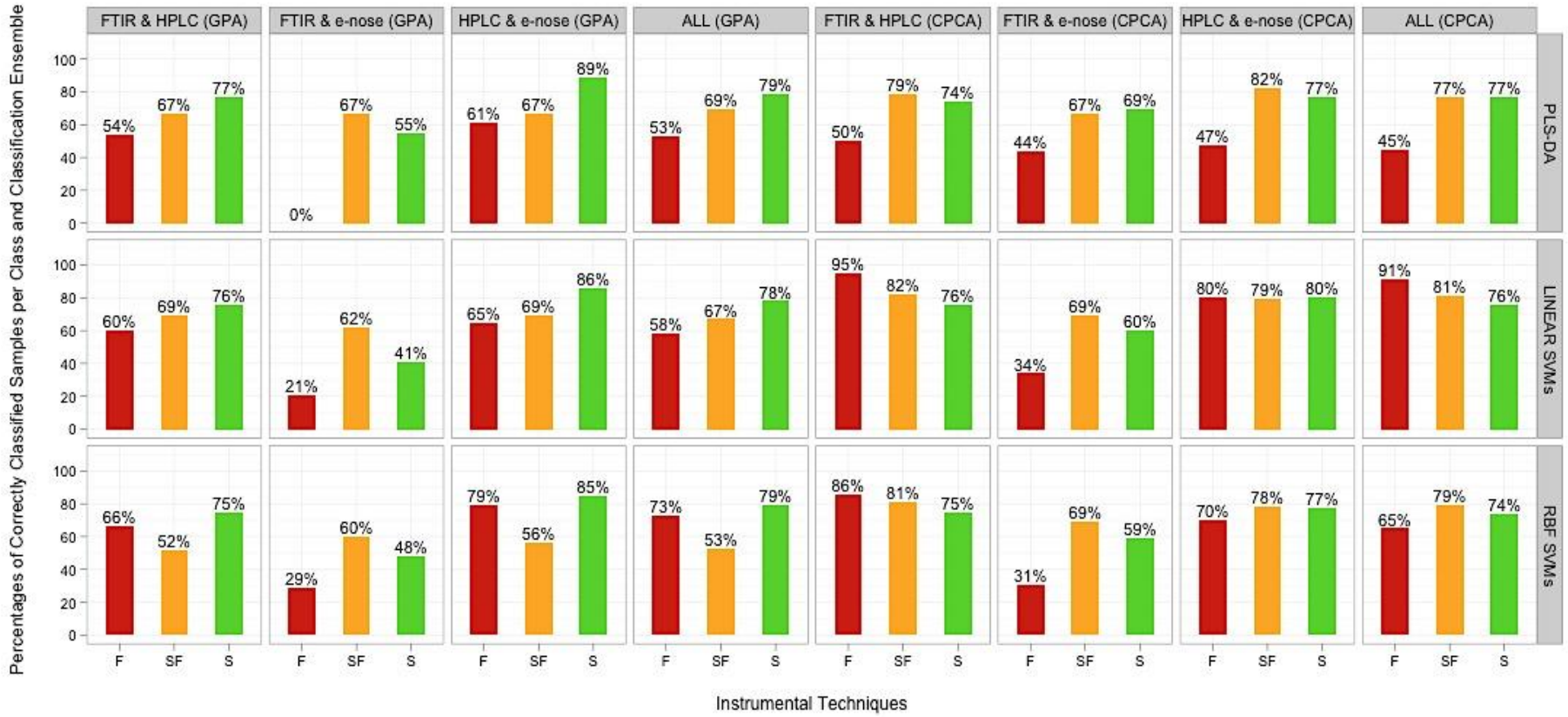


Figure 5-26 Class prediction rates of the integrated datasets for case study 4

The figure illustrates the percentages of correctly classified samples per each distinct class, when the integrated datasets of case study 4 are imported in the analysis pipeline. Data fusion was performed using GPA and CPCA. It is noteworthy that CPCA produces significantly higher percentages of correctly classified fresh samples (the minority class) than GPA.

The permutation results for the datasets of case study 4 using RBF SVMs and PLS-DA are illustrated in the histograms of Figure 5-27 and Figure 5-28 respectively. For most experimental data under study, the non-permuted overall accuracies are found well above the 95% confidence values; in particular, the majority of non-permuted results are even greater than the 99% confidence intervals. Thus, the application of permutation testing has confirmed that all obtained classification results by RBF SVMs are indeed statistically significant. Furthermore, in the case of PLS-DA, all overall accuracies besides the one obtained by e-nose were also established as significant. However, the non-permuted value of the e-nose dataset, which is equal to 49%, was found bellow the upper bound of the 95% confidence interval; therefore, the result can be ascribed purely to chance and the null hypothesis cannot be rejected. However, it is noteworthy that the observed classification accuracy for the e-nose data is found extremely close to the upper 95% confidence bound; even so, the non-permuted $\%CC$ is established as non-significant.

In addition, based on the superimposed density curves of Figure 4-13 and the boxplots of Figure 5-30, it is obvious once more that the density estimations of the two classifiers obtain completely different distributions and spread. Table 3 and Table 4 summarise the most important descriptive statistics of these distributions. As with case study 1 (see Section 4.3.3), we can conclude that the RBF SVMs constitute more powerful classifiers in comparison to the PLS-DA models since they generate consistently better results.

Finally, the execution times for all permutation tests of case study 4 are illustrated in Figure 5-31. Once more thorough permutation testing, which consists of 100 individual permutation tests, is completed only within a few hours.

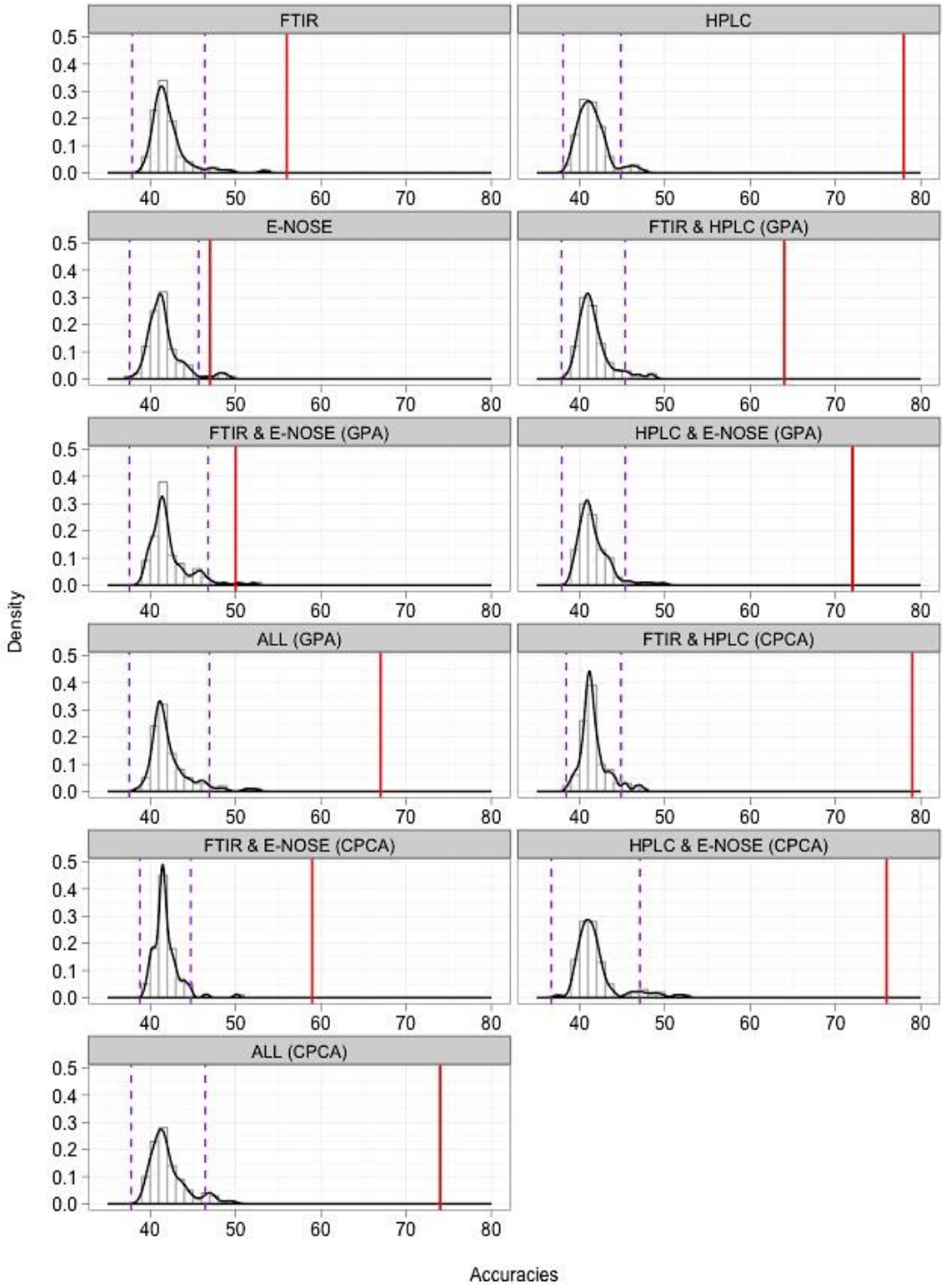


Figure 5-27 Distribution plots of the permutation tests on the datasets of case study 4 using RBF SVMs

The figure depicts the histograms and density curves of the permuted results for the RBF SVMs, when applied on the datasets of case study 4. The outcome of permutation testing for the datasets of case study 4 verifies that all of the obtained overall accuracies are statistically significant.

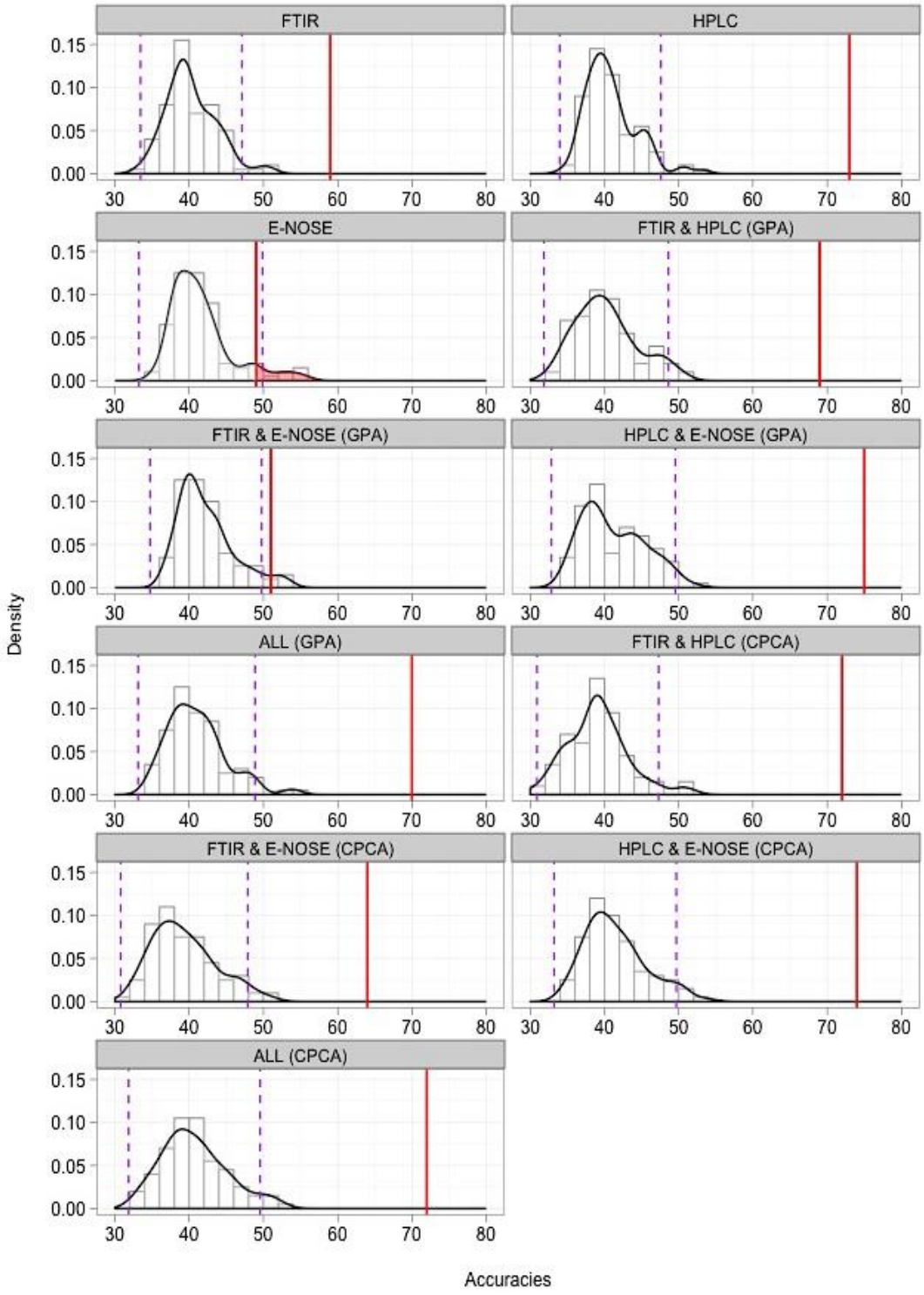


Figure 5-28 Distribution plots of the permutation tests on the data of case study 4 using PLS-DA

The figure depicts the histograms and density curves of the permuted results for the PLS-DA ensembles, when applied on the datasets of case study 4. In the case of e-nose, the red highlighted area represents the proportion of the distribution that is equal or greater than the observed non-significant value.

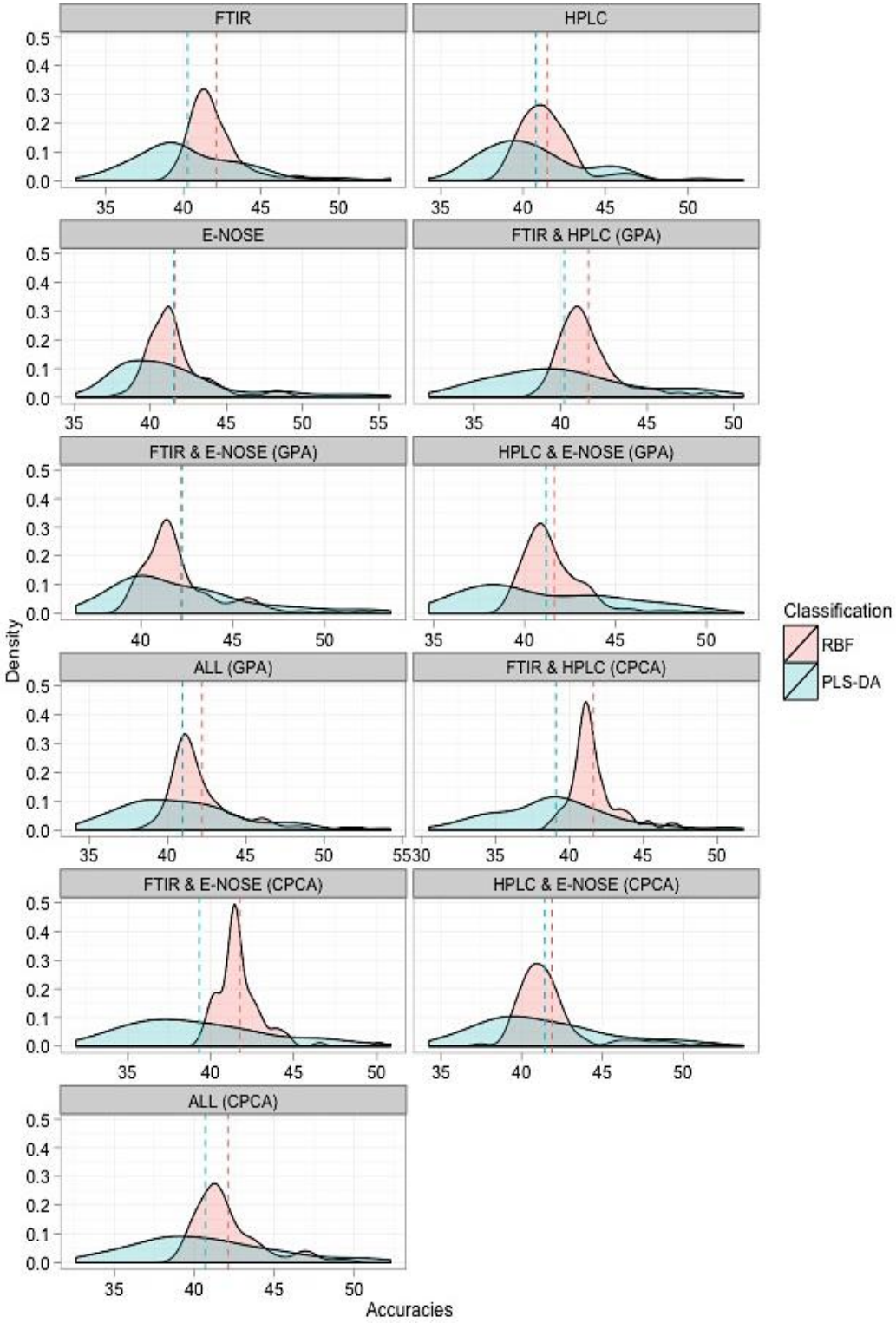


Figure 5-29 Superimposed density plots of the permutation tests on the datasets of case study 4 using PLS-DA and nonlinear (RBF) SVMs

The figure provides a visual comparison of the permutation distributions when different classification models are applied on the datasets of case study 4; the distributions for PLS-DA and SVMs are depicted in a semi-transparent blue and red colour respectively. In these plots, the dashed lines represent the mean values of each density curve and are coloured accordingly.

	RBF SVMs					
Datasets	**Original %CC**	**Mean Value**	**Median Value**	**Min Value**	**Max Value**	**Upper 95%C.I.**
FTIR	54%	42%	42%	39%	53%	46%
HPLC	78%	41%	41%	39%	47%	45%
E-NOSE	47%	42%	41%	38%	49%	46%
FTIR & HPLC (GPA)	64%	42%	41%	39%	48%	45%
FTIR & E-NOSE (GPA)	50%	42%	42%	39%	52%	47%
HPLC & E-NOSE (GPA)	72%	42%	41%	39%	50%	45%
ALL (GPA)	67%	42%	41%	39%	52%	47%
FTIR & HPLC (CPCA)	79%	42%	41%	39%	47%	45%
FTIR & E-NOSE (CPCA)	59%	42%	41%	39%	50%	45%
HPLC & E-NOSE (CPCA)	76%	42%	41%	37%	52%	47%
ALL (CPCA)	74%	42%	42%	39%	50%	46%

Table 15 Descriptive statistics of the permutation distributions obtained by RBF SVMs (case study 4)

The results presented in Table 11 have been rounded towards the nearest integer.

	PLS-DA					
Datasets	**Original %CC**	**Mean Value**	**Median Value**	**Min Value**	**Max Value**	**Upper 95%C.I.**
FTIR	59%	40%	40%	33%	51%	47%
HPLC	73%	41%	40%	34%	53%	48%
E-NOSE	49%	42%	41%	35%	56%	50%
FTIR & HPLC (GPA)	69%	40%	40%	32%	51%	49%
FTIR & E-NOSE (GPA)	51%	42%	41%	36%	54%	50%
HPLC & E-NOSE (GPA)	75%	41%	40%	35%	52%	50%
ALL (GPA)	70%	41%	41%	34%	54%	49%
FTIR & HPLC (CPCA)	72%	39%	39%	31%	52%	47%
FTIR & E-NOSE (CPCA)	64%	39%	39%	32%	51%	48%
HPLC & E-NOSE (CPCA)	74%	41%	41%	34%	54%	50%
ALL (CPCA)	72%	41%	40%	33%	52%	50%

Table 16 Descriptive statistics of the permutation distributions obtained by PLS-DA (case study 4)

The results presented in Table 12 have been rounded towards the nearest integer.

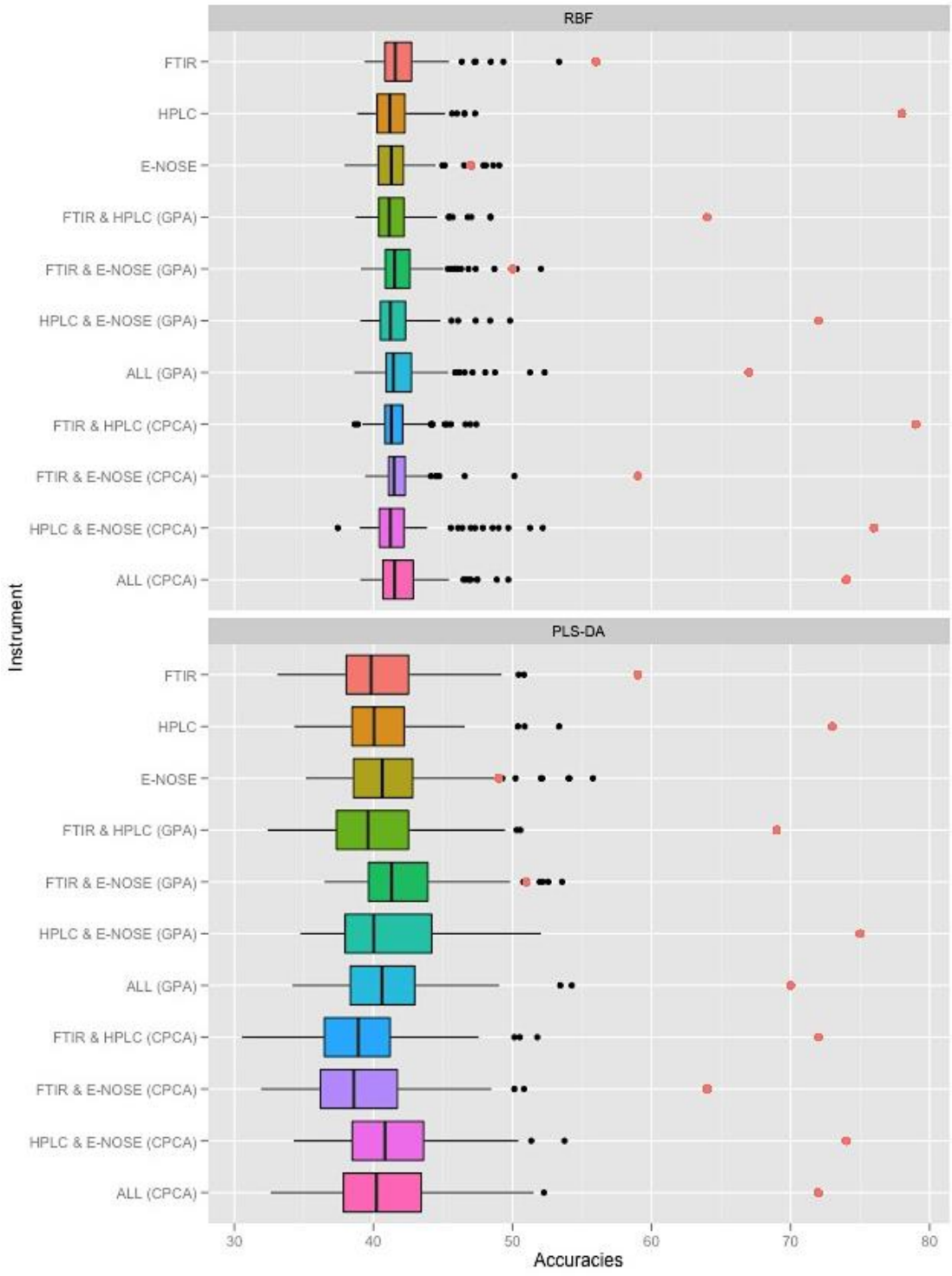


Figure 5-30 Boxplots representing the outcome of permutation testing when RBF SVMs and PLS-DA are applied on the datasets of case study 4

The boxplots provide a powerful visual aid for a straightforward comparison of the descriptive statistics of a given permutation distribution. Each boxplot illustrates the "five-number summary": namely, the minimum, first (lower) quartile, median, third (upper) quartile and maximum value. In addition, the observed non-permuted values are highlighted in a red colour.

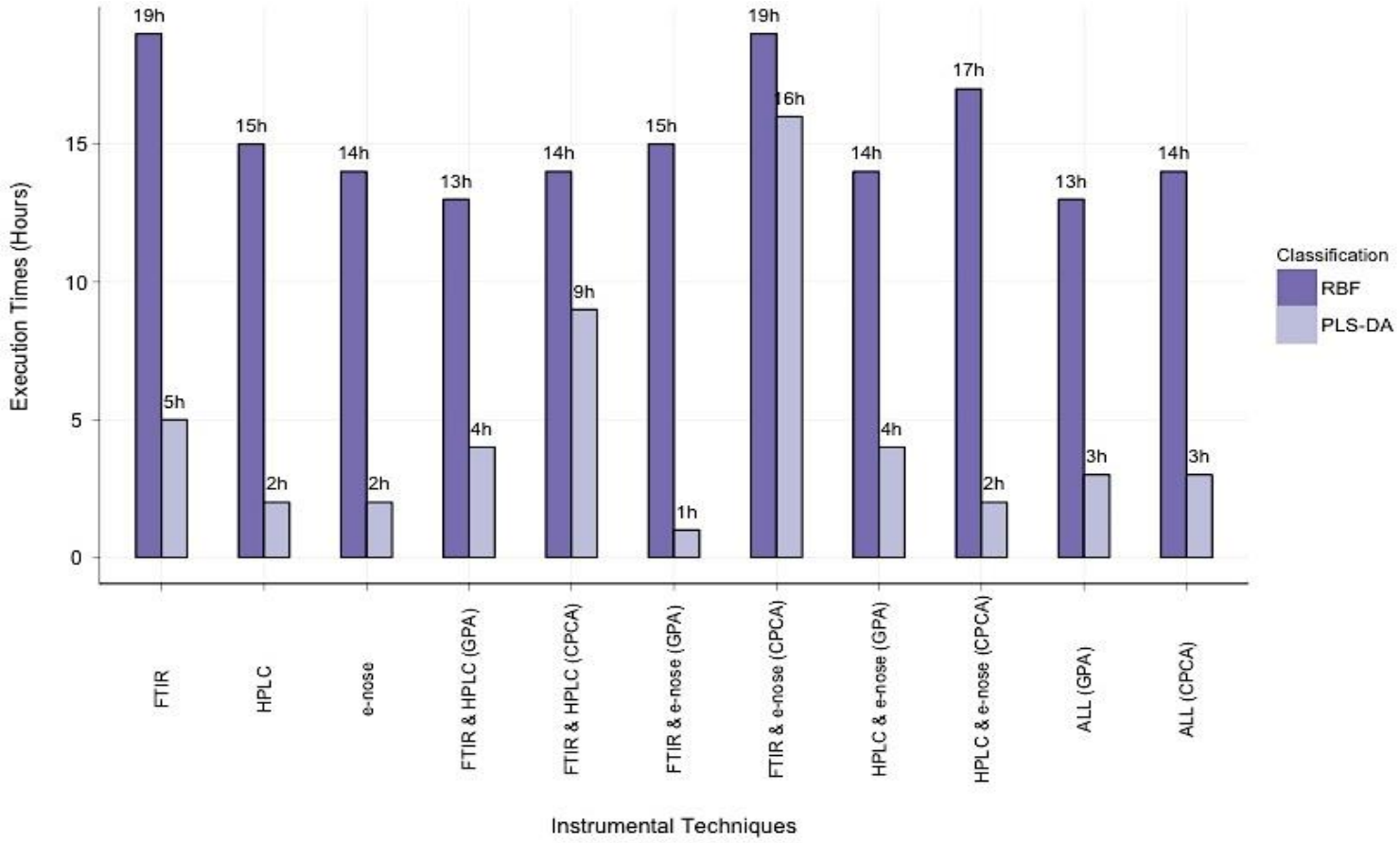


Figure 5-31 Execution times of the permutation tests on the datasets of case study 4

The figure displays the execution times of 100 permutation tests using PLS-DA and RBF SVMs for each standalone and integrated dataset of case study 4. The execution times are based on a fully optimised analysis pipeline featuring parallel programming (master/slave architecture) over eight processors (see Section 5.2.4) as well as fast approximation algorithms for the optimisation of the classifiers' hyperparameters via bootstrapping. The execution times have been rounded towards the nearest integer.

5.4 Comparison of the individual case studies

In this chapter, the suite of machine learning and validation tools developed in Chapter 4 was applied to three further independent case studies. This Section provides an overall comparison of the performance of the three implemented classification ensembles (PLS-DA, linear and nonlinear SVM ensembles) among the different case studies, when applied on standalone and integrated datasets. In addition, this Section aims to reveal any common underlying patterns, similarities and/or differences between the different case studies and the classifiers. The overall trend across all four case studies is presented in the graph of Figure 5-32.

For the FTIR data, all case studies besides the "Survey of minced beef" demonstrate the exact same pattern; clearly, linear classifiers (PLS-DA and linear SVMs) favour the standalone FTIR data both prior and after PCA. On the contrary, the overall performance of kernel-based (RBF) SVMs appears to be inferior to the other two linear classification ensembles. As thoroughly discussed in Section 4.3.2, Xu *et al.* (2006) support that relatively simplistic chemometric algorithms such as PLS-DA are more efficient when applied on traditional analytical techniques such as spectroscopy, where the data are linear and well understood. Therefore, the nonlinear projection into a high-dimensional feature space via a kernel is found unsuitable in this instance. Finally, for the standalone FTIR data that have been subjected to PCA, the accuracies of nonlinear SVMs approximate those of the linear models.

In addition, HPLC proved to be the most diagnostic instrumental technique since it consistently produces the best recorded classification accuracies. In all case studies that include HPLC, the percentages of correctly classified samples ($\%CC$) are significantly higher when SVMs are applied on the datasets as opposed to PLS-DA; the HPLC data demonstrate outstanding results especially in the cases of RBF SVMs. Furthermore, the HPLC datasets that have been subjected to PCA also follow a similar trend, with exception of case study 1.

As far as the e-nose data (prior and post PCA) are concerned, in case study 1 the overall accuracies are significantly higher when SVMs, and especially RBF SVMs, are applied. On the contrary, in case study 4, standalone e-nose data present better predictions for PLS-DA. Even though the overall accuracies appear to be divergent, the classifiers are always strongly influenced in this case by the majority class – whether semi-fresh or spoiled samples. Even so, the e-nose datasets overall did not present any discriminative information, while the produced classification accuracies proved to be statistically non-significant.

Finally, the Raman data of case study 3 produced similar $\%CC$ values to FTIR. However this case study suffered from a major impediment; the highly imbalanced datasets (see Sections 5.2.2 and 5.3.2) resulted in biased decision boundaries, which classified correctly only the sampled of the majority class. Thus, the Raman data have illustrated very poor generalisation performance.

As far as the integrated datasets are concerned, both data fusion methods have demonstrated their own strengths and limitations. Even so, CPCA proved to be a more prominent data fusion technique than GPA. More specifically, CPCA clearly improves the outcome of the integration by combining the strongest assets of the initial datasets, while GPA appears to be consistently dominated by the weakest experimental technique. In the case of integrated datasets, SVMs, and in particular linear SVMs, demonstrated the most promising classification accuracies.

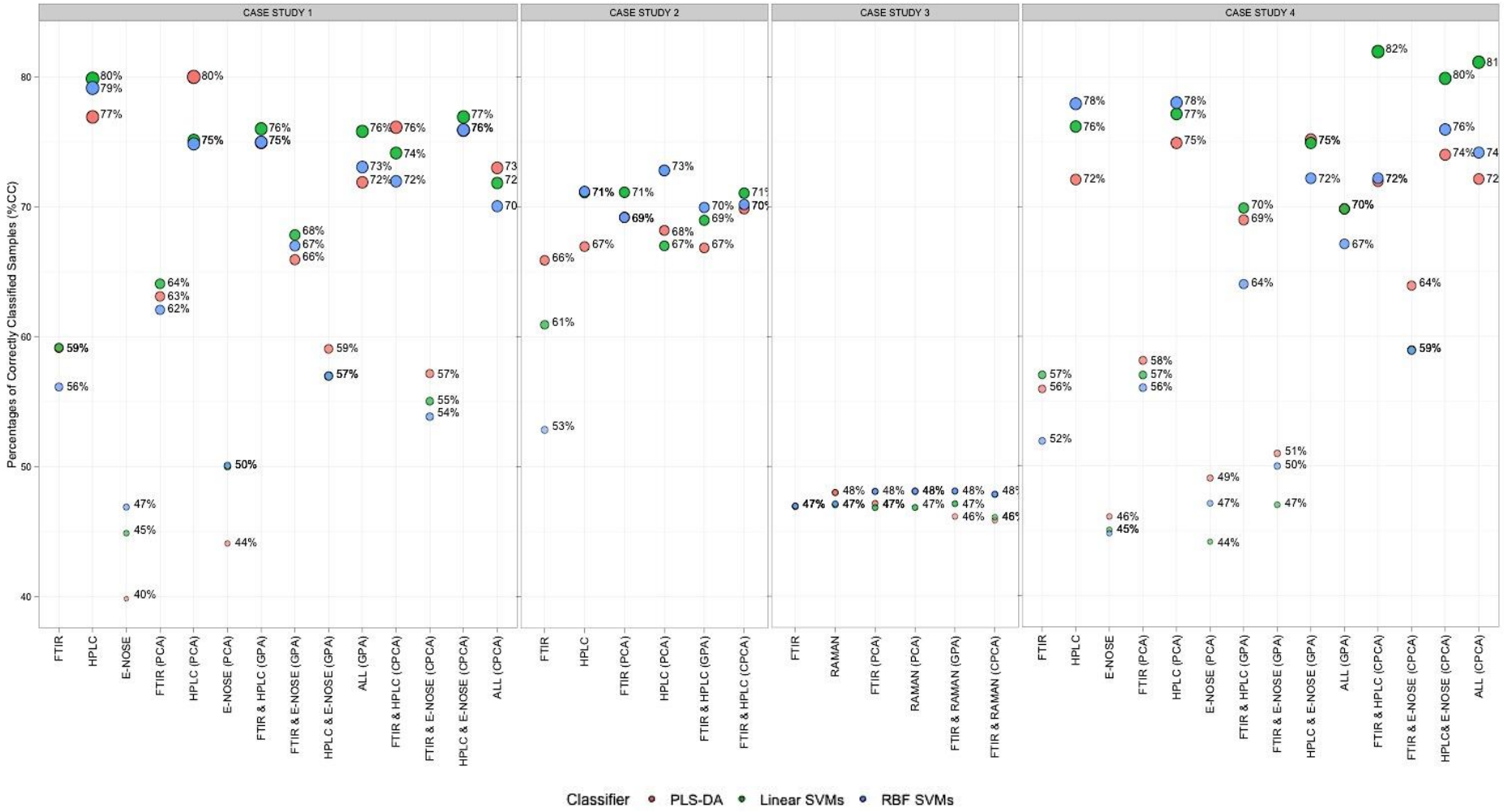


Figure 5-32 Investigating the common trends across all four individual case studies

5.5 Conclusion

In order to verify the generalisation ability of the implemented multivariate analysis pipeline, the constructed statistical tools were tested upon three new individual real-world case studies featuring different types of data (beef fillets, minced beef, pork), which have been analysed using a variety of instruments (FTIR, HPLC, Raman and e-nose) under different temperatures and packaging. By analysing the new case studies, we were able to show that significant results can be obtained on further datasets. In addition, by a direct comparison of their results, some noteworthy conclusions were drawn.

HPLC proved to be the best instrumental technique for the chosen application of assessing meat freshness. Standalone HPLC data, both prior and after the application of PCA, consistently demonstrated the highest percentages of correctly classified samples (%CC). Thus, we can conclude that the provided HPLC data contained abundances of several specific chemical compounds associated with and denoting spoilage. Conversely, the FTIR, Raman and e-nose data were the measurements of raw sensors with no prior feature selection or mapping to specific compounds. Also, it is obvious that the HPLC data present higher classification accuracies for kernel-based (RBF) SVMs. On the contrary, the overall accuracies of the simple spectroscopic data by FTIR are clearly profiting by the application of linear classifiers, and especially by the application of traditional chemometric methods such as PLS-DA. In this instance, the nonlinear mapping by RBF SVMs has been found unfit. Finally, the e-nose data did not demonstrate any discriminative information, and its classification results proved to be statistically non-significant.

As far as the integrated datasets are concerned, CPCA was consistently found to be a better data fusion technique than GPA. More specifically, CPCA clearly improves the outcome of the integration by combining the strongest features of the initial datasets, while GPA appears to be dominated by the weakest experimental technique.

6 Development of improved visualisation methods for chemometrics applications

6.1 Introduction

A range of novel visualisation tools has been used throughout this thesis. This chapter provides a thorough description of the visualisation techniques, graphics libraries and web technologies underlying these tools. Even though there has been tremendous progress in the field of chemometrics, often the visualisation tools used to demonstrate the results are out-dated and the graphs occasionally difficult to comprehend. One of the aims of this project was to pursue the development of new visualisation methods that enhance the interpretability of the project's data and results, with a view to employing them for the construction of static images, interactive web-based graphs, and dynamically generated reports.

6.2 Materials and Methods

6.2.1 The importance of Data Visualisation

Data visualisation can be defined as "the science of visual representation of data, which contains information abstracted in some schematic form, including attributes or variables for the units information" (Friendly, 2001). Over the past years, software systems have attempted to integrate heterogeneous data at both the data source and the semantic level (Goesmann *et al.*, 2003). However, the sheer volume and complexity of the data under study makes it almost impossible to illustrate using traditional visualisation methods.

The chief aim of visualisation is to make the data accessible. The assessment of a good visualisation tool is based on a three-fold relationship among the designer, the reader and the data, as presented in the schematic diagram of Figure 6-1. According to Illinsky *et al.* (2011), in order for a visualisation graphic to be successful, first of all it needs to be informative. In addition, the designer has to convey the information in the most persuasive and efficient (possible) way, by including only precise information while keeping the visual "noise" at a minimum (Steele *et al.*, 2010). Finally, the graphs must be aesthetically pleasing; according to Illinsky *et al.* (2011), visualisation "leverages the incredible capabilities and bandwidth of the humans' visual system to move a huge amount of information into the brain very quickly". The overall level of success of a visualisation tool is highly dependent on the balance in between these relationships. Designers often fail to achieve a balance between functionality and aesthetics, hence leading into extremely complex, abstruse, biased and/or misleading results.

In this thesis, the graphs that represent the results of the multivariate analysis pipeline have been meticulously designed to satisfy all the afore-mentioned attributes; equal importance has been given to the implementation of the analysis pipeline as well as the development of efficacious comprehensive and reproducible graphs.

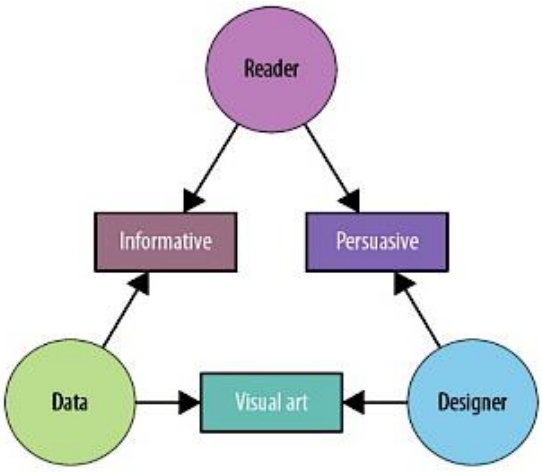


Figure 6-1 The "designer-reader-data trinity" of data visualisation

The visualisation outcome chiefly depends on which of these relationships is dominating over the others. Above all, the visualisation techniques have to be informative in addition to efficient. Furthermore, a visual aid should convey information to the readers in the most persuasive and aesthetically pleasing way. The figure has been extracted from Illinsky *et al.* (2011).

6.2.2 Generating static graphs

As described in the previous chapters, the bulk of the work was implemented using the **R** project for statistical computing (see Section 2.2.6). Even though **R** does not qualify as a scripting language or web technology, it is a powerful and widely popular graphics tool for the construction of "publication-quality diagrams and plots" (Murrell, 2005).

In this project, **R** has been extensively used as a means of constructing high-quality graphics that enhance the interpretability of the analysis results acquired by the multivariate analysis pipeline. Since **R** consists of a plethora of built-in and add-on packages of great variability, it is often extremely difficult to succeed in attaining optimal functionality; therefore, this section attempts to narrow down the enormity of the range of functions down to those of most relevance to the applications covered in this thesis.

6.2.2.1 R packages and functions

The **plot()** function, as provided by the built-in **graphics** package (**R** Development Core Team, 2012), is the core function for plotting **R** objects, and constitutes the backbone for all other graphics packages and plotting functions. In this work, the function was used to construct simple two-dimensional scatter plots. The corresponding three-dimensional scatterplots are available through the popular **scatterplot3d** (Ligges and Mächler, 2003) package and the homonymous function.

A major innovation in the **R** graphics field has been achieved with the introduction of the **ggplot2** package (Wickham, 2009; Wickham and Chang, 2012). Nowadays, the **ggplot2** package is widely recognised among the **R** users as the "best graphics package for **R**" (http://www.inside-r.org/packages/ggplot2/reviews/simply-best-graphics-package-r) due to its clean and subtle aesthetics, ease of use and intuitive syntax. The package has been developed based on the Grammar of Graphics, written by Wilkinson (Wilkinson, 2005; Wilkinson *et al.*, 2005). Each new plot consists of several independent reusable components, built in a layered grammar (Figure 6-2), where each of these features provides an additional functionality. Thus, **ggplot2** is

extremely powerful since it grants the users the ability to construct graphics precisely tailored to their needs rather than using a set of pre-defined plots. The majority of graphs presented in this PhD are solely based on the **ggplot2** package; the functionality of the package has been customised in order to highlight the results and the outcome along every step of the multivariate analysis pipeline. The generated graphs cover a wide range of different plots, from simple scatterplots to histograms and barplots with complex layouts, among others.

The **ggplot2** package was also applied as a means of highlighting the density of points in a two-dimensional scatterplot. A similar functionality is also provided for the basic **plot()** function by the **KernSmooth** package (Wand *et al.*, 2011). In this instance, the scatter plots are enhanced with smoothed colour density representation based on the algorithm by Wand and Jones (Wand *et al.*, 1995).

Finally, the **ellipse** package was utilised to generate two-dimensional scatterplots with 95% confidence ellipses for each distinct input class. The methods were used to provide PC scores plots that convey more information than the simple scatter plots normally used.

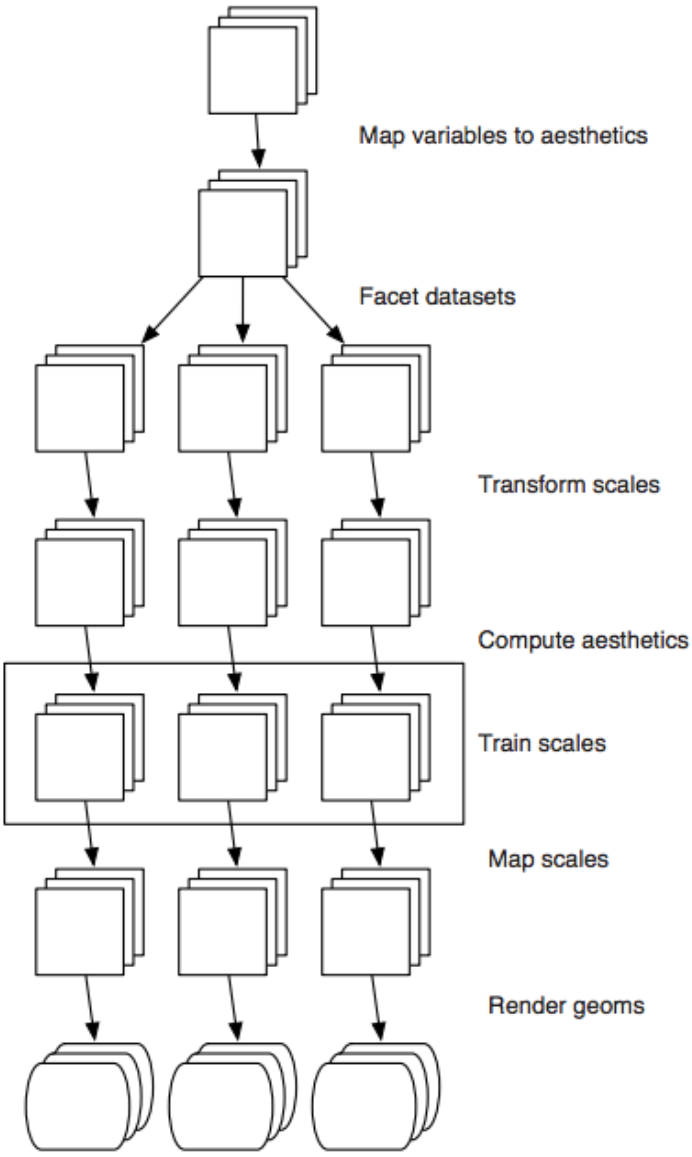


Figure 6-2 Construction process of a ggplot2 graph – the layered grammar approach

The figure depicts the construction process of a graph by the **ggplot2** package. Each square in the figure stands for a single layer; a layer is formed by the combination of data, mappings, statistical information and the geometric object (geom), which controls the type of graph to be plotted. A plot may contain multiple layers; in this figure, the plot consists of three distinct layers and three panels. The components that usually make up a **ggplot2** plot include the data, the mappings from variables to aesthetics, scales, coordinate and facet specifications, layers of annotations and geometric objects (geoms), among others. The figure has been extracted from Wickham (2009).

To construct the contour and surface plots (Section 2.3.2.3) of the optimisation process, the two afore-mentioned packages were once more employed. More specifically, the **persp()** function of the **graphics** package was applied to draw three-dimensional surface (perspective) plots; the functionality of the built-in **R** function was extended to allow user-defined colour palettes as well as the addition of points on the surface plot. In addition, the **contour()** function was employed for the construction of contour plots, or alternatively the addition of contour lines to an existing plot. The contour plots may also be filled by using the **filled.contour()** function. The **ggplot2** package also provides the users with a contour function, by simply adding the **stat_contour()** layer in the geometric object (geom component). Finally, the **lattice** package (Sarkar, 2008) provides the **contourplot()** function that implements similar plots upon a grid.

The grid plots of Section 2.3.2.3 and Section 3.3.2 were based on the **levelplot()** function by the **lattice** package and the **image.plot()** function by the **fields** package (Furrer *et al*., 2012) respectively. Once more, the functionality of the command was extended in order to support the visualisation of points and simplices in the graphs. In addition, the **ggplot2** package provides similar functionality by adding the **geom_raster()** or **scale_fill_gradientn()** components; however, in the case of **ggplot2**, the resolution of the grid is not as great as the afore-mentioned packages.

Finally, the spectroscopic data were visualised using the **emu** package (Harrington, 2011); similar functionality is provided by the newly introduced **ChemoSpec** package (Hanson, 2012) for the visualisation of spectroscopic data and chromatograms.

6.2.2.2 Generating dynamic reports *via* R

In the context of reproducible research, the **Sweave** (Leisch, 2002; Leisch, 2005) framework has also been investigated for the construction of dynamic reports *via* **R**. Traditionally, the execution of **R** scripts is conducted first, followed by the gathering and reporting of the results. This approach is suitable for a small number of repetitions, however it is not very practical when the outcome of the analyses needs to be reproduced a plethora of times. **Sweave** supports the incorporation of **R** commands and/or scripts within **LATEX** documents (Leisch, 2005). Therefore, the functionality of high-quality typesetting and data analysis as provided by **LATEX** and **R** respectively, is integrated into a single unified statistical document (Leisch, 2005). The users can generate dynamic reports, which contain on-the-fly **R** output such as figures, tables, documentation, even **R** commands, among many others. In case that the input data or analysis change, the contents of the generated reports are automatically updated; thus, the greatest appeal provided by **Sweave** is reproducibility. Furthermore, the noweb syntax adopted by **Sweave** (Leisch, 2005) is extremely simple and straightforward to learn.

The **Sweave** framework is currently a built-in **R** feature provided by the **utils** package (**R** Development Core Team, 2012). Instead of storing and running the commands from within an ***.R** script, a ***.Rnw** file is used instead as illustrated in Figure 6-3. **Sweave** translates the initial document into a **LATEX** file. Finally, the **pdflatex()** (or **latex()**) command compiles the **LATEX** file and generates a new report (such as .pdf or .eps or both), which contains the output as acquired by **R**.

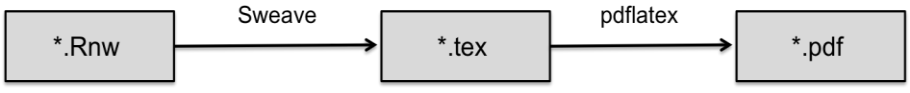


Figure 6-3 The workflow from Sweave to an automatically generated PDF file
Initially, a ***.Rnw** file is created that includes the **R** scripts in addition to the **LATEX** commands for high-quality typesetting. Finally, the ***.tex** file is compiled into a ***.pdf** file.

6.2.3 Web technologies and Scripting languages

Over the past years, the World Wide Web (WWW or W3) (Berners-Lee, 1996; Berners-Lee *et al*., 2001; Berners-Lee and Fischetti, 2008) has profoundly changed the rapidity of acquisition and storage of knowledge, as well as promoting the development of hitherto unimaginably intricate methods of dealing with the vast enormity of newly acquired knowledge and/or data. Web development and design is indeed an immense and complex field. With the exponential development of the informatics field over the past few years, nowadays there can be traced literally thousands upon thousands of different web technologies, methodologies, disciplines and standards, programming languages, and software packages. Some of the core and commonly used web technologies are depicted in Figure 6-4.

The Web as we know it nowadays is actually the second phase in the Web's evolution; namely, Web 2.0 (DiNucci, 1999; O'Reilly, 2005; O'Reilly, 2009). Web 2.0 is a collection of technologies, business strategies, and social trends; compared to its predecessor (Web 1.0), it is more dynamic, interactive, customised and media-intensive. Whereas with Web 1.0 the end-users remained passive simply viewing the contents of a web page, Web 2.0 allows its users to view, interact but also contribute to the content of a web page. Currently, a transition from Web 2.0 towards Web 3.0 is already emerging; Web 3.0 will include personalised and user-behaviour features, thus providing "a portable personal semantic web".

Web and grid technologies have developed at a dazzling rate, offering a multitude of advantages to the life sciences as a bonus from the computational sciences. These technologies have moved from their classical and somewhat static architectures to more dynamic and service-oriented ones as illustrated in Figure 6-4. Rich Internet Applications (RIAs) (Fraternali *et al.*, 2010) became increasingly important and popular during the past decade, and currently play a prominent role towards the evolution of Web 2.0. RIAs resemble to a vast degree the characteristics and functionality of desktop applications, even though they are web-based, since they feature responsive user interfaces and interactive capabilities. Thus, web-based programs become easier to use and more functional to use compared to the static and limited web pages of Web 1.0. Clients in the RIA model handle user-interface-related

activity, while simultaneously the server processes and stores data in addition to returning the results back to the user. A RIA runs on a web browser, usually without the necessity of installing any software on the client side. In addition, RIAs are typically based on, or feature, asynchronous communication such as Asynchronous **JavaScript** and **XML** (**AJAX**). Therefore, clients can interact with the webpage without having to wait for results from the server.

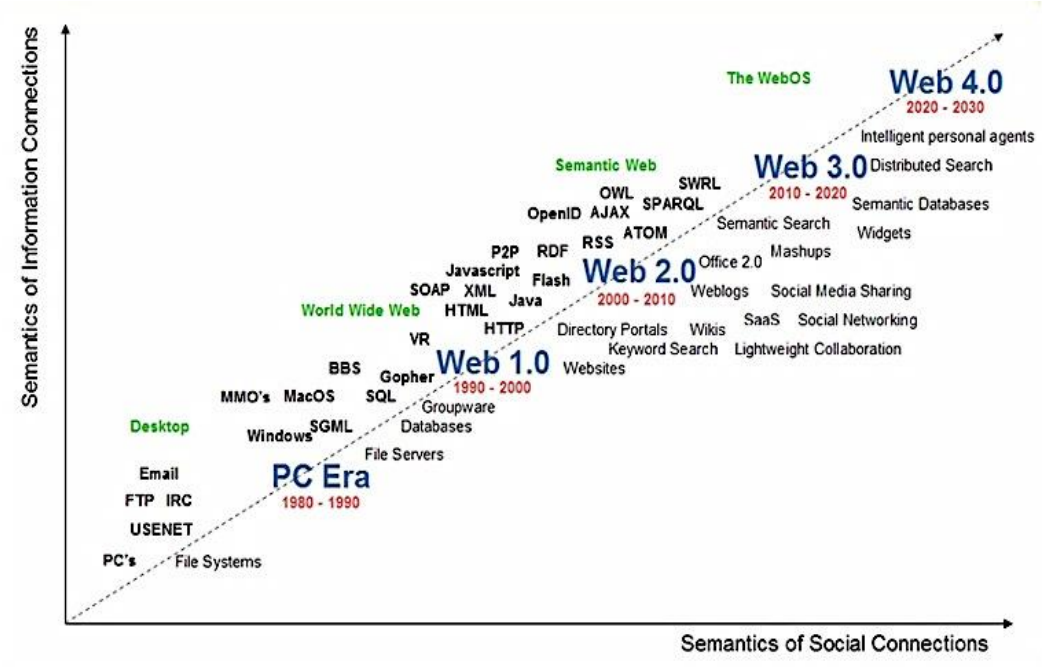


Figure 6-4 The progress of web technologies and programming languages over time

The figure illustrates the progress of web technologies, from the PC era to current times where Web 2.0 is the dominant web philosphy, all the way to proposed future web technologies such as Web 3.0 and Web 4.0. Along with the advance of web technologies, various different programming languages are displayed based upon each era. The shift of interest from static technologies such as HTML to dynamic, interactive and content-rich languages such as AJAX is depicted. The figure has been extracted from http://thepaisano.files.wordpress.com/2008/03/webtimeline.jpg

Asynchronous **JavaScript** and **XML** (**AJAX**) (Garrett, 2005) was introduced as a means of web application development. **AJAX** is not a novel language or a single technology; rather, it is "a powerful combination of several different vigorous technologies, each flourishing in its own right" (Garrett, 2005; Lin *et al.*, 2008; Wang *et al.*, 2008). **AJAX** incorporates:

1. **DOM** for providing dynamic display and interaction
2. **CSS** and **XHTML** for carrying out standardised-format presentation
3. **XML** and **XSTL** for carrying out data exchange and processing
4. The **XMLHttpRequest** object for asynchronous data retrieving and handling
5. **JavaScript** for data processing and binding all the technologies together

The adoption of **AJAX** has drastically enhanced interactive functionality in websites due its asynchronous nature. The classic web application model makes use of synchronous interactions in a request-wait-response client-server model. In this process, as illustrated in Figure 6-5, the client (user) triggers initially an **HTTP** request to the web server through a web interface. Subsequently, the server analyses the request sent by the client, and carries out any processing tasks such as retrieving data, among many others. Due to the synchronous nature of the model, the client has to unnecessarily wait while the server processes the submitted data. For every requested task, the application is locked up and the waiting time increases exponentially. In most cases, the browser displays blank pages while processing the data and the users must wait until the entire **HTML** page is reloaded. Finally, the server returns a response along with the requested results back to the user. This model is adapted from the Web's original use as a hypertext medium, but it is not found fit for software applications since it makes a lot of technical sense, but does not make for a great user experience.

AJAX on the other hand, which is based on asynchronous interactions, follows a completely different approach when compared to the classic web application model, as presented in Figure 6-5. **AJAX** allows the users to continue interacting with the web interface without any interruptions or page reloads, while simultaneously, messages are exchanged with the web server in the background. The **XMLHttpRequest** object, which constitutes the backbone of the **AJAX** methodology, uses a request-response model that supports the indiscernible exchange of data between client and server in the background. Therefore, the users avoid communicating directly with the server by stimulating an **HTTP** request as in the case of the classic model. On the contrary, due to the asynchronous nature of this methodology, the displayed contents of the browser can never be scintillated, delayed

or disappeared (Wang *et al.*, 2008). Finally, data are returned to the **AJAX** engine without the necessity of completing the entire processing. The remaining supporting technologies incorporated into the **AJAX** method are in charge of creating direct, richer and more dynamic web interfaces that look and act like local desktop applications.

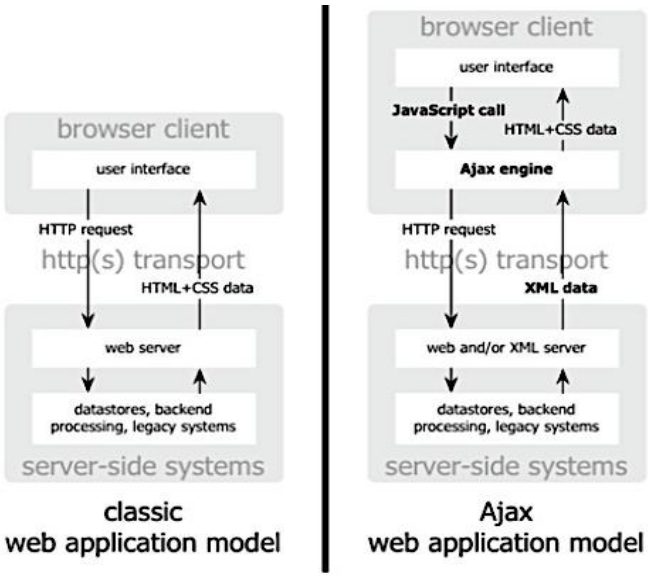


Figure 6-5 Comparison between the classic and the AJAX web application model

The figure provides a direct comparison between the traditional synchronous web application approach (left) and the asynchronous **AJAX** web application model (right). The figure has been extracted from Lin *et al.*, (2008).

jQuery (http://jquery.com/) is a light and exceptionally fast "write less, do more" **JavaScript** library, which simplifies client-side scripting and offers feature-rich functionality. Former long and complex **JavaScript** commands are simplified *via* **jQuery** to within a single or a few lines. For instance, **HTML** traversing, event handling, **AJAX** requests and replies, as well as animating become more rapid and user-friendly.

In addition, the **DataTables** plug-in (http://www.datatables.net/) for the **jQuery** library has been employed for the construction of fully interactive feature-rich **HTML** tables, as opposed to the traditional static **HTML** tables within a web interface. **DataTables** support features such as pagination, on-the-fly filtering and sorting, among many others.

JavaScript Object Notation (**JSON**) is a lightweight, text-based, data interchange format (Crockford, 2006). Even though **JSON** derives straight from **JavaScript**, it is language-independent, hence it can be used by many different programming languages. **JSON** was employed in combination with **jQuery** and **AJAX** for the asynchronous and indiscernible exchange of data in the background between the web-browser and the web-server.

On the server-side, **Perl** (http://www.perl.org/) version 5.8.0 was used as the supporting scripting language. Even though **Perl** is not part of the RIA approach, it still constitutes a powerful tool in the field of bioinformatics.

6.2.4 The iWebPlots package

Over the past years, there have been numerous efforts to implement interactive graphics *via* **R**. In contrast to static graphs, with interactive plots the users have the ability to select, query and interact with the area defined by the plot as well as make use of dynamic features such as rotations and zooming. Only a miniscule number of **R** packages that implement this functionality have been released and they usually act as standalone programs. Therefore, the probability of them being embedded within web pages is highly diminished. This drawback can usually be overridden, when the plots are used in combination with other programming languages and/or plug-ins that add the interactive functionality.

The **iWebPlots** package (Chatzimichali and Bessant, 2011) is a novel **R** package for the creation of interactive web-based plots, developed during this project. The first version of the package is available on CRAN, the official **R** repository, at http://CRAN.R-project.org/package=iWebPlots. However, the functionality of the package has been expanded since the first release, therefore new releases will be uploaded in the near future.

In this package, interactivity is implemented using the fundamental **HTML** image maps methodology. An image-map can be defined as an image with clickable regions, commonly referred to as “hot spots”. These interactive areas may consist of

rectangles, circles and/or irregular polygons. **iWebPlots** is mainly dependent on the **geneplotter Bioconductor** (Gentleman *et al.*, 2004; Gentleman and Biocore, 2012) package, which, among its other features, generates interactive heatmaps for microarray data. The functionality of the interactive image maps, as provided by the **geneplotter** package, was further extended towards building a variety of different types of plots such as scatterplots, histograms and barplots, among many others.

The functions of **iWebPlots** take as input a matrix of coordinates with associated metadata, which are subsequently drawn in a plot using the **R** platform. The image that contains the plot is subsequently saved as a bitmap or PNG image. Prior to adding the image maps' interactive features, the coordinates of the plot have to be converted from the Cartesian coordinate system into the user graphics coordinate system, which is measured in pixels. Once the corresponding pixels are calculated, an **HTML** layer is constructed upon the PNG image. At the end of this process, every point in the plot corresponds to a clickable area within the map.

Additional features provided by the **iWebPlots** package include dynamic tooltips and text annotations as well as asynchronous alternations between two- dimensional and three-dimensional scatterplots. Finally, each plot can be interlinked with a static (**HTML**) or fully interactive data table (**DataTables**), which displays additional information about the data in the graph. The text of the tooltips and the data tables is customised directly by the users.

The greatest strength of the **iWebPlots** package is its simplicity and ease of use; the package is based on pure **HTML** image maps methodology, so additional specialised software such as applets, libraries, graphical user interfaces (GUIs) or plug-ins are required by the end user. This allows the generated web-based plots to run problem-free in every web browser with minimum execution times. The output of **iWebPlots** can be embedded within any **HTML** page or web project outside the realms of this project. Furthermore, the users can easily modify and expand the functionality of the package and of the generated **HTML** pages, therefore great extensibility is ensured.
There are other **R** packages that implement similar functionality. The **googleVis** (Gesmann and de Castillo, 2011) package provides an **R** interface for the popular

Google Chart Tools (https://developers.google.com/chart/). The users can create dynamic, interactive, feature-rich web-based graphs *via* **R**, without uploading their data directly to Google. However, the package does have certain limitations; in order to visualise the graphs in the browser, there is a need for Internet connection in addition to the installation of the Flash plug-in. Another package, **iPlots** (Urbanek and Theus, 2003) is a powerful **R** package for the creation of interactive graphics that offers features such as querying, linked highlighting, and colour brushing, among others. Even so, the **iPlots** package runs as a standalone graphical user interface (GUI) within the **R** console, and cannot be embedded within any other project or interface. In addition, the core of the package is written in **Java**, hence a fully working **Java** installation on the local machine is expected.

6.2.5 Constructing a web interface for demonstrative purposes

As part of this project, a web interface has been constructed for demonstrative purposes as the front-end of the multivariate analysis pipeline. This approach attempts to evaluate whether the results of the analyses can be directly embedded within any **HTML** page and/or web project. As presented in Section 6.2.1, the web interface was chiefly built by using two powerful **JavaScript**-based technologies; namely, **jQuery** and **AJAX**. In this work, **AJAX** requests are triggered from the web front-end, more specifically from **jQuery** scripts embedded within static **HTML** pages, towards specific **Perl** files located at the server-side. These requests may also carry additional information such as associated data and input parameters. Subsequently, the **Perl** scripts execute the relevant **R** scripts, also stored on the web server, after passing these parameters. Once the **R** script terminates the execution of the scripts, it returns the output of the analyses (calculation results, plots, files, *et al.*) back to **Perl**. Finally, **Perl** sends an asynchronous **AJAX** response along with the requested results back to the original client-based script. The web interface is partially updated and the output from **R** is displayed to the user, without the need of reloading the entire page. Due to **AJAX** and its asynchronous functionality, the exchange of requests, responses and data is performed within seconds.

In addition to the static figures generated by **R**, the **iWebPlots** package (see Section 6.2.4) was also employed on the server-side for the construction of fully interactive web-based graphs. The graphs were easily incorporated in the main web interface for users to view. The functionality of **iWebPlots** in this context was successfully tested on the outcome of supervised and unsupervised techniques such as Principal Component Analysis (PCA).

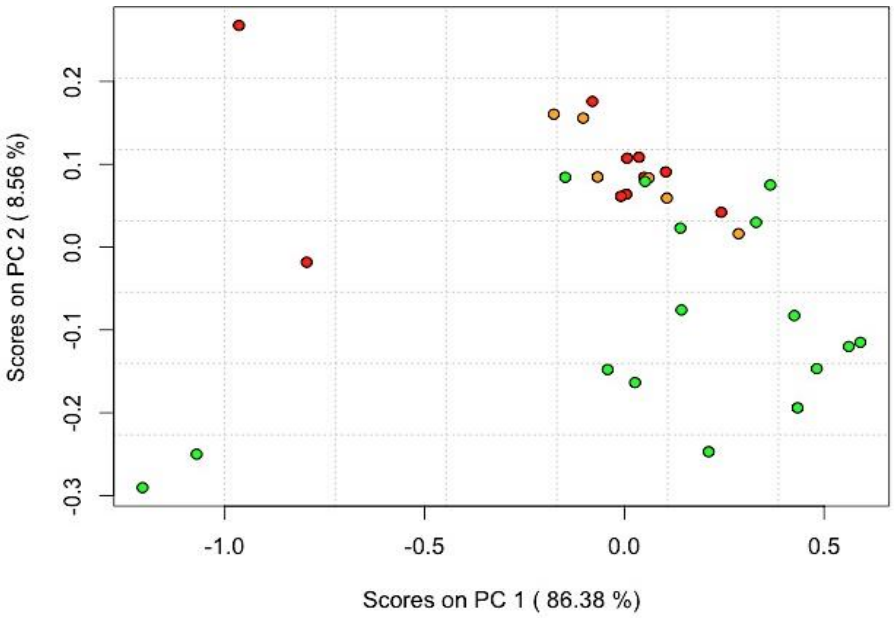
Finally, the **DataTables** plug-in has been used in order to construct a fully interactive data table. The table stores and displays associated metadata for the objects displayed in the plot; for instance, in the case study 1 (shelf life of beef fillets), such information may contain the name, class (sensory scores) and/or instrumental technique for each sample depicted in the graph. These data are exported from **R** in a **JSON** format that is received by **AJAX** and parsed by **jQuery**. One of the most powerful attributes of **DataTables** is the provided search mechanism. More specifically, the table can be filtered using either the default search box or by clicking one of the points in the interactive map. This "on-click" event generates an **AJAX** request, which is posting the selected sample's name as a parameter against the entries of the data table. As a result, the table contents are filtered and only the matches are displayed. Finally, additional attributes include pagination, sorting as well as colouring of the rows based on a predefined criterion such as the class values (fresh, semi-fresh, spoiled samples).

6.3 Results and Discussion

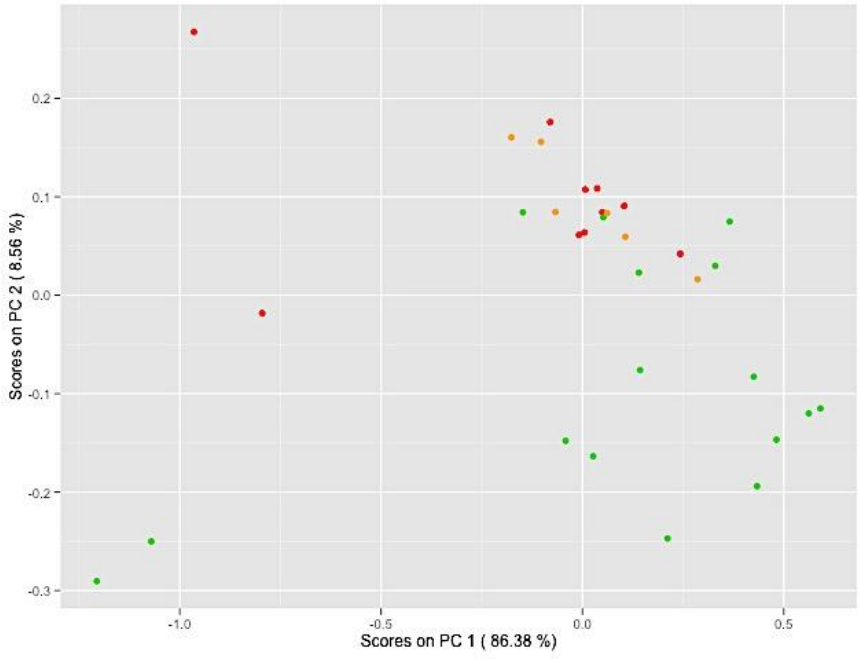
All the graphs throughout this thesis, which demonstrate the results of the multivariate analysis pipeline, have been generated by in-house scripts according to the methodology described in this chapter. Despite the variegated background and origins of these methods, the purpose of the implementation aimed to satisfy a common goal: to enhance the interpretability of the results as obtained by the multivariate analysis pipeline.

For the first step of the multivariate analysis pipeline, data visualisation was employed as a means of exploratory data analysis in order to highlight any underlying trends and outliers. Initially, the output of PCA was visually represented using the base **R** graphics system, as presented in the static scatterplot of Figure 6-6(a). The generated two-dimensional plots were further enhanced with smoothed colour density representation based on the algorithm by Wand and Jones (1995) for the detection of any possible clusters (Figure 6-7(a)). It was therefore apparent that the limited functionality of the traditional plotting methods in addition to the predefined set of built-in parameters resulted in aesthetically poor graphs that did not enhance interpretability.

Therefore, the powerful **ggplot2 R** package was extensively employed for the construction of publication-quality multi-layered graphics. Based on Figure 6-6(b) and Figure 6-7(b), it is obvious that the information depicted in the graphs has not changed; however, the plots are currently satisfying all the key points of Section 6.2.1, by producing effective, persuasive and aesthetically pleasing graphs.



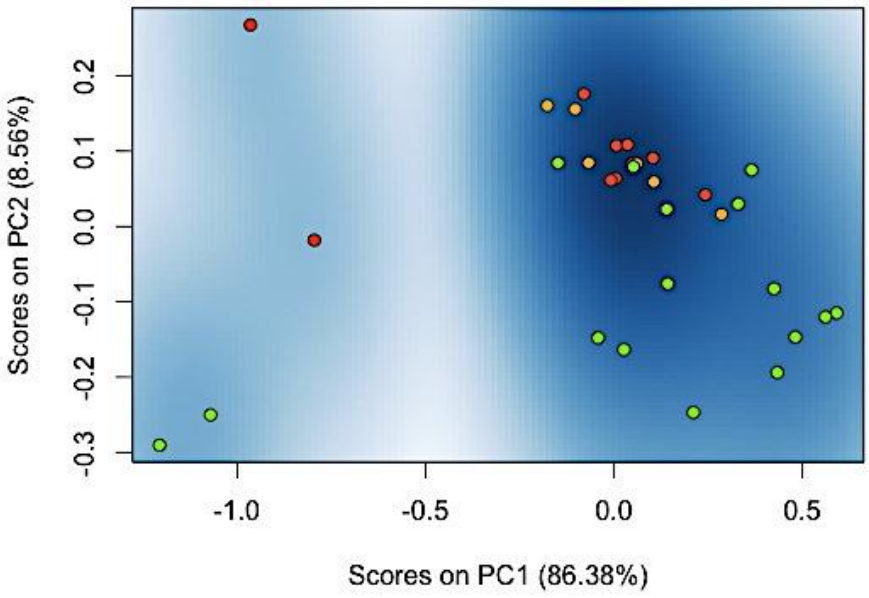
c) graphics package



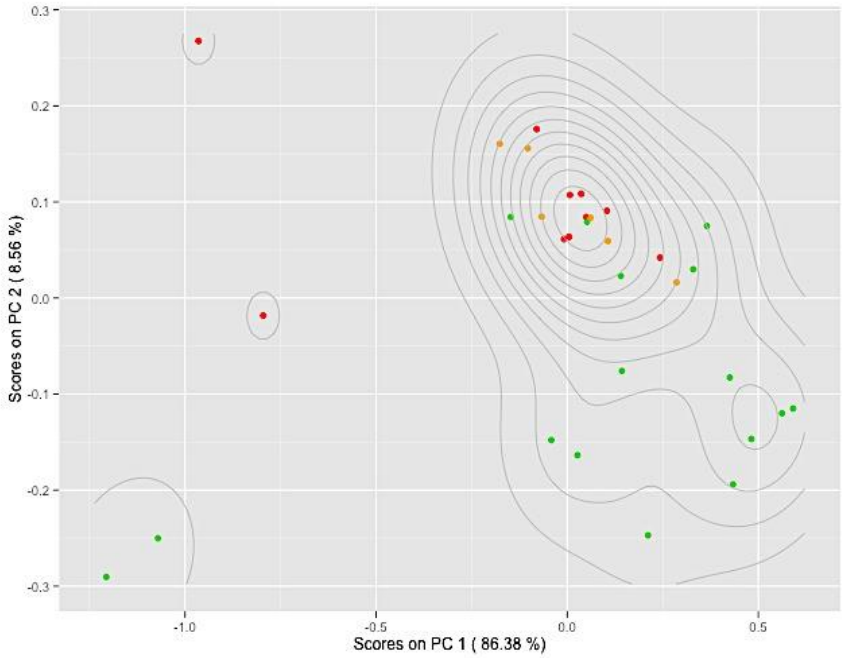
d) ggplot2 package

Figure 6-6 Scatterplots produced by the graphics and ggplot2 package respectively

The figure provides a direct comparison between the two-dimensional scatterplots generated by the built-in **graphics** package and the add-on **ggplot2** package respectively. The scatterplots demonstrate the outcome of PCA when applied on the standalone mean-centered FTIR data of case study 1. Colour representation was used to identify the three classes according to their relevant sensory scores: fresh (red colour), semi-fresh (orange colour) and spoiled (green colour). For comparison purposes, only the 32 common samples (see Section 2.3) are depicted in the graphs. It is obvious, that **ggplot2** forms more aesthetically pleasing graphs compared to the default R graphics system.



a) KernSmooth package



b) ggplot2 package

Figure 6-7 Scatterplots with density estimation using the KernSmooth and ggplot2 packages

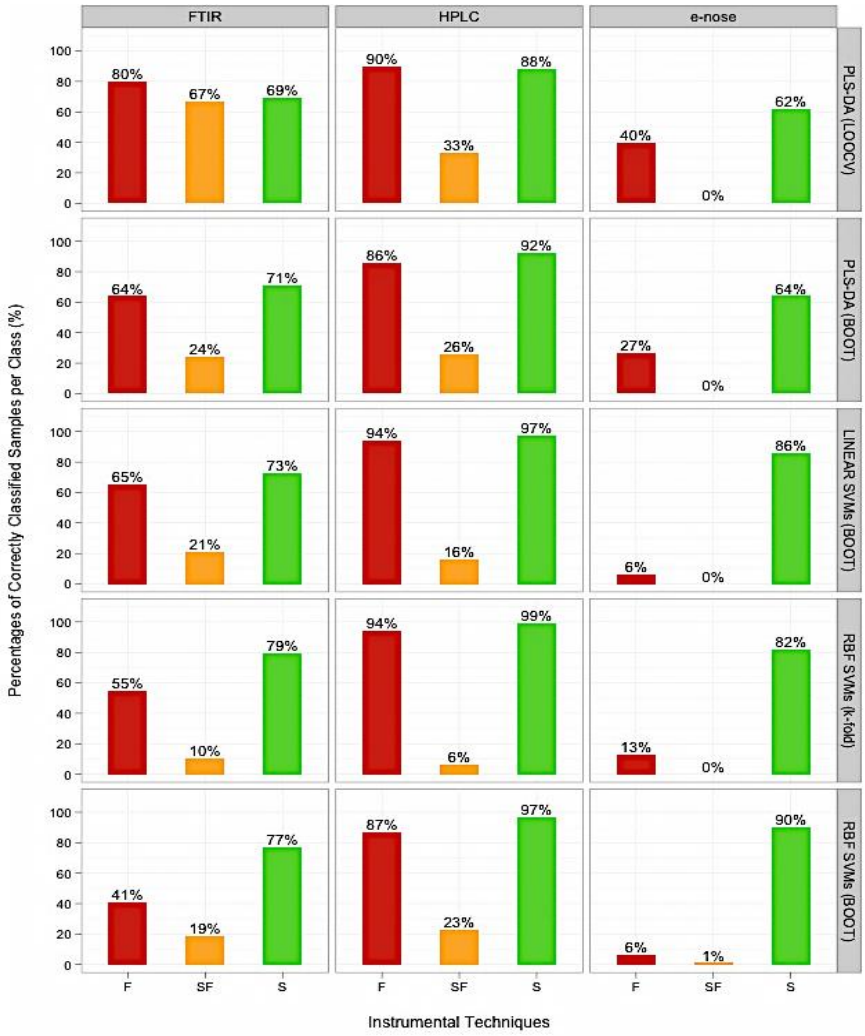
Two-dimensional smoothed colour density scatterplots display the outcome of PCA when applied on standalone mean-centered FTIR data. Colour representation was used once more to identify the three distinct classes. Kernel density estimation techniques were applied to highlight the formation of any possible clusters. The scatter plots were initially enhanced with smoothed colour density representation using the **KernSmooth R** package (Wand *et al.*, 2011). However, the density contour plots drawn by the **ggplot2** package are relatively more intuitive and easier to comprehend than the basic plot.

In addition to aesthetics, faceting is the second most important functionality provided by the **ggplot2** package. Facets are tables of graphics that represent subsets of data, commonly depicting the same type of graph. Faceting is an extremely powerful tool that enables direct and straightforward comparisons using several different parameters simultaneously. Figure 6-8 highlights the necessity for the application of specialised feature-rich libraries, using as an example the per-class percentages of correctly classifies samples from Section 2.3.2.2; in this instance, the two different graphical techniques that have been employed display exactly the same information. The first approach displays information using a static table of data. Even though this approach is extremely informative and efficient, it is relatively difficult for the reader to perform a direct visual comparison among the different cases, and extract any trends and/or underlying relationships. On the contrary, the powerful **ggplot2** library makes use of different visualisation attributes, such as shapes, colour palettes and facets to grant the users with the ability to compare all different cases simultaneously. This functionality can be further extended to integrate several individual facets, which may consist of several different types of plots, together into a unified view. This approach, enables the simultaneous comparison of several different graphs in addition to parameters, and hence the comparison of several different aspects of the same study (see Figure 5-32).

All the aspects that have been described thus far can be further exploited with the application of **Sweave** for the construction of dynamically generated reports. Figure 6-9 demonstrates the simplicity upon which the **Sweave** framework is built. The **R** scripts are wrapped within a small set of **LATEX** commands, which generate a pdf report in a dynamic and automatic fashion. The same **Sweave** document such as the one of Figure 6-9 can be successfully applied unaltered to several other input files in order to generate the exact same plot but on different data. Therefore, this approach verifies that reproducible documentation as well as graphs and figures can be produced upon demand.

In order to test the applicability and generic nature of the package, it was thoroughly examined on various different Case Studies. The package constructs dynamic barplots, histograms and scatterplots, among others. Figure 6-10 demonstrates the most commonly applied visual aid for the outcome of hierarchical cluster analysis (HCA) – the dendrogram – as generated directly by the **iWebPlots** package. The lines connecting the nodes in the dendrogram represent the distance (the degree of dissimilarity) between the leaves or clusters. In this instance, the HCA algorithm has been applied on the PCA scores of the HPLC dataset for case study 1. Similar to the PCA score plots of Section 2.3.1, it can be concluded from the figure that mainly spoiled and fresh samples present good clustering, whereas semi-fresh samples are usually grouped with either one of the two. The leaves of the cluster are all interactive, while they can be easily coloured upon the user's request.

The web pages offer fully interactive plots and data tables with dynamic and asynchronous features (Figure 6-11) as described in Section 6.2.5. The users can interact with each point in the plot, which automatically filters the corresponding entry in the data table, whereas rollover and click upon events display dynamic tooltips with associated metadata. In addition, the users can highlight one, multiple or all available classification groups (fresh, semi-fresh, spoiled). Furthermore, the samples' names can be dynamically enabled or disabled in the plots, whereas the users can alternate between two-dimensional and three-dimensional plots as easily. The scaling of the data and the axes can be altered at any time asynchronously without reloading the whole page. Finally, the users can view the spectroscopic data on the web interface upon demand.



Datasets	FTIR			HPLC			e-nose		
PLS-DA (LOOCV)	80%	67%	69%	90%	33%	88%	40%	0%	62%
PLS-DA (BOOT)	64%	24%	71%	86%	26%	92%	27%	0%	64%
k-fold Linear SVM	65%	21%	73%	94%	16%	87%	6%	0%	86%
k-fold RBF SVM	55%	10%	79%	94%	6%	99%	13%	0%	82%
BOOT RBF SVM	41%	19%	77%	87%	23%	97%	6%	1%	90%

Figure 6-8 Comparison of static data representation and powerful feature-rich visualisation

The figure illustrates the significant differences between traditional visualisation approaches such as the static tables and the proposed visualisation tools. Even though the static table is informative, it is relatively difficult to perform a direct comparison across the different entries. On the contrary, the application of **ggplot2** provides an efficient and straightforward way to simultaneously compare several parameters under study (instrumental techniques and classification/validation models, distinct classes, overall accuracies).

```
\documentclass[a4paper]{article}

\title{Permutation Results}
\author{Chatzimichali Eleni}

\begin{document}

<<echo=FALSE>>=
library(ggplot2)

permRes_all <- read.csv("permRes_all.csv")
Instrument_Names <- unique(permRes_all$Instrument)
permRes_all$Instrument <- factor(permRes_all$Instrument, levels = rev(Instrument_Names))
permRes_all$Classification <- factor(permRes_all$Classification, levels = c("RBF", "PLS-DA"))
@
\begin{center}
\setkeys{Gin}{width=1.0\textwidth}
<<echo=FALSE,fig=TRUE,width=9,height=15>>=

ggplot(data=permRes_all, aes(x = Instrument, y=Accuracies, fill = rev(Instrument))) +
  geom_boxplot()+geom_point(aes(x = Instrument, y=Real_Value, colour="red"), size=2.3) +
  facet_wrap(~Classification, nrow=2) + coord_flip() + opts(legend.position="none",
  axis.title.y = theme_text(size=13, angle=90), axis.title.x = theme_text(size=13),
  axis.text.y = theme_text(size = 12), axis.text.x=theme_text(size=12))

@
\end{center}
\end{document}
```

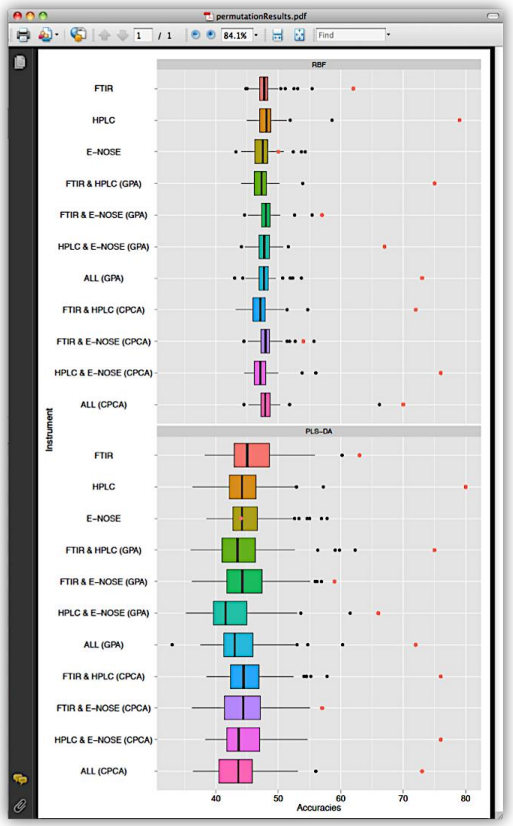


Figure 6-9 Sweave example for the dynamic construction of a PDF report directly from R

The figure illustrates the simplicity of the **Sweave** format; all **R** functions are fused together with the **LATEX** commands into a single ***.Rnw** file. In this example, **Sweave** generates a dynamic pdf report containing a **ggplot2** boxplot for the comparison of permutation results for different classification ensembles. The relevant data are read from a comma-delimited Excel file. If the input data were to be changed by providing a different file, the actual **R** script could still be used without alterations in order to reproduce the final figure; thus, great reproducibility is accomplished.

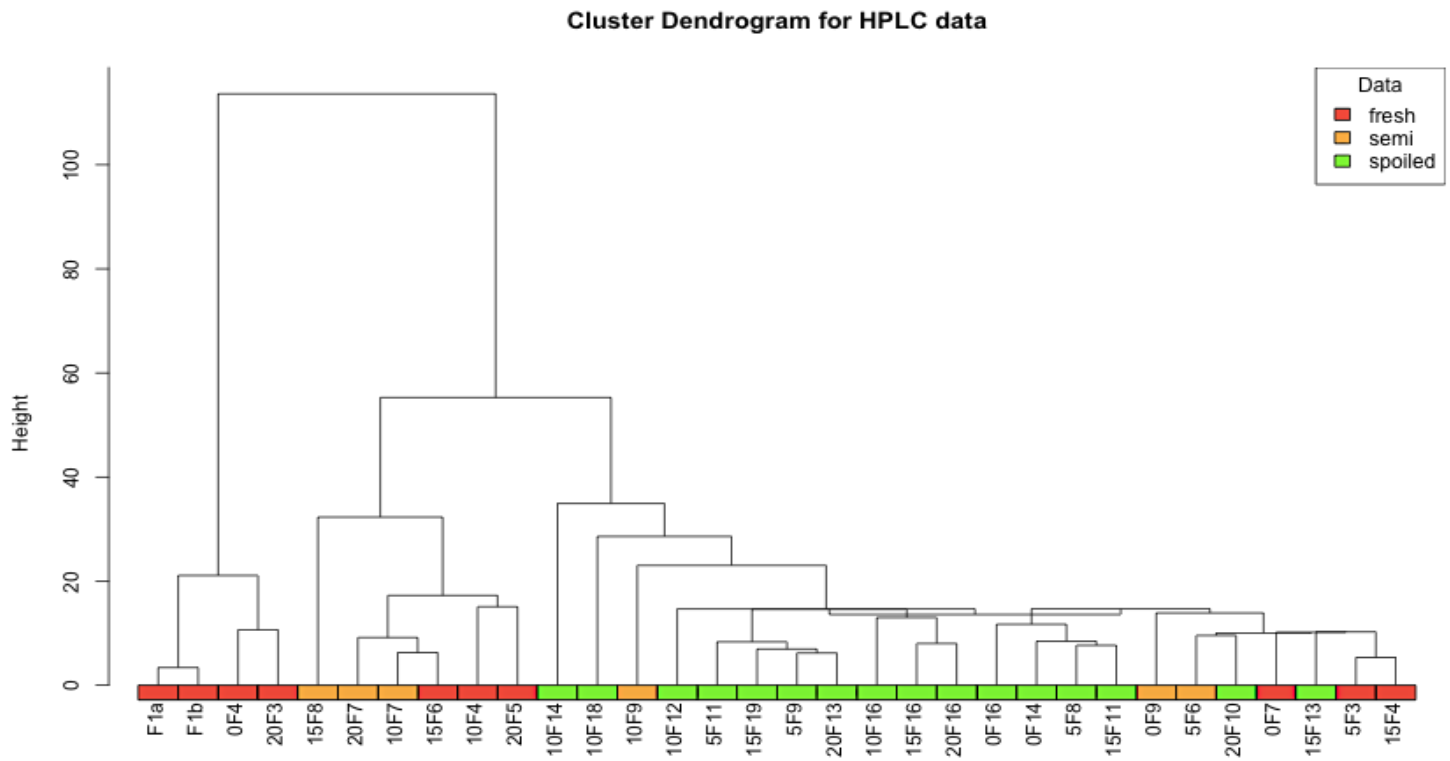
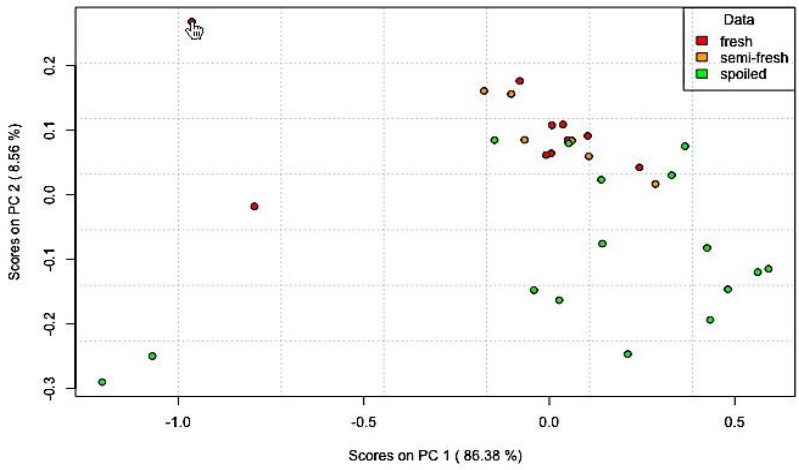


Figure 6-10 Interactive dendrogram generated by the iWebPlots package representing the outcome of HCA when applied on the HPLC data of case study 1

The figure displays the constructed dendrogram when HCA is applied on the HPLC dataset of case study 1. Once more, only the 32 common samples are used as a visual aid. In this example, "Euclidean distance" was applied as the distance algorithm and "centroid method" as the linkage algorithm. The samples have been automatically coloured by the **iWebPlots** package; different colour representations are used to differentiate the distinct classification groups (red colour is used for fresh, orange for semi-fresh and green for spoiled samples) In addition, the package provides full interactivity of the dendrogram's leaves; the users can click or rollover the coloured boxes and/or labels, and view additional information about the sample under study. This interactive web-based plot can be embedded in any web interface and/or web project.



Show 50 entries Search:

Sample number	Sample name	Classification
1	F1a	Fresh
2	F1b	Fresh
3	0F4	Fresh
4	0F7	Fresh
5	0F9	Semi-fresh
6	0F14	Spoiled
7	0F16	Spoiled
8	5F3	Fresh
9	5F6	Semi-fresh
10	5F8	Spoiled
11	5F9	Spoiled
12	5F11	Spoiled
13	10F4	Fresh
14	10F7	Semi-fresh
15	10F9	Semi-fresh
16	10F12	Spoiled
17	10F14	Spoiled
18	10F16	Spoiled
19	10F18	Spoiled
20	15F4	Fresh
21	15F6	Fresh
22	15F8	Semi-fresh
23	15F11	Spoiled
24	15F13	Spoiled
25	15F16	Spoiled
26	15F19	Spoiled
27	20F3	Fresh
28	20F5	Fresh
29	20F7	Semi-fresh
30	20F10	Spoiled
31	20F13	Spoiled
32	20F16	Spoiled
Sample number	**Samples**	**Classification**

Showing 1 to 32 of 32 entries

Figure 6-11 Partial view of the implemented web interface for the FTIR dataset of case study 1

The figure depicts a partial view of the web interface and the PCA scores of the mean-centered FTIR data for case study 1. The interface includes an interactive plot, as generated directly by **R**, with an active **HTML** image map. The users can click upon each point in the plot to view relevant information about the sample in the plot. In addition, a fully interactive and dynamic data table stores the associated metadata. (Both) the table and plot are interconnected, allowing asynchronous exchange of data and a user-friendly impression.

6.4 Conclusion

The work reported in this chapter mainly investigated three distinct fields of the **R** graphics system: high-quality static images with clean aesthetics, interactive graphs that can be embedded within a web interface, in addition to dynamically generated reports that contain reproducible documentation and graphics. All the graphs presented throughout this thesis have been generated by in-house scripts.

In addition to the implementation of the multivariate analysis pipeline, great importance has also been given to the graphical representation of the research findings in the most effective and informative way. The generated graphs proved to be extremely efficient as means of data visualisation, while simultaneously they provide persuasive and aesthetically pleasing graphs that enhance interpretability.

Furthermore, even though **R** is a powerful graphics tool, it grants limited interactivity to the users. For the graphical visualisation of the analysis' steps, the derivative bonus was the implementation of the **iWebPlots R** package for the creation of interactive web-based plots. The generated plots were successfully incorporated as part of a user-friendly web interface, built for demonstrative purposes. Finally, automatically generated graphs by the analysis were exported in dynamic reports, thus providing great reproducibility.

7 Conclusion and Recommendations

7.1 Summary

This PhD aimed at the construction of a novel suite of software tools for the accurate, rapid and inexpensive detection of bacterial spoilage in meat and the evaluation of meat freshness. This research was carried out as part of the SYMBIOSIS-EU project, funded by European Commission Framework 7.

Chapter 1 described the background of analytical techniques in combination with chemometric methods for the evaluation of meat freshness in the context of systems biology. In addition, the chapter presents a thorough background of state-of-the art classification, validation and evaluation techniques in an attempt to highlight their advantages and limitations.

Chapter 2 introduced a first working implementation of the multivariate classification pipeline, designed for the analysis of the single-instrument datasets from a single case study ("Shelf life beef fillets stored in air at 0, 5, 10, 15 and 20°C"). Two different types of classification models were employed, namely PLS-DA and SVMs. Classification ensembles were implemented in addition to standalone classifiers, while various validation methods were investigated for the optimisation of their training parameters. Out of all the generated models, the ensembles of SVMs proved to be the most powerful since they demonstrated the highest overall accuracies ($\%CC$) in the majority of cases. In addition, the ensembles of PLS-DA proved to be well-suited for the simple spectroscopic data obtained by FTIR, which are characterised by nearly linear separation between the three distinct classes. On the contrary, the single classifiers of PLS-DA in combination with LOOCV exaggerated the generalisation accuracy and led to overoptimistic results. Bootstrapping proved to be the most thorough among the different validation methods since it produced the best bias-variance trade-off with only a minimum number of overfitting instances. Despite its accurate performance, this approach led to long execution times due to the

large number of individual models involved, especially when using SVMs as these take a long time to optimise.

Chapter 3 presents a solution for speeding up the optimisation of the SVM hyperparameters *via* bootstrapping, particularly for the cases of nonlinear SVMs with the RBF kernel. To date, a naïve approach for the tuning process of the RBF SVMs has been used, whereby a grid-search is applied over a wide range of hyperparameters. Even though this simplistic technique is widely popular and extensively applied by the scientific community, it is very computationally intensive. A new heuristic approach was presented in which the Box complex algorithm for constrained nonlinear optimisation was used in the place of the time-consuming grid-search. The approximation algorithm was incorporated in the multivariate analysis pipeline where in combination with parallel programming, speeded-up the computationally demanding SVM tuning process by up to ~ 90× times. It is important to note that, with the new methodology, the previously unfeasible permutation testing was successfully applied and executed within an average of a few hours. The validity of the newly introduced algorithm was confirmed by comparing the accuracies obtained by the Box complex algorithm with the ones by the grid-search on the same set of data.

Once the analysis pipeline was built and optimised, its functionality was further extended to include the fusion of multiple heterogeneous datasets obtained by various analytical techniques. Chapter 4 describes the application of multi-block and morphometric integration methods as a means of determining whether better generalisation performance is achieved when integrated as opposed to standalone datasets are used. Furthermore, in order to confirm the reproducibility and generic nature of the implemented suite of tools, the analysis pipeline was successfully applied to three further independent real-world case studies ("Shelf life of minced beef stored in air, MAP, and in active packaging at 0, 5, 10 and 15°C", "Survey of minced beef" and "Pork stored in air and MAP"). In addition to the individual analysis results, Chapter 5 also investigates the common trends across all four investigated case studies. In all cases, the FTIR data presented significantly higher classification accuracies for the linear classifiers, and especially for PLS-DA. Even so, it was verified that the ensembles of SVMs qualify as more powerful and robust

classification models since they produced consistently higher overall accuracies ($\%CC$) than PLS-DA. Among the analytical techniques, HPLC proved to be the most diagnostic technique for the assessment of meat freshness, with classification accuracies around 80%. On the contrary, e-nose did not demonstrate any discriminative information, and its results proved to be statistically non-significant. Thus, we can conclude that the provided HPLC data contained abundances of several specific chemical compounds associated with and denoting spoilage. Conversely, the FTIR, Raman and e-nose data were the measurements of raw sensors with no prior feature selection or mapping to specific compounds. At a per-class level, the semi-fresh samples were consistently difficult to classify, whether they constituted the majority or minority class. Furthermore, in the majority of cases, CPCA produced higher overall and per-class accuracies ($\%CC$) than GPA. For case study 4, the $\%CC$ obtained by CPCA managed to exceed not only the outcome of GPA but also the overall accuracy recorded by standalone HPLC, equal to 80%. This proves that the comprehensive fusion of valuable information from complementary analytical techniques may indeed result in greater performance. This observation proves that the fusion of the most discriminative information from complementary analytical techniques may indeed result in greater classification performance.

In this research, in addition to the implementation of the analysis pipeline, great importance was also given in the development of powerful visualisation techniques as a means of enhancing the interpretability of the obtained results. Chapter 6 highlights the necessity for designing informative yet also aesthetically pleasing graphs. The graphical methods and web technologies presented in the chapter were used to construct the graphs throughout the thesis, demonstrate the outcome of the various analyses. The generated graphs, ranging from high-quality static images to reproducible reports and interactive web-based plots, proved to be more efficient as means of data representation than other traditional visualisation methods.

The generated suite of tools, as presented throughout this thesis, has covered with equal emphasis a wide range of chemometric fields, including data aggregation, integration, analysis and visualisation. The software tools show the most promise in terms of performance, flexibility and simplicity. Exceptional attention and precision has been given to the application of rigorous validation and evaluation techniques as a means of generating as accurate, robust and unbiased models as possible; as stated in Brereton (2006) and Westerhuis *et al*. (2008), even though proper validation of machine learning models has received special attention over the past decade, it is most often lacking in the recent applications. Furthermore, a novelty of this research was the application of advanced optimisation techniques, which resulted in a striking speedup of the end-to-end analyses without however compromising the integrity of the validation and evaluation process; the minimisation of the computational cost and complexity was so impressive, it reached the point where all end-to-end analyses within the pipeline were executed within a few hours on a personal computer, obviating any need of a server or supercomputer. In addition, the analysis pipeline was built in the context of reproducible research in an extremely simple, straightforward and user-friendly way. The functionality of the implemented statistical tools can be further fused into a unified form of a single **R** package. The package can be uploaded on CRAN, the official **R** repository, where it will be freely available to others users. Having confirmed the generic nature and applicability of the developed tools by testing them on new real-world case studies, the package can be equally efficient when applied by other scientists to areas of inquiry far wider and more diverse than the present study. Finally, the package provides a great degree of extendibility, since it allows its users to conduct completely personalised analyses in addition to modifying and expanding its functionality.

In summary, the project aims and objectives set out in Chapter 1 have been successfully met. In addition, the project has generated ideas for further work, which are explored in the following Section.

7.2 Recommendations for Future Work

7.2.1 Improved classification of semi-fresh samples

A major outcome of this research was the fact that semi-fresh samples were consistently difficult to classify, which limited the $\%CC$ values obtained throughout the project. The reason behind the high misclassification rates may be two-fold and could be analysed as follows.

From a chemometric point of view, the semi-fresh samples most frequently represented the minority class. However, it was demonstrated that the decision boundaries of most classifiers, but especially those of SVMs, are highly influenced by imbalanced data and hence become biased towards the majority class. As an attempt to minimise the dominating behaviour of the majority class, the classification models were re-built using different weights for each designated class during the training and testing process; however, this approach did not notably improve the classification results. Several other approaches available in the literature provide solutions to overcome this major impediment of machine learning algorithms. Most of the proposed methods are based on under-sampling the majority class and/or over-sampling the minority class with replacement (Nojima *et al*., 2012). Synthetic composition of samples that derive from the minority class has also been extensively applied among other heurists (Nojima *et al*., 2012). However, a novel but quite prominent algorithm is the Synthetic Minority Oversampling Technique (SMOTE) (Chawla *et al*., 2002). The SMOTE algorithm performs under-sampling of the majority class and over-sampling of the minority class by creating synthetic instances using the k-nearest neighbour (kNN) rather than performing over-sampling with replacement. The success of the algorithm according to Chawla *et al*. (2002) lies in the fact that the synthetic examples are created in feature space rather than data space; thus, according to Liu *et al.* (2006) the SMOTE algorithm outperforms all the other proposed heuristics. Therefore, it may prove to be extremely fruitful when applied on the highly imbalanced datasets of this work.

In addition, the difficulty to correctly predict the semi-fresh samples can only indicate that the identification of semi-fresh samples by the sensory panel was not entirely based on its chemical composition, thus they are prone to subjective assessment. This project has solely focused on the investigation of multivariate classification techniques and problems. However, in addition to analytical measurements and sensory assessment, microbiological analyses were also conducted on the meat samples according to Argyri (2010) and Papadopoulou *et al.* (2011). The functionality of the multivariate analysis pipeline can be therefore expanded to include calibration cases by using as input the provided microbiological counts. The microbiological analysis included total viable counts (TVC), *Pseudomonas spp.*, *Br. thermosphacta*, *Enterobacteriaceae*, lactic acid bacteria (LAB), yeasts and moulds, and pH.

7.2.2 Improvement of the SVM optimisation algorithm

Furthermore, the application of the Box complex algorithm has proven to be extremely fruitful in the process of tuning nonlinear SVMs with bootstrapping, leading to exceptional speedup rates without compromising the accuracy and integrity of the results. However, on very rare instances, the algorithm selected unsuitable combinations of hyperparameters as optimal. Even though the algorithm is robust since it is tolerant to noisy problems, it does suffer from one major impediment; it is highly dependent on the randomly selected initial point, upon which the first complex (constrained simplex) is constructed. Therefore, if the initial point is chosen at random within an unacceptable region, it may fail to converge and terminate its functionality. Therefore, the investigation of an approach for the identification of potentially successful starting points may prove to be extremely beneficial. As an alternative, a restart algorithm may be implemented, which randomly selects the initial points, performs Box complex, and subsequently extracts the combination of hyperparameters that presented the best performance.

7.2.3 Feature extraction

Currently, feature extraction was implemented in the analysis pipeline only as part of PCA. However, a more thorough investigation of feature extraction algorithms may help increase the generalisation accuracy and apply further dimensionality reduction of the multivariate datasets. In any feature extraction algorithm, the variables are ranked based on their contribution to the classifier, and the variables that present high ranks are considered optimal. The implementation of feature extraction may be straightforward for relatively simple classification models, however the difficulty increases for the kernel-based SVMs, where the high-dimensional feature space is quite complex. A widely applied feature extraction method that may prove to be fruitful in the case of SVMs is the SVM-RFE algorithm (Guyon *et al*. 2002; the algorithm is based on the iterative backwards sequential selection method using weights as the variable criterion (Xu *et al.*, 2006).

REFERENCES

Almeida, J. A. S., Barbosa, L. M. S., Pais, A. A. C. C. and Formosinho, S. J. (2007), "Improving hierarchical cluster analysis: A new method with outlier detection and automatic clustering", *Chemometrics and Intelligent Laboratory Systems,* vol. 87, no. 2, pp. 208-217.

Alvarez-Ordoñez, A. and Prieto, M. (2012), "Technical and Methodological Aspects of Fourier Transform Infrared Spectroscopy in Food Microbiology Research", in *Fourier Transform Infrared Spectroscopy in Food Microbiology,* Springer, pp. 1-18.

Ammor, M. S., Argyri, A. and Nychas, G. E. (2009), "Rapid monitoring of the spoilage of minced beef stored under conventionally and active packaging conditions using Fourier transform infrared spectroscopy in tandem with chemometrics", *Meat Science,* vol. 81, no. 3, pp. 507-514.

Andrade, J. M., Gómez-Carracedo, M. P., Krzanowski, W. and Kubista, M. (2004), "Procrustes rotation in analytical chemistry, a tutorial", *Chemometrics and Intelligent Laboratory Systems,* vol. 72, no. 2, pp. 123-132.

Argyri, A. A. (2010), *Quantifying meat spoilage with an array of biochemical indicators* (Doctor of Philosophy thesis), Cranfield University, Cranfield Health.

Argyri, A. A., Doulgeraki, A. I., Blana, V. A., Panagou, E. Z. and Nychas, G. -. E. (2011), "Potential of a simple HPLC-based approach for the identification of the spoilage status of minced beef stored at various temperatures and packaging systems", *International journal of food microbiology,* vol. 150, no. 1, pp. 25-33.

Argyri, A. A., Jarvis, R. M., Wedge, D., Xu, Y., Panagou, E. Z., Goodacre, R. and Nychas, G. -. E. (2013), "A comparison of Raman and FT-IR spectroscopy for the prediction of meat spoilage", *Food Control,* vol. 29, no. 2, pp. 461-470.

Argyri, A. A., Panagou, E. Z., Tarantilis, P. A., Polysiou, M. and Nychas, G. -. E. (2010), "Rapid qualitative and quantitative detection of beef fillets spoilage based on Fourier transform infrared spectroscopy data and artificial neural networks", *Sensors and Actuators B: Chemical,* vol. 145, no. 1, pp. 146-154.

Arthur, D. and Vassilvitskii, S. (2007), "K-Means++: the Advantages of Careful Seeding", pp. 1035.

Baralis, E. and Fiori, A. (2008), "Exploring heterogeneous biological data sources", pp. 647.

Barker, M. and Rayens, W. (2003), "Partial least squares for discrimination", *Journal of Chemometrics,* vol. 17, no. 3, pp. 166-173.

Bauer, E. and Kohavi, R. (1999), "An empirical comparison of voting classification algorithms: Bagging, boosting, and variants", *Machine Learning,* vol. 36, no. 1, pp. 105-139.

Belousov, A., Verzakov, S. and Von Frese, J. (2002), "A flexible classification approach with optimal generalisation performance: support vector machines", *Chemometrics and Intelligent Laboratory Systems,* vol. 64, no. 1, pp. 15-25.

Belousov, A. I., Verzakov, S. A. and von Frese, J. (2002), "Applicational aspects of support vector machines", *Journal of Chemometrics,* vol. 16, no. 8-10, pp. 482-489.

Ben-Hur, A. and Weston, J. (2010), "A User's Guide to Support Vector Machines", in Carugo, O. and Eisenhaber, F. (eds.) Humana Press, , pp. 223-239.

Bennett, K. P. and Campbell, C. (2003), "Support Vector Machines: Hype or Hallelujah?", *SIGKDD Explorations,* vol. 2, pp. 2000.

Berger, S. I., Iyengar, R. and Ma'ayan, A. (2007), "AVIS: AJAX viewer of interactive signaling networks", *Bioinformatics,* vol. 23, no. 20, pp. 2803-2805.

Berners-Lee, T., Hendler, J. and Lassila, O. (2001), "The semantic web", *Scientific American,* vol. 284, no. 5, pp. 28-37.

Berners-Lee, T., (1996), *WWW: Past, Present, and Future.*

Berners-Lee, T. and Fischetti, M. (2008), *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor,* Paw Prints.

Bessant, C., Brereton, R.,G. and Dunkerley, S. (1999), "Integrated processing of triply coupled diode array liquid chromatography electrospray mass spectrometric signals by chemometric methods", *Analyst,* vol. 124, no. 12, pp. 1733-1744.

Boardman, M. and Trappenberg, T. (2006), "A Heuristic for Free Parameter Optimization with Support Vector Machines", *Neural Networks, 2006. IJCNN '06. International Joint Conference on,* pp. 610.

Boser, B. and Vapnik, V. N. (1992), "A training algorithm for optimal margin classifiers", *Proceedings of the fifth annual workshop on Computational learning theory,* , pp. 144-152.

Boser, B. E., Guyon, I. M. and Vapnik, V. N. (1992), "A training algorithm for optimal margin classifiers", *Proceedings of the fifth annual workshop on Computational learning theory,* Pittsburgh, Pennsylvania, United States, ACM, New York, NY, USA, pp. 144.

Bottou, L., Cortes, C., Denker, J. S., Drucker, H., Guyon, I., Jackel, L. D., LeCun, Y., Muller, U. A., Sackinger, E., Simard, P. and Vapnik, V. (1994), "Comparison of classifier methods: a case study in handwritten digit recognition", *Pattern Recognition, 1994. Vol. 2 - Conference B: Computer Vision & Image*

Processing., Proceedings of the 12th IAPR International. Conference on, Vol. 2, pp. 77.

Boulesteix, A. -. (2004), "PLS dimension reduction for classification with microarray data", *Statistical applications in genetics and molecular biology,* vol. 3, no. 1.

Boulesteix, A. -., Lambert-Lacroix, S., Peyre, J. and Strimmer, K. (2011), *plsgenomics: PLS analyses for genomics*, available at: http://CRAN.R-project.org/package=plsgenomics.

Box, M. J. (1965), "A New Method of Constrained Optimization and a Comparison With Other Methods", *The Computer Journal,* vol. 8, no. 1, pp. 42-52.

Bragin, E. (2012), *Interactive genome browser based on html5 and related web technologies* (Master of Science thesis), Cranfield Univerity, Cranfield Health.

Brereton, R. G. (2006), "Consequences of sample size, variable selection, and model validation and optimisation, for predicting classification ability from analytical data", *TrAC Trends in Analytical Chemistry,* vol. 25, no. 11, pp. 1103.

Brereton, R. G. (2009), *Chemometrics for pattern recognition*, Wiley.

Brereton, R.G. and Lloyd, G.R., (2009), *Support Vector Machines for classification and regression*, The Royal Society of Chemistry.

Burges, C. J. C. (1998), "A tutorial on support vector machines for pattern recognition", *Data mining and knowledge discovery,* vol. 2, no. 2, pp. 121-167.

Butcher, E. C., Berg, E. L. and Kunkel, E. J. (2004), "Systems biology in drug discovery", *Nature biotechnology,* vol. 22, no. 10, pp. 1253-1259.

Byvatov, E. and Schneider, G. (2003), "Support vector machine applications in bioinformatics", *Appl Bioinformatics,* vol. 2, no. 2, pp. 67-77.

Calì, A., Calvanese, D., De Giacomo, G. and Lenzerini, M. (2004), "Data integration under integrity constraints", *Information Systems,* vol. 29, no. 2, pp. 147-163.

Canty, A. "An S-Plus Library for Resampling Methods", .

Canty, A. and Ripley, B. D. (2012), *boot: Bootstrap R (S-Plus) Functions*, .

Carney, S. L. (2003), "Leroy Hood expounds the principles, practice and future of systems biology", *Drug discovery today,* vol. 8, no. 10, pp. 436-438.

Cawley, G. C. and Talbot, N. L. C. (2007), "Preventing Over-Fitting during Model Selection via Bayesian Regularisation of the Hyper-Parameters", *J.Mach.Learn.Res.*, vol. 8, pp. 841-861.

CellML , *The CellML project - CellML*, available at: http://www.cellml.org.

Chancelier, J. P., Delebecque, F., Gomez, C., Goursat, M., Nikoukhah, R. and Steer, S. (1990), *Scilab*, available at: http://www.scilab.org/.

Chang, Y. W., Hsieh, C. J., Chang, K. W., Ringgaard, M. and Lin, C. J. (2010), "Training and testing low-degree polynomial data mappings via linear SVM", *The Journal of Machine Learning Research*, vol. 11, pp. 1471-1490.

Chang, C. and Lin, C. (2011), "LIBSVM: A library for support vector machines", *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27:1-27:27.

Chapelle, O. and Vapnik, V.N., (2000), *Model Selection for Support Vector Machines*.

Chapelle, O., Vapnik, V. N., Bousquet, O. and Mukherjee, S. (2002), "Choosing Multiple Parameters for Support Vector Machines", *Machine Learning*, vol. 46, no. 1, pp. 131-159.

Chatzimichali, E. A. and Bessant, C. (2011), *iWebPlots: Interactive web-based plots*, available at: http://cran.r-project.org/web/packages/iWebPlots/index.html.

Chawla, N. V., Bowyer, K. W., Hall, L. O. and Kegelmeyer, W. P. (2002), "SMOTE: Synthetic Minority Over-sampling Technique", *Journal of Artificial Intelligence Research*, vol. 16, pp. 321-357.

Chawla, N. V., Bowyer, K. W., Hall, L. O. and Kegelmeyer, W. P. (2002), "SMOTE: Synthetic Minority Over-sampling Technique", *Journal of Artificial Intelligence Research*, vol. 16, pp. 321-357.

Chawla, N. V. (2003), "C4.5 and imbalanced data sets: investigating the effect of sampling method, probabilistic estimate, and decision tree structure", *In Proceedings of the ICML'03 Workshop on Class Imbalances,* .

Chih-Wei Hsu and Chih-Jen Lin (2002), "A comparison of methods for multiclass support vector machines", *Neural Networks, IEEE Transactions on*, vol. 13, no. 2, pp. 415-425.

Chong, L. and Ray, L. B. (2002), "Whole-istic biology", *Science*, vol. 295, no. 5560, pp. 1661.

Chu, A., Ahn, H., Halwan, B., Kalmin, B., Artifon, E. L., Barkun, A., Lagoudakis, M. G. and Kumar, A. (2007), "A decision support system to facilitate management of patients with acute gastrointestinal bleeding", *Artif Intell Med,* .

Ciosek, P., Brzózka, Z., Wróblewski, W., Martinelli, E., Di Natale, C. and D'Amico, A. (2005), "Direct and two-stage data analysis procedures based on PCA, PLS-DA and ANN for ISE-based electronic tongue—Effect of supervised feature extraction", *Talanta*, vol. 67, no. 3, pp. 590-596.

Clarke, B., Fokoué, E. and Zhang, H. H. (2009), *Principles and theory for data mining and machine learning*, Springer.

Cortes, C. and Vapnik, V. N. (1995), "Support-vector networks", *Machine Learning*, vol. 20, no. 3, pp. 273-297.

Craig, A., Cloarec, O., Holmes, E., Nicholson, J. K. and Lindon, J. C. (2006), "Scaling and Normalization effects in NMR spectroscopic metabonomic data sets.", Analytical Chemistry, vol. 78, no. 7, pp. 2262-2267

Craig Venter, J., Adams, M. D., Myers, E. W., Li, P. W., Mural, R. J., Sutton, G. G., Smith, H. O., Yandell, M., Evans, C. A., Holt, R. A., Gocayne, J. D., Amanatides, P., Ballew, R. M., Huson, D. H., Wortman, J. R., Zhang, Q., Kodira, C. D., Zheng, X. H., Chen, L., Skupski, M., Subramanian, G., Thomas, P. D., Zhang, J., Gabor Miklos, G. L., Nelson, C., Broder, S., Clark, A. G., Nadeau, J., McKusick, V. A., Zinder, N., Levine, A. J., Roberts, R. J., Simon, M., Slayman, C., Hunkapiller, M., Bolanos, R., Delcher, A., Dew, I., Fasulo, D., Flanigan, M., Florea, L., Halpern, A., Hannenhalli, S., Kravitz, S., Levy, S., Mobarry, C., Reinert, K., Remington, K., Abu-Threideh, J., Beasley, E., Biddick, K., Bonazzi, V., Brandon, R., Cargill, M., Chandramouliswaran, I., Charlab, R., Chaturvedi, K., Deng, Z., Di Francesco, V., Dunn, P., Eilbeck, K., Evangelista, C., Gabrielian, A. E., Gan, W., Ge, W., Gong, F., Gu, Z., Guan, P., Heiman, T. J., Higgins, M. E., Ji, R. -., Ke, Z., Ketchum, K. A., Lai, Z., Lei, Y., Li, Z., Li, J., Liang, Y., Lin, X., Lu, F., Merkulov, G. V., Milshina, N., Moore, H. M., Naik, A. K., Narayan, V. A., Neelam, B., Nusskern, D., Rusch, D. B., Salzberg, S., Shao, W., Shue, B., Sun, J., Zhen Yuan Wang, Wang, A., Wang, X., Wang, J., Wei, M. -., Wides, R. and Xiao, C. (2001), "The sequence of the human genome", *Science*, vol. 291, no. 5507, pp. 1304-1351.

Cristianini, N. and Shawe-Taylor, J. (2000), *An introduction to support vector machines: and other kernel-based learning methods*, Cambridge University Press.

Crockford, D. (2006), "The application/json media type for javascript object notation (json)", .

Cuellar, A. A., Lloyd, C. M., Nielsen, P. F., Bullivant, D. P., Nickerson, D. P. and Hunter, P. J. (2003), "An Overview of CellML 1.1, a Biological Model Description Language", *Simulation*, vol. 79, no. 12, pp. 740-747.

Cytoscape , *Cytoscape: Analyzing and Vizualizing Network Data - Cytoscape website*, available at: http://www.cytoscape.org/.

Dahl, T. and Næs, T. (2004), "Outlier and group detection in sensory panels using hierarchical cluster analysis with the Procrustes distance", *Food Quality and Preference*, vol. 15, no. 3, pp. 195-208.

Dantzig, G. B. (1987), "simplex method for solving linear programs", in Eatwell, J., Milgate, M. and Newman, P. (eds.) *The New Palgrave: A Dictionary of Economics*, Palgrave Macmillan, Basingstoke,.

Davison, A. C. and Hinkley, D. V. (1997), "Bootstrap Methods and Their Applications", .

Dietterich, T. G. (2000), "Ensemble Methods in Machine Learning", in *Multiple Classifier Systems*, Springer Berlin Heidelberg, , pp. 1-15.

Dietterich, T. G. (2000), "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization", *Machine Learning*, vol. 40, no. 2, pp. 139-157.

Dijksterhuis, G. B. and Gower, J. C. (1992), "The interpretation of Generalized Procrustes Analysis and allied methods", *Food Quality and Preference*, vol. 3, no. 2, pp. 67-87.

Dijksterhuis, G. (1994), "Procrustes analysis in studying sensory-instrumental relations", *Food Quality and Preference*, vol. 5, no. 1-2, pp. 115-120.

Dimitriadou, E., Hornik, K., Leisch, F., Meyer, D. and Weingessel, A. (2011), *e1071: Misc Functions of the Department of Statistics (e1071), TU Wien*, available at: http://cran.r-project.org/web/packages/e1071/index.html (accessed 2011).

Dimitriadou, E., Hornik, K., Leisch, F., Meyer, D. and Weingessel, A. (2010), *e1071: Misc Functions of the Department of Statistics (e1071), TU Wien*, .

DiNucci, D. (1999), "Fragmented future", *Print*, vol. 53, no. 4, pp. 32.

Doriano "Paisano" Carta , *Web Timeline*, available at: http://thepaisano.files.wordpress.com/2008/03/webtimeline.jpg (accessed 2011).

Dryden, I. L. (2012), "Package "shapes"", , no. R Foundation for Statistical Computing, Vienna, Austria.

Duan, K., Keerthi, S. S. and Poo, A. N. (2003), "Evaluation of simple performance measures for tuning svm hyperparameters", *Neurocomputing*, vol. 51, pp. 41-59.

Duan, K. -. and Keerthi, S. S. (2005), "Which is the best multiclass SVM method? An empirical study", *Proceedings of the Sixth International Workshop on Multiple Classifier Systems*, pp. 278.

Dunkley, T. P. J., Hester, S., Shadforth, I. P., Runions, J., Weimar, T., Hanton, S. L., Griffin, J. L., Bessant, C., Brandizzi, F., Hawes, C., Watson, R. B., Dupree, P. and Lilley, K. S. (2006), "Mapping the Arabidopsis organelle proteome", *Proceedings of the National Academy of Sciences of the United States of America*, vol. 103, no. 17, pp. 6518-6523.

Efron, B. and Tibshirani, R. (1993), *An introduction to the bootstrap*, Chapman \& Hall.

Ellis, D. I., Broadhurst, D. and Goodacre, R. (2004), "Rapid and quantitative detection of the microbial spoilage of beef by Fourier transform infrared

spectroscopy and machine learning", *Analytica Chimica Acta,* vol. 514, no. 2, pp. 193-201.

Ellis, D. I., Broadhurst, D., Kell, D. B., Rowland, J. J. and Goodacre, R. (2002), "Rapid and Quantitative Detection of the Microbial Spoilage of Meat by Fourier Transform Infrared Spectroscopy and Machine Learning", *Applied and Environmental Microbiology,* vol. 68, no. 6, pp. 2822-2828.

Ellis, D. I. and Goodacre, R. (2001), "Rapid and quantitative detection of the microbial spoilage of muscle foods: current status and future trends", *Trends in Food Science & Technology,* vol. 12, no. 11, pp. 414-424.

Ercolini, D., Russo, F., Torrieri, E., Masi, P. and Villani, F. (2006), "Changes in the spoilage-related microbiota of beef during refrigerated storage under different packaging conditions", *Applied and Environmental Microbiology,* vol. 72, no. 7, pp. 4663-4671.

Finkelstein, A., Hetherington, J., Li, L., Margoninski, O., Saffrey, P., Seymour, R. and Warner, A. (2004), "Computational challenges of systems biology", *Computer,* vol. 37, no. 5, pp. 4+26-33.

Finney, A. and Hucka, M. (2003), "Systems biology markup language: Level 2 and beyond", *Biochemical Society transactions,* vol. 31, no. 6, pp. 1472-1473.

Foody, G. M. and Mathur, A. (2004), "A relative evaluation of multiclass image classification by support vector machines", *Geoscience and Remote Sensing, IEEE Transactions on,* vol. 42, no. 6, pp. 1335-1343.

Fraternali, P., Rossi, G. and Sánchez-Figueroa, F. (2010), "Rich internet applications", *Internet Computing, IEEE,* vol. 14, no. 3, pp. 9-12.

Friendly, M. (2001), "Milestones in the history of thematic cartography, statistical graphics, and data visualization", *Accessed: March,* vol. 18, pp. 2010.

Furrer, R., Nychka, D. and Sain, S. (2012), *fields: Tools for spatial data,* .

Gabriel, E., Fagg, G. E., Bosilca, G., Angskun, T., Dongarra, J. J., Squyres, J. M., Sahay, V., Kambadur, P., Barrett, B., Lumsdaine, A., Castain, R. H., Daniel, D. J., Graham, R. L. and Woodall, T. S. (2004), "Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation", *Proceedings, 11th European PVM/MPI Users' Group Meeting*, September, Budapest, Hungary, pp. 97.

Garfinkel, D. (1985), "Computer-based modeling of biological systems which are inherently complex: problems, strategies, and methods.", *Biomedica biochimica acta,* vol. 44, no. 6, pp. 823-829.

Garrett, J. J. (2005), "Ajax: A new approach to web applications", .

Ge, H., Walhout, A. J. M. and Vidal, M. (2003), "Integrating 'omic' information: A bridge between genomics and systems biology", *Trends in Genetics*, vol. 19, no. 10, pp. 551-560.

Geman, S., Bienenstock, E. and Doursat, R. (1992), "Neural networks and the bias/variance dilemma", *Neural computation*, vol. 4, no. 1, pp. 1-58.

Gentleman, R. C., Carey, V. J., Bates, D. M., Bolstad, B., Dettling, M., Dudoit, S., Ellis, B., Gautier, L., Ge, Y. and Gentry, J. (2004), "Bioconductor: open software development for computational biology and bioinformatics", *Genome biology*, vol. 5, no. 10, pp. R80.

Gentleman, R. and Biocore *geneplotter: Graphics related functions for Bioconductor*, .

Gentleman, R. C., Carey, V. J., Bates, D. M., Bolstad, B., Dettling, M., Dudoit, S., Ellis, B., Gautier, L., Ge, Y. and Gentry, J. (2004), "Bioconductor: open software development for computational biology and bioinformatics", *Genome biology*, vol. 5, no. 10, pp. R80.

Gesmann, M. and de Castillo, D. (2011), "googleVis: Interface between R and the Google Visualisation API", *The R Journal*, vol. 3, no. 2, pp. 40-44.

Giraldo, B., Garde, A., Arizmendi, C., Jane, R., Benito, S., Diaz, I. and Ballesteros, D. (2006), "Support vector machine classification applied on weaning trials patients", *Conf Proc IEEE Eng Med Biol Soc*, vol. 1, pp. 5587-5590.

Glasmachers, T. (2008), *Gradient Based Optimization of Support Vector Machines*Fakult\at f\ur Mathematik, Ruhr-Universit\at Bochum, Germany, .

Goesmann, A., Linke, B., Rupp, O., Krause, L., Bartels, D., Dondrup, M., McHardy, A. C., Wilke, A., Pühler, A. and Meyer, F. (2003), "Building a BRIDGE for the integration of heterogeneous data from functional genomics into a platform for systems biology", *Journal of Biotechnology*, vol. 106, no. 2-3, pp. 157-167.

Good, P. I. (2006), *Permutation, Parametric and Bootstrap Tests of Hypotheses*, 3rd ed, Springer-Verlag New York Inc, Dordrecht.

Goodacre, R., Broadhurst, D., Smilde, A. K., Kristal, B. S., Baker, J. D., Beger, R., Bessant, C., Connor, S., Capuani, G., Craig, A., Ebbels, T., Kell, D. B., Manetti, C., Newton, J., Paternostro, G., Somorjai, R., Sjöström, M., Trygg, J. and Wulfert, F. (2007), "Proposed minimum reporting standards for data analysis in metabolomics", *Metabolomics*, vol. 3, no. 3, pp. 231-241.

Google Chart Tools(2012), available at: https://developers.google.com/chart/.

Gosling, J. (2010), *Java programming language*, available at: http://www.java.com/en/.

Gower, J. C. (1975), "Generalized procrustes analysis", *Psychometrika*, vol. 40, no. 1, pp. 33-51.

Gower, J.C., (2010), *Procrustes methods*, John Wiley & Sons, Inc.

Gram, L., Ravn, L., Rasch, M., Bruhn, J. B., Christensen, A. B. and Givskov, M. (2002), "Food Spoilage – interactions between food spoilage bacteria", *International journal of food microbiology*, vol. 78, no. 1, pp. 79-97.

Gratzer, G. (1995), *Math into LATEX: an introduction to LATEX and AMS-LATEX*, Birkhauser Boston.

Groeger, J. S., Lemeshow, S., Price, K., Nierman, D. M., White, P., Klar, J., Granovsky, S., Horak, D. and Kish, S. K. (1998), "Multicenter outcome study of cancer patients admitted to the intensive care unit: a probability of mortality model", *J Clin Oncol*, vol. 16, no. 2, pp. 761-770.

Guo, Q., Wu, W., Massart, D. L., Boucon, C. and de Jong, S. (2002), "Feature selection in principal component analysis of analytical data", *Chemometrics and Intelligent Laboratory Systems*, vol. 61, no. 1-2, pp. 123-132.

Gutierrez-Osuna, R. , *Lecture 13: Validation*, available at: http://research.cs.tamu.edu/prism/lectures/iss/iss_l13.pdf.

Guyon, I., Weston, J., Barnhill, S. and Vapnik, V. (2002), "Gene selection for cancer classification using support vector machines", *Machine Learning, 46(1-3);389-422,2002*, vol. 46, no. 1-3, pp. 389-422.

Hanafi, M., Kohler, A. and Qannari, E. (2011), "Connections between multiple co-inertia analysis and consensus principal component analysis", *Chemometrics and Intelligent Laboratory Systems*, vol. 106, no. 1, pp. 37-40.

Hand, D., Mannila, P. and Smyth, P. (2001), "Principles of Data Mining", .

Hanson, B. A. (2012), *ChemoSpec: Exploratory Chemometrics for Spectroscopy*, available at: http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html.

Harrington, J. (2011), *emu: Interface to the Emu Speech Database System*, .

Harrington, P. B. (2006), "Statistical validation of classification and calibration models using bootstrapped Latin partitions", *TrAC Trends in Analytical Chemistry; Use and abuse of chemometrics*, vol. 25, no. 11, pp. 1112-1124.

Hassani, S., Martens, H., Qannari, E. M., Hanafi, M., Borge, G. I. and Kohler, A. (2010), "Analysis of -omics data: Graphical interpretation- and validation tools in multi-block methods", *Chemometrics and Intelligent Laboratory Systems*, vol. 104, no. 1, pp. 140-153.

Hastie, T., Tibshirani, R. and Friedman, J. (2009), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*, Springer.

Hegde, P. S., White, I. R. and Debouck, C. (2003), "Interplay of transcriptomics and proteomics", *Current opinion in biotechnology,* vol. 14, no. 6, pp. 647-651.

Hestenes, M. R. and Stiefel, E. (1952), "Methods of Conjugate Gradients for Solving Linear Systems", *Journal of Research of the National Bureau of Standards,* vol. 49, no. 6, pp. 409-436.

Hesterberg, T., Moore, D. S., Monaghan, S., Clipson, A. and Epstein, R. (2005), "Bootstrap methods and permutation tests", *Introduction to the Practice of Statistics,* vol. 5, pp. 1-70.

Hongyan, G. (2009), "A simple multi-sensor data fusion algorithm based on principal component analysis", *Computing, Communication, Control, and Management, 2009. CCCM 2009. ISECS International Colloquium on,* Vol. 2, pp. 423.

Hood, L. (2003), "Systems biology: Integrating technology, biology, and computation", *Mechanisms of ageing and development,* vol. 124, no. 1, pp. 9-16.

Hosmer, D. and Lemeshow, S. (2000), "'Applied Logistic Regression'", *Wiley Series in Probability and Statistics,* vol. second.

Hsu, C. W., Chang, C. C. and Lin, C. J. (2003), *A practical guide to support vector classification,* Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan.

Hubert, L. and Schultz, J. (1976), "Quadratic assignment as a general data analysis strategy", *British Journal of Mathematical and Statistical Psychology,* vol. 29, no. 2, pp. 190-241.

Ideker, T., Galitski, T. and Hood, L., (2001), *A new approach to decoding life: Systems biology*.

Iliinsky, N. and Steele, J. (2011), *Designing Data Visualizations: Representing Informational Relationships,* O'Reilly Media.

Institute for Systems Biology , *Systems Biology: the 21st Century Science*, available at: http://www.systemsbiology.org/Intro_to_ISB_and_Systems_Biology/Systems_Biology_--_the_21st_Century_Science.

Izenman, A. J. (2008), *Modern multivariate statistical techniques: regression, classification, and manifold learning,* Springer.

Jackson, J. E. (1991), *A user's guide to principal components,* Wiley-Interscience.

Jaggi, M. (2008), *Geometry of Support Vector Machines*, available at: http://www.m8j.net/data/List/Files-142/Geometry%20of%20Support%20Vector%20Machines.pdf (accessed 2011).

Jardon, M. (2006), *Systems Biology: an Overview*, available at: http://www.scq.ubc.ca/systems-biology-an-overview/.

Joyce, A. R. and Palsson, B. Ø. (2006), "The model organism as a system: Integrating 'omics' data sets", *Nature Reviews Molecular Cell Biology*, vol. 7, no. 3, pp. 198-210.

jQuery Development Team (2010), *jQuery: The Write Less, Do More, JavaScript Library*, available at: http://jquery.com/.

Karatzoglou, A., Meyer, D. and Hornik, K. (2006), "Support Vector Machines in R", *Journal of Statistical Software*, vol. 15, no. 9, pp. 1-28.

Kavraki, L. (2007), *Dimensionality Reduction Methods for Molecular Motion*, available at: http://cnx.org/content/m11461/1.10/.

Keerthi, S. S. and Lin, C. J. (2003), "Asymptotic behaviors of support vector machines with Gaussian kernel", *Neural Comput.*, vol. 15, no. 7, pp. 1667-1689.

Kitano, H. (2002), "Computational systems biology", *Nature*, vol. 420, no. 6912, pp. 206-210.

Kohavi, R. (1995), "A study of cross-validation and bootstrap for accuracy estimation and model selection", *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 2*, Montreal, Quebec, Canada, Morgan Kaufmann Publishers Inc, San Francisco, CA, USA, pp. 1137.

Kolda, T. G., Lewis, R. M. and Torczon, V. (2003), "Optimization by direct search: New perspectives on some classical and modern methods", *SIAM Review*, vol. 45, pp. 385-482.

Kopka, H. and Daly, P. W. (2004), *Guide to LATEX*, Addison-Wesley Professional.

Kotsiantis, S., Kanellopoulos, D. and Pintelas, P., *Handling imbalanced datasets: A review*.

Kressel, U. H. -. (1999), "Advances in kernel methods", in Schölkopf, B. a. B., Christopher, J. C. and Smola, A. J. (eds.) MIT Press, Cambridge, MA, USA, pp. 255-268.

Lagarias, J. C., Reeds, J. A., Wright, M. H. and Wright, P. E. (1998), "Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions", *SIAM Journal of Optimization*, vol. 9, pp. 112-147.

Lawton, G. (2008), "New Ways to Build Rich Internet Applications", *Computer*, vol. 41, no. 8, pp. 10-12.

Leisch, F. (2002), "Sweave. Dynamic generation of statistical reports using literate data analysis.", .

Leisch, F. (2005), "Sweave user manual", .

Ligges, U. and Mächler, M. (2003), "Scatterplot3d - an R Package for Visualizing Multivariate Data", *Journal of Statistical Software,* vol. 8, no. 11, pp. 1-20.

Liu, Y., An, A. and Huang, X. (2006), "Boosting Prediction Accuracy on Imbalanced Datasets with SVM Ensembles", in Ng, W., Kitsuregawa, M., Li, J., et al (eds.) Springer Berlin Heidelberg, , pp. 107-118.

Ma, J., Krishnamurthy, A. and Ahalt, S. (2004), "SVM training with duplicated samples and its application in SVM-based ensemble methods", *Neurocomputing,* vol. 61, no. 0, pp. 455-459.

Mead, J. A., Shadforth, I. P. and Bessant, C. (2007), "Public proteomic MS repositories and pipelines: Available tools and biological applications", *Proteomics,* vol. 7, no. 16, pp. 2769-2786.

Mesarovic, M. D., Sreenath, S. N. and Keene, J. D. (2004), "Search for organising principles: understanding in systems biology.", *Systems biology,* vol. 1, no. 1, pp. 19-27.

Meyer, D., Leisch, F. and Hornik, K. (2003), "The support vector machine under test", *Neurocomputing,* vol. 55, no. 1-2, pp. 169.

Meyer, V. R. (2013), “Practical high-performance liquid chromatography”, Willey.com

Molina, F., Dehmer, M., Perco, P., Graber, A., Girolami, M., Spasovski, G., Schanstra, J. P. and Vlahou, A. (2010), "Systems biology: opening new avenues in clinical research", *Nephrology Dialysis Transplantation,* vol. 25, no. 4, pp. 1015-1018.

Muin, M. and Fontelo, P. (2006), "Technical development of PubMed interact: An improved interface for MEDLINE/PubMed searches", *BMC Medical Informatics and Decision Making,* vol. 6.

Muin, M. and Fontelo, P. (2006), "Technical development of PubMed interact: An improved interface for MEDLINE/PubMed searches", *BMC Medical Informatics and Decision Making,* vol. 6.

Murrell, P. (2005), *R graphics,* Chapman & Hall/CRC.

Murugesan, S. (2007), "Understanding Web 2.0", *IT Professional,* vol. 9, no. 4, pp. 34-41.

Nelder, J. A. and Mead, R. (1965), "A Simplex Method for Function Minimization", *Comput J,* vol. 7, no. 4, pp. 308-313.

Nelder, S. and Singer, J. (2009), "Nelder-Mead algorithm", *Scholarpedia,* vol. 4, no. 2, pp. 2928.

Nicolaou, N., Xu, Y. and Goodacre, R. (2011), "Fourier Transform Infrared and Raman Spectroscopies for the Rapid Detection, Enumeration, and Growth Interaction of the Bacteria Staphylococcus aureus and Lactococcus lactis ssp. cremoris in Milk", *Analytical Chemistry*, vol. 83, no. 14, pp. 5681-5687.

Noble, W. S. (2006), "What is a support vector machine?", *Nat Biotechnol*, vol. 24, no. 12, pp. 1565-1567.

Nojima, Y., Mihara, S. and Ishibuchi, H. (2012), "Application of parallel distributed genetics-based machine learning to imbalanced data sets", *Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on*, pp. 1.

O'Reilly, T. (2005), "Web 2.0: compact definition", *Message posted to http://radar.oreilly.com/archives/2005/10/web_20_compact_definition.html*.

O'Reilly, T. (2009), *What is web 2.0*, O'Reilly Media.

Panagou, E. Z., Mohareb, F. R., Argyri, A. A., Bessant, C. M. and Nychas, G. E. "A comparison of artificial neural networks and partial least squares modelling for the rapid detection of the microbial spoilage of beef fillets based on Fourier transform infrared spectral fingerprints", *Food Microbiology*, vol. In Press, Corrected Proof.

Papadopoulou, O., Panagou, E. Z., Tassou, C. C. and Nychas, G. -. E. (2011), "Contribution of Fourier transform infrared (FTIR) spectroscopy data on the quantitative determination of minced pork meat spoilage", *Food Research International*, vol. 44, no. 10, pp. 3264-3271.

Paulson, L. D. (2005), "Building rich web applications with Ajax", *Computer*, vol. 38, no. 10, pp. 14-17.

Qin, S. J., Valle, S. and Piovoso, M. J. (2001), "On unifying multiblock analysis with application to decentralized process monitoring", *Journal of Chemometrics*, vol. 15, no. 9, pp. 715-742.

R Development Core Team (2012), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.

Ramadan, Z., Jacobs, D., Grigorov, M. and Kochhar, S. (2006), "Metabolic profiling using principal component analysis, discriminant partial least squares, and genetic algorithms", *Talanta*, vol. 68, no. 5, pp. 1683-1691.

Ramadan, Z., Zhang, P., Jacobs, D., Tavazzi, I. and Kochhar, S., (2007), *An NMR- and MS-based metabonomic investigation of saliva metabolic changes in feline odontoclastic resorptive lesions (FORL)-diseased cats*, Springer Boston.

Raychaudhuri, S., Stuart, J. M. and Altman, R. B. (2000), "Principal components analysis to summarize microarray experiments: application to sporulation time series.", *Pacific Symposium on Biocomputing.Pacific Symposium on Biocomputing*, , pp. 455-466.

Reisinger, F., Corpas, M., Hancock, J., Hermjakob, H., Birney, E. and Kahlem, P., (2008), *ENFIN - An integrative structure for systems biology*.

Richet, Y. (2010), *JMathTools*, available at: http://jmathtools.berlios.de/doku.php.

Rossini, K., Verdun, S., Cariou, V., Qannari, E. M. and Fogliatto, F. S. (2012), "PLS discriminant analysis applied to conventional sensory profiling data", *Food Quality and Preference; Sensometrics 2010*, vol. 23, no. 1, pp. 18-24.

Roussel, S., Bellon-Maurel, V., Roger, J. M. and Grenier, P. (2003), "Fusion of aroma, FT-IR and UV sensor data based on the Bayesian inference. Application to the discrimination of white grape varieties", *Chemometrics and Intelligent Laboratory Systems*, vol. 65, no. 2, pp. 209-219.

Sagotsky, J. A., Zhang, L., Wang, Z., Martin, S. and Deisboeck, T. S. (2008), "Life Sciences and the web: A new era for collaboration", *Molecular Systems Biology*, vol. 4.

Sarkar, D. (2008), *Lattice: Multivariate Data Visualization with R*, Springer, New York.

Sattlecker, M. (2011), *Optimization of Machine Learning Methods for Cancer Diagnostics using Vibrational Spectroscopy* (Doctor of Philosophy thesis), Cranfield, Cranfield Health.

Schmid, U., Rösch, P., Krause, M., Harz, M., Popp, J. and Baumann, K. (2009), "Gaussian mixture discriminant analysis for the single-cell differentiation of bacteria using micro-Raman spectroscopy", *Chemometrics and Intelligent Laboratory Systems*, vol. 96, no. 2, pp. 159-171.

Schölkopf, B. and Smola, A. J. (2001), *Learning with kernels: Support vector machines, regularization, optimization, and beyond*, MIT press.

Scholkopf, B., Burges, C. and Vapnik, V. N. (1996), "Incorporating invariances in support vector learning machines", *Proceedings of the 1996 International Conference on Artificial Networks, Bochum, Germany Lecture notes in computer science*, vol. 1112, pp. 47-52.

Search Tool for the Retrieval of Interacting Genes/Proteins , *Search Tool for the Retrieval of Interacting Genes/Proteins (STRING 7) Website*, available at: http://string.embl.de/.

Sebastien Bihorel, M. B. (2012), *Package 'neldermead'*, available at: http://cran.r-project.org/web/packages/neldermead/neldermead.pdf (accessed 10/2012).

Simply the best graphics package for R(2012), available at: http://www.inside-r.org/packages/ggplot2/reviews/simply-best-graphics-package-r.

Shah, A. A., Barthel, D., Lukasiak, P., Blazewicz, J. and Krasnogor, N. (2008), "Web and grid technologies in bioinformatics, computational and systems biology: A review", *Current Bioinformatics*, vol. 3, no. 1, pp. 10-31.

Shlens, J. (2005), *A Tutorial on Principal Component Analysis*, .

Smilde, A. K., van, d. W., Bijlsma, S., van der Werff-van, d. V. and Jellema, R. H. (2005), "Fusion of Mass Spectrometry-Based Metabolomics Data", *Analytical Chemistry*, vol. 77, no. 20, pp. 6729-6736.

Smilde, A. K., Westerhuis, J. A. and de Jong, S. (2003), "A framework for sequential multiblock component methods", *Journal of Chemometrics*, vol. 17, no. 6, pp. 323-337.

Smola, A. J. (1998), *Learning with kernels*, Citeseer.

Smola, A. J. and Schölkopf, B. (2004), "A tutorial on support vector regression", *Statistics and computing*, vol. 14, no. 3, pp. 199-222.

Smolinska, A., Blanchet, L., Coulier, L., Ampt, K. A. M., Luider, T., Hintzen, R. Q., Wijmenga, S. S. and Buydens, L. M. C. (2012), "Interpretation and Visualization of Non-Linear Data Fusion in Kernel Space: Study on Metabolomic Characterization of Progression of Multiple Sclerosis", *PLoS ONE*, vol. 7, no. 6, pp. e38163.

Society of sensory professionals , *Generalized Procrustes Analysis*, available at: http://www.sensorysociety.org/ssp/wiki/Generalized_Procrustes_Analysis/ (accessed 2011).

Spendley, W., Hext, G. R. and Himsworth, F. R. (1962), "Sequential Application of Simplex Designs in Optimisation and Evolutionary Operation", *Technometrics*, vol. 4, no. 4, pp. 441-461.

Staelin, C. (2003), "Parameter selection for support vector machines", *Hewlett-Packard Company, Tech.Rep.HPL-2002-354R1*, .

Steele, J. and Iliinsky, N. (2010), *Beautiful Visualization*, O'Reilly Media.

Stegmann, M. B. and Gomez, D. D. (2002), *A brief introduction to statistical shape analysis*, , University of Denmark, DTU.

Steinmetz, V., Sevila, F. and Bellon-Maurel, V. (1999), "A Methodology for Sensor Fusion Design: Application to Fruit Quality Assessment", *Journal of Agricultural Engineering Research*, vol. 74, no. 1, pp. 21-31.

Sun Microsystems , *JavaFX - Rich Internet Applications Development - RIAs JavaFX*, available at: http://javafx.com/.

Suykens, J. A. K., Gestel, T. V., Brabanter, J. D., Moor, B. D. and Vandewalle, J. (2002), *Least Squares Support Vector Machines*, World Scientific, Singapore.

SYMBIOSIS-EU , *SYMBIOSIS-EU website*, available at: http://www.symbiosis-eu.net/.

Synnergren, J., Olsson, B. and Gamalielsson, J. (2009), "Classification of information fusion methods in systems biology", *In Silico Biology*, vol. 9, no. 3, pp. 65-76.

Thalib, L., Kitching, R. L. and Bhatti, M. I. (1999), "Principal component analysis for grouped data - A case study", *Environmetrics*, vol. 10, no. 5, pp. 565-574.

Tierney, L. (2003), *Simple Parallel Statistical Computing in R*, available at: http://homepage.stat.uiowa.edu/~luke/talks/uiowa03.pdf.

Tom Christiansen, Nathan Torkington, and other authors as noted. (2010), *The Perl Programming Language*, available at: http://www.perl.org/.

Tufail, M. and Ormsbee, L. (2009), "Optimal Water Quality Management Strategies for Urban Watersheds Using Macrolevel Simulation and Optimization Models", *Journal of Water Resources Planning and Management*, vol. 135, no. 4, pp. 276-285.

Tzung-Pei Hong, Yeong-Chyi Lee and Min-Thai Wu (2005), "Using the master-slave parallel architecture for genetic-fuzzy data mining", *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, Vol. 4, pp. 3232.

Urbanek, S. and Theus, M. (2003), "iPlots: high interaction graphics for R", *Proceedings of the 3rd International Workshop on Distributed Statistical Computing*, Citeseer, .

Valentini, G. and Dietterich, T. G. (2004), "Bias-variance analysis of support vector machines for the development of svm-based ensemble methods", *Journal of Machine Learning Research*, vol. 5, pp. 725-775.

van den Berg, R. A., Hoefsloot, H. C. J., Westerhuis, J. A., Smilde, A. K. and van der Werf, M. J. (2006), "Centering, scaling, and transformations: Improving the biological information content of metabolomics data", *BMC Genomics*, vol. 7.

Van Looy, S., Meeus, J., Wyns, B., Cruyssen, B., De Keyser, F. and Boullart, L. (2005), "'Feature selection in the prediction of Infliximab dose increase'", *Proceedings of the 9th IASTED International Conference Artificial Intelligence and Soft Computing*, .

Van Looy, S., Verplancke, T., Benoit, D., Hoste, E., Van Maele, G., De Turck, F. and Decruyenaere, J. (2007), "A novel approach for prediction of tacrolimus blood concentration in liver transplantation patients in the intensive care unit through support vector regression", *Crit Care*, vol. 11, no. 4, pp. R83.

Vapnik, V. N. (2000), *The nature of statistical learning theory*, Springer.

Varmuza, K. and Filzmoser, P. (2009), *Introduction to Multivariate Statistical Analysis in Chemometrics*, CRC Press.

Vera, G., Jansen, R. and Suppi, R. (2008), "R/parallel - speeding up bioinformatics analysis with R", *BMC Bioinformatics*, vol. 9, no. 1, pp. 390.

Verplancke, T., Van Looy, S., Benoit, D., Vansteelandt, S., Depuydt, P., De Turck, F. and Decruyenaere, J. (2008), "Support vector machine versus logistic regression modeling for prediction of hospital mortality in critically ill patients with haematological malignancies", *BMC Medical Informatics and Decision Making*, vol. 8, no. 1, pp. 56.

Vinayagam, A., Konig, R., Moormann, J., Schubert, F., Eils, R., Glatting, K. H. and Suhai, S. (2004), "Applying Support Vector Machines for Gene Ontology based gene function prediction", *BMC Bioinformatics*, vol. 5, pp. 116.

von Mering, C., Jensen, L. J., Kuhn, M., Chaffron, S., Doerks, T., Krüger, B., Snel, B. and Bork, P. (2007), "STRING 7 - Recent developments in the integration and prediction of protein interactions", *Nucleic acids research*, vol. 35, no. SUPPL. 1, pp. D358-D362.

Wand, M. and R port and updates by Ripley B. (2011), *KernSmooth: Functions for kernel smoothing for Wand & Jones (1995)*, available at: http://cran.r-project.org/web/packages/KernSmooth/index.html (accessed 2011).

Wand, M. P. and Jones, M. C. (1995), *Kernel smoothing*, Chapman \& Hall.

Wang, H. and Yang, J. (2008), "Research and application of web development based on ASP.NET 2.0+Ajax", *Industrial Electronics and Applications, 2008. ICIEA 2008. 3rd IEEE Conference on*, pp. 857.

Wang, X. and Paliwal, K. K. (2003), "Feature extraction and dimensionality reduction algorithms and their applications in vowel recognition", *Pattern Recognition*, vol. 36, no. 10, pp. 2429-2439.

WATSON, J. D. and CRICK, F. H. C. (1953), "Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid", *Nature*, vol. 171, no. 4356, pp. 737-738.

Weber, C. M., Cauchi, M., Patel, M., Bessant, C., Turner, C., Britton, L. E. and Willis, C. M. (2011), "Evaluation of a gas sensor array and pattern recognition for the identification of bladder cancer from urine headspace", *Analyst*, .

Westerhoff, H. V. and Palsson, B. O. (2004), "The evolution of molecular biology into systems biology", *Nature biotechnology*, vol. 22, no. 10, pp. 1249-1252.

Westerhuis, J.A., Hoefsloot, H.C.J., Smit, S., Vis, D., Smilde, A.K., van Velzen, E., van Duijnhoven, J. and van Dorsten, F., (2008), *Assessment of PLSDA cross validation*, Springer Boston.

Westerhuis, J. A., Kourti, T. and MacGregor, J. F. (1998), "Analysis of multiblock and hierarchical PCA and PLS models", *Journal of Chemometrics*, vol. 12, no. 5, pp. 301-321.

Wickham, H. (2009), *ggplot2: elegant graphics for data analysis*, Springer New York.

Wickham, H. and Chang, W. (2012), "ggplot2: An implementation of the Grammar of Graphics", .

Wilkinson, L., Anand, A. and Grossman, R. (2005), "Graph-theoretic scagnostics", *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*, pp. 157.

Wilkinson, L., Wills, D., Rope, D., Norton, A. and Dubbs, R. (2005), *The Grammar of Graphics*, Springer.

Wise, B. M., Gallagher, N. B., Bro, R. and Shaver, J. M. (2003), *PLS_Toolbox 3.0 for use with MATLAB*, available at: https://noppa.tkk.fi/noppa/kurssi/ke-90.5100/materiaali/pca___pls.pdf.

Wold, H. (1975), "Soft Modeling by Latent Variables; the Nonlinear Iterative Partial Least Squares Approach", *Perspectives in Probability and Statistics, Papers in Honour of M.S. Bartlett*, , pp. 520-540.

Wold, S., Esbensen, K. and Geladi, P. (1987), "Principal component analysis", *Chemometrics and Intelligent Laboratory Systems; Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists*, vol. 2, no. 1, pp. 37-52.

Wold, S., Hellberg, S., Lundstedt, T., Sjöström, M. and Wold, H. (1987), "PLS modeling with latent variables in two or more dimensions", *Proceedings PLS-meeting*, Frankfurt am Main, pp. 1.

Wold, S. and Sjöström, M. (1998), "Chemometrics, present and future success", *Chemometrics and Intelligent Laboratory Systems*, vol. 44, no. 1–2, pp. 3-14.

Wold, S., Sjöström, M. and Eriksson, L. (2001), "PLS-regression: a basic tool of chemometrics", *Chemometrics and Intelligent Laboratory Systems*, vol. 58, no. 2, pp. 109-130.

Wright, M. H. (2012), *Nelder, Mead, and the other Simplex Method*, available at: http://www.math.uiuc.edu/documenta/vol-ismp/42_wright-margaret.pdf.

Wu, W., Guo, Q., Jong, S. d. and Massart, D. L. (2002), "Randomisation test for the number of dimensions of the group average space in generalised Procrustes analysis", *Food Quality and Preference*, vol. 13, no. 3, pp. 191.

Xu, Y., Zomer, S. and Brereton, R. G. (2006), "Support vector machines: a recent method for classification in chemometrics", *Critical Reviews in Analytical Chemistry*, vol. 36, no. 3-4, pp. 177-188.

Xu, Y. and Brereton, R. G. (2005), "A comparative study of cluster validation indices applied to genotyping data", *Chemometrics and Intelligent Laboratory Systems*, vol. 78, no. 1–2, pp. 30-40.

Xu, Y. and Goodacre, R. (2012), "Multiblock principal component analysis: an efficient tool for analyzing metabolomics data which contain two influential factors", *Metabolomics,* vol. 8, no. 1, pp. 37-51.

Yu, H. (2011), *Rmpi: Interface (Wrapper) to MPI (Message-Passing Interface)*, available at: http://cran.r-project.org/web/packages/Rmpi/index.html (accessed 2011).

Zeger, S. L., Irizarry, R. and Peng, R. D. (2006), "On time series analysis of public health and biomedical data", *Annu Rev Public Health,* vol. 27, pp. 57-79.

Zhijie Lin, Jiyi Wu, Qifei Zhang and Hong Zhou (2008), "Research on Web Applications Using Ajax New Technologies", *MultiMedia and Information Technology, 2008. MMIT '08. International Conference on,* pp. 139.

Zimmerman, J. E., Kramer, A. A., McNair, D. S. and Malila, F. M. (2006), "Acute Physiology and Chronic Health Evaluation (APACHE) IV: hospital mortality assessment for today's critically ill patients", *Crit Care Med,* vol. 34, no. 5, pp. 1297-1310.