UPPSALA
UNIVERSITET

UPTEC X 17 028
Examensarbete 30 hp
September 2017

Improvments and evaluation of data processing in LC-MS metabolomics

for application in in vitro systems pharmacology

Alice Anlind

UPPSALA
UNIVERSITET

Teknisk- naturvetenskaplig fakultet
UTH-enheten

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

Improvments and evaluation of data processing in LC-MS metabolomics

Alice Anlind

The resistance of established medicines is rapidly increasing while the rate of discovery of new drugs and treatments have not increases during the last decades (Spiro et al. 2008). Systems pharmacology can be used to find new combinations or concentrations of established drugs to find new treatments faster (Borisy et al. 2003). A recent study aimed to use high resolution Liquid chromatography–mass spectrometry (LC-MS) for in vitro systems pharmacology, but encountered problems with unwanted variability and batch effects(Herman et al. 2017). This thesis builds on this work by improving the pipeline and comparing alternative methods and evaluating used methods. The evaluation of methods indicated that the data quality was often not improved substantially by complex methods and pipelines. Instead simpler methods such as binning for feature extraction performed best. In-fact many of the preprocessing method commonly used proved to have negative or neglect-able effects on resulting data quality. Finally the recently introduced Optimal Orthonormal System for Discriminant Analysis (OOS-DA) for batch removal was found to be a good alternative to the more complex Combat method.

Handledare: Mats Gustafsson & Efthymia Chantzi
Ämnesgranskare: Claes Andersson
Examinator: Jan Andersson
ISSN: 1401-2138, UPTEC X 17 028

Avancerade algoritmer för kartläggning av cellens beståndsdelar, stjälper mer än hjälper

Korrekt kartläggning av cellers beståndsdelar kan göra individanpassad medicin en verklighet. men är databehandlingen korrekt? Denna rapport visar hur många algoritmer som används för databehandling förstör eller är verkningslösa. Enkla metoder visar sig också kunna prestera lika bra eller bättre än komplicerade metoder.

När du tar din medicin får du en dos som verkar fungera bra för de flesta. I verkligheten har vi alla olika förutsättningar och det är därför inte ovanligt med både biverkningar och verkningslösa mediciner. Det bästa vore om varje medicinering var anpassad till just dina celler och förutsättningar. För att kunna göra det måste vi kunna läsa av dina celler för att förstå vad som skulle fungerar för just dig. Nya biotekniska framsteg gör att vi nu har maskiner som kan läsa av alla cellens beståndsdelar inom en dag till ett överkomligt pris. Att tolka förändringar i dessa beståndsdelar blir komplext och man utvecklar därför algoritmer för att med hjälp av datorer kunna tolka och förenkla insamlad data. Maskinerna är också känsliga för variation i behandlingen av proverna som stör ut den signal som innehåller biomedicinskt relevant information. Även här utvecklas algoritmer för att försöka tvätta bort oönskade störningar i insamlad data.

Jag har studerat sådana algoritmer för att jämföra olika varianter och se hur bra de är på att göra sitt jobb. Databehandlingen sker ofta via komplicerade program som kräver mycket datakraft. Jag testade bland annat en mycket enklare metod som inte krävde så mycket datakraft. Den enklare metoden presterade lika bra eller bättre än de alternativa mer komplexa metoderna på mina insamlade data. Om den enklare metoden presterar bra även på andra datamängder av samma typ skulle stora besparingar kunna göras med avseende på både tid och kostnader.

Vid tvättning av insamlade data är det vanligt att forskarna utvecklar sina egna metoder. Det är sällan som dessa metoder testas ordentligt. Jag testade ett par metoder från en tidigare publicerad vetenskaplig artikel och implementerade även egna metoder för rengöring. De flesta metoderna har väldigt liten eller negativ effekt på den data som behandlats i detta projekt. Att se begränsningar i metoder är svårt och därför borde forskarna lägga mer tid på att bevisa och motivera att deras metoder gör vad de beskriver.

Table of Contents

Abbreviations

EM	Expectation-maximum
FTMS	Fourier transform mass spectrometry
GB	Gigabyte
I	Intensity
LC	Liquid chromatography
LC-MS	Liquid chromatography mass spectrometry
LOESS	Locally weighted scatter plot smoothing / local regression
LV	Latent variable
m/z	mass/charge
MS	Mass spectrometry
OOS-DA	Optimal Orthonormal System for Discriminant Analysis
PCA	Principal component analysis
PCX	Principal component X
PLS-DA	Partial least squares Discriminant Analysis
QC	Quality Control
rt	retention time
TIC	Total Ion Count
ToF	Time of Flight
UPLC	Ultra high pressure liquid chromatography

1 Background

During the last decades the rate at which new drugs are released have not increased, despite major discoveries and cost reductions in life science. With our current speed it has been estimated that it will take 300 years to double the number of drugs available (Spiro *et al.* 2008). Finding treatment to diseases and improving already existing treatments remains slow and challenging. Several currently effective treatments such as antibiotics are loosing effectiveness as resistance is increasing. If the speed of developing new treatments does not increase, treatable diseases might become untreatable.

Since the development of new drugs has not been an slow method of finding new treatments, alternative method should be tried. One idea is to find more effective ways to use currently available drugs. By studying cells subject to different concentrations and combinations of drugs the field of *in vitro* systems pharmacology aims to find connections and similarities between drugs and learn more about combination effects. By knowing which drugs work in similar ways or are connected current treatments can be optimized and old drugs can be used in new ways (Borisy *et al.* 2003). One particularly interesting opportunity is to customize the dose and drugs for each individual treatment in what is known as personalized medicine (Kaddurah-Daouk & Weinshilboum 2014). The current system of generalized treatments could result in too low doses to some patients while giving too high doses to others. Personalized treatments could therefore enable maximum drug effect while minimizing the side effects. Different variants of personalized medicine are already being used in health care in for example pharmacogenomics (Hess *et al.* 2015). However there are still challenges to overcome such as not being cost effective enough and shortcomings of genomic techniques such as not being able to account for environmental differences.

Technological advancements in the last decade has resulted in new biotechnological and data analysis tools that has enabled genome wide scale maps of gene expression, protein abundance and recently metabolites. *In vitro* systems pharmacology is utilizing these methods to create maps of the effect of drugs on cellular pathways and biology.

One early and successful project was the Connectivity map (CMap) database (Lamb *et al.* 2006). The CMap database contains results of standardized experiments designed to study the changes in mRNA gene expression induced by a large library of different drug compounds. It has successfully been used to re-purpose drugs (Iorio *et al.* 2010) and identifying previously unknown drug mechanisms (Iorio *et al.* 2010, Gullbo *et al.* 2011, Hassan *et al.* 2011, Fryknäs *et al.* 2013). However large scale mRNA experiments are expensive, and the relevance to biology can be hard to deduce from mRNA levels (ThermoFisher Scientific). A cheaper alternative with higher relevance to biology is metabolomics which has become popular in *in vitro* systems pharmacology in recent years (Wishart 2016). Metabolites are easier to interpret than mRNA and protein levels since they are closer to physiology (Kell & Goodacre 2014). However the information from different parts of the cell are complementary and a combination of all techniques are needed to get a complete picture of the cell.

For metabolomics there are mainly two techniques used, NMR and MS (Zhang *et al.* 2012). NMR is known to be more reproducible while MS has a higher sensitivity and resolution. In LC-MS the number of detectable metabolites is significantly higher and quantities about one million times smaller can be detected (Alonso *et al.* 2015, Nassar *et al.* 2017).

The first steps towards a CMap like database for metabolomics were conducted by Aftab *et al.* using NMR (Aftab *et al.* 2014). The NMR study was able to confirm clustering of drug families as expected indicating that metabolomics is a viable tool for *in vitro* systems pharmacology. A more recent similar study was conducted using LC-MS but failed to reproduce similar types of drug family clustering (Herman *et al.* 2017), partly due to problems with experimental variability and batch effects.

For large-scale studies, the experimental work is often divided into batches. Batch effects refers to variability in the data that is only dependent on which batch the samples were prepared and measured in. There are several methods to remove batch effects in high-dimensional datasets. Initial methods were built around building latent variable (LV) models of the data and identifying the latent variables with batch effect (Alter *et al.* 2000, Benito *et al.* 2004). While there have been documented cases of LV models working in practice, critique was raised regarding the high number of samples needed in each batch and regarding removing all variation in the latent variables (Nielsen *et al.* 2002, Johnson *et al.* 2007). As a response the Combat algorithm was developed based on empirical Bayes statistics which was designed to avoid these limitations (Leek *et al.* 2010). For metabolomic data both Combat and LV based methods have been applied (Fukushima *et al.* 2014, Andgan 2016). However Leek *et al.* argue that an underestimated method that outperforms both latent variable methods and Combat is to redo the experimental work (Leek *et al.* 2010).

In this thesis the established LV methods, PCA and PLS-DA, the new latent variable method OOS-DA proposed by Herman et al. and Combat are compared. Since batch effect removal has mostly been performed for microarray mRNA gene expression data it is possible that there are characteristics of the metabolomics LC-MS data that would lead to other conclusions. Furthermore OOS-DA was previously only compared to PCA, in the study by Herman *et al.*, which is an unsupervised method. By adding comparison to another supervised LV method, PLS-DA, and to the supervised empirical Bayes method, Combat, further insights might be gained.

2 Aim

The aim of this thesis was to replicate and improve a computational pipeline for data processing of LC-MS metabolomics data. This pipeline should be adapted and optimized for use in *in vitro* systems pharmacology studies.

The goal was to include three main parts in the pipeline; feature extraction, preprocessing and batch effect removal. Each part should include comparisons between different techniques and methods. Some method specific user defined parameters should be optimized and the final results should be evaluated. One additional goal was to apply the pipeline to other datasets.

3 Theory

3.1 LC-MS

An LC-MS system aims to detect and quantify molecules by separating them based on mass charge ratio (m/z) and the chemical separation specified by the experimental set-up of the LC system. An LC-MS system consists of three connected parts, the liquid chromatography system, the ionization device and the mass spectrometer, see Figure 1.

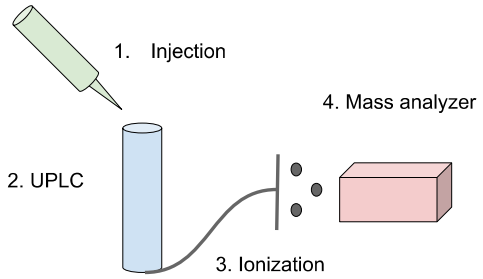


Figure 1: Schematic of a LC-MS system

Assume you would like to detect and quantify metabolites in a set of samples. Samples are first prepared by extracting metabolites before injection into the LC-MS system. Once the sample is injected the LC system separates the molecules by solubility in the mobile phase. Highly soluble metabolites will elute from the column first. The time of elution is refereed to as the retention time. The elution is passed on to an ionizer which charges each molecule and transforms it into gas phase before injection into the MS. The MS detects the m/z value and the intensity of each ionized molecule. A response (detected m/z) from one type of ionized molecule is called a signal. All the signals registered for one injection into the MS are summarized in a m/z spectrum, see Figure 2. One m/z spectrum is created about every half second. Such a time interval is often referred to as a scan as the MS scans the contents of a small time frame of elutions from the LC. The full output from the MS is a list of several m/z spectras which can be summarized in the form of an intensity map, see Figure 2.

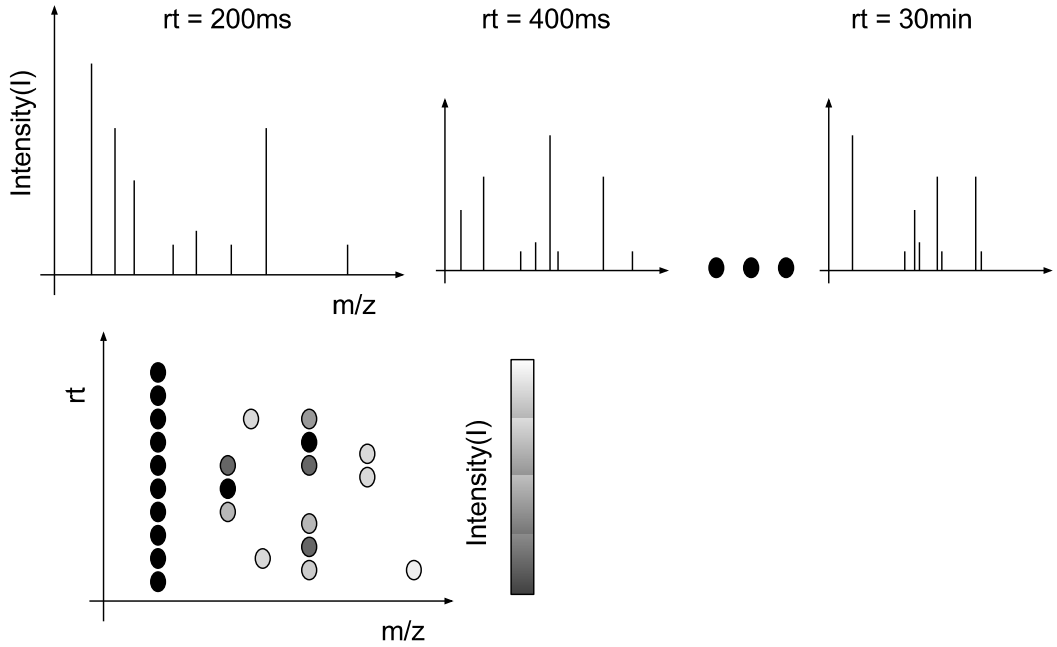


Figure 2: Each scan has an retention time (rt) and a m/z spectrum. All scans can be visualized together by combining them into an map, where each intensity is visualized by a grayscale value.

The LC system is often a Ultra High Pressure Liquid Chromatography (UPLC) that is run either as reverse-phase chromatography(RP) or hydrophilic interaction liquid chromatography (HILIC) (Zhou & Yin 2016). The RP set-up enables good separation for semi-polar and lipid molecules while HILIC is best suited for polar metabolites. The mobile phase used is often a analytical gradient of several solvents. Most experiments use their own optimized gradient protocols while the solvents chosen are often the same (Ranninger *et al.* 2016, Narduzzi 2017, Herman *et al.* 2017). For RP systems common solvents are, water, acetonitrile and methanol. Mobile phase modifiers, such as formic acid, are often added to the mobile phase to improve stability and sensitivity (Zhou & Yin 2016). Due to the varying set-up of the mobile phase, the retention time (rt) across experiments is not comparable.

There are many different types of ionization principles to choice from; Atmospheric Pressure Ionisation (APCI), Atmospheric Pressure Photoionisation (APPI), Electrospray (ESI) (Barwick *et al.* 2006). The ESI method is perhaps the most common ionization principle (Ranninger *et al.* 2016, Narduzzi 2017, Herman *et al.* 2017). ESI is a soft ionizer which means that the fragmentation of the molecules are limited. The ionized product of a molecule, M, is often the molecule with a added cation (Barwick *et al.* 2006). The most common products of ESI is: $[M+H]^+$, $[M+Na]^+$ or $[M+nH]^{n+}$. Ionization is done via spraying the molecules in solvent through a needle tip at high voltage (typically 1-10keV) and temperature (300-500C) (Barwick *et al.* 2006, Narduzzi 2017, Herman *et al.* 2017). This produces highly charged droplets which evaporate into the ionized molecule in gas phase, which is the form accepted by the mass spectrometers.

The basis in MS is to use the the conversion of potential energy to kinetic energy of an charged particle in an electric field. All potential energy is converted into kinetic energy as the particle is accelerated through the electric field. This results in the m/z ratio being deduced as the following equation:

$$E_p = E_k \Leftrightarrow zU = \frac{1}{2} m v^2 \Leftrightarrow \frac{m}{z} = \frac{2U}{v^2}$$

where E_p is the potential energy, E_k the kinetic energy, z the charge of the particle, U the electric potential difference, m the mass of the particle and v the velocity. The electric potential is given by the MS and the velocity is measured in the MS. Different MS types use slightly different methods of deducing the velocity of the molecules.

Two common mass spectrometer types are Time of Flight MS (ToF-MS) and OrbiTrap systems. ToF-MS accelerates the molecules with a electric field through a potential and the measured time to reach the detector from passing the potential will give the m/z value of the molecule (Wolff & Stephens 1953). The number of molecules are given by the intensity of the signal hitting the detector. OrbiTrap systems will catch ions into trajectories around an inner spindle-like electrode. The trajectory of the ions can then be transformed into m/z values by means of Fourier transform of the frequency (Hu *et al.* 2005). It has been shown that the OrbiTrap systems have a higher resolution than ToF (Perry *et al.* 2008). OrbiTrap systems are more expensive than ToF and has less database entries of fragmentation patterns to be used for metabolite identification (Perry *et al.* 2008). Furthermore there are several additional settings, options and parameters to tune. One such is if the MS is run in positive or negative mode. Positive mode means that only positively charged molecules are measured, while negative means that molecules with negative charge are measured. Another common term is targeted MS. Targeted MS have spiked in isotope standards that are used to better quantify molecules. In this thesis all datasets are run in positive mode and are untargeted.

3.2 LC-MS data processing

The raw data from LC-MS consists of thousands of m/z spectras for each sample which are hard to interpret. By data processing the data can be transformed to easier interpret the biology of the data. The two main steps used in LC-MS data processing are feature extraction and preprocessing.

3.2.1 Feature extraction

The feature extraction procedure aims to simplify the data by grouping several signals into features. There are many different algorithms and ideas on how to extract features. The following methods will be used and presented bellow, Binning, XCMS and OpenMS. The binning algorithm was developed as part of this thesis project, XCMS is a commonly used package and OpenMS is the method applied by Herman *et al.* (Herman *et al.* 2017). A graphic representation of the algorithms can be seen in Figure 3.

Binning

The binning algorithm first makes all spectra have the same number of features by having a common vector of bins (intensity intervals) with the same m/z values for all retention time points

and samples. Then the resulting intensity map for one sample is reduced to a single m/z spectrum by adding together the individual spectra (rows of the resulting intensity map). This means that there is no need for alignment of the samples due to differences in the retention times.

XCMS

XCMS, perhaps the most popular program for LC-MS data processing, tries to collect signals into peaks by trying to matching a Gaussian function to the collected data. There are two methods available matchedFilter and centWave. The matchedFilter (Smith *et al.* 2006) algorithm uses small bins of m/z to form chromatograms of the rt-dimension. In the rt-dimension a Gaussian filter identifies peak shapes which are integrated to get the total peak signal. In centWave (Tautenhahn *et al.* 2008) a mass trail is found by finding ranges in rt-dimension where a signal at a certain m/z value is continuously present. An Gaussian filter is then used on this region to identify peak location and area of integration. For alignment each peak is linked up with all other peaks that are within a given resolution in both m/z and rt direction. This is followed by a user defined number of cycles of retention time correction step, where small differences in rt between samples are corrected, followed by an updated alignment. For our data only one cycle was done as the rt shifts between samples were as small as a few seconds.

OpenMS

In the paper by Herman *et al.* a step further is taken by trying to add peaks together with their pipeline using the software framework OpenMS (Sturm *et al.* 2008). Depending on ionization one metabolite might result in several peaks. By grouping peaks together into candidate metabolites the hope is that the data will be easier to interpret. To find which peaks to group together an support vector machines (SVMs) was pre-trained on a dataset of known peaks of chemical compounds. The result of these trained SVMs is used on the data to predict which peaks form a candidate metabolite. In the pipeline of Herman *et. al.* the alignment across samples is done via clustering approaches to solve the complexity of having several regions in each peak.

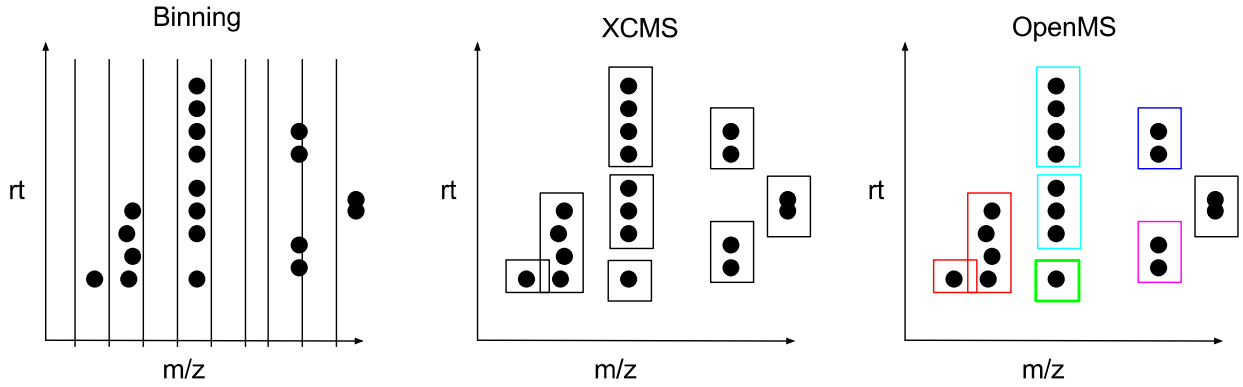


Figure 3: Schematic of different feature extraction methods, from the most simplistic (Binning) to the most complex (OpenMS). Binning divides signals (black dots) into uniformly distributed bins in the m/z dimension. XCMS fits gaussian curves to the collected MS-data to find peak areas. OpenMS merges several peak areas into potential metabolites.

This results in features that are comparable across samples. The data for one sample can therefore be simplified into an intensity vector with the values of the intensity for each feature. The full dataset can be collected into a intensity matrix, I, with samples as columns and features as rows.

3.2.2 Preprocessing

The data can be subject to computer based analysis at this stage but usually a preprocessing step is performed before. Since LC-MS is a high resolution technique able to detect metabolites in the range of nano grams, both technical and biological experimental variability (noise) is often highly abundant. The preprocessing is often tailored by each user to solve variability issues in the specific dataset. Blank samples only containing solvents are often used to identify and subtract background. Normalization of intensity distributions across samples is often applied to correct for experimental variance between samples. A common method is to simply scale the intensity matrix to make the medians, of the intensities for all features, between samples equal. However the steps are often poorly evaluated and presented in many published reports and the effects of this poor practice will be apparent later on in this thesis.

3.3 Latent variable model batch effect removal

The idea of LV model batch effect removal is to find latent variables that contain the batch effects in the dataset and remove them.

Consider a dataset represented by a matrix $\mathbf{X}$ containing n samples as columns $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}]$ and k variables as rows. Assume that the dataset $\mathbf{X}$ consist of batch effects, $\mathbf{X^b}$, and true signals, $\mathbf{X^t}$:

$$\boldsymbol{X} = \boldsymbol{X}^t + \boldsymbol{X}^b \quad (1)$$

Let us assume that each sample $\mathbf{x_i}$ can be expressed as a linear combination of some latent variables $t_i(i)$:

$$\boldsymbol{x_i} = \boldsymbol{p_1} t_1(i) + \boldsymbol{p_2} t_2(i) + \ldots + \boldsymbol{p_n} t_n(i) \quad (2)$$

where $\mathbf{p_j}$ is a loading vector being a column vector with k elements. The values of the vector t_j of the latent variables are called the score values of xi. The full dataset can now be expressed as:

$$\boldsymbol{X} = \boldsymbol{P}\,\boldsymbol{T}^T \quad (3)$$

where $\mathbf{P}$ is a matrix containing all loading vectors as columns, and $\mathbf{T}$ is a matrix containing all score vectors as columns.

Denote the loading vectors containing the batch effects as $\mathbf{P^b}$, and their respective scores as $\mathbf{T^b}$. The true signal can now be found as:

$$\boldsymbol{X}^t = \boldsymbol{X} - \boldsymbol{X}^b = \boldsymbol{X} - \boldsymbol{P}^b\left(\boldsymbol{T}^b\right)^T \quad (4)$$

Three different methods of finding the latent variables containing the batch effect were tried: PCA, PLS-DA and OOS-DA. The theory of which will be presented in the following subsections.

3.3.1 PCA

Principal component analysis (PCA) is a method to linearly compress a dataset while retaining as much variance as possible.

Consider a mean centred dataset, **X**, with n samples as columns, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ and k variables as rows. The covariance between any two variables in **X** can be found in the covariance matrix, **C.** Assuming the variables have zero mean, the covariance matrix **C** can be empirically approximated as:

$$\hat{C} = \frac{1}{N-1} X X^T \quad (5)$$

The eigenvalue λ_i associated with eigenvector $\mathbf{p}_i$ of **C** correspond to the variance of all samples in **X** along the direction defined by $\mathbf{p}_i$. The score values of one sample $\mathbf{x}_j$ are the coordinates of $\mathbf{x}_j$ when expressed in the coordinate system defined by the eigenvectors (loading vectors) $\mathbf{p}_i$.

Thus the dataset **X** can be reduced to fewer dimensions by projecting **X** onto the loading matrix, $\mathbf{P}=[\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_l]$, where l is the number of latent variables. The resulting score matrix, $\mathbf{T}=[\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_l]$, can be calculated as:

$$T = X^T P \quad (6)$$

3.3.2 PLS-DA

In this thesis project the 'plsregress' function in Matlab was used to perform PLS regression modelling. This function implements the SIMPLS method , the interested reader is encouraged to visit de Jongs article for more in-depth explanations and details (de Jong 1993).

Consider a mean centred dataset, $\mathbf{X}_0=[\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n]$, with n samples as columns and k variables. Each sample i has a mean centred response variables stored in $\mathbf{Y}_0=[\mathbf{y}_1,\mathbf{y}_2,\ldots,\mathbf{y}_m]$ as columns. The idea of PLS is to perform linear regression in latent variable representations of $\mathbf{X}_0$ and $\mathbf{Y}_0$ to find a linear regression model (coefficients) that can accurately predict $\mathbf{Y}_0$ from $\mathbf{X}_0$. For PLS-DA each class is considered a response variable that can either be 1, if the sample belongs to a class of interest, or 0 if it does not belong to that class.

3.3.3 OOS-DA

Optimal Orthonormal System for Discriminant Analysis (OOS-DA) aims to find latent variables of a dataset that best separates the data based on pre-defined class labels (Okada & Tomita 1985).

Consider a dataset, **X**, containing n samples as column vectors, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$, where each sample, $\mathbf{x}_i$, contains k variables. Each sample also has a class label c, where c is a numeric index, c=1, 2, ... , C, where C is the number of classes. To introduce the idea behind OOS-DA let the centroid of a subset S among samples collected in **X** be defined as:

$$x^s = \frac{1}{N_s} \sum_{i \in S} x_i \quad (7)$$

where N_s is the size of the subset. Now define the between-class scatter matrix B as:

$$\boldsymbol{B}=\frac{\sum_{c=1}^{C} N_c(\boldsymbol{x}^c-\boldsymbol{x}^g)(\boldsymbol{x}^c-\boldsymbol{x}^g)}{N} \quad (8)$$

where N_c, is the number of samples with class label c, N is the total number of samples, $\boldsymbol{x}^c$ is the centroid of class c and $\boldsymbol{x}^g$ is the global centroid. The global centroid is the centroid of the entire dataset **X**.

Define the within-class scatter matrix for each class c, $\mathbf{W_c}$, as:

$$\boldsymbol{W}_c=\frac{\sum_{i=1}^{N_c} (\boldsymbol{x}_i-\boldsymbol{x}^c)(\boldsymbol{x}_i-\boldsymbol{x}^c)}{N_c} \quad (9)$$

The different within-class scatter matrices, $\mathbf{W_c}$, are as a last step joined to form the general within variance as:

$$\boldsymbol{W}_o=\frac{\sum_{c=1}^{C} \boldsymbol{W}_c N_c}{N} \quad (10)$$

A separation matrix, **S**, which reflects how the between-class scatter matrix is related to the within-class scatter matrix can now be defined as:

$$\boldsymbol{S}=\boldsymbol{W}_0^{-1}\boldsymbol{B} \quad (11)$$

The goal of OOS-DA is to sequentially for each dimension find the eigenvectors of the scatter matrix **S** with the largest corresponding eigenvalue. The set of eigenvectors extracted this way will be used as orthogonal loading vectors $\mathbf{p}_i$ of **X**. The score score vectors $\mathbf{t}_n$ can then be calculated from the linear transformation $\mathbf{t}_n=\mathbf{P}^T\mathbf{x}_n$, where $\mathbf{P}=[\mathbf{p}_1, \mathbf{p}_2, \ldots \mathbf{p}_k]$.

3.4 Combat

Combat is a batch effect removal technique developed to counteract the limitations of LV based approaches. Johnson *et al.* stated that LV approaches often require more than 25 samples per batch making it unsuitable when batches are small (Johnson *et al.* 2007). Further more they argue that if the latent variables (the score values) are uncorrelated (which is true for both PCA, PLS-DA and OOS-DA) the first latent variables will be very important for the performance. The batch effect removal itself is also blind, in that it removes all variation in the direction associated with the latent variables, not only the batch variation. This means that if there is a large batch variation and a smaller true variation in the same direction, the true variation will be lost in LV based approaches. The idea of Combat is to use a linear model to model the observed data in terms of multiplicative

and additive effects and then performing both subtraction and scaling (division) aimed at removing these effects. The fitting of the model parameters is based on so called empirical Bayes methods.

3.5 Dimensionality optimization

One key issue related to the batch effect removal step is the choice of the number of dimensions to remove from the data in the attempt to remove the batch effects. To automate this process dimensionality optimization was performed where the results obtained by means of different number of dimensions removed were compared.

To evaluate the performance of a particular parameter setting a separation score was calculated as a measurement of the separation of some pre-specified classes. The classes specified should be known to separate well biologically speaking and should not be the same as the classes for the biological question since that would lead to overfitting.

3.6 Separation score

The separation score is an measure of the separation between classes and is defined as:

$$s = \frac{b}{\hat{w}} \quad (12)$$

where b is the between class variability score and $\hat{w}$ denotes the mean of the within class variability score of each class c, equation 13 and 15.

The separation between classes, b, is quantified as the mean distance of class medians around the global median, see Figure 4 and equation 15. If the samples inside a class has high variability then the between class variability could still be large even when there is only a small separation. To account for this the between class variability is divided by the within class variability, see Figure 4 and equation 13 and 14.

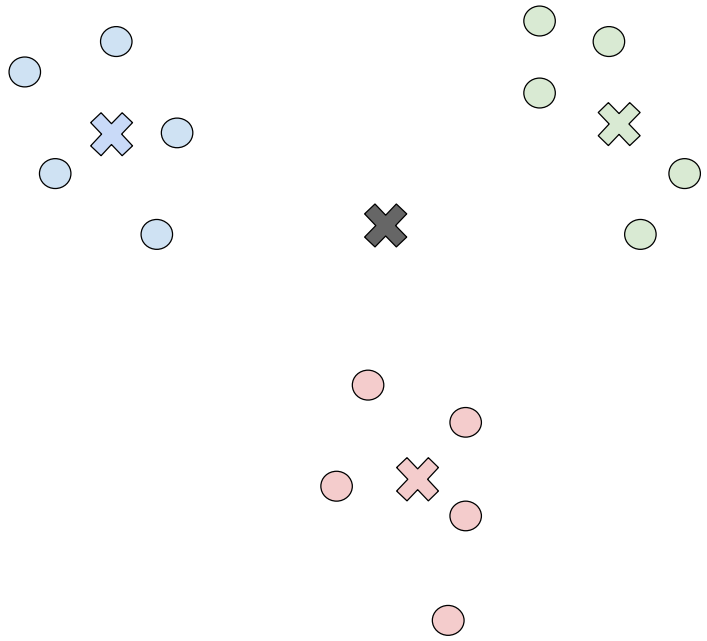


Figure 4: Schematic for within class variability (w) and between class variability (b). The within class variability for blue class is the mean distance from the members (circles) to the mean (cross). The between class variability is the mean distance between the global mean (grey cross) and the class means (red, blue and green cross).

$$\hat{w} = \frac{1}{C}\sum_{c=1}^{C} w_c \quad (13)$$

here C is the number of classes and w_c is the within-variance score within class c, calculated as:

$$w_c = \frac{1}{N_s}\sum_{s=1}^{N_s} \frac{|\boldsymbol{x}_s - \widetilde{\boldsymbol{x}}_c|}{|\widetilde{\boldsymbol{x}}_c|} \quad (14)$$

where N_s is the number of samples in class, c, $\widetilde{\boldsymbol{x}}_c$ is the class median and $\boldsymbol{x}_s$ is sample s belonging to class, c.

$$b = \frac{1}{N_c}\sum_{c=1}^{N_c} \frac{|\widetilde{x}_c - \widetilde{x}|}{|\widetilde{x}|} \quad (15)$$

Here $\widetilde{x}$ is the global median and N_c is the number of classes.

The drug dataset seemed to be sensitive to outliers within the classes, which were biological replicates of untreated cells (Controls) and cells treated with Mebendazole (Mebendazole). Small changes in outliers of these classes had a larger impact on the score than small changes in all of the

samples. To avoid this problem outliers were excluded from each class before separation score calculations. For each class the samples were evaluated as the Euclidean distance to the sample median vector. The 20% of replicates that had the largest distance to the median was removed before separation score calculations.

4 Methods

A computational pipeline for processing LC-MS data was built inspired by the pipeline of Herman *et al.* (Herman *et al.* 2017). The pipeline was initially tested on the dataset produced by Herman *et al.*. An additional dataset was found in the European Bioinformatics Institute (EMBL-EBI) affiliated database of metabolomic experiments, MetaboLights (Haug *et al.* 2013). The dataset MTBLS209 was selected since it had verified status, available blank and quality control (QC) samples and run order information (Narduzzi 2017).

4.1 Experimental data and set-up

4.1.1 Drug dataset

One goal of the dataset produced by Herman *et. al.* was to study if human cell cultures treated with drugs of the same drug class would cluster together based on induced metabolomic changes. The experiment was done by incubating non-cancerous mammary gland derived cells *in vitro* with a drug, belonging to one of 4 different drug classes or a pesticide from a group of pesticides, for 24 hours, see Table 1. It should be noted that support for this drug classes could only be found for the tyrosine kinase inhibitors (Thomas *et al.* 2012). The drugs were then harvested and prepared to be run in an LC-MS system. The incubation was done in 4 batches due to experimental limitations in the number of samples. The assignment of drug samples into the 4 batches was done by random, see Table 1. Each drug treatment was done in a biological triplicate. Untreated cells (controls) and cells treated with the drug Mebendazole was done in biological triplicate for each batch for quality control. After each batch the cells were freeze dried. When all batches were complete the freeze dried samples were rehydrated and analysed by a Thermofisher Qexactive OrbiTrap MS coupled to a UPLC system. For further quality control 25 blank samples and 10 pool samples were added randomly in the run order. The blank samples only contained the rehydration solution. Pool samples was created by allocating a small part of all samples into one pool. Separate pools only containing the samples from one batch was also prepared. For more details see the study by Herman *et. al.* (Herman *et al.* 2017). In the following sections this dataset will be referred to as the "drug dataset".

Table 1: Drug classes used as columns with drugs in columns. Colours represents which batch the drug was prepared in. Mebendazole was prepared in triplicate in all four batches. This table is copied with permission from Herman *et. al.* (Herman *et al.* 2017).

Estrogen pathway perturbators	Anti-inflammatory	Tyrosine kinase inhib.	HDAC inhib.	Pesticides
17-ß-estradiol	Dexamethasone	Erlotinib	Vinblastine	Octylmethoxycinnamate
Diethylstilbestrol	Hydrocortisone	Gefitinib	Vincristine	Diuron
Bisphenol A	Prednisolone	Lapatinib	Albendazole	Glyphosate
Genistein	Aspirin		Mebendazole	Atrazine
Quercetin	Ibuprofen		Trichostatin A	Dimephenthioate
Tamoxifen	Chloroquine			Thiram
	NVP-BEZ235			

4.1.2 Grape dataset

The original aim behind the grape dataset was to study spatial differences across different grape species. The grapes were collected at a set maturity (18 brix) and immediately stored at -80 °C. The different tissue types, skin (b), pulp (p) and seeds (s) were extracted manually and ground under -80°C. The preparation before the LC-MS analysis was done according to the optimized protocol by Theodoridis *et al.* with slight modifications (Theodoridis *et al.* 2012, Narduzzi 2017). The samples were analysed with both targeted and untargeted approaches, in both positive and negative mode of the MS, but in this thesis only the positive untargeted data will be used. Blank samples, samples containing a standardized mix of metabolites and pool samples were identified in files and meta data but no clear definition of the preparation of these samples were given. Similarly run order was extracted from the sample names produced by the LC-MS system. In the following sections this dataset will be refereed to as the "grape dataset".

4.2 Computational pipeline

The computational pipeline was implemented in Matlab 2015b (The MathWorks Inc., Natick, MA, 2000) for fast development speed using provided and custom functions to clean the data, see Figure 5 for an overview. Before preprocessing the raw data need to be transformed into features and be aligned between samples. This is refereed to as the feature extraction step in this thesis. The data by Herman had already been processed by a complex pipeline implemented in the framework OpenMS (Herman *et al.* 2017). Due to the complexity of this pipeline we compared two alternative feature extraction methods, XCMS and a custom binning approach (Smith *et al.* 2006, Tautenhahn *et al.* 2008). After the feature extraction the resulting dataset was cleaned by means of several functions developed for the preprocessing steps. These preprocessing steps were based on known technical variations and individual observations of the given datasets. After preprocessing, batch effects could be observed in the drug dataset. The tissue differences were used to simulate batch effects in the grape dataset. Four different methods of batch effect removal using PCA, PLS-DA, OOS-DA and Combat were investigated. To choose the number of dimensions removed for batch effect removal, dimensionality optimization using an objective function across a grid of possible number of dimensions removed in two consecutive steps of the pipeline was employed.

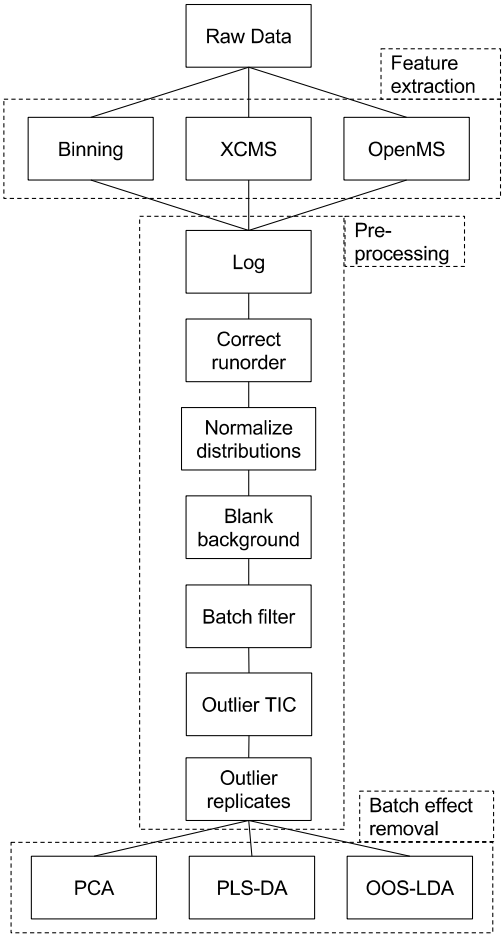


Figure 5: Schematic over the computional pipeline developed. The pipeline was organized into the three blocks, feature extraction, preprocessing and batch effect removal.

4.3 Feature Extraction

Feature extraction was performed using three different methods, Binning, XCMS and OpenMS. The feature extraction in OpenMS was conducted as in the article by Herman *et al.*. All other methods were implemented and run as a part of this thesis project. The runs using XCMS were performed according to the manual, and the implementation of the binning approach was written in Matlab see following sections for summary or the code for details (Anlind 2017).

4.3.1 Binning

For the binning algorithm the number of bins to use had to be specified. In order to ensure the resolution of the Orbitrap MS to be at the level of 0.05 Da, the number of bins was specified to 20000. To reduce noise all signals with an intensity below 100 was ignored.

First a common feature vector of m/z values at the bins middle point was specified. The m/z values are the m/z range (0-1000) divided into 20,000 equally large pieces with size equal to the resolution (0.05). The m/z range can be set by the user to depending on the settings used by the MS instruments. The range may also be decreased in the lower m/z regions to reduce noise from small highly abundant molecules. Then the algorithm will go through each signal and match the m/z of the signal with the m/z of the bin that is within a distance of 0.025. The intensity will then be added to the already existing values in the bins intensity values. If the signal is outside the range it will be ignored.

Let:
- mz(x) denote the m/z value of bin x or signal x,
- I(x) denote the intensity vector at bin x,
- int(s) denote the intensity value of signal s,

Procedure:

1. For each signal, s
    1.1. **If** int(s) <100
        1.1.1. Do nothing
    1.2. **Else**
        1.2.1. Find bin, b where |mz(bin) - mz(s)| < 0.025
        1.2.2. I(b) <- I(b) + int(s)

4.4 Preprocessing pipeline

The preprocessing pipeline in R used by Herman et al. was used for inspiration. The new pipeline developed for this thesis project was written in Matlab and is designed to be more robust to outliers. The full pipeline code is available at Github (Anlind 2017). The ordering of the different functions in the pipeline is important, but depending on the actual aims, different orders might be more or less suitable. The logarithm function was set first in the pipeline since it will alter the relative importance of features on the separation score used. The outlier removing functions were set at the end of the pipeline to ensure that no new outliers appeared when correcting the data.

4.4.1 Log

The range of intensity values obtained from an MS experiment are known to have high variation in metabolomic datasets, typically ranging from 10^4 to 10^8 . Therefore changes in the highly abundant features will be so large that changes in less abundant features will have limit impact on the separation score, even if there is a high fold change. This was corrected by replacing the intensities with the natural logarithm of the intensities. The logarithm was chosen as it corrects for heteroscedasticity, results in pseudo scaling and make multiplicative models additive (van den Berg *et al.* 2006). However it should be noted that van den Berg showed that auto-scaling or range

scaling might be more suitable choices. These methods were not investigated due to a lack of time, but are encouraged for future work.

However the logarithm is not defined for zero meaning that features that have an intensity of zero will become undefined after logarithmic transformation. A zero intensity in this context means that there is no signal present. Since the logarithm of one equals zero, all intensities with the value zero were set to one before logarithmic transformation.

Procedure:

2. Set all zero values of intensity matrix, I (corresponding to non-detected features) equal to 1

3. Replace the intensity matrix, I, with the logarithm of I, log(I).

4.4.2 Correction of intensity based on run order

Due to temporal drifts in LC-MS properties, it is known that for MS runs with a large number of samples the intensity of the signal might be affected by the run order of the samples. The effect of run order is seen as a trend either increasing or decreasing based on run order. The systematic change due to run order can vary across features. To correct for this the trend as a function of run order was identified and removed from the data, Figure 6, for each feature.

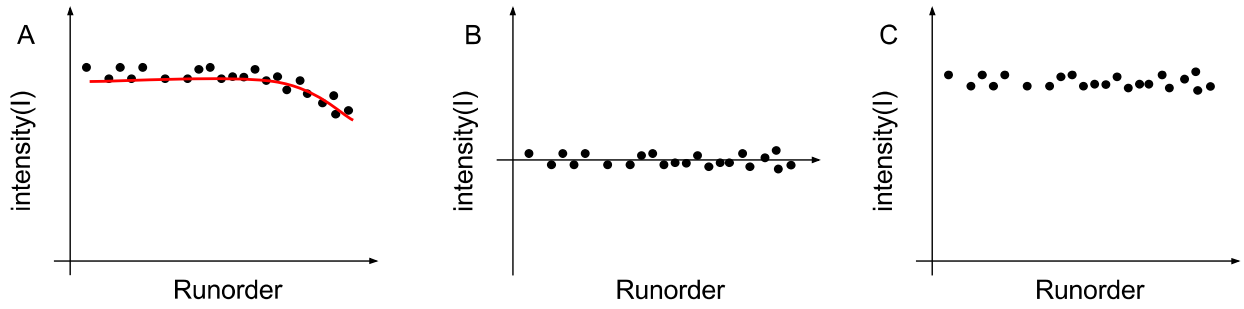


Figure 6: Schematic description of runorder correction. A. First a trend curve (red line) is identified. B. The predicted trend is then removed leaving only the residuals that are centered around zero. C. The data is rescaled to original intensity.

The resulting data would then be centred around zero loosing the information about general abundance of the feature. To correct for this the mean intensity across the samples are added to rescale the adjusted data back to the original intensity scale. The trend curve was found using Matlab's curve fitting function, fit, with the following call, fit(run order, intensity, 'smoothingspline', 'SmoothingParam', 1e-4) (MathWorks 2017). The LOESS (Cleveland & Devlin 1988) algorithm was also considered as an alternative, but it resulted in lower separation scores and had significantly longer computational time for the drug dataset, see Table 2. The drug dataset had about three times as many samples per feature which might contributed to the change between the methods. Meaning that LOESS scales poorly as the number of samples increase.

Table 2. Comparison of Smoothing spline and LOESS for runorder correction.

Method	Score drug	Time/feat.(ms)	Score grape	Time/feat.(ms)
Smoothing spline	0.449	15	7.46	22
LOESS	0.359	71	7.31	16

Neither LOESS nor the Smoothing spline approach are algorithms that are robust against outliers. Outliers in this case would represent large deviation in one feature for only one sample. This could for example happen if a drug was added to only a few samples. To make the algorithm robust, the outliers were identified and removed before curve fitting . Outliers were identified as residuals to the fitted curve that were greater than 1.5 standard deviations of all residuals (MathWorks 2017).

Procedure:

1. For each feature
    1. Fit a curve to intensity based on run order
    2. Remove samples with residuals greater than 1.5 standard deviations of residuals
    3. Fit a new curve using the reduced dataset
    4. Subtract the curve
    5. Add the mean of the original data.

4.4.3 Normalization of intensity distributions

Some samples may have systematically lower or higher expression of most features. While this can be of biological origin, it is often more likely to be a technical error due to either LC-MS, injection size or experimental preparation. To correct for this there are several methods to choose from such as factor scaling, quantile normalization and CyclicLoess (Ballman *et al.* 2004). CyclicLoess and quantile normalization is considered one of the more complex methods that has fewer assumptions than factor scaling . CyclicLoess and quantile normalization is based on the idea that any pair of samples should have the same statistical distribution of intensities. Both methods have been showed to perform similarly (Bolstad *et al.* 2003). As Herman et. al. used CyclicLoess I decided to use the same method for reproducibility. It should be noted that there are papers suggesting that quantile normalization is a better choice than CyclicLoess (Bolstad *et al.* 2003).

CyclicLoess is available in the R package "limma" but it is not available for Matlab (Ritchie ME *et al.* 2015). Hence the fast version of CyclicLoess was re-implemented in Matlab with slight modifications to get better performance and faster computational time. Comparing LOESS and Smoothing Spline showed that Smoothing Spline produced better separation scores at a lower computational time once again, see Table 3. Robust, outlier proof methods are not included in the R implementation of fast CyclicLoess. For LC-MS metabolomics data in *in vitro* systems

pharmacology the robustness is vital as for high concentration of drugs, outliers are always expected. The expected outliers are the drugs added to the cell culture which will be very high in only a few samples. For this reason the median was used to calculate the reference sample used in the fast algorithm instead of mean sample used by Ballman (Ballman *et al.* 2004).

Table 3. Comparison of Smoothing Spline and LOESS for CyclicLoess on the grape dataset.

Method	Score	Time total(ms)
Smoothing Spline	7.72	10
LOESS	6.16	199

The algorithm compares each sample with a reference sample, which is the median sample in the method implemented as part of this thesis project. The idea is that the majority of the intensities should be equal when comparing the sample and the median sample. To express this the trend of sample, s, plotted against the median sample should form a diagonal from the origin. The algorithm finds the trend in a modified version of this plot and de-trends any other trend than the diagonal from origin, see Figure 7.

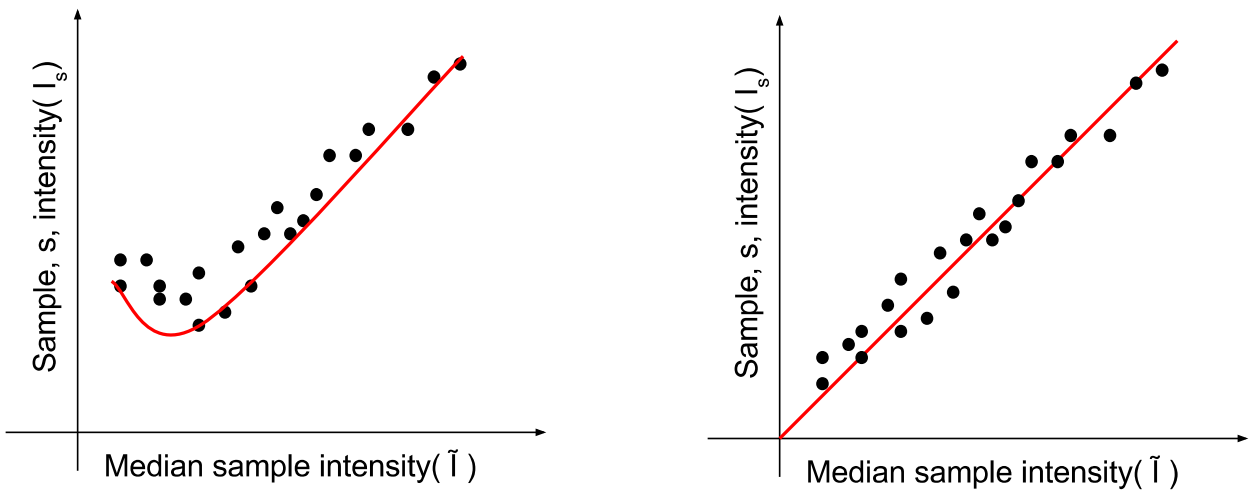


Figure 7: Schematic of CyclicLoess. Trend (red) is correct to fit the y=x function.

Procedure:

1. For each sample

    1. Fit a curve that deviates from the diagonal
    2. Remove samples with residuals greater than 1.5 standard deviations of residuals
    3. Fit a new curve using the reduced dataset

4. Subtract the curve

4.4.4 Blank background removal

When running LC-MS it is considered gold standard to add blank samples every 5-10 samples for both quality control and cleaning of the column. Hence the information in the blanks will indicate what features are carried in the background either as artefact (called carry over) from earlier samples or as a background from non-biological origins. Examples of sources of non-biological background are the mobile phase, the solvents and left-overs from the sample preparation. No matter if the signals are carry overs or a background signals they will introduce unwanted variance into the data. Two different methods of removing this variation was evaluated in this project called blank filter and blank subtraction. Blank filter was studied previously by Herman *et al.*.

Blank filter

The blank filter aims to exclude features that are highly expressed in the blank compared to the samples. Let $max_s(samples)$ denote the maximum intensity of the feature across all samples. Let $median_f(blanks)$ denote the median intensity of the blanks for feature f. The blank filter keeps feature f if:

$$\frac{max_s(samples)}{100} > median_f(blanks) \quad (16)$$

The actual ratio of 100 of the $max_s(samples)$ used was found to give the highest score when compared to both 10 and 1000.

Procedure blank filter:

1. For each feature, f

    1. **IF** $max_s(samples)/100 > median_f(blanks)$

        1. Consider f a blank signal and remove f from dataset

Blank subtraction

If there are carry over effects in the data this will lead to biological signals bleeding into the blanks. Hence the blank filter would identify these signals as contaminants and remove the entire feature from the dataset, potentially missing important biological variation. To solve this I made a new algorithm called blank subtraction. In blank subtraction the idea is to identify if a sample's intensity for a feature belongs to the same distribution as the blank samples. If the sample intensity is within 5 standard deviations of the blank median it is considered to be a contaminant and set to zero. Thus the feature f is considered to be a contaminant if the following strict inequality holds:

$$I - \hat{I}_b > std(I_b) \cdot 5 \quad (17)$$

where I is the intensity of the sample for that feature, $\hat{I}_b$ is the median of the blanks for this

feature and $std(I_b)$ is the standard deviation of the blanks for this feature. Several multiplication factors for the standard deviation were tried, ranging from 0 to 5, where 5 achieved the best score, see Table 4. 5 was the highest multiplicative factor tested. If the signals are normally distributed a factor of 5 means that there is a probability of 1 in 3.5 million that a non-contaminant is falsely labelled as contaminant. Higher factors are therefore hard to motivate theoretically and might thus lead to overfitting. However it should be noted that a common threshold used in limit of quantification is 10 standard deviations, meaning that higher thresholds should have been tried if there was time (Armbruster & Pry 2008).

Table 4. Comparison of thresholds in standard deviations(std) for blank subtraction on the grape dataset.

Number of std	Score
0	5.78
1	6.58
2	7.51
3	8.40
4	9.25
5	9.72

If the inequality in equation 17 is not fulfilled the measured signal I is considered to consist of both a blank background, and a true signal, I_t:

$$I = I_t + I_b \quad (18)$$

The true signal I_t is thus found by subtracting the median sample of the blanks.

Procedure blank subtraction:

1. For each sample, s
    1. For each feature, f
        1. **IF** abs(I(f,s) - median(blanks)) > std(blanks)*5
            1. I(f,s) <- I(f,s) - median(blanks)
        2. **ELSE**
            1. I(f,s) <- 0

4.4.5 Outlier TIC

Sometimes samples are subject to so large technical or experimental errors that preprocessing cannot clean them. Instead it is advisable to simply exclude these samples from the dataset. One

way to detect these outlier samples is to compare the sum of all intensities, called total ion count (TIC), and identify abnormal values. This is often done by manually by removing samples outside the whiskers of a box plot of the TICs. In this work an algorithm was implemented to automate this process.

The limits for outlier detection are traditionally set to 2 standard deviations (95% confidence interval assuming normal distributions) for the data. While this threshold is often good there are cases where a higher or lower threshold might be useful. Therefore the option to change the number of standard deviations was added to the function, see Figure 8. For the drug dataset a threshold of 2 standard deviations was used while a threshold of 3 was used for the grape dataset. This was due to the high variability of the different species in the grape dataset leading to the removal of entire species if the threshold was set at 2.

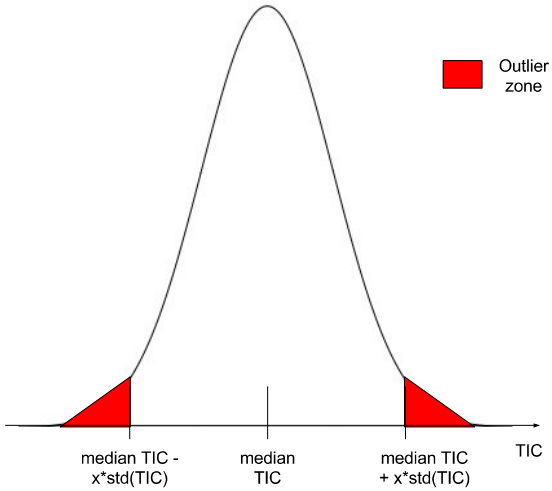


Figure 8: Schematic of outlier removal by TIC. Samples that lie within the outlier zone (red) are excluded. The limit for the outlier zone is set in terms of standard deviations (std) for the median.

Procedure:

Let mediandev(x) denote the median standard deviation of x.

1. For each sample, s

    2. **IF** abs(TIC(s) - TIC(samples)) > mediandev(samples)*2

        1. Consider s outlier, remove s from dataset

4.4.6 Outlier replicates

When creating three or more replicates it is possible to assess the quality of those replicates to identify samples with poor similarity between replicates. A fair assumption to make is that the

correlation, in this work Pearson correlation, should be high between each replicate, r_i , and the median of all replicates, $\tilde{r}$. In cases of large technical or experimental variability the correlation across replicates will be low. If a replicate, r_i, has a Pearson correlation coefficient $\rho_{r_i,\tilde{r}}$ lower then a threshold value, t, it should be removed from the dataset, see equation 22. The threshold was set to 0.8 for the grape dataset and 0.7 for the drug dataset.

$$\rho_{r_i,\tilde{r}} < t \quad (22)$$

Procedure:

1. For each group or replicates, g
    1. For each replicate, r
        1. **IF** Pearson correlation < t
            1. Consider r an outlier, remove r from the dataset

4.4.7 Batch filtering

For large experiments the samples often needs to be prepared in batches due to experimental limitations. This leads to unwanted variance that needs to be removed to exclude that the results are not due to effects of preparation batch. A similar approach as used in Herman *et al.* was implemented. The batch filter algorithm aims to find features that are highly present in one batch and had a low presence in all other samples. The threshold for highly present features was set to having a signal higher than zero in more than 80% of the samples. The threshold for low presence was set to 20%. Other levels were tried but the effect was limited in our datasets.

Procedure:
For each batch, *b*
    for each feature, *f*
        if intensity >0 for >80% of samples in batch *b* **AND** intensity>0 for <20% of all other samples
            consider feature *f* a batch specific contaminant and remove it

4.5 Batch effect removal

After all preprocessing steps were completed the first two principal components from PCA of the drug dataset separated the data into the four experimental batches the samples were prepared in, see Figure 9. Three different methods for identifying the latent variables explaining the batch effect were compared; PCA, PLS-DA and OOS-DA. The results from employing these methods were compared with the corresponding results obtained using Combat.

After the batch effects based on experimental batches were removed a new batch effect emerged in the processed data from OpenMS, see Figure 10. The new batch effects was identified as depending on if the run order was before or after the 95th sample. However this new batch effect could only be

found when using OpenMS and thus could not be seen using the other two feature extraction methods, XCMS and binning, see Figure 11 and Figure 12. In XCMS another kind of batch effect was identified but its origin could not be traced, see Figure 11. For binned processed data there was no clear new separation seen after batch effect removal, see Figure 12.

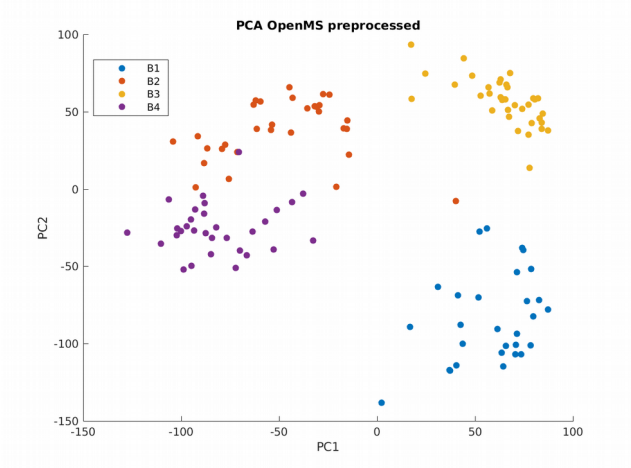


Figure 9: Batch effects in drug dataset after preprocessing shown by means of PCA using the first two principal components. Here OpenMS was used as feature extraction method.

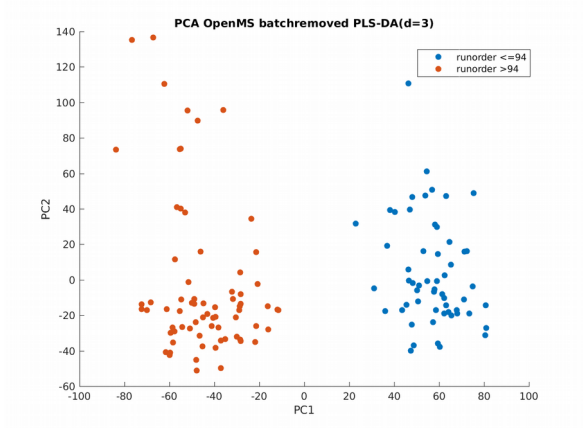


Figure 10: Secondary batch effects in drug dataset with OpenMS as feature extraction method. The data seperates into two batches based on runorder before 95 (blue) or after (red).

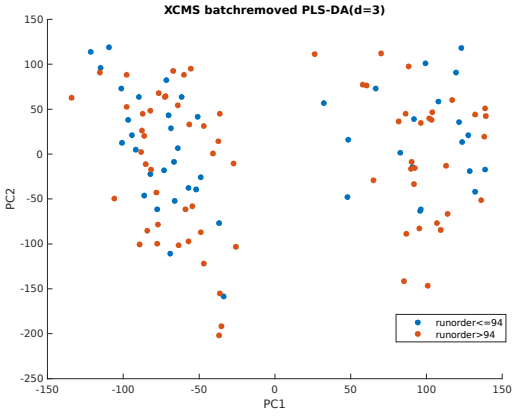


Figure 11: A clear split can be seen after batch effect removal using XCMS for the drug data. The split is is not based on runorder like in the OpenMS drug data.

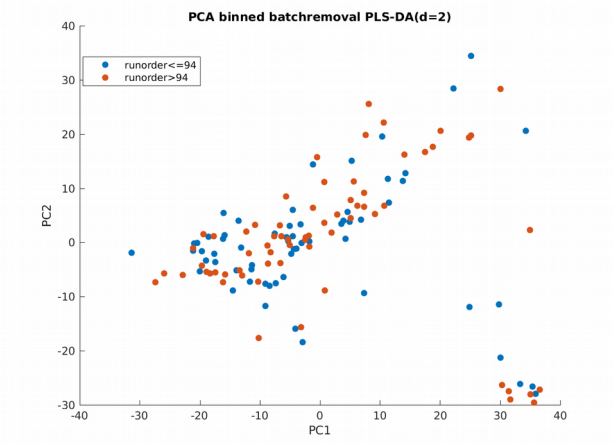


Figure 12: No clear separation based on runorder can be seen in when using binned drug data.

For the grape dataset there were no batch effects found. Instead the tissue type was used to simulate a batch effect to be removed, see Figure 13. Ideally this should result in the revealing of species specific effects that after batch effect removal should be independent of tissue type.

Figure 13: Separation based on tissue type is the dominant separator for the grape dataset.

4.5.1 Dimensionality optimization

The choice of the number of dimensions to remove from the data was selected by dimensionality optimization, using a grid. The separation score was used as objective function.

In the grape dataset no clear candidate classes could be found, the only classes annotated were the different grape species and tissues. The separation between all species were used for the analysis even though this was the original biological question. This means that there is a risk of overfitting, but unfortunately no better classes could be found.

In the drug dataset two good candidate classes are the Mebendazole and Control samples as they are present in all batches, with triplicates within each batch. Controls should separate well from Mebendazole assuming that the Mebendazole affects the metabolome. Since Mebendazole is a drug used to treat colon cancer this assumption should hold true (Nygren *et al.* 2013).

There was an imbalance in the number of samples in the Control and Mebendazole samples. Previous experience from Herman *et al.* and initial experiments showed a bias based on class sizes in the separation score. To counteract this imbalance in the analyses performed, replicates where created as linear combinations of all replicates in the under-represented class:

$$\boldsymbol{I}_r = \boldsymbol{I}_c \boldsymbol{W}_r \quad (20)$$

where $\boldsymbol{I}_r$ is the intensity matrix of the simulated replicates with samples as columns, $\boldsymbol{I}_c$, is the intensity matrix of the real replicates with samples as columns. $\boldsymbol{W}_r$ is a weight matrix where each column is a uniformly random vector where all elements are between 0 and 1, with the sum of all elements in each column equal to 1.

To increase stability of the random simulation, the score calculation was performed 100 times and the mean score was used. The standard deviations were found to be in the magnitude of 0.01.

4.6 Software and Hardware

All feature extraction methods and dimensionality optimization of data processing was run on UPPMAX using provided batch scripts (Anlind 2017). All other script were run on commodity hardware using R studio 2015 and Matlab 2015b under Ubuntu LTS 16.04.

5 Results & Discussion

5.1 Feature extraction

A comparison of feature extraction methods was only done for the drug dataset. This was due to the grape dataset being added too late in the project. Results show that the "Binning" method for feature extraction had the highest separation score after all steps of data analysis, see Table 5. It should however be noted that the separation was poor for all methods. It is therefore possible that a good separation cannot be found due to poor data quality. This is however an indication that the binning approach might be viable for feature extraction.

Table 5. Score values for the drug dataset using three different feature extraction methods, as judged at three different parts of the pipeline. The Binning method had the highest score in at all steps.

Step	Binned	XCMS	OpenMS
Feature Extraction	0.33	0.31	0.32
Preprocessing	0.71	0.43	0.41
Batch removal	0.71	0.46	0.41

5.2 Preprocessing pipeline

Most newly incorporated preprocessing methods showed no or actually negative improvements on the separation score in both datasets, see Table 6 and 7. For the grape dataset the logarithm had a strong negative effect while the normalization of intensity distributions and the blank subtraction had strong positive impacts. However for the drug dataset all these functions had mostly neglect-able effects on the separation score. Overall effects on the separation score was mostly neglect-able for the drug dataset. The exception is when binning is used for feature extraction where positive

effects can be seen at blank subtraction. This might be due to the binning being closer to the raw data making processing of the data easier.

Methods that had poor or neglect-able performance, marked with footnote 1, were excluded before progressing in the pipeline. Poor performance was not only evaluated on separation score values. A high separation can be due to non-biological factors, such as outliers increasing the separation score. To determine if that was the case a more in-depth analysis was made for each method which will be explained in the following sections. The result of this analysis resulted in normalized intensity distributions being applied for grape dataset despite having a negative separation score impact. When there was no score impact the methods were not used to minimize the risk of destroying the data and batch effects before batch effect removal.

The final preprocessing pipeline for the grape dataset included: Logarithm, Normalize Distributions, Blank Subtraction, TIC Outlier, Replicate Outlier. For the drug dataset the methods was tailored depending on the feature extraction method. For the binned data the pipeline was: Logarithm, Blank Subtraction, TIC Outlier, Replicate Outlier. For XCMS the pipeline was: Log, TIC Outlier, Replicate Outlier. For OpenMS the pipeline was: Log, Blank Filter, TIC Outlier, Replicate Outlier.

Table 6. Separation score(s) and improvement(Δs) of separation score for the preprocessing on grape dataset. s = seperation score between all biological replicate classes, s_{wt} = average seperation score between biological replicate classes within tissues types.

Method	s	Δs	s_{wt}	Δs_{wt}
Raw	17.16	-	6.09	-
Log	7.54	-9.62	4.38	-1.71
Correction runorder	7.53[1]	-0.01[1]	4.18[1]	-0.20[1]
Normalize Distributions	7.75	+0.22	4.24	-0.14
Blank Filter	7.14[1]	-0.61[1]	3.93[1]	-0.31[1]
Blank Subtraction	9.72	+1.97	5.24	+1.00
Outlier TIC	9.72	+0.00	5.24	+0.00
Outliers replicate	9.73	+0.01	5.23	-0.01

[1] This method was excluded due to poor performance. subsection for more details.

Table 7. Separation score(s_{MC}) and improvement(Δs_{MC}) of separation score for the preprocessing methods on the drug dataset. s_{MC}(B) = score for dataset processed with binning, s_{MC}(X) = score for dataset processed with XCMS, s_{MC}(O) = score for dataset processed with OpenMS, Feature Ex. = Feature Extraction, Corr. Run = Correct runorder, Norm. Dst. = Normalize distributions, Blank F. = Blank filter, Blank Sub. = Blank Subtraction, Batch F. = Batch Filter, OL TIC = Outlier TIC, OL Rep. = Outlier Replicates.

Method	s_{MC}(B)	Δs_{MC}(B)	s_{MC}(X)	Δs_{MC}(X)	s_{MC}(O)	Δs_{MC}(O)
Feature Ex.	0.33	-	0.31	-	0.32	-
Log	0.25	-0.08	0.48	+0.17	0.45	+0.13
Corr. Run.	0.24	-0.01	0.48[1]	+0.00[1]	0.45	+0.00
Norm. Dst.	0.24[1]	+0.00[1]	0.45[1]	-0.03[1]	0.45[1]	+0.00[1]
Blank F.	-[1,2]	-[1,2]	0.48[1]	+0.00[1]	0.47	+0.02
Blank Sub.	0.68	+0.44	0.46[1]	-0.02[1]	0.45[1]	+0.00[1]
Batch F.	0.61[1]	-0.07[1]	0.48[1]	+0.00[1]	0.47[1]	+0.00[1]
OL. TIC	0.74	+0.06	0.42	-0.06	0.47	+0.00
OL. Rep.	0.71	-0.03	0.43	+0.01	0.42	-0.05

[1] This method was excluded due to poor performance. See subsection for more details.
[2] The score could not be calculated due to too few remaining features.

5.2.1 Log

If no logarithm is applied the data will heavily favour changes that have a high intensity value. Hence the logarithmic transformation should always be applied. In our dataset we can only see an decrease for the grape dataset and the binned drug data, see Table 6 and 7. This would indicate that there were high intensity features that contributed more to the separation than the intensities with lower intensity values. High intensity features are typically added chemicals or very highly abundant substances that often have high variance. Except for solvents and mobile phase it could have been the drugs added for the drug dataset. In the binned data there are a high number of features (>15000). As the number of features increases the risk of the data being noisy in lower and middle domains rises, hence the logarithm amplifies this noise which could be an additional reason.

5.2.2 Correction of intensity based on run order

The correction based on run order addresses that there might be a systematic change in intensity in some features based on run order. Ideally it should not change the intensities if there is no systematic difference and correct it if there is one. For all datasets only minor changes in separation could be seen. This would indicate that the variation, if there is one, does not affect the separation or that the underlying biologically related separation and the technical separation cancel out. By investigating at the features where the samples had the highest mean change of intensity one can get an idea of what corrections were done to the data.

For the grape dataset, the run order correction was run on each tissue separately. The run order correction on the p tissue introduced new unwanted variation into the data, see Figure 14. Before run order correction it seems to be two underlying classes with different expression. One at around 6.5 and another at 4. After run order correction these classes got additional unwanted variation. By investigating the fitted trend curve it can be seen that the slight bias for early sample order for the class with higher intensities lead to a sort of overfitting, see Figure 15. Similar but weaker phenomena were seen for the remaining tissues s and b, see Appendix A.

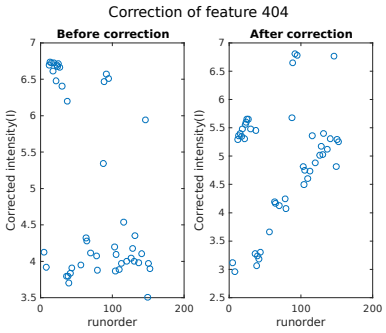


Figure 14: Effect of runorder correction of grape dataset tissue p.

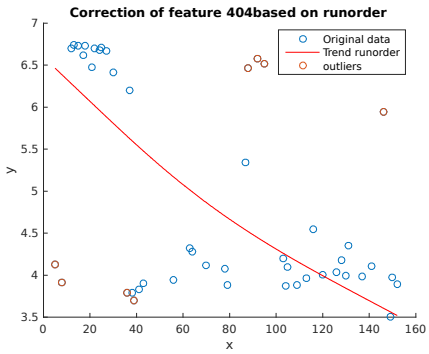


Figure 15: The trend (red) is overfitted for feature 404 in grape dataset tissue p.

For the drug dataset each feature extraction method was investigated separately. The run order correction has little to no effect on the highest changed feature in XCMS, see Figure 16. Since the effect was also small on the separation score it was decided that the correction of run order should not be applied for XCMS subset.

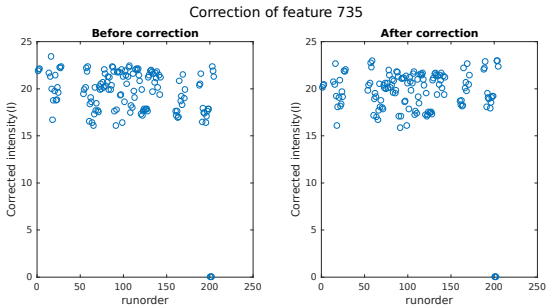


Figure 16: Runorder correction of the drug dataset with XCMS as feature extraction method has an neglectable effect of the features.

For Binned and OpenMS large variation due to run order was noted, and the trend line was well fitted to this trend for both dataset, see Figure 17 and Figure 18. Hence the run order correction was able to remove unwanted variation in these dataset despite low separation score increase. The run order correction was applied for both OpenMS and Binned versions of drug dataset.

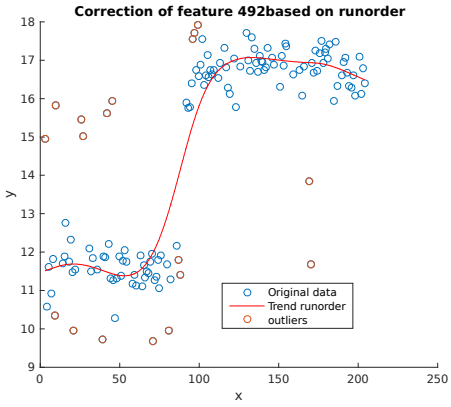


Figure 17: Trend is able to identify runorderbased variation in the feature 492 in the drugdataset processed with OpenMS.

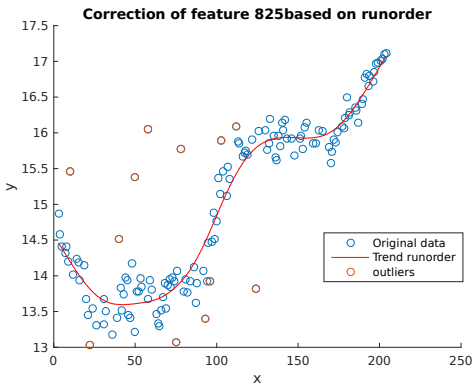


Figure 18: Trend is able to identify runorderbased variation in feature 845 in the drugdataset processed with Binning.

5.2.3 Normalization of intensity distributions

The normalization of intensity distributions has the goal of making the intensity distribution similar to the distribution for a typical sample. The methods used for normalization are mainly based on assumption that all samples have about the same distribution of signal intensity. One case where this does not hold true is between tissue types, which is why the method was employed separately for each tissue type in the grape dataset. The separation score decreased for the grapes, but by looking into the distributions of tissue b it can be observed that the samples are being normalized as desired, see Figure 19.

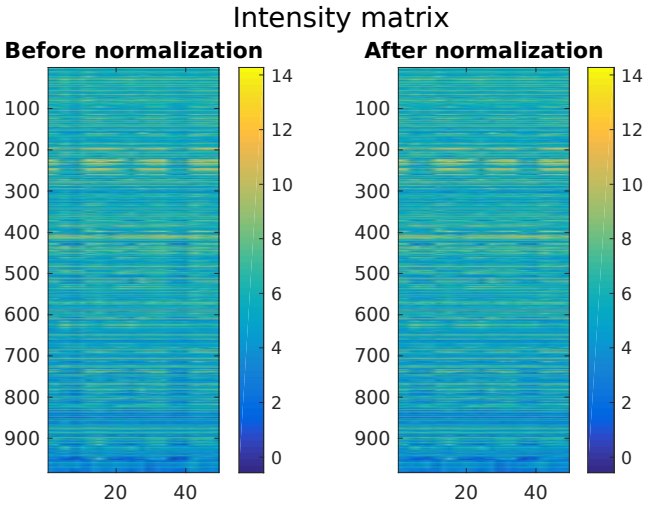


Figure 19: For grape dataset, tissue b, normalisation of distribution succefully corrects for outliers in sample 9 and those close to sample 40.

For the drug dataset processed using OpenMS as feature extraction method the opposite is true where the distributions are well aligned before normalization but becomes vastly different afterwards, see Figure 20. Similar results were seen with XCMS and Binning, see Appendix B.

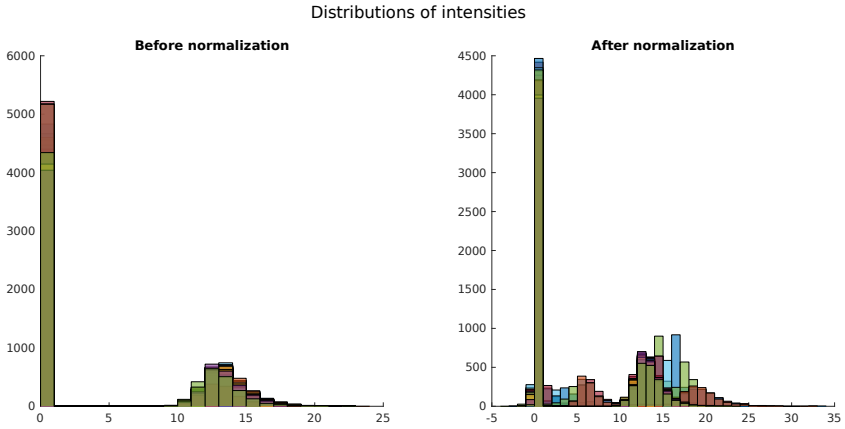


Figure 20: Normalization of intensity distributions in the drug dataset using OpenMS as feature extraction method. After normalization the samples intensity distributions are more different than before normalization.

It is not clear why this strange result is obtained for the drug dataset. In some way the original signal must be very different from the median sample. Due to time restriction there was not time for a more in-depth investigation of this phenomenon. However it does highlight the importance of investigating the effects on the data for this normalization method as unexpected results might appear.

5.2.4 Blank background removal

For removal of blank background two different approaches were tried, blank subtraction and blank filtering. For the grape dataset and drug dataset processed with binning the blank subtraction had resulted in a vast increase in separation score while the blank filter had a negative effect on separation score, see Tables 6 and 7. But for the drug dataset processed with XCMS and OpenMS the effects on separation score where mostly neglect-able for both methods, where a slight increase was seen for blank filter on OpenMS data, see Table 6 and 7.

In the binned drug data and the grape dataset almost all features, 100% and 98%, are filtered out by the blank filter method. This is a sign of strong carry over effects where the signals from previous samples were carried over to the next sample. Here blank subtraction has a slightly lower percentage affected features, see Table 8. Since blank subtraction handles each signal individually more parts of the data can be kept than in the blank filter.

For the drug dataset processed with XCMS and OpenMS the opposite is true, where the blank subtraction affects more features than the blank filter, see Table 8. Since the blank subtraction is dynamic not all samples are affected within one feature which could contribute to the difference. Out from these numbers and the separation score it is hard to evaluate if the effect of either method

is positive or negative. For this reason the separation score was set as used as the guide. For OpenMS by blank filtering there was a slight increase of 0.03 and therefore that method was used.

Table 8. Impact on feature for blank filter and blank subtraction.

Dataset	Features affected(blank sub)	Features affected(blank filter)
Grape	885(90%)	966(98%)
Drug Binned	12 787(83%)	15 401(100%)
Drug XCMS	5 264(67%)	3 078(39%)
Drug OpenMS	4 959(77%)	1 475(23%)

5.2.5 Batch filter

For the batch filter little to no effect was seen in the score. This might be due to that only about 1% of the features were removed, see Table 9. The small fractions of features removed could be a sign of that the batch effects are more complex than can be captured by the coverage approach used by the batch filter.

Table 9. Number of features removed by batch filter.

Dataset	Features removed
Drug Binned	370(2%)
Drug XCMS	69(1%)
Drug OpenMS	40(1%)

5.3 Batch effect removal

5.3.1 Grape data

Dimensionality optimization in grape data showed clear increase in separation score for all three methods. PCA and PLS-DA reached a maximum score of 1.06 using 1 dimension for removal, see Figure 21. For OOS-DA the even higher scores could be observed at using between 35-50 dimensions. However the scores have a high variation in-between every other increase or decrease of dimensions, which could be a sign of instability of OOS-DA in that region. To test the stability, OOS-DA batch effect removal was repeated 10 times for each dimension value and the mean and standard deviations were plotted, see Figure 22. It can now be seen that the standard deviations increase substantially when the dimensions are higher than 30, see Figure 22. The cause for this is unknown but it could be due the randomization of vectors used to generate uncorrelated vectors when identifying the latent variables. To minimize risk of overfitting no dimensions higher than 30 were used for batch effect removal with OOS-DA on the grape dataset. Instead a maximum score was found using 28 dimensions resulting in a score of 1.12, Figure 21. OOS-DA provided a slightly higher score than PLS-DA and PCA.

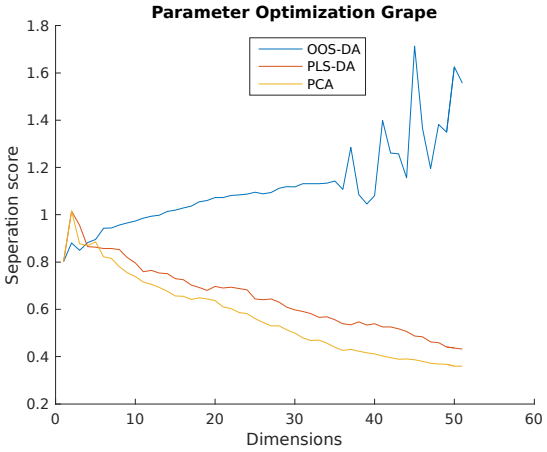


Figure 21: PLS-DA (red) and PCA (yellow) methods achieve the same maximum separation score of 1.01 at dimension=1. OSS-LDA have the highest maximum, but show numerical instability at dimensions >30.

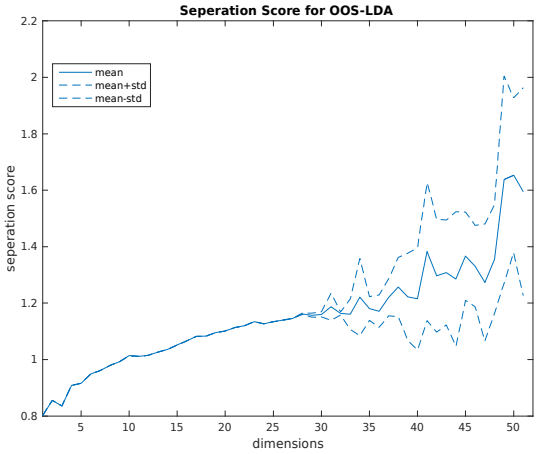
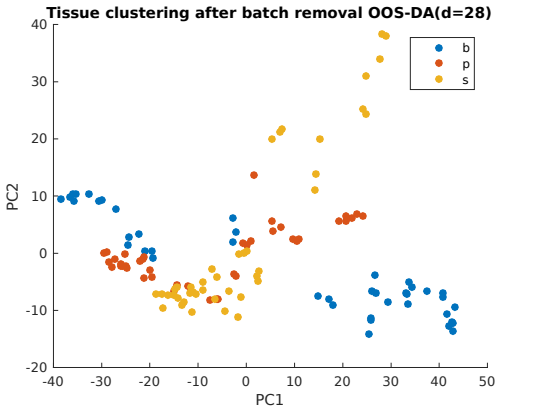
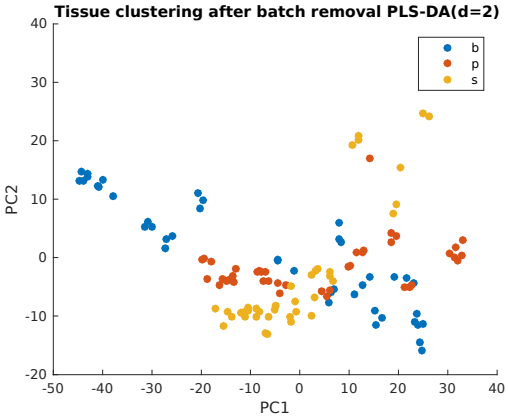


Figure 22: Reproducability of OOS-DA for 10 iterations/dimension show unstable scores at dimensions >30.

PCA had remaining tissue clustering as identified by the PCA plot, see Figure 23. OOS-DA and PLS-DA on the other hand showed only small signs of tissue clustering, Figure 23. The ultimate goal was to see if samples would cluster into underlying species after batch effects were removed. A visual inspection of 3 randomly chosen species, 41b maillard (41B)**,** Iasma Eco 3 (F3P51) and Gewurztraminer (GWT), showed that only OOS-DA and PLS-DA was able to achieve some clustering of species, see Figure 24. However the clustering is not perfect and the batch effect was not fully removed successfully in neither OOS-DA nor PLS-DA.

Batch removal with Combat resulted in a separation score of 1.20, which is higher than all other methods. The PCA plot confirms that the tissue information was successfully removed, see Figure 25. Species clustering was tighter than for OOS-DA and PLS-DA, but there is still overlap between species indicating an incomplete separation, see Figure 25. Therefore Combat provided the best batch effect removal for the grape dataset.





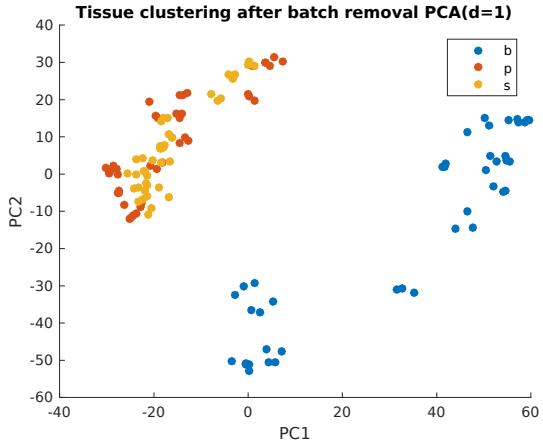
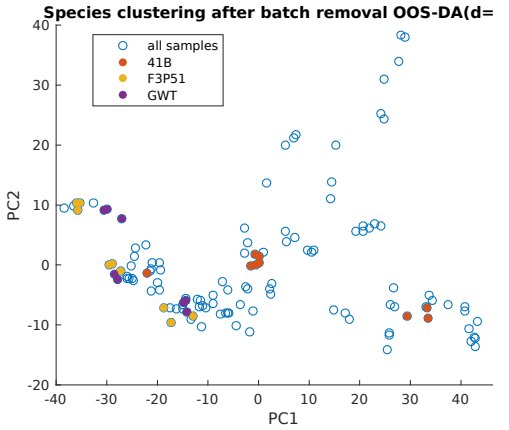
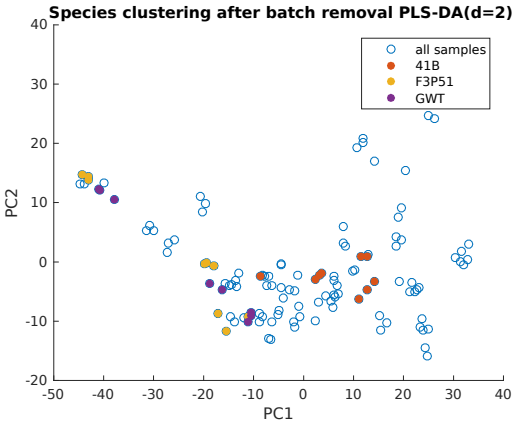


Figure 23: OOS-DA and PLS-DA results in the most overlapping tissue groups, indicating the best batch effect removal. For PCA tissue 'b' (blue) still form a separate group.





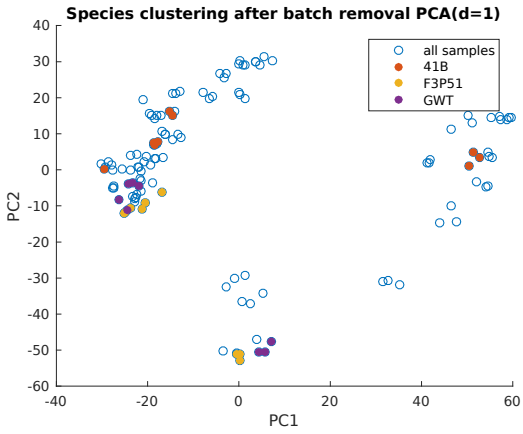


Figure 24: Species clustering show that there is a somewhat successful species clustering for F3P51 and GWT when OOS-DA or PLS-DA is used for batch effect removal. For PCA the clustering is split due to incomplete batch effect removal.

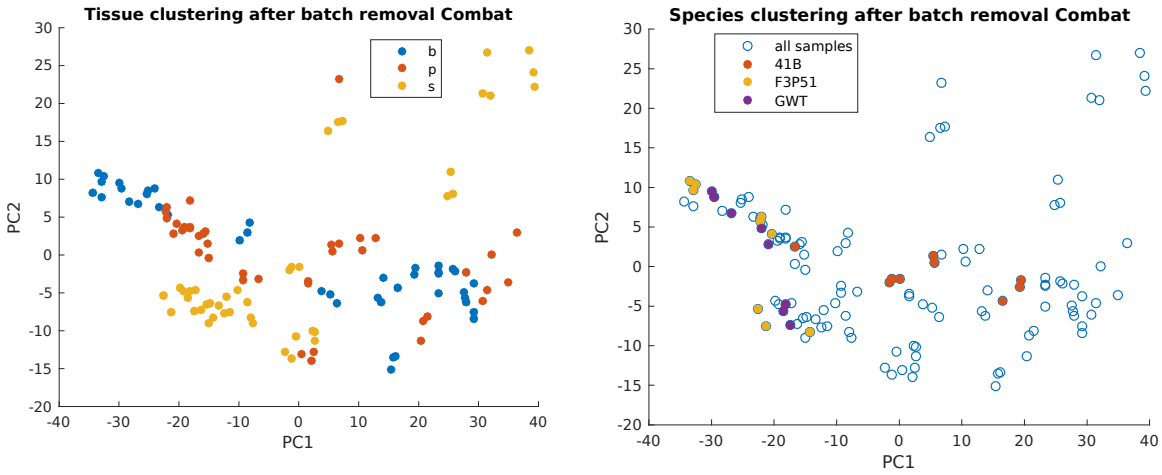


Figure 25: Batch effect removal using Combat was able to remove tissue information and clustering of species can be observed. There is still overlap across species after batch effect removal using Combat.

5.3.2 Drugs

During dimensionality optimization using the drug dataset individual points between 40-50 dimensions for batch effects in the grid of binned dataset using OOS-DA method had the highest separation scores, see Appendix A. Just like with the grape dataset these maximum points are probably due to numerical instability in OOS-DA at dimensions higher than 30 for this dataset. The dimensionality optimization was therefore limited to a maximum of 30 in both dimensions to reduce the chance of instability.

Only batch effect removal with OOS-DA on data with XCMS used as feature extraction method that was able to achieve an increase in separation score, see Figure 26. The maximum separation score of 0.46 was found while using first 23 dimensions for the first batch effect, the four sample preparation batches, and 1 dimension with the second batch effects, the split classes. This is surprising as the split effect could not be seen in the prior plots for PCA. The score improvement of batch effect removal was however only 0.03, which is just slightly above the standard deviation of 0.01 for the score. This means that the separation increase is mostly insignificant.

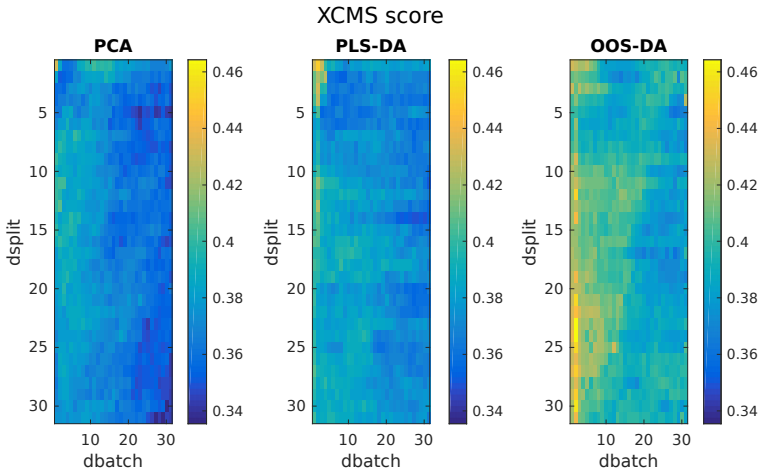


Figure 26: Seperation score for batch effect removal on drug dataset used with XCMS. Only OOS-DA is able has a score increase at dimensions higher than 0.

For other feature extraction methods OOS-DA had an small negative effect on the score. For PCA and PLS-DA the score decreased rapidly as dimensions increased, see Figure 27 and Figure 28. The negative effect on the score for OOS-DA was significantly lower than when using PCA or PLS-DA in all cases. Hence if a latent method is to be used, OOS-DA is the best choice.

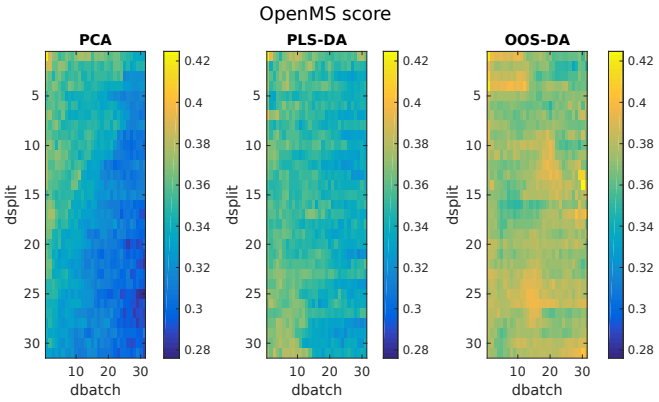


Figure 27: Seperation score for batch effect removal on drug dataset used with OpenMS. Small negative score impact is seen for OOS-DA while larger negative impact is seen for PLS-DA and PCA.

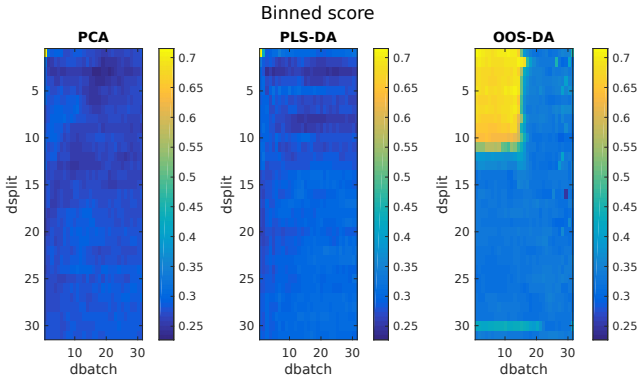


Figure 28: Seperation score for batch effect removal on drug dataset used with OpenMS. Negelectable score impact is seen for OOS-DA below 10 dimensions in both batch and split dimensions. While rapid decrease is observed for PLS-DA and PCA at all dimensions higher than 0.

Batch effect removal with Combat resulted in decreasing scores for all feature extraction methods, see Table 10. The method performed among the worst in terms of separation score together with PCA.
OOS-DA was able to get an slight increase or neglect-able effect on separation score independently on feature extraction method.

Table 10. Separation score(S) for different batch removal methods. The score is given for all 3 different feature extraction methods. The method with highest score is OOS-DA and no batch removal. The lowest scores is found in Combat or PCA depending on feature extraction method.

Method	S_{Binned}	S_{XCMS}	S_{OpenMS}
No batch removal	0.71	0.43	0.42
PCA	0.35	0.41	0.39
PLS-DA	0.43	0.45	0.40
OOS-DA	0.71	0.46	0.42
Combat	0.40	0.42	0.38

The batch effect removal with using OOS-DA at the optimal dimensions, dbatch=4 (sample preparation batches) and dsplit =12 (split batches), which had the highest score was further investigated. The batch effect removal is incomplete where batch 2 still clusters separately from the other three batches, see Figure 29. While investigating the clustering of the two separation score classes Control and Mebendazole, no clear separation between them can be seen, see Figure 29. This indicates that batch effect removal is both incomplete and not enough to clean up the drug

dataset. Instead as suggested by Leek *et al.* in terms of high variability and batch effects, the experiments might have to be re-done instead.

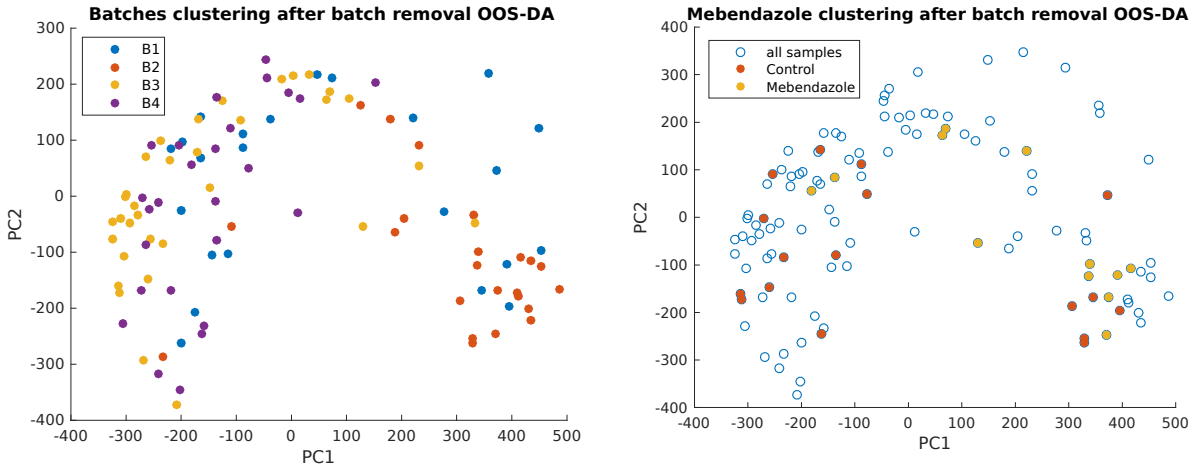


Figure 29: Batch effect removal with optimal settings, OOS-DA (dbatch=4, dsplit=12) and feature extraction methods (Binning). Batch effect from batch 2 (red) is not completely removed, and there is no visible clustering between control and Mebendazole samples.

6 Conclusions

The comparison and evaluation of computational data analysis methods for processing of LC-MS data in the context of metabolomics studied in this thesis project lack a one size fits all solution. While some methods showed universally poor results, both in terms of plots and scores, others showed varying result depending on dataset, feature extraction methods and dimensionality. It can be concluded that correction based on run order performed poorly, but the run order plots for the features indicate that good results could be achieved if the fitting of the trend curve is improved. The blank subtraction outperformed or performed on par with the blank filtering method, offering a good option for a more robust and accurate background removal. For some methods the results also offer insights into the data. Such as the occurrence of carry over effects indicated by the background removal or potential run order effects as seen by the correct run order plots. Hence preprocessing methods should be used with care and be thoroughly investigated to ensure that the effects of the preprocessing is improving data quality.

Furthermore this work showed that evaluation of LC-MS data is hard when the quality of the data is poor. In this thesis project one high quality dataset was identified, the grape dataset. Here the clustering of replicates could be seen in the first latent variables directly after feature extraction. The second dataset studied, drug dataset, was found to have poor quality as replicates did not group together even after extensive preprocessing and batch effect removal. While high quality datasets ensure that there are signals that can reveal the biological meaning in the dataset, cleaning an already quite clean data is questionable. On the other hand for a low quality dataset the need for

cleaning is high, but one cannot ensure that it is possible to extract any biological conclusion. The only safe way to evaluate methods are controlled experiments made to evaluate that particular method. Taking time and money for these experiments might feel wasteful but is ultimately needed to improve further experiments and evaluation.

The evaluation of feature extraction methods performed in this thesis project was inconclusive as the feature extraction methods were only tried for the drug dataset. It can however be stated that the only significant change in separation score between the methods was favouring the least computationally expensive method, binning, at all stages. The short computational time for binning enables an extensive optimization of the number of bins, which might altogether be as, or more, important than the choice of feature extraction method. Further experiments should therefore be done to investigate the effects of parameter optimization for binning and comparing it to the more well known methods such as XCMS.

For batch effect removal the newly implemented OOS-DA method had a performance close to Combat for the grape dataset and a better performance for the drug dataset. Improvements are still needed to improve stability for using high number of dimensions. If the stability in higher dimensions can be secured there is an potential that OOS-DA might add improved removal of batch effect compared to Combat. However as seen in the current performance of both Combat and OOS-DA the underlying effects are often hard to fully remove and repeating the experiments might still be the most effective means of batch effect removal.

The aim of this thesis was to replicate and improve a computational pipeline for LC-MS raw data processing. The resulting computational pipeline is available on Github (Anlind 2017). The pipeline has modular parts that were all investigated in this report. By investigation on two different datasets partial insights where obtained regarding their pitfalls. Preprocessing methods where found to often lack outstanding performance, and a new improved blank background removal was developed. A simple binning method for feature extraction showed potential to outperform today's established methods. Finally OOS-DA was shown to be a good contender for batch effect removal, but improvements are needed for use in higher dimensions on LC-MS data.

7 References

Aftab O, Engskog MKR, Haglöf J, Elmsjö A, Arvidsson T, Pettersson C, Hammerling U, Gustafsson MG. 2014. NMR Spectroscopy-Based Metabolic Profiling of Drug-Induced Changes In Vitro Can Discriminate between Pharmacological Classes. Journal of Chemical Information and Modeling 54: 3251–3258.

Alonso A, Marsal S, Julià A. 2015. Analytical Methods in Untargeted Metabolomics: State of the Art in 2015. Frontiers in Bioengineering and Biotechnology, doi 10.3389/fbioe.2015.00023.

Alter O, Brown PO, Botstein D. 2000. Singular value decomposition for genome-wide expression data processing and modeling. Proceedings of the National Academy of Sciences 97: 10101–10106.

Andgan. 2016. Pipeline to process UPLC-MS/MS data with XCMS andgan/metabolomics_pipeline. WWW document 16 February 2016: https://github.com/andgan/metabolomics_pipeline. Accessed 25 April 2017.

Anlind N. 2017. masterthesis. WWW document 13 May 2017: https://github.com/Sn0flingan/masterthesis. Accessed 13 May 2017.

Armbruster DA, Pry T. 2008. Limit of Blank, Limit of Detection and Limit of Quantitation. The Clinical Biochemist Reviews 29: S49–S52.

Ballman KV, Grill DE, Oberg AL, Therneau TM. 2004. Faster cyclic loess: normalizing RNA arrays via linear models. Bioinformatics (Oxford, England) 20: 2778–2786.

Barwick V, Langley J, Mallet T, Stein B, Webb K. 2006. Best Practice Guide for Generating Mass Spectra. LGC Limited

Benito M, Parker J, Du Q, Wu J, Xiang D, Perou CM, Marron JS. 2004. Adjustment of systematic microarray data biases. Bioinformatics 20: 105–114.

Bolstad BM, Irizarry RA, Astrand M, Speed TP. 2003. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. Bioinformatics (Oxford, England) 19: 185–193.

Borisy AA, Elliott PJ, Hurst NW, Lee MS, Lehár J, Price ER, Serbedzija G, Zimmermann GR, Foley MA, Stockwell BR, Keith CT. 2003. Systematic discovery of multicomponent therapeutics. Proceedings of the National Academy of Sciences 100: 7977–7982.

Cleveland WS, Devlin SJ. 1988. Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting. Journal of the American Statistical Association 83: 596–610.

de Jong S. 1993. SIMPLS: An alternative approach to partial least squares regression. Chemometrics and Intelligent Laboratory Systems 18: 251–263.

Fryknäs M, Gullbo J, Wang X, Rickardson L, Jarvius M, Wickström M, Hassan S, Andersson C, Gustafsson M, Westman G, Nygren P, Linder S, Larsson R. 2013. Screening for phenotype selective activity in multidrug resistant cells identifies a novel tubulin active agent insensitive to common forms of cancer drug resistance. BMC cancer 13: 374.

Fukushima A, Kusano M, Mejia RF, Iwasa M, Kobayashi M, Hayashi N, Watanabe-Takahashi A, Narisawa T, Tohge T, Hur M, Wurtele ES, Nikolau BJ, Saito K. 2014. Metabolomic Characterization of Knockout Mutants in Arabidopsis: Development of a Metabolite Profiling Database for Knockout Mutants in Arabidopsis1[W][OPEN]. Plant Physiology 165: 948–961.

Gullbo J, Fryknäs M, Rickardson L, Darcy P, Hägg M, Wickström M, Hassan S, Westman G, Brnjic S, Nygren P, Linder S, Larsson R. 2011. Phenotype-based drug screening in primary ovarian carcinoma cultures identifies intracellular iron depletion as a promising strategy for cancer treatment. Biochemical Pharmacology 82: 139–147.

Hassan S, Laryea D, Mahteme H, Felth J, Fryknäs M, Fayad W, Linder S, Rickardson L, Gullbo J, Graf W, Påhlman L, Glimelius B, Larsson R, Nygren P. 2011. Novel activity of acriflavine against colorectal cancer tumor cells. Cancer Science 102: 2206–2213.

Haug K, Salek RM, Conesa P, Hastings J, de Matos P, Rijnbeek M, Mahendraker T, Williams M, Neumann S, Rocca-Serra P, Maguire E, González-Beltrán A, Sansone S-A, Griffin JL, Steinbeck C. 2013. MetaboLights—an open-access general-purpose repository for metabolomics studies and associated meta-data. Nucleic Acids Research 41: D781–D786.

Herman S, Khoonsari PE, Aftab O, Krishnan S, Strömbom E, Larsson R, Hammerling U, Spjuth O, Kultima K, Gustafsson M. 2017. Mass spectrometry based metabolomics for in vitro systems pharmacology: pitfalls, challenges, and computational solutions. Metabolomics 13: 79.

Hess GP, Fonseca E, Scott R, Fagerness J. 2015. Pharmacogenomic and pharmacogenetic-guided therapy as a tool in precision medicine: current state and factors impacting acceptance by stakeholders. Genetics Research, doi 10.1017/S0016672315000099.

Hu Q, Noll RJ, Li H, Makarov A, Hardman M, Graham Cooks R. 2005. The Orbitrap: a new mass spectrometer. Journal of Mass Spectrometry 40: 430–443.

Iorio F, Bosotti R, Scacheri E, Belcastro V, Mithbaokar P, Ferriero R, Murino L, Tagliaferri R, Brunetti-Pierri N, Isacchi A, di Bernardo D. 2010. Discovery of drug mode of action and drug repositioning from transcriptional responses. Proceedings of the National Academy of Sciences of the United States of America 107: 14621–14626.

Johnson WE, Li C, Rabinovic A. 2007. Adjusting batch effects in microarray expression data using empirical Bayes methods. Biostatistics (Oxford, England) 8: 118–127.

Kaddurah-Daouk R, Weinshilboum RM. 2014. Pharmacometabolomics: Implications for Clinical Pharmacology and Systems Pharmacology. Clinical Pharmacology & Therapeutics 95: 154–167.

Kell DB, Goodacre R. 2014. Metabolomics and systems pharmacology: why and how to model the human metabolic network for drug discovery. Drug Discovery Today 19: 171–182.

Lamb J, Crawford ED, Peck D, Modell JW, Blat IC, Wrobel MJ, Lerner J, Brunet J-P, Subramanian A, Ross KN, Reich M, Hieronymus H, Wei G, Armstrong SA, Haggarty SJ, Clemons PA, Wei R, Carr SA, Lander ES, Golub TR. 2006. The Connectivity Map: using gene-expression signatures to connect small molecules, genes, and disease. Science (New York, NY) 313: 1929–1935.

Leek JT, Scharpf RB, Bravo HC, Simcha D, Langmead B, Johnson WE, Geman D, Baggerly K, Irizarry RA. 2010. Tackling the widespread and critical impact of batch effects in high-throughput data. Nature Reviews Genetics 11: 733–739.

MathWorks. 2017. Robust Fitting. WWW document 25 April 2017: https://www.mathworks.com/examples/curvefitting/mw/curvefit-ex56454479-robust-fitting. Accessed 25 April 2017.

Narduzzi L. 2017. MTBLS209:The measurement of spatial distribution of grape metabolites in berry tissues using LC-MS. WWW document 23 January 2017: https://www.ebi.ac.uk/metabolights/MTBLS209. Accessed 25 April 2017.

Nassar AF, Wu T, Nassar SF, Wisnewski AV. 2017. UPLC–MS for metabolomics: a giant step forward in support of pharmaceutical research. Drug Discovery Today 22: 463–470.

Nielsen TO, West RB, Linn SC, Alter O, Knowling MA, O'Connell JX, Zhu S, Fero M, Sherlock G, Pollack JR, Brown PO, Botstein D, van de Rijn M. 2002. Molecular characterisation of soft tissue tumours: a gene expression study. The Lancet 359: 1301–1307.

Nygren P, Fryknäs M, Agerup B, Larsson R. 2013. Repositioning of the anthelmintic drug mebendazole for the treatment for colon cancer. Journal of Cancer Research and Clinical Oncology 139: 2133–2140.

Okada T, Tomita S. 1985. An optimal orthonormal system for discriminant analysis. Pattern Recognition 18: 139–144.

Perry RH, Cooks RG, Noll RJ. 2008. Orbitrap mass spectrometry: instrumentation, ion motion and applications. Mass Spectrometry Reviews 27: 661–699.

Ranninger C, Schmidt LE, Rurik M, Limonciel A, Jennings P, Kohlbacher O, Huber CG. 2016. Improving global feature detectabilities through scan range splitting for untargeted metabolomics by high-performance liquid chromatography-Orbitrap mass spectrometry. Analytica Chimica Acta 930: 13–22.

Ritchie ME, Phipson B, Wu D, Hu Y, Law CW, Shi W, Smyth GK. 2015. limma powers differential expression analyses for RNA-sequencing and microarray studies. Nucleic Acids Research 43:

Smith CA, Want EJ, O'Maille G, Abagyan R, Siuzdak G. 2006. XCMS: Processing Mass Spectrometry Data for Metabolite Profiling Using Nonlinear Peak Alignment, Matching, and Identification. Analytical Chemistry 78: 779–787.

Spiro Z, Kovacs IA, Csermely P. 2008. Drug-therapy networks and the prediction of novel drug targets. Journal of Biology 7: 20.

Sturm M, Bertsch A, Gröpl C, Hildebrandt A, Hussong R, Lange E, Pfeifer N, Schulz-Trieglaff O, Zerck A, Reinert K, Kohlbacher O. 2008. OpenMS – An open-source software framework for mass spectrometry. BMC Bioinformatics 9: 163.

Tautenhahn R, Böttcher C, Neumann S. 2008. Highly sensitive feature detection for high resolution LC/MS. BMC Bioinformatics 9: 504.

Theodoridis G, Gika H, Franceschi P, Caputi L, Arapitsas P, Scholz M, Masuero D, Wehrens R, Vrhovsek U, Mattivi F. 2012. LC-MS based global metabolite profiling of grapes: solvent extraction protocol optimisation. Metabolomics 8: 175–185.

ThermoFisher Scientific. Ion AmpliSeq Transcriptome Human Gene Expression Kit - Thermo Fisher Scientific. WWW document: https://www.thermofisher.com/order/catalog/product/A26325. Accessed 1 June 2017.

Thomas A, Rajan A, Giaccone G. 2012. Tyrosine Kinase Inhibitors in Lung Cancer. Hematology/Oncology Clinics of North America 26: 589–605.

van den Berg RA, Hoefsloot HC, Westerhuis JA, Smilde AK, van der Werf MJ. 2006. Centering, scaling, and transformations: improving the biological information content of metabolomics data. BMC Genomics 7: 142.

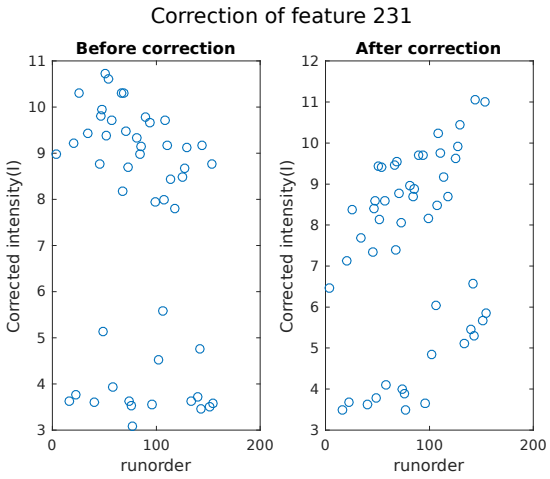
Wishart DS. 2016. Emerging applications of metabolomics in drug discovery and precision medicine. Nature Reviews Drug Discovery 15: 473–484.

Wolff MM, Stephens WE. 1953. A Pulsed Mass Spectrometer with Time Dispersion. Review of Scientific Instruments 24: 616–617.

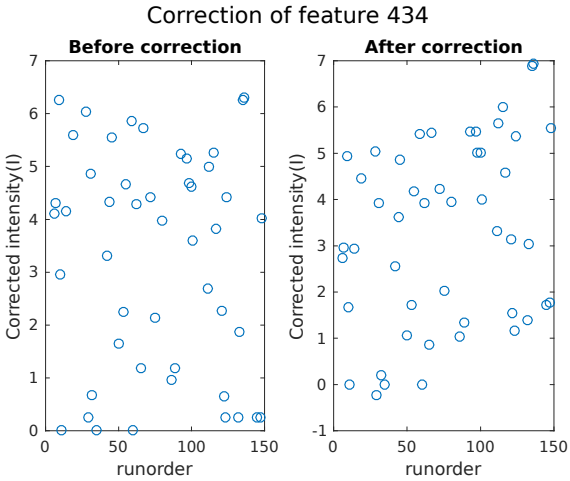
Zhang A, Sun H, Wang P, Han Y, Wang X. 2012. Modern analytical techniques in metabolomics analysis. The Analyst 137: 293–300.

Zhou J, Yin Y. 2016. Strategies for large-scale targeted metabolomics quantification by liquid chromatography-mass spectrometry. Analyst 141: 6362–6373.

Appendix A

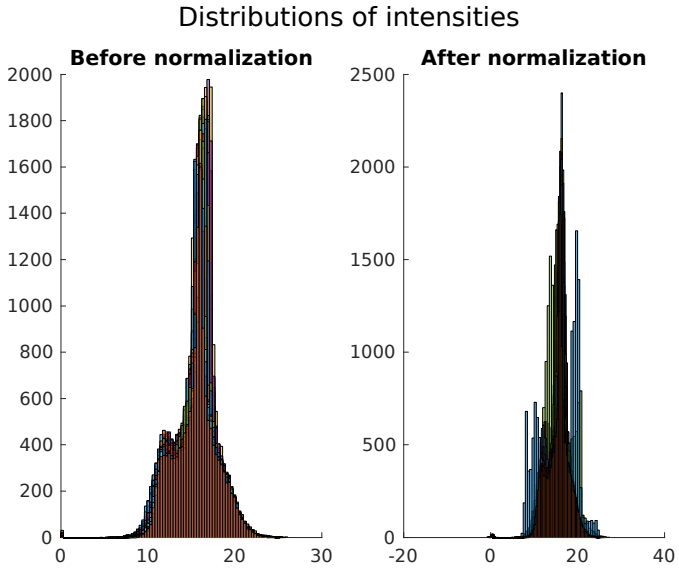


For Tissue b

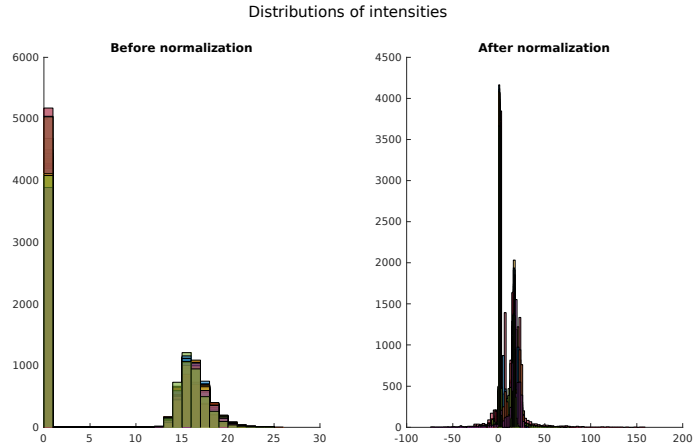


For tissue s

Appendix B



For Binning



For XCMS

Appendix C

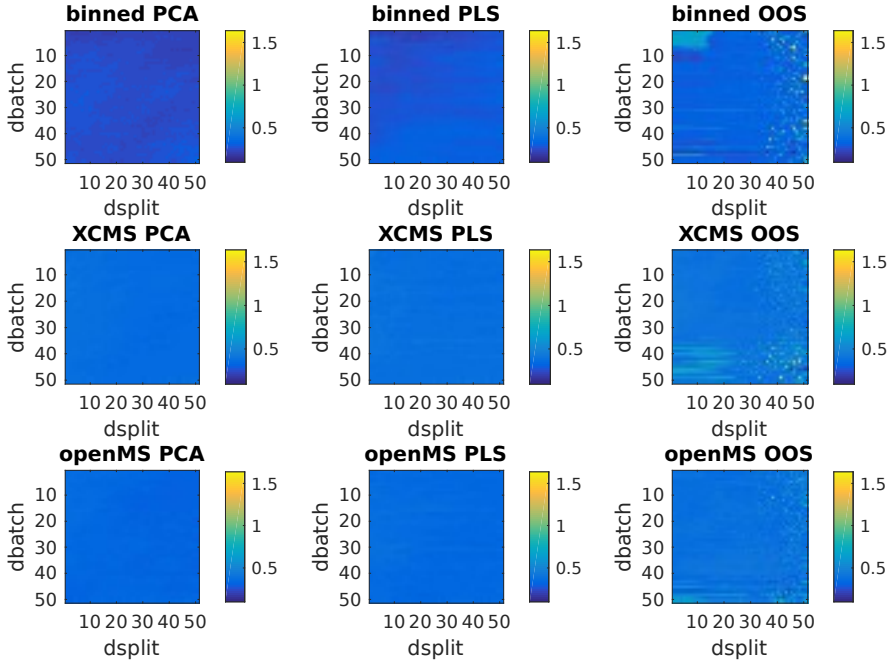


Figure: Parameter optimization showed binned batch removed with OOS-DA to give the highest score. At dimensions higher than 30 OOS-DA show signs of instability as indicated by the grainy surface.