Analysis of LC-MS data in untargeted nutritional metabolomics

Anton Mattsson

School of Science

Thesis submitted for examination for the degree of Master of Science in Technology.

Kuopio 12.8.2019

Supervisor

Prof. Harri Lähdesmäki

Advisors

Dr. Kati Hanhineva

Dr. Jussi Paananen

Aalto University
School of Science

Aalto University
School of Science

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi
Abstract of the master's thesis

Author Anton Mattsson

Title Analysis of LC-MS data in untargeted nutritional metabolomics

Degree programme Life Science Technologies

Major Bioinformatics **Code of major** SCI3092

Supervisor Prof. Harri Lähdesmäki

Advisors Dr. Kati Hanhineva, Dr. Jussi Paananen

Date 12.8.2019 **Number of pages** 52 **Language** English

Abstract

Liquid chromatography-mass spectrometry based untargeted metabolomics is a technique that can measure the levels of thousands of compounds from virtually any biological sample. This thesis was done for the research group of nutritional metabolomics at the University of Eastern Finland. While there exists software for analyzing raw LC-MS data, the output of such software often requires additional preprocessing and quality control procedures that are integral to the workflow of the research group. This thesis covers many of these steps in detail, while also providing a broad overview of metabolomics and LC-MS instrumentation.

The most important steps for curating data that is output from a LC-MS data collection software are drift correction, removal of low-quality features and imputation of missing values. We use cubic spline regression to model and correct for the systematic drift of signal intensity during an LC-MS run. Next, low-quality features are identified using several quality metrics measuring the relative magnitude of analytical variation. Finally, missing values are imputed by predicting them using a random forest fit on the observed part of the dataset.

The main outcome of the thesis is an R package that automates data analysis of LC-MS experiments. The package provides a simple interface for the common preprocessing steps and several statistical analysis techniques for finding the most interesting features of the data, along with an arsenal of visualizations for quality control, exploratory visualization and assessment of study results. The package is licensed under the open source MIT license and is available for anyone to use. In addition, this thesis presents a new algorithm for finding molecular features originating from the same compound.

Keywords Metabolomics, Liquid chromatography , Mass spectrometry, Data analysis, R

Aalto University
School of Science

Aalto-yliopisto, PL 11000, 00076 AALTO
www.aalto.fi
Diplomityön tiivistelmä

Tekijä Anton Mattsson

Työn nimi LC-MS datan analysointi kohdentamattomassa ravitsemuksellisessa metabolomiikassa

Koulutusohjelma Life Science Technologies

Pääaine Bioinformatics **Pääaineen koodi** SCI3092

Työn valvoja Prof. Harri Lähdesmäki

Työn ohjaajat PhD Kati Hanhineva, PhD Jussi Paananen

Päivämäärä 12.8.2019 **Sivumäärä** 52 **Kieli** Englanti

Tiivistelmä

Nestekromatografiaan ja massaspektrometriaan (liquid chromatography-mass spectrometry, LC-MS) perustuvalla kohdentamattomalla metabolomiikalla voidaan mitata tuhansien molekyylien pitoisuuksia lähes mistä tahansa biologisesta näytteestä. Tämä diplomityö tehtiin ravitsemuksellisen metabolomiikan tutkimusryhmään Itä-Suomen yliopistossa. Vaikka ryhmällä on käytössään ohjelmia LC-MS instrumenttien raakadatan käsittelyyn, ohjelmien tulostiedostojen data vaatii usein jatkokäsittelyä ja laadunvalvontaa. Nämä toimenpiteet ovat äärimmäisen tärkeitä metabolomiikkatutkimusten luotettavuuden kannalta. Tämä diplomityö antaa hyvän yleiskuvan metabolomiikasta sekä nestekromatografia-massaspektrometriasta ja käy yksityiskohtaisesti läpi tärkeimmät jatkokäsittelyvaiheet.

Datan laatu varmistamisen kannalta tärkeimmät vaiheet ovat liukuman korjaus, huonolaatuisten piirteiden poisto ja puuttuvien arvojen paikkaus. LC-MS ajon aiheuttama liukuma signaalien vahvuudessa mallinnetaan ja korjataan kuutiosplinireg ressiolla. Huonolaatuisten signaalien tunnistuksessa käytetään useita laatumittareita, jotka mittaavat suhteellista analyyttista mittausvirhettä. Puuttuvat arvot paikataan ennustamalla ne satunnaismetsällä, joka koulutetaan datan havaituilla arvoilla.

Työn tärkein tulos on R-paketti, joka automatisoi LC-MS tutkimusten analyysin. Paketti tarjoaa helppokäyttöisen rajapinnan datan käsittelyyn ja moniin tilastollisiin testeihin. Lisäksi paketissa on tarjolla suuri joukko visualisointifunktioita, joita voi käyttää sekä laadunvalvonnassa, datan tutkimisessa, että tutkimustulosten arvioinnissa. Paketti on lisensoitu avoimen lähdekoodin MIT-lisenssillä, joten se on vapaasti käytettävissä. Lisäksi, työssä esitellään uusi algoritmi saman yhdisteen aiheuttamien molekyylipiirteiden löytämiseen.

Avainsanat Metabolomiikka, nestekromatografia, massaspektrometria, data-analyysi, R

Preface

I want to thank my advisors, my supervisor and my fiancée for supporting me during the time I worked on this thesis. The outcome would not look like the one presented without their valuable input!

Kuopio, 26.7.2019

Anton Mattsson

Contents

Abbreviations

CASMI	Critical Assessment of Small Molecule Identification
CV	Coefficient of variation
DNA	Deoxyribonucleic acid
EIC	Extracted ion chromatogram
ESI	Electrospray ionization
GC	Gas chromatography
HILIC	Hydrophilic interaction chromatography
LC	Liquid chromatography
LOD	Limit of detection
LOESS	Locally estimated scatterplot smoothing
MAD	Median absolute deviation
MIT	Massachusetts Institute of Technology
MS	Mass spectrometry
NMR	Nuclear magnetic resonance
PCA	Principal component analysis
QC	Quality control
QTOF	Quadrupole time-of-flight
RNA	Ribonucleic acid
RPC	Reverse-phase chromatography
RSD	Relative standard deviation
TIC	Total ion chromatogram
TOF	Time-of-flight
t-SNE	t-distributed stochastic neighbor embedding
UEF	University of Eastern Finland

1 Introduction

Metabolomics, the study of metabolic products, is a rising field of study that has the potential of unlocking some of the deepest secrets of cell biology and physiology. Indeed, Patti et al. refer to metabolomics as "the apogee of the omics trilogy" in their review. [Patti et al., 2012] Metabolomics measures the levels of metabolites, small molecules transformed in metabolism to characterize the phenotype of an individual cell or organism. It has been shown that metabolomics is able to uncover links between food consumption and risk of disease, as well as characterize some of the many believed functions of gut microbiota. [Scalbert et al., 2014]

As a relatively new field of study, metabolomics still lacks standardization on many areas: there are significant differences between the procedures employed across research groups. These differences range from sample preparation procedures to instruments and software used to analyze the samples. While variety in vendor providers and competition can increase instrument quality, it is worrying that the results of a metabolomics experiments depend on the software used to process the raw data received from the instrument. At the same time, the field of metabolomics offers great opportunity for data analysis method development. [Aretz and Meierhofer, 2016]

Metabolomics can be divided into two sets of methodologies: targeted metabolomics measures a predefined set of metabolites and untargeted metabolomics aims to measure all observable metabolites in a given samples. [Johnson et al., 2016]

This thesis was written for the research group of food and nutritional metabolomics at University of Eastern Finland (UEF). Our research group uses liquid chromatography-mass spectrometry (LC-MS) based metabolomics to discover links between dietary choices and risk of disease, such as the lowered risk of diabetes associated with consumption of eggs [Noerman et al., 2018] and to analyze composition of food products, such as effects of sourdough fermentation on rye and wheat bread [Koistinen et al., 2018].

Untargeted LC-MS based metabolomics experiments can detect thousands of molecular features that represent metabolites. Thus, efficient preprocessing and analysis of these datasets requires powerful computational methods. [Johnson et al., 2016] In addition, LC-MS instruments are highly sensitive, which makes quality control extremely important, further highlighting importance of automated computational approaches. [Broadhurst et al., 2018]

Our research group has over 10 members working on metabolomics experiments, but has limited resources for data analysis, with only a single dedicated bioinformatician. The goal of the thesis is to improve the data analytical workflow of the research group by automating numerous routine tasks of data preprocessing and analysis as well as providing a tool that is simple to use for people inexperienced with data science and/or programming. In addition to saving time, the new solution should provide better reproducibility of results and facilitate future development. In addition, a new algorithm is developed for clustering molecular features originating from the same compound (see Section 4.2).

This thesis consists of two parts: a literature review and a software package. The literature review presents an overview of metabolomics, followed by a closer review

of data analysis in LC-MS based metabolomics as employed at the research group at UEF. The review emphasizes quality control and the later stages of data analysis of LC-MS metabolomics experiments, while data collection steps performed by external software are described briefly. The software package implements methods described in the literature review in R programming language.

The outline of the thesis is as follows. First, Section 2 offers an introduction to metabolomics covering the general concept, as well as most common applications, especially in nutritional sciences and an overview of the most relevant technological platforms, discussing their strengths and weaknesses. Section 3 covers the principles of liquid chromatography-mass spectrometry and presents some of the major challenges of the platform. Section 4 presents the data analytical processes required to overcome these challenges and presents the new feature clustering algorithm. Section 5 highlights visualization techniques employed in quality control of LC-MS datasets, with emphasis on two commonly used dimensionality reduction techniques, principal component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE). Section 6 discusses the goals and design principles of the R package and presents its most important features. Finally, Section 7 summarizes the thesis with a discussion of the results and their relevance for our research group and broader scientific community.

2 Metabolomics

2.1 Introduction

Metabolomics studies metabolites, small molecules that are substrates and products of metabolism. These include lipids, sugars, steroids, organic acids and phytochemicals among many others. Metabolites are involved in a broad variety of essential cellular activities including energy production, signal transduction and apoptosis. [Johnson et al., 2016] For instance, N-methyltransferase and the metabolic state regulate epigenetic activity in human embryonic stem cells [Sperber et al., 2015]. Some metabolites, such as adenosine triphosphate, acetyl coenzyme A and S-adenosyl methionine can regulate post-translational modifications, which in turn alter protein activity. [Wellen et al., 2009, Nakahata et al., 2008] The spectrum of biochemical activity of metabolites is vast, and studying metabolites can for example reveal the fundamental causes of many diseases [Wishart, 2016].

Metabolomics belongs to the omics family of analytical chemistry methods, where the other most relevant omics are genomics, transcriptomics and proteomics (see Figure 1). The omics technologies roughly follow the central dogma of molecular biology, which states that information primarily flows from DNA to DNA or RNA and from RNA to proteins [Crick, 1970]. Genomics studies the genes that constitute the genome, transcriptomics studies the mRNA molecules to discover how the genes are transcribed, while proteomics studies proteins that are translated from the mRNA molecules. [Debnath et al., 2010] As the fourth part of the omics family, metabolomics takes one step further towards the phenotype and focuses on profiling of metabolites that are the product of enzymatic activity of proteins [Johnson et al., 2016]. Wishart goes as far as to state that genomics tell us what might happen, whereas metabolomics tells us what is actually happening

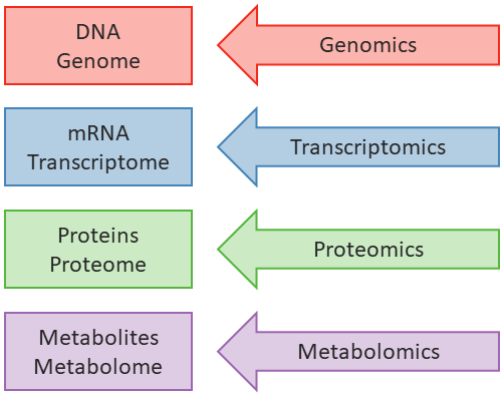


Figure 1: A diagram depicting the omics technologies and their objects of study

[Wishart, 2016]. However, as Karczewski & Snyder point out in their review, integration of multiple omics data is crucial to understand the fundamental mechanisms of common but complex diseases such as diabetes, obesity, autism and schizophrenia [Karczewski and Snyder, 2018].

The collection of metabolites found in an organism or a tissue under study is referred to as its metabolome. Metabolomes of living organisms from bacteria to humans consist of both endogenous compounds that originate from the organism itself and exogenous compounds that originate from outside the organism. Human metabolome, for example, contains exogenous compounds from micro-organisms such as gut microbiota, as well as dietary compounds and xenobiotic compounds such as drugs. [Johnson et al., 2016] Wishart (2016) summarizes the metabolome as a fusion of downstream output from the genome and upstream input of the environment [Wishart, 2016].

The metabolomes of distinct organisms tend to overlap, especially in the endogenous part. Both humans and bacteria need nucleotides, amino acids and fatty acids to construct DNA, RNA, proteins and membranes that are essential to all cellular activity. In addition, nearly all organisms either consume or produce glucose. Thus, there are approximately 100 compounds that are highly abundant in nearly all organisms: amino acids, nucleotides and intermediates of cellular respiration. [Jang et al., 2018] Despite the similarities, metabolomes often express great inter-individual variation, mostly quantitative, but also qualitative, which poses a major challenge in metabolomics studies, especially in human studies [Johnson et al., 2016]. This variation is due to factors like sex, genetic variation, age and health status, along with dietary habits, which have the strongest impact on human metabolome. There are over 25 000 known compounds in different foods, of which most are further metabolized by the human metabolism. Thus, dietary habits influence the human metabolome simply by adding metabolites from the metabolomes of the organisms humans consume, in addition to the changes in human metabolism affected by metabolites originating from food. [Scalbert et al., 2014]

The current understanding of human metabolism, summarized as a map of interconnected chemical reactions is already complex, but believed to be far from complete [Johnson et al., 2016, Wishart, 2016]. As a measure of the scale of human

metabolome and the recent advances in database coverage, the latest version Human Metabolome Database released in 2018 currently features over 114 100 compounds, in comparison to 2180 compounds in 2007, 6408 compounds in 2009 and 40 153 compounds in 2013 [Wishart et al., 2018].

2.2 Applications

Metabolomics analyses can deal with virtually any type of biological samples: tissues, biofluids such as urine or plasma, individual cells, biological waste or even breath and fumes, *in vivo* or *in vitro* [Johnson et al., 2016, Wishart, 2016]. The ability to study localized effects can reveal how drugs are metabolized in specific organs or provide crucial information about disease mechanisms. For example, a clever way to analyze the metabolism of an organ is to conduct a paired analysis of arterial blood samples (blood entering the organ) and venous blood samples (blood exiting the organ), which can be used to construct an input-output model of the organ. Additionally, the ability of metabolomics to dive to cellular and even sub-cellular level can prove crucial in heterogeneous tissues such as the brain and tumors. [Johnson et al., 2016]

The advances in both analytical equipment and computational tools has established metabolomics as a widely used tool in biomarker discovery [Johnson et al., 2016]. Koeth et al. were able to identify plasma trimethylamine N-oxide as a marker of cardiovascular disease [Koeth et al., 2013], and urinary taurine was linked to ionizing radiation by Pannkuk et al. [Pannkuk et al., 2015]. To reach beyond biomarker discovery, modern biomedical studies first utilize metabolomics to associate metabolites and pathways with certain clinical conditions and validate results with functional and mechanistic studies to provide more information about the underlying mechanisms of the condition. This often proves challenging, as discovering the biological role of a metabolite requires researchers to be able to figure out how the metabolite is connected to other compounds in metabolic pathways. Therefore, metabolomics is viewed as an important tool in biomedicine: it offers a broad view over the mechanisms of metabolism. [Johnson et al., 2016]

Nutritional metabolomics is a field with a goal to link metabolites to intake of specific diets and discover relations between dietary habits and health. For example, a recent study revealed potential metabolites that link high egg consumption to lowered risk of type 2 diabetes [Noerman et al., 2018], while another studied the relationship between Nordic diet and type 2 diabetes [Shi et al., 2018]. Animal studies can also be used, as in the study by Airaksinen et al. revealing high-fat diet inducing changes in carnitine and lipid metabolism in adipose tissue in mice [Airaksinen et al., 2018]. Nutritional metabolomics also studies composition of food products under different conditions, such as the effect of sourdough fermentation on rye and wheat composition [Koistinen et al., 2018] and differences between metabolic profiles of organic and conventionally grown strawberries [Kårlund et al., 2015]. Nutritional metabolomics, and metabolomics in general is regarded as an important future step in personalized and preventive medicine [Wishart, 2016]

The field of metabolomics can be divided into two main methodologies: untargeted and targeted metabolomics. In targeted metabolomics, the experimentation measures

the levels of a predefined set of metabolites. The advantages of targeted approaches are higher sensitivity and selectivity as well as a higher level of quantification. Untargeted metabolomics aims to measure every detectable metabolite from any given sample, with no a priori assumptions about the findings and no predefined set of metabolites to be measured. This allows for novel and unanticipated findings as well as an unbiased view over multiple metabolic pathways. However, untargeted approaches result in complex datasets that require advanced computational tools for identification of compounds and correlating signals across samples. In addition, it is not possible to measure and identify all metabolites with current technology. These limitations are a major challenge of untargeted approaches, and will be addressed in more detail in Section 3.3 Targeted approaches can be used to validate and expand upon results from untargeted experiments. [Johnson et al., 2016]

2.3 Technologies

Three technologies have emerged as the principal tools of metabolomics: nuclear magnetic resonance (NMR) spectroscopy, gas chromatography-mass spectrometry (GC-MS) and liquid chromatography-mass spectrometry (LC-MS). Table 1 summarizes key advantages and disadvantages of these technologies.

A major advantage of NMR compared to MS methods is its high scalability: the cost of NMR analysis per sample is a fraction of that of an LC-MS analysis. This is in part due to fact that NMR experiments require smaller amount of sample preparation. In addition, NMR is faster and more robust than MS, especially LC-MS, which makes it poised for use in a clinical setting, while LC-MS is currently mainly limited to research settings. In addition, the ability of NMR to quantify the concentration of metabolites and identify the structure of unknown compounds is unparalleled by MS based methods. [Markley et al., 2017] The high scalability and robustness of NMR is well demonstrated by the launch of a commercial, consumer-targeted NMR-based blood test by Nightingale Health and Aava Medical Centre in the end of the year 2018 [Aava Medical Centre, 2018, Nightingale Health, 2018]. However, mass spectrometry, especially LC-MS, offers superior sensitivity to NMR, meaning that metabolites with lower concentrations can be detected. This level of sensitivity enables LC-MS to detect a much larger spectrum of compounds than NMR and has lead to LC-MS being the most popular platform in modern metabolomics. [Markley et al., 2017] GC-MS offers a compromise between robustness and sensitivity, and is the only technology that is able to analyze gases, which makes it the only choice when analyzing samples from breath or fumes. [Wishart, 2016]

Due to the large differences in the capabilities, the technologies are considered complementary instead of purely competitive, and combining NMR and MS can provide both increased coverage of the metabolome and more accurate identifications compared to what either of the technologies could achieve alone [Marshall and Powers, 2017].

To summarize: Metabolomics is a versatile tool for analyzing the chemical composition of multiple types of biological samples. Untargeted analysis has the greatest potential of unbiased metabolic phenotyping. Despite its weaknesses,

Technology	Advantages	Disadvantages
NMR	• Quantitative • Non-destructive • Fast (2–3 min per sample) • Requires no derivatization • Requires no separation • Detects most organic classes • Allows identification of novel chemicals • Most spectral features are identifiable • Robust, mature technology • Can be fully automated • Compatible with liquids and solids • Long instrument lifetime (over 20 years)	• Not sensitive (LOD = 5 µM) • High start-up cost (> 900 000 €) • Large instrument footprint • Cannot detect or identify salts and inorganic ions • Cannot detect non-protonated compounds • Requires larger sample volumes (0.1–0.5 mL)
GC-MS	• Robust, mature technology • Modest start-up cost (~ 140 000 €) • Quantitative (with calibration) • Modest sample volume (0.1–0.2 mL) • Good sensitivity (LOD = 0.5 µM) • Large body of software and databases for metabolite identification • Detects most organic and some inorganic molecules • Excellent separation reproducibility • Many spectral features are identifiable • Can be mostly automated • Compatible with gases and liquids	• Destructive (sample not recoverable) • Requires sample derivatization • Requires separation • Slow (20–40 min per sample) • Not compatible with solids • Novel compound identification is difficult
LC-MS	• Superb sensitivity (LOD = 0.5 nM) • Very flexible technology • Detects most organic and some inorganic molecules • Small sample volumes (10–100 µL) • Can be done without separation (direct injection) • Has the potential to detect the largest portion of metabolome • Can be mostly automated • Compatible with solids and liquids	• Destructive (sample not recoverable) • Not very quantitative • Higher start-up cost (> 250 000 €) • Slow (15–40 min per sample) • Usually requires separation • Poor separation resolution and lower reproducibility versus GC-MS • Less-robust instrumentation than NMR or GC-MS • Not compatible with gases • Most spectral features are not yet identifiable • Novel compound identification is difficult • Short instrument lifetime (<9 years)

Table 1: Table of advantages and disadvantages of NMR, GC-MS and LC-MS. LOD: limit of detection. Adapted from [Wishart, 2016]

LC-MS is the most common technology used in untargeted metabolomics thanks to its unrivaled sensitivity. For the rest of this thesis, the focus will be on untargeted LC-MS metabolomics and ways to deal with the weaknesses of the current LC-MS equipment and software.

3 Liquid chromatography-mass spectrometry

LC-MS is a coupling of liquid chromatography and mass spectrometry. Liquid chromatography is used to separate the metabolites of a sample by their polarity, while the mass spectrometer is responsible for measuring the mass and abundance of each metabolite. A mass spectrometer is typically composed of three parts: an ion source is used to ionize the metabolites, a mass analyzer is used to measure the mass-to-charge ratio of metabolites and finally a detector is used to measure abundance of metabolites. [Zhou et al., 2012] Applying LC before MS has the advantage of separating isomers that would otherwise be indistinguishable and minimizes ion suppression, where a more easily ionizable species masks the presence of a less ionizable one [Berg et al., 2013]. The diagram in Figure 2 showcases the principle of LC-MS. The different parts of LC-MS equipment are presented in further detail below.

3.1 Liquid chromatography

The principle behind chromatography is that molecules that are dissolved in liquids can also be absorbed onto or interact with solids (for example, molecules dissolved in coffee can stain cloth). Furthermore, molecules are absorbed by solid surfaces based on their chemical properties such as size and polarity. In liquid chromatography, the goal is to separate metabolites based on their polarity. This is achieved by passing the solution through a column that consists of two parts: a solid stationary phase and a liquid mobile phase. [Bird, 1989]

In the most classic setting, normal phase liquid chromatography, a polar stationary phase and initially hydrophobic mobile phase are used and as a result, polar metabolites are absorbed by the stationary phase and hydrophobic metabolites are

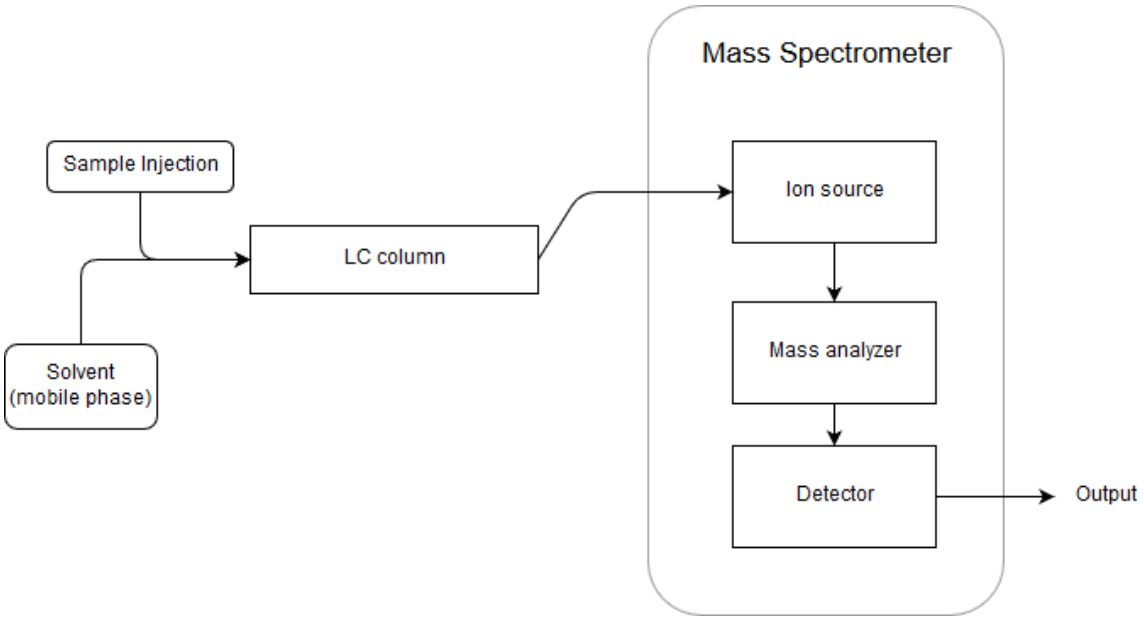


Figure 2: A diagram of the principle of LC-MS

quickly washed out of the column. Next, the polarity of the liquid phase is gradually increased and metabolites start to dissolve back to the liquid phase, so that the most polar metabolites are eluted last. This technique is called gradient elution and is achieved by mixing together polar and hydrophobic solvents. The time that a metabolite spends in the column is referred to as retention time. Reverse-phase (RP) LC is - as the name suggests - a reverse method of normal phase LC: a hydrophobic solid phase and an increasingly hydrophobic liquid phase are used to separate hydrophobic compounds by their polarity. [Bird, 1989]

In modern LC-MS based metabolomics, two different columns are commonly used to be able to separate both hydrophilic and hydrophobic metabolites. RP chromatography is used to separate hydrophobic metabolites such as phenolic acids, flavonoids, glycosylated steroids, alkaloids, and other glycosylated species. Polar compounds like sugars, amino sugars, amino acids, vitamins, carboxylic acids, and nucleotides are usually separated by hydrophilic interaction chromatography (HILIC) instead of normal phase LC since HILIC is more compatible with electrospray ionization (ESI) used in the mass spectrometer. [Zhou et al., 2012, Hemström and Irgum, 2006] HILIC is a slightly more complicated chromatography method that uses hydrophilic stationary phases with reversed-phase type liquid phase consisting of 5-40% water and a water-miscible organic solvent (a solvent that forms a homogenous solution with water in any concentration) such as acetonitrile. The polar stationary phase will cause the liquid phase to separate into a stagnant water layer and a more mobile layer consisting of the bulk organic solvent. The partition of metabolites by their polarity will then occur mainly in the liquid-liquid interface rather than on the solid-liquid interface between the stationary and the mobile phase. [Hemström and Irgum, 2006]

3.2 Mass spectrometry

Ion source The purpose of an ion source is to ionize the metabolites to be measured, since all mass spectrometers require the metabolites to have a charge - neutral molecules cannot be measured. Multiple techniques of ionization have been developed for mass spectrometry, namely electrospray ionization (ESI), atmospheric pressure chemical ionization, atmospheric pressure photoionization, and fast atom bombardment. Of these, ESI is by far the most used method in metabolomics, since its "soft ionization" produces a large number of ions and often preserves the metabolites intact, which aids initial identification. [Zhou et al., 2012]

In electrospray ionization, the samples are first pumped through a thin charged metal capillary and nebulized at the tip of the capillary, forming a spray of small charged droplets. The droplets are subsequently evaporated, and the residual charge is transferred to the metabolites to be analyzed. The ionized metabolites are then transferred to the mass analyzer in high vacuum via a series of small apertures and focusing voltages. ESI instrumentation can be set to detect either positive or negative ions and switching between positive and negative ionization mode within an analytical run is possible. [Pitt, 2009] In LC-MS, samples are usually analyzed in both positive and negative ionization mode, since they provide complementary information as some metabolites often have a tendency to ionize either positively or negatively [Berg et al., 2013, Zhou et al., 2012]

Mass analyzer The task of a mass analyzer is to determine mass-to-charge ratios (m/z) of metabolites, which can be achieved with a variety of techniques. Unlike in ionization, there is no clear favorite in mass analyzers for metabolomics, but different techniques are used and combined by different vendors. [Zhou et al., 2012]

A quadrupole analyzer is composed of four parallel metal rods. By combining constant and varying voltages, the instrument can permit only ions with a narrow band of m/z values to pass along the axis of the rods at a time. Varying the voltages allows scanning across a range of m/z values. Quadrupole analyzers can also be used as collision cells, where ions are fragmented by collisions with an inert gas such as nitrogen or argon. Analyzing the fragments in addition to the initial ion yields a so-called MS/MS or MS^2 spectrum. The fragmentation patterns facilitate metabolite identification. A quadrupole analyzer with MS^2 capabilities features three separate quadrupoles, with the second functioning as a collision cell and is called a triple quadrupole mass spectrometer. [Pitt, 2009]

Time-of-flight (TOF) analyzers accelerate ions through a high voltage in a flight tube. The m/z value of an ion affects its velocity in the high voltage, and thus the time taken to travel down the flight tube to reach the detector. By pulsing the initial accelerating voltage, the output of the detector as a function of time can be converted into a mass spectrum. TOF analyzers achieve a higher sensitivity and mass accuracy compared to quadrupole analyzers. [Pitt, 2009]

Ion trap analyzers first trap ions in a space confided by tree hyperbolic electrodes using static and varying voltages. Next, they eject ions towards the detector in a sequence based on their m/z values, creating a mass spectrum. An ion trap can also

be used to fragment specific ions, while ejecting the rest, which allows analysis of fragmentation patterns in MS^2 spectra. [Pitt, 2009]

Different mass analyzers can be combined into a tandem mass spectrometer. A common example is the combination of quadrupole and TOF analyzers, where the third quadrupole of a triple quadrupole mass spectrometer is replaced by a TOF analyzer. This combination is called a hybrid quadrupole time-of-flight (QTOF) mass spectrometer. [Zhou et al., 2012, Pitt, 2009] The data used in this work results from experiments conducted at the Metabolomics Center of UEF, which has both a QTOF-based mass spectrometer and an ion trap-based mass spectrometer. In most projects, the samples are analyzed in four different settings: the metabolites are separated using both a HILIC and an RP column, and ionized in both positive and negative mode ESI.

3.3 Current challenges

Some of the challenges in LC-MS metabolomics are directly related to the innate properties of the LC-MS instrument itself. First, LC-MS is only able to detect metabolites that can be ionized: molecules that cannot be charged remain undetected. The ionization procedure also presents a problem related to the level of quantitativity of results: the concentration of the metabolites is not measured directly, but a linear relation between concentration and the signal intensity measured by the MS detector is assumed. Thus, LC-MS data only allows comparison of concentrations of the same metabolite across samples, not comparison of concentrations across metabolites, as some metabolites are more easily ionized than others. [Zhou et al., 2012, Aretz and Meierhofer, 2016] Furthermore, signal intensity does not always scale linearly with metabolite concentration. Deviations from linearity are mainly caused by changes in LC column properties or the so-called ion suppression effect, where a less volatile compound hinders evaporation of droplets in the ESI phase, and thus suppresses ions of other compounds with similar retention time. [Berg et al., 2013]

LC-MS instruments often present a systematic drift in signal intensities during longer runs. The drift effect is caused by changes in the column conditions, contaminations and varied environmental conditions such as temperature and humidity. Due to the strong effect of instrument and environmental conditions, analysis of same samples on different periods of time can yield different results. This is problematic for studies with a large sample size (> 200), since data from multiple LC-MS runs might not be fully comparable. In addition to signal intensities, there is observable drift in metabolite retention times and, to a lesser extent, in mass-to-charge ratios. [Zhou et al., 2012, Berg et al., 2013]

Analytical variability in LC-MS data can be assessed and accounted for using quality control (QC) samples. The most common strategy is to create QC samples by pooling together a small aliquot of all samples to be studied. This pooled sample is then injected multiple times in the beginning of an experiment to stabilize the column, and subsequently injected on a regular interval. These regular injections of an identical QC sample allow the assessment of both systematic (drift effect) and

unsystematic analytical variability and many approaches to account for both have been developed. [Zhou et al., 2012, Broadhurst et al., 2018]

Other issues of LC-MS based metabolomics are caused by the complexity of the biological samples. Many metabolites have a tendency to capture adducts, including cations such as ammonium, sodium and potassium ions, as well as anions such as formate or acetate ions. This naturally affects mass of a portion of the molecules, resulting in multiple peaks detected by the LC-MS instrument that actually originate from the same metabolite. These extra peaks can cause problems in the data analysis phase, especially in metabolite identification. [Pitt, 2009] In addition, the redundant peaks aggravate the multiple testing problem in statistical analysis, where p-values of hypothesis tests are corrected using methods such as the Benjamini-Hochberg false discovery rate procedure. This in turn lowers statistical power, resulting in an inflated type II error rate.

Identification of metabolites is perhaps the greatest challenge of LC-MS based metabolomics. Identification of compounds currently requires extensive manual work, despite of efforts to automate the task. [Patti et al., 2012] The primary tool for automatic metabolite identification is matching MS/MS spectra against spectral databases, either local databases constructed using chemical standards or public databases such as METLIN [Guijas et al., 2018] or Human Metabolome Database [Wishart et al., 2018]. Unfortunately, the identification of metabolites from public databases is hindered by the lack of standardization. LC-MS experiments are carried out using a wide variety of instrumentation and software, and the fragmentation pattern and retention time of a metabolite might vary across platforms. Thus, identifications from public databases are not completely reliable. [Aretz and Meierhofer, 2016]. In the recent years, there have been efforts to tackle the identification problem with machine-learning based algorithms, motivated in part by the Critical Assessment of Small Molecule Identification (CASMI) challenge organized by a collaboration of researchers around the world [Nikolic et al., 2018]. However, the performance of these algorithms is still far from the stage where the top candidate proposed by an algorithm could be used as a putative identification. For example, a collaboration of researchers from Friedrich-Schiller University and Aalto University have dominated the challenge of automatic structural identification using only in silico fragmentation [Nikolic et al., 2018], but the latest algorithm proposed by the collaboration in 2018 only achieves an accuracy of nearly 25% on the first candidate [Bach et al., 2018]. Thus, while many tools can help reduce the manual workload, they are still far from replacing it.

4 Preprocessing LC-MS data

4.1 From raw data to peak table

The output of an LC-MS experiment consists of raw data files, containing MS spectra. The data has three dimensions: retention time, mass-to-charge ratio and signal intensity. A data collection software is used to produce a so-called peak table or a peak list from the raw data. Instead of continuous spectra, a peak table has only one numerical value, called abundance, for each peak in each sample. Some peaks are then combined into one molecular feature. As discussed in the previous section, a single metabolite can result in multiple peaks. Thus, the features of the resulting dataset are referred to as "molecular features" instead of "metabolites". [Zhou et al., 2012] Conversion of raw data files to a peak table is commonly conducted using software provided by the LC-MS vendor, or using open-source tools such as MZmine 2 [Pluskal et al., 2010], XCMS [Smith et al., 2006] or MS-DIAL [Tsugawa et al., 2015], which is commonly used at the UEF.

Signal filtering The starting point of the processing of raw data are total ion chromatograms (TIC), the total signal intensity over all m/z values as a function of retention time. First, the signal intensity of TICs is filtered to remove small magnitude noise by smoothing the signal. This can be achieved using moving median or mean filters, or a more sophisticated option, a Savitzky-Golay filter, which fits local high-order polynomials to the signal. The requirement for a filter is to be able to smooth the signal without eliminating relevant peaks. Next, a baseline correction algorithm is used to correct for a baseline shift, where the intensity of the baseline noise increases as a function of retention time. Similarly to signal filtering, a low-order Savitzky-Golay filter can be employed to eliminate the baseline shift.

The top centre of Figure 3 represents the baseline correction on a TIC. Before any further preprocessing steps, the total ion chromatograms are split into small intervals of m/z values, retaining the chromatogram of signal intensity as a function of retention time for each m/z window. These parts of the dataset are called extracted ion chromatograms (EIC). [Zhou et al., 2012] Traditionally, EICs were constructed using fixed-width bins of m/z values. Modern algorithms make use of the density function of signal intensities over m/z values to construct EICs that correspond to peaks in signal intensity. [Zhou et al., 2012, Tsugawa et al., 2015, Myers et al., 2017]

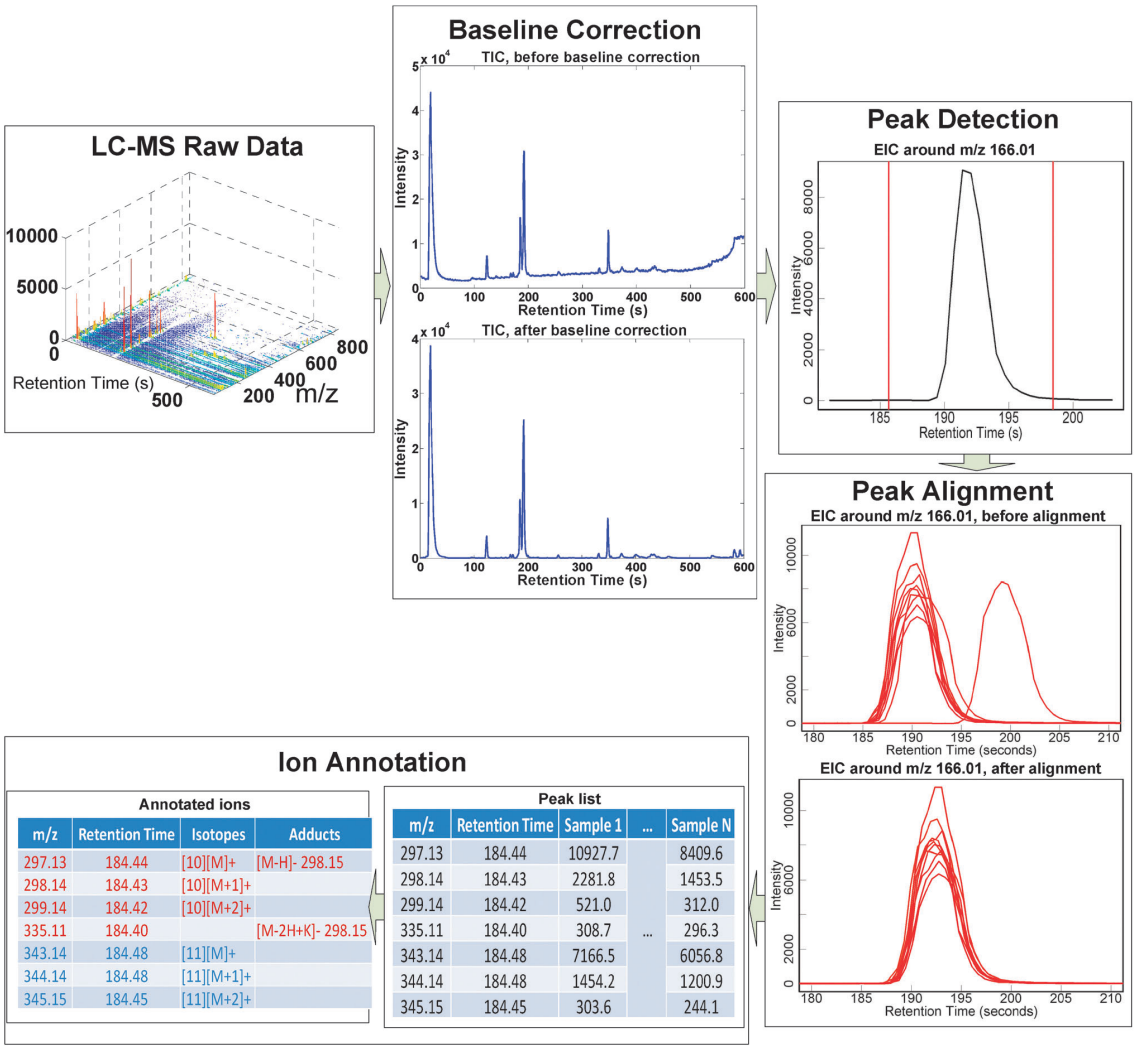


Figure 3: A diagram of the preprocessing process of raw LC-MS data. Reproduced with permission from [Zhou et al., 2012]

Peak detection Perhaps the most critical part of the data collection procedure is the peak detection phase, as errors in peak detection propagate trough the complete data analysis phase up to statistical analysis, metabolite identification and biological interpretation [Myers et al., 2017]. The goal is to scan EICs and identify peaks in

the signal intensity that correspond to individual molecular features. Separating peaks from background noise is a difficult task, as the edges of peaks are often not well defined. Various algorithms for peak detection have been developed and are available in proprietary vendor software and open-source tools. [Zhou et al., 2012] For example, MS-DIAL detects peaks based on changes in signal amplitude and first and second derivatives. Edges of peaks are identified where both signal amplitude and first derivative exceed set thresholds. The top of a peak is identified when the sign of the first derivative changes and the second derivative exceeds a threshold. The thresholds to use can be computed from the data using preset formulas or set manually. [Tsugawa et al., 2015] Myers et al. showed that results from different peak picking software can differ significantly [Myers et al., 2017]. Thus, it is common practice at UEF to manually inspect the spectra of most interesting peaks before biological interpretation of study results.

Peak matching and alignment The retention time of an ion can shift between samples, even if they are analytical replicates. After the peaks in individual samples have been identified, there is a need to match and align the peaks across samples, so the abundances of the corresponding features can be compared (see bottom left of Figure 3). Again, many different approaches have been developed to match similar peaks across samples and correct for the retention time shift. After peak matching and alignment it is common that peaks present in only a small subset of samples are regarded as noise and discarded. [Tsugawa et al., 2015, Zhou et al., 2012]

Ion annotation As discussed before, a single metabolite can be represented by multiple peaks in LC-MS data. LC-MS software provide tools to account for redundant peaks caused by isotopes, dimers and adducts including cations such as ammonium, sodium and potassium ions, as well as anions such as formate or acetate ions. Two peaks originating from the same compound often have similar-shaped profiles as function of retention time. Thus, peaks are suspected to originate from the same compound if their EICs are correlated, and their mass difference matches the mass of a neutron (isotopes) or a known adduct or if the mass of one of the peaks is doubled (dimer). These peaks are then combined into one molecular feature, resulting in a complete peak table. [Zhou et al., 2012]

4.2 Novel method for combining molecular features

Unfortunately, the procedures for identifying peaks originating from the same metabolite implemented in current software are not perfect but fail to detect some peaks originating from the same compound. This section presents an additional algorithm for detecting molecular features originating from the same compounds. The development of the algorithm was done in co-operation with David Broadhurst, Professor of Data Science & Biostatistics in the School of Science, and director of the Centre for Integrative Metabolomics & Computational Biology at the Edith Covan University.

Features originating from the same compound are assumed to be strongly correlated across samples and have a small difference in their retention time. This

motivates the first step of the algorithm: the algorithm identifies pairs of correlated features within a specified retention time window. Both the correlation threshold and the size of the retention time window are specified by the user. For illustration, a correlation coefficient threshold of 0.9 and a retention time window of ±1 second is used. Pearson correlation coefficient is used, as the relationship between features originating from the same compound is assumed linear.

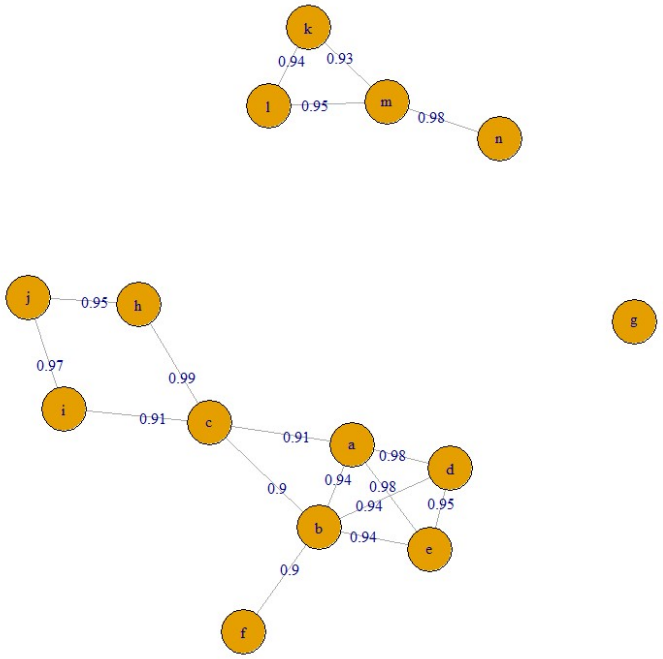
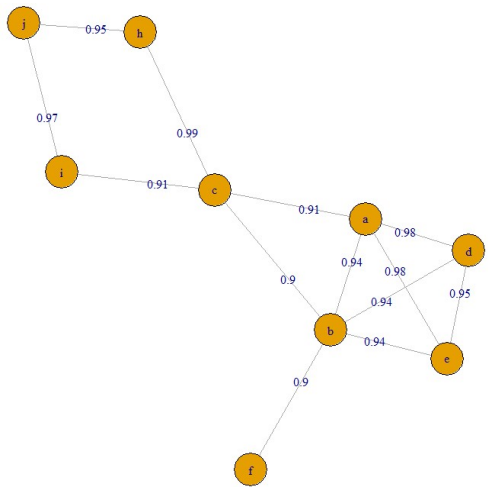
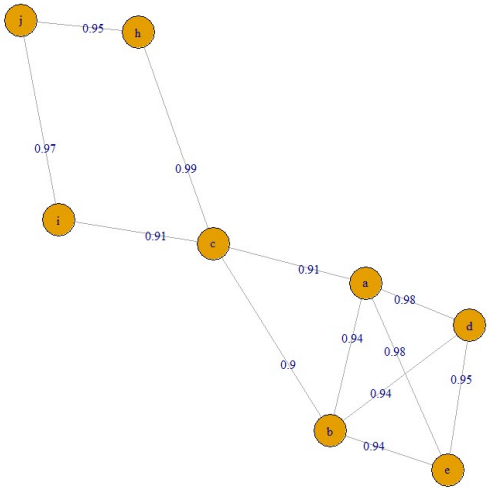


Figure 4: A sample network of intercorrelated features

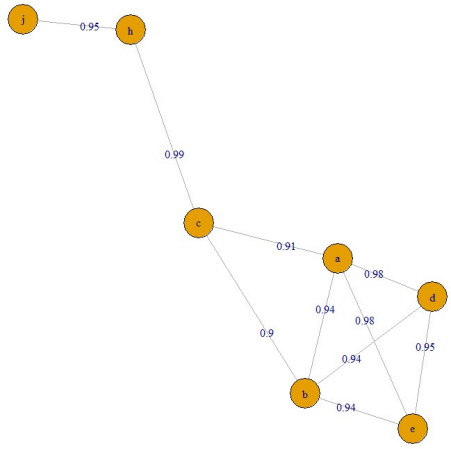
Next, an undirected graph of all the connections between features is generated, where each node represents a feature, each edge an aforementioned connection and edge weight the corresponding Pearson correlation coefficient, see Figure 4 for an example. The graph is then decomposed to its connected components, groups of nodes where all the nodes of the component are reachable from any other node. The components are then pruned, i.e. nodes are removed from the component until all the nodes have a sufficiently high degree (the number of edges of the node). This step requires a third user-defined parameter, degree threshold, defined as a percentage of the maximum possible degree. For example, in a component of five nodes, the maximum degree is 4. With a degree threshold of 0.8, each node is required to have at least $0.8 \cdot 4 = 3.2 \approx 3$ edges (the number of edges required is rounded to the nearest integer). If this criterion is not met, the node with the lowest degree is discarded until the criterion is met. In the case of a tie, the node with the lowest sum of edge weights is discarded. Note that nodes that are initially discarded can form new clusters among themselves, and single nodes can for a "cluster" of one node. Figure 5 illustrates the process of the algorithm on the largest component of the graph in Figure 4. Figure 6 shows the state of the graph after clustering, with each final cluster colored differently.



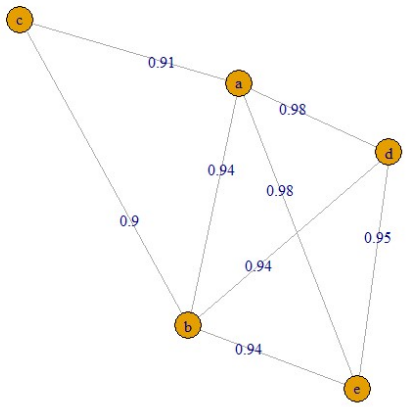
(a) The original cluster, clearly node f will be dropped.



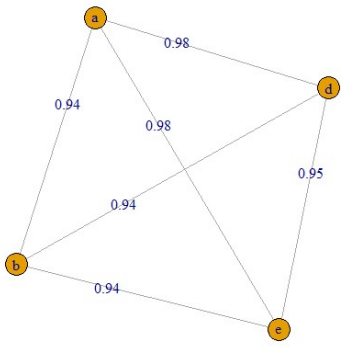
(b) Nodes i, j and h all have degree 2. Since node i has the lowest sum of edge weights (correlations), it will get dropped next.



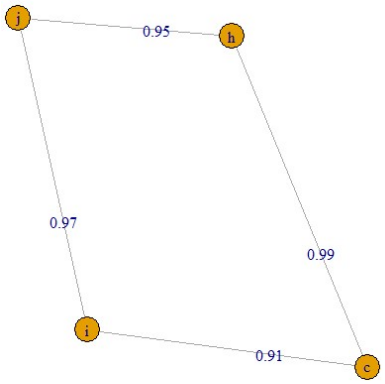
(c) Nodes j and h are the next nodes to be dropped.



(d) Node c will get dropped next, as it has the lowest degree.



(e) Finally, each node is connected to all other nodes, so the cluster is finished.



(f) The dropped nodes will form another cluster, since the required threshold for the degree $0.8 \cdot 3 = 2.4 \approx 2$ is fulfilled for each node.

Figure 5: Progress of the peak cluster algorithm with degree threshold set to 0.8.

After the clustering, the feature with the largest median peak area is retained for each cluster. All the features that are clustered together are recorded for future reference.

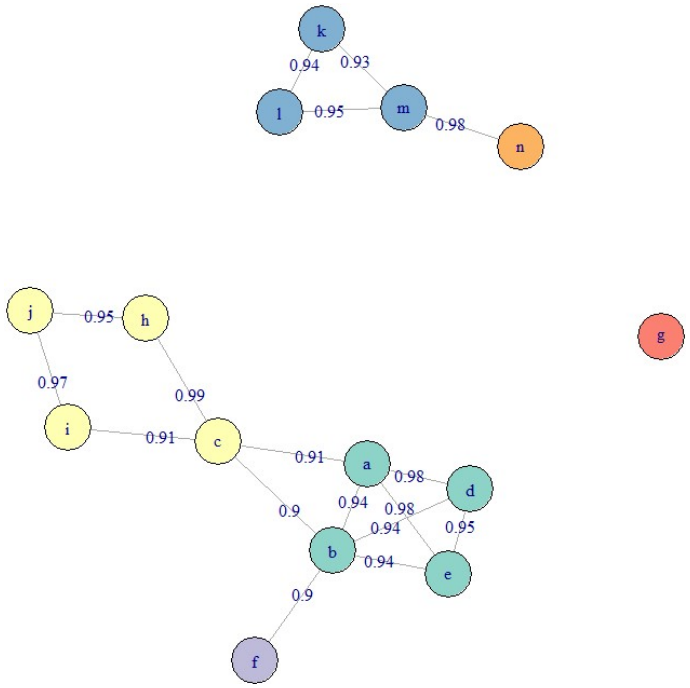


Figure 6: The sample network from Figure 4, with the final clusters indicated by color. Degree threshold was set to 0.8.

Testing Adjusted Rand index was used as a performance metric of the clustering. The benefit of adjusted Rand index is that, unlike many other metrics of clustering performance, it takes into account the amount of correct classifications that arise by chance. [Hubert and Arabie, 1985] First, we need to build a contingency table of the clustering given by the algorithm $X = \{X_1, X_2, \ldots, X_r\}$ and the true clustering $Y = \{Y_1, Y_2, \ldots, Y_s\}$, where X_1 denotes the features that are clustered to cluster number 1, n_{ij} denotes the number of features that belong to both X_i and Y_j. The contingency table is then of following form:

X	Y_1	Y_2	$\ldots$	Y_s	Sums
X_1	n_{11}	n_{12}	$\ldots$	n_{1s}	a_1
X_2	n_{21}	n_{22}	$\ldots$	n_{2s}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
X_r	n_{r1}	n_{r2}	$\ldots$	n_{rs}	a_r
Sums	b_1	b_2	$\ldots$	b_s	

The contingency table values are then used to compute the adjusted Rand index:

$$\text{Adjusted Rand Index} = \frac{\overbrace{\sum_{ij}\binom{n_{ij}}{2}}^{\text{Index}} - \overbrace{\left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right] / \binom{n}{2}}^{\text{Expected Index}}}{\underbrace{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right]}_{\text{Max Index}} - \underbrace{\left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right] / \binom{n}{2}}_{\text{Expected Index}}}$$

A value of 1 signifies perfect clustering, while a value of 0 means that the clustering did not perform better than was expected by chance. Adjusted Rand index can also yield negative values, if the clustering performs worse than random guessing. Note that the cluster indexes do not need to match: if two features are both clustered to cluster "2", and belong to the true cluster "4", they are assumed to be correctly clustered. [Hubert and Arabie, 1985]

To evaluate the performance of the algorithm, the clustering method was applied to two LC-MS metabolomics datasets: a study featuring bread samples of breads of different cereals [Koistinen et al., 2018] and a human study with plasma samples [Noerman et al., 2018]. The cereal dataset contained 26287 features, of which 276 were identified, while the plasma dataset contained 3688 features, of which 432 were identified. The features were identified by a human expert, and not all identifications have a high level of certainty. This can reduce the credibility of the testing procedure.

Features that 1) are measured using the same column and ionization mode and 2) originate from the same metabolite are the most interesting. For conciseness, these features are referred to as "duplicated features", and correspond to the "true positive" class. The cereal dataset and plasma dataset contained 19 and 47 duplicated features, respectively (see Table 2). The low proportion of identified compounds is common in LC-MS datasets, since identification requires extensive manual labour. The small number of identified compounds, and the even smaller number of identified features that originate from the same metabolite inevitably presents problems in testing the performance of the algorithm.

Dataset	Features	Identified	Duplicated
Cereal	26287	276	19
Plasma	3688	432	47

Table 2: The numbers of features in the two datasets used for testing feature clustering

The clustering algorithm was applied separately to the four parts of the complete datasets, and the performance was evaluated by computing clustering performance metrics on the identified part of the datasets (see Figure 7). The parameters of the algorithm were set to initial trial values using previous knowledge. Correlation coefficient threshold was set to 0.9, retention time window to 1 second and degree

threshold to 80% of the maximum degree. In addition to the initial guess, other parameter values were tried using random search for parameter values in reasonable intervals.

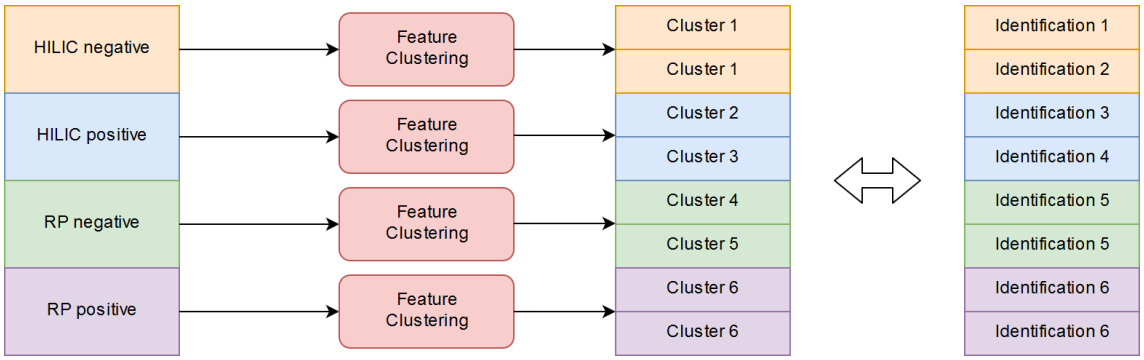


Figure 7: A diagram of the testing procedure of the feature clustering algorithm. The clustering is performed separately on the four parts of the dataset (HILIC and RP columns and positive and negative ionization). After clustering, the features are combined, and performance is evaluated by comparing the clustering to the original identifications.

For the cereal dataset, the performance of the algorithm was satisfactory. The default parameters yielded an adjusted Rand index of 0.618 and the algorithm correctly clustered 75.6% of the duplicate features. Unfortunately, the algorithm does seem to provide multiple false positives as well, as only 61% of the features that are clustered together actually originate from same compounds. The random search was conducted for 40 combinations of the parameters, with the correlation coefficient threshold varying between 0.8 and 0.95, retention time window between 0.5 and 2 seconds and degree threshold between 70% and 90%. None of the alternative parameter settings outperformed the default values, but no clear patterns between performance and parameter values were observed.

The results on the plasma samples were completely unsatisfactory, though. The default parameters resulted in an adjusted Rand index of 0, the algorithm failed to cluster all the duplicated features and all the features that were clustered together originated from different compounds. This raises the question: how can the performance vary so greatly between datasets? The answer lies in the distribution of retention time differences and correlation coefficients between pairs of duplicated features, depicted in Figure 8 for the plasma dataset. It is easy to observe that both the retention time difference and correlation coefficient vary greatly, and there are few pairs of features that have a high correlation coefficient and low retention time difference.

To further investigate the duplicated features in the datasets, the retention time difference and the correlation coefficient between all the duplicated features were computed and visualized in Figure 9. Clearly the duplicated features in the cereal dataset follow the assumptions behind the algorithm much better. The differences between the datasets can be explained by the different nature of the study designs and

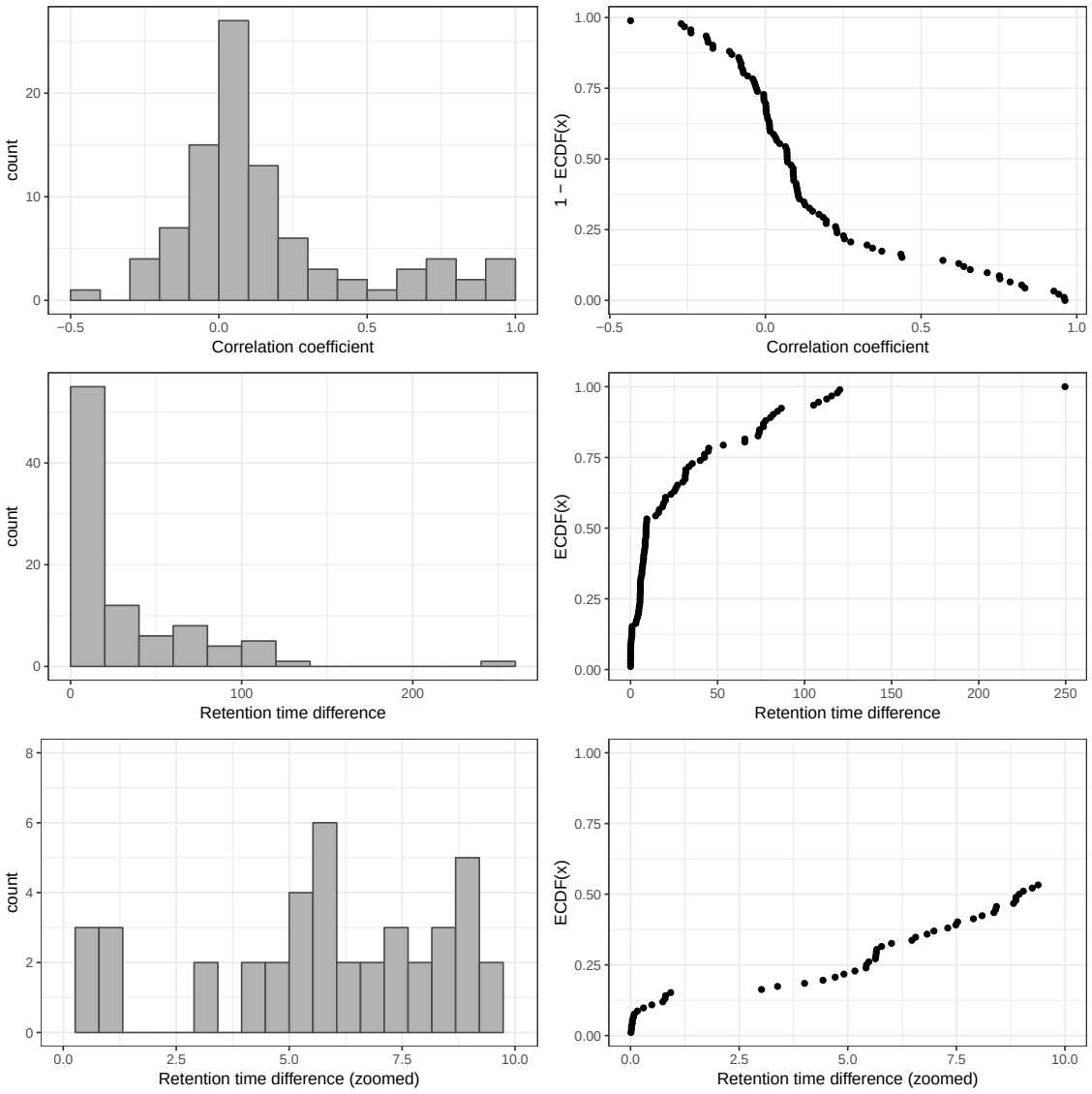
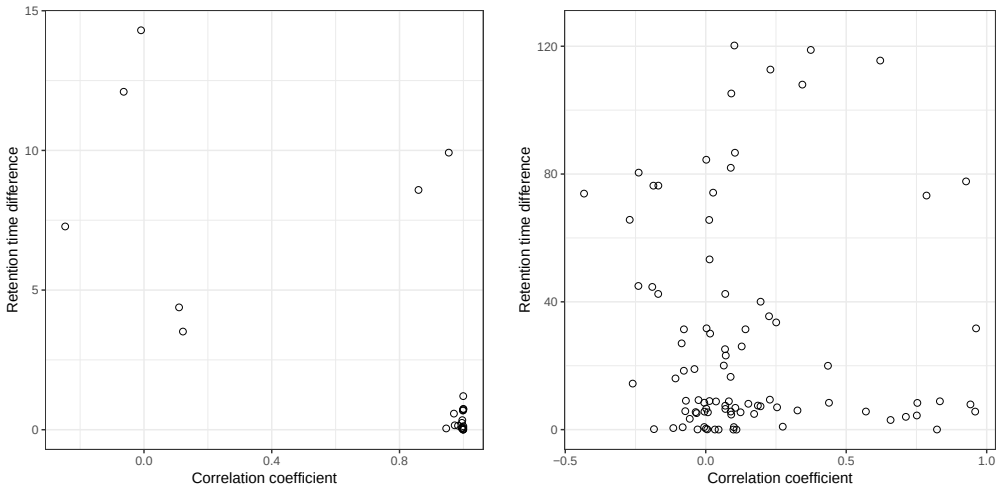


Figure 8: Distributions of retention time difference and correlation coefficient between duplicated features of the plasma dataset. For the correlation coefficients, 1 - empirical cumulative distribution is displayed, while the raw empirical cumulative distribution is used for retention time difference. Retention time differences are expressed in seconds.

sample material. The cereal dataset features samples from different bread varieties, which differ greatly in composition [Koistinen et al., 2018]. Thus, correlation patterns might be easier to observe. The plasma dataset consists of over 30-year-old plasma samples, and as is common in human studies, the differences between individuals are smaller than the differences between samples from different bread varieties, which can obscure the correlation patterns. [Noerman et al., 2018]. In addition, the age of the plasma samples might play a role in the inconsistencies in data quality.



(a) Cereal dataset (b) Plasma dataset

Figure 9: Retention time difference and correlation coefficient between duplicated features

4.3 Drift correction

As discussed earlier in Section 3.3, signal intensity drift is a major issue in LC-MS experiments. Moreover, the amplitude and direction of drift can vary across signals. Thus, no single function can be used to correct a complete batch of LC-MS signals. Many approaches have been proposed in the literature, such as linear regression or bracketed linear regression [van der Kloet et al., 2009], locally estimated scatterplot smoothing (LOESS) regression [Dunn et al., 2011] and its successor cubic spline regression [Kirwan et al., 2013], support vector regression [Kuligowski et al., 2015] and cluster based regression using cubic splines [Brunius et al., 2016]. All of the algorithms provide tools to deal with the varying drift patterns in LC-MS datasets, and none has been clearly proven superior to others [Broadhurst et al., 2018]. The approach used in this work is feature-wise cubic spline correction [Kirwan et al., 2013].

As a reminder from Section 3.3, LC-MS experiments commonly feature quality control (QC) samples, injections of an identical sample that can be used to assess and correct analytical variability such as signal intensity drift. The feature-wise cubic spline correction algorithm consists of two steps: first, the drift function is modeled based on the drift of QC samples, then all the samples are corrected using the predicted values from that function. Cubic spline regression fits a piece-wise cubic polynomial with continuous first and second order derivatives, where every QC sample serves as a knot. Overfitting is tackled by using smoothing splines, adding a penalization term to the common least squares optimization.
This leads to a drift function f that solves:

$$\text{minimize} \sum_{i=1}^{N_{QC}} (y_i - f(x_i))^2 + \lambda \int f''(x)^2 dx, \quad \lambda > 0, \tag{1}$$

where N_{QC} is the number of QC samples, y_i and x_i are the signal abundance and injection order of the i:th QC sample, respectively and λ is a tuning parameter. The second term penalizes overall "curvature" or "wiggliness" of the function, which helps reduce overfitting. Setting the tuning parameter $\lambda = 0$ results in a "perfect" fit with zero error, a curve that passes through all QC samples, while setting $\lambda = \infty$ results in a straight line. In the R implementation used in this work, λ is not set directly, but through a scale-free smoothing parameter, which is then transformed to a suitable λ [Ripley and Maechler, 2019]. The smoothing parameter is chosen using leave-one-out cross validation from a range between 0.5 and 1.5, which has been deemed reasonable during analysis on multiple LC-MS datasets. Smoothing parameter values below 0.5 tend to lead to clear overfitting, while a value of 1.5 is high enough to guarantee that the curve is practically linear.

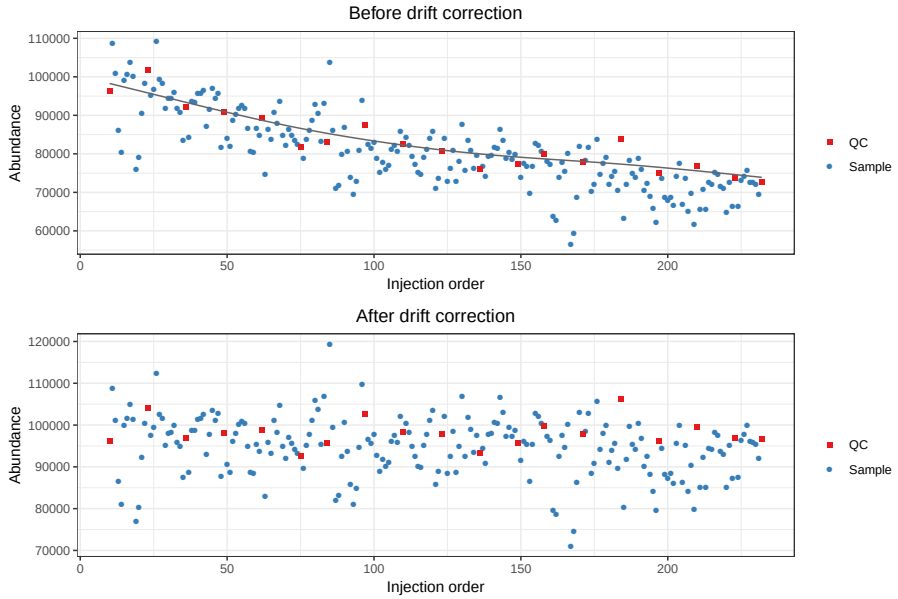


Figure 10: A signal before and after drift correction by smoothing cubic spline regression. The systematic effect of the drift is clearly removed. Smoothing parameter: 0.668

Once the drift function f is defined, the signal abundances are corrected according to this curve following a procedure defined by Brunius [Brunius et al., 2016]. Each sample receives a correction factor c_j, where j is the index of the sample of form:

$$c_j = \frac{f(x_1)}{f(x_j)}, \tag{2}$$

where $f(x_1)$ is the drift function value at the first sample, which is typically the first QC sample in a dataset and $f(x_j)$ is the drift function value at the j:th sample. The abundances of the samples are then multiplied by the corresponding correction factors. Figure 10 shows an exemplar signal before and after drift correction. Figure 11 showcases the robustness of the approach towards QC samples that deviate strongly from the global trend.

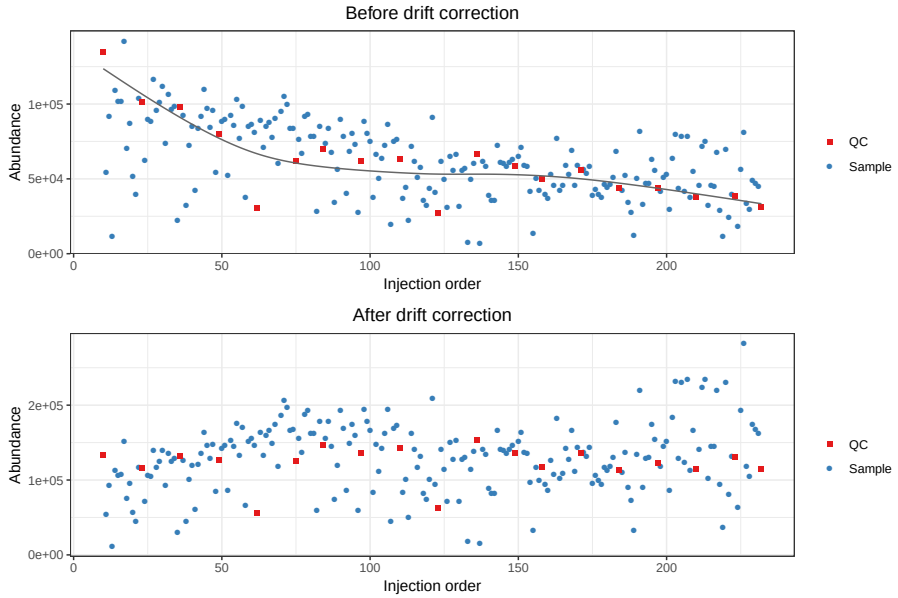


Figure 11: A signal with two deviating QC samples before and after drift correction by smoothing cubic spline regression. Due to the smoothing parameter, the correction method is robust against the two deviating QC samples and corrects for the global drift trend correctly. Smoothing parameter: 0.581

For this drift correction procedure to work properly, it is highly recommended that both the first and the last sample are QC samples. If this is not the case, the cubic spline is forced to extrapolate beyond the range of the QC samples, which can lead to unexpected shapes of the drift function. In addition, the interval between QC samples should be narrow enough to ensure accurate approximation of the drift pattern. At the same time, a too narrow interval between QC samples will increase the number of QC samples, which prolongs the LC-MS run and ultimately leads to stronger drift. An interval of 10-12 samples between every QC sample is commonly used at UEF.

4.4 Quality metrics

Even after drift correction, a large percentage of the features from LC-MS experiments contain significant proportions of analytical variation. In addition, some features often contain large amounts of missing values. Thus, there is a need for quantitative quality metrics that can be used for flagging or removing low-quality features, increasing the signal-to-noise ratio of the data set. Broadhurst et al. have proposed a set of quality metrics that are equipped to deal with this issue. [Broadhurst et al., 2018]

The first quality metric applied is QC detection rate. This can be defined as simply the number of QC samples where the feature was detected, i.e. it is not missing divided by the number of total QC samples for a given feature, expressed as a percentage. The logic behind using QC detection rate as a quality metric is that if a feature cannot be reliably detected in QC samples, it is regarded as noise.

Broadhurst et al. suggest an acceptance limit of 70%. [Broadhurst et al., 2018] This limit seems quite harsh for studies where the groups differ significantly, such as studies of samples from different types of grain, or different berries. In these cases, some group might have metabolites that are not present in others, which means that their concentration in the QC samples might fall below the resolution of the mass spectrometer. In these cases, a lower acceptance limit can thus be agreeable.

The rest of the quality metrics measure two things: internal spread of the QC samples, and spread of the QC samples compared to the spread of the biological samples. Internal spread is measured with relative standard deviation (RSD), also known as coefficient of variation (CV), defined as

$$RSD = \frac{s_{QC}}{\bar{x}_{QC}},$$

where s_{QC} is the standard deviation of the QC samples and $\bar{x}_{QC}$ is the sample mean of the signal in the QC samples. RSD can also be replaced by a non-parametric, robust version based on the median and median absolute deviation (MAD):

$$RSD^* = \frac{1.4826 \cdot MAD_{QC}}{\text{median}(x_{QC})}.$$

The spread of the QC samples compared to the biological samples is measured using a metric called D-ratio:

$$D - ratio = \frac{s_{QC}}{s_{biological}},$$

where $s_{biological}$ is the standard deviation of biological samples. Similar to RSD, D-ratio has a non-parametric, robust alternative:

$$D - ratio^* = \frac{MAD_{QC}}{MAD_{biological}}.$$

For both RSD and D-ratio, lower values represent better quality. The robust versions of RSD and D-ratio are used in this work as they are less affected by a single outlying QC sample and since the original metrics work best if the data is normally distributed, which is often not the case with LC-MS data. [Broadhurst et al., 2018]

Once the quality metrics are computed for each feature, they can be used to filter out low-quality features. Broadhurst et al. recommend that only features with RSD < 0.2 and D-ratio < 0.4 should be retained. [Broadhurst et al., 2018] Since the robust alternatives are used in this work, an additional acceptance condition is added: Features with classic RSD, RSD* and basic D-ratio all below 0.1 are retained. This additional condition prevents the removal of features with very low values in all but a few samples. These features tend to have a very high value of D-ratio*, since the median absolute deviation of the biological samples is not affected by the large concentration in a handful of samples, but the samples with high concentration affect the composition of the pooled QC samples. This causes D-ratio* to overestimate the significance of random errors in measurements of QC samples. Thus, other quality

metrics are applied, with a highly conservative limit of 0.1 to ensure that only good quality features are retained this way.

The aforementioned quality metrics can also be used for experiment level quality assessment by visualizing the distribution of the quality metrics across all features. This approach can be used for overall quality control as well as to assess the effect of different sample preparation techniques or other procedures to overall data quality.

4.5 Imputation

LC-MS metabolomics datasets tend to have a significant amount of missing values, that can be caused by a variety of reasons. Most often the feature is either truly not present in the sample, or the concentration is below the detection limit of the instrument. However, missing values can also result from errors made by software in the peak detection and alignment phases (See Section 4.1). Since many statistical procedures require a complete data matrix, it is necessary to employ a strategy for imputing the missing values. Unfortunately, the multiple sources of missingness present a challenge for data imputation. [Kokla et al., 2019]

In a recent study, Kokla compared multiple methods for missing value imputation on LC-MS metabolomics datasets, including four naive imputation methods where all the missing values are imputed by the same value:

- Zero: missing values considered truly missing.
- Mean of the feature.
- Minimum value of the feature.
- Half the minimum value of the feature: missing values considered to be caused by concentrations below the detection limit.

In addition, Kokla compared five imputation methods where the missing values are predicted using other information in the dataset.

- Singular value decomposition: all missing values are initially imputed by zero and the data matrix $\mathbf{X}$ is subsequently decomposed as $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ and the most relevant eigenvectors $\mathbf{V}^T$ (corresponding to principal components of the data, see Section 5.1) are used to linearly estimate the missing values [Hastie et al., 1999].
- Probabilistic principal component analysis [Nyamundanda et al., 2010] and Bayesian principal component analysis [Takemasa et al., 2003]: two variations of singular value decomposition that also rely on similar dimensionality reduction, but introduce a probabilistic model minimizing non-relevant principal directions.
- Random forest: missing values are predicted by a random forest fit on the non-missing part of the dataset [Stekhoven and Bühlmann, 2011].

- K-nearest neighbors: missing values are imputed with the mean of the k nearest neighbors [Batista and Monard, 2003].

Kokla concluded that random forest imputation performs best on LC-MS datasets from the UEF [Kokla et al., 2019]. Thus, random forest imputation is used in this work and is presented here in more detail. Since random forests are an ensemble method based on decision trees, we will continue with a brief introduction to decision trees.

Decision trees Decision trees are supervised models that can be used in either regression or classification tasks. The fundamental principle of constructing decision trees is to recursively partition the data into rectangular regions, hypercubes, where each final region is uniform in terms of predictions, i.e. any point in the region receives the same prediction. [Fratello and Tagliaferri, 2019] Figure 12 showcases an arbitrary example, where the abundance of L-serine (on the vertical axis) is predicted by the abundance of glyceric acid and taurine.

Decision tree training commonly involves two steps: a growing phase and an optional pruning phase. In the growing phase, the first step is to choose a feature that best separates the data in two parts. What is considered the best split is defined by a split criterion, a measure of uniformity of the data. The goal of each split is to find maximally uniform subgroups. For example the implementation in the R package rpart (used to construct the decision tree in Figure 12) uses the following formula for regression: $SS_T - (SS_L + SS_R)$, where $SS_T = \sum(y_i - \overline{y})^2$ is the sum of square for the data at the node, and SS_L and SS_R are the square sums of the left and right children of the node. This quantity is maximized at each split, i.e. the variance in the children is minimized. In case of discrete variables, or a classification task, Gini impurity is used:

$$I_G = \sum_{j=1}^{k} p_j(1 - p_j),$$

where k is the number of classes of the predicted variable. Gini impurity measures the probability that a random element in the subgroup is falsely classified, if it was randomly labeled according to the distribution of class labels in the subgroup. Gini impurity is minimized at each split and reaches zero for completely pure classes.

The splitting process is repeated for each subgroup until a stopping criterion is met. Common stopping criteria are to limit the minimum number of samples per node or set a minimum level of impurity, after which the data is no longer split. Without a stopping criterion, the tree would separate each sample into a single leaf, leading to gross overfitting. [Fratello and Tagliaferri, 2019]

The optional pruning phase advances iteratively bottom-up. At each iteration, a split in the complete tree is collapsed to a single leaf node. This procedure results in a series of decision trees of decreasing size. The performance of each such tree is then evaluated, for example using cross validation, and the best performing tree is retained.

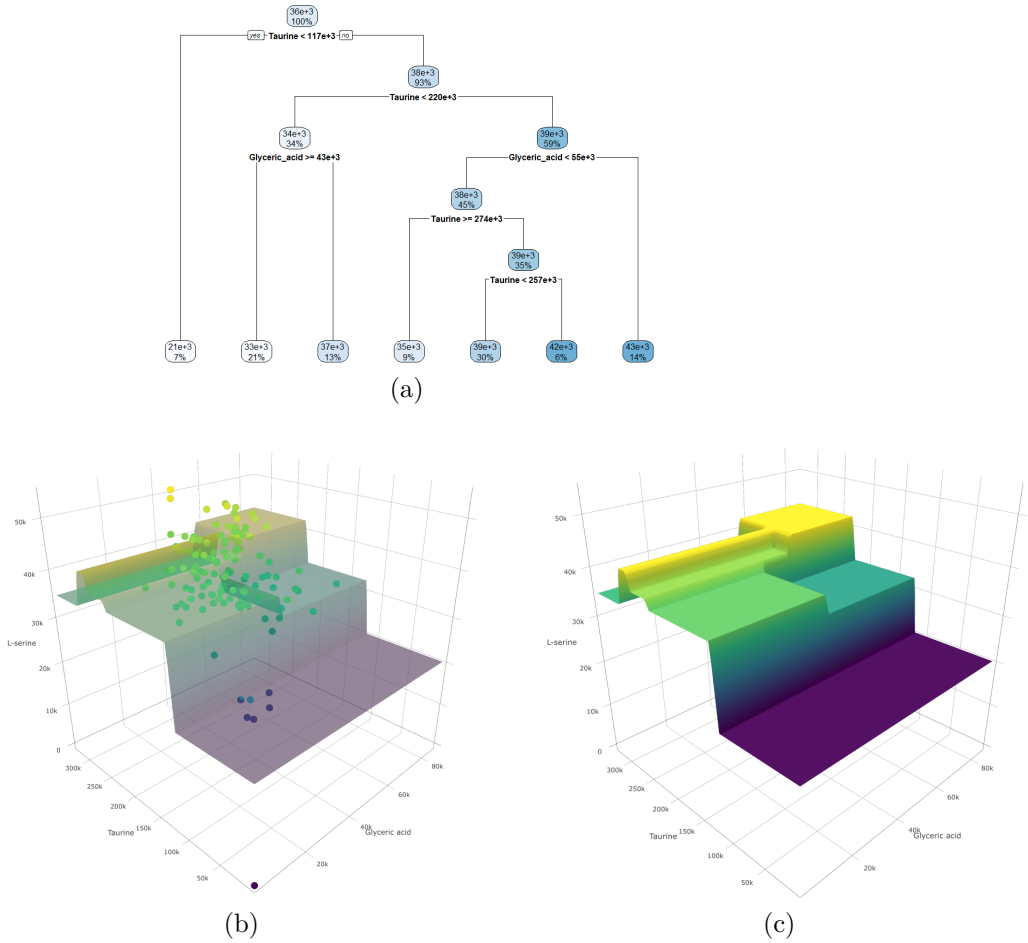


Figure 12: A decision tree trained to predict abundances of L-serine from abundances of Glyceric acid and Taurine. The data is originated from an LC-MS experiment with a HILIC column and negative ionization. a) The decision tree b) The observations and the prediction surface of the decision tree c) The prediction surface of the decision tree

Unfortunately, while easy to interpret, decision trees often perform poorly in real life applications for two reasons. First, despite pruning, it is hard to find a balance between overfitting and accuracy. Second, even small changes in the training data can induce major changes in the decision tree structure, making decision trees not robust against noise. Luckily, these weaknesses can be avoided using random forests.

Random forest Random forest is an ensemble method where, as the name suggests, multiple decision trees are grown, and the ensemble of the trees is used in classification or regression. In the case of classification, each tree casts a unit vote and in regression, the mean of the predictions of the individual trees is used as the final result. The motivation of random forest is that growing multiple trees can make use of the unstable nature of decision trees: minor alterations in the training data can lead to significantly different classifiers. [Breiman, 2001]

Breiman shows that the performance of random forests is strongly dependent on two measures. First, the performance of individual trees obviously affects the performance of the forest. Second, correlation between the predictions of different trees decreases the performance of the forest. In the ideal case, the predictions of individual trees would be independent and increasing the number of trees would increase the performance of the forest as long as each individual tree performed better than random guessing. Thus, the strategies for growing random forests need to present some way of inducing independence between trees. [Breiman, 2001]

Many different strategies exist for inducing independence between the decision trees of a random forest. Commonly each tree has access to a subset of samples and/or features, sampled with replacement (bootstrapping) or without replacement. In this work, the recommended settings of the randomForest R package are used: bootstrapping sample of the same size as the original data is used for each tree. In addition, each node of the decision tree only has access to a random subset of one third of the features to determine the best split. This strategy is similar to the one recommended by Breiman [Breiman, 2001]

Breiman presents multiple beneficial qualities of random forests. First, theoretically, adding trees to a random forest does not lead to overfitting, but instead the generalization error of the forest converges towards a limiting error rate. Second, thanks to their structure, random forests are resistant to noise in the training data. Third, random forests can assess variable importance in prediction by calculating the rise in error rate if the variable was excluded from the dataset. In addition, random forests can assess their own performance using out-of-bag estimates for the error rate. In this approach, the error is calculated by predicting values for each data point in the training data using only the trees that do not contain the respective data point. Breiman shows that the out-of-bag error rates are a reliable measure of the error, although they tend to overestimate the true generalization error, since only a subset of trees is used instead of all the trees.

missForest missForest is an algorithm for imputing missing values using random forest. Thanks to the properties of random forest, missForest can impute missing values from both categorical and continuous variables and can deal with non-linear and interactive effects in the data. missForest iteratively imputes missing values of each variable of the dataset using other variables to predict the missing values. A pseudocode representation of the algorithm is given below in the following notation:

- $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_p)$ an $n \times p$ data matrix, where n is the number of observations and p is the number of variables
- For an arbitrary variable $\mathbf{X}_s$:
  - $y_{\text{obs}}^{(s)}$ the observed values of the variable
  - $y_{\text{mis}}^{(s)}$, the missing values of the variable
  - $x_{\text{obs}}^{(s)}$, the values of other variables in the rows where the value of $\mathbf{X}_s$ is observed

  - $x_{\text{mis}}^{(s)}$, the values of other variables in the rows where the value of $\mathbf{X}_s$ is missing

- γ, a stopping criterion

Algorithm 1: Impute missing values with RF [Stekhoven and Bühlmann, 2011]

Input: $\mathbf{X}$, an $n \times p$ matrix, a stopping criterion γ
Output: The imputed matrix $\mathbf{X}^{\text{imp}}$

```
Make initial guess for missing values
k ← vector of sorted indices of columns in X w.r.t. increasing amount of
  missing values
while not γ do
    X_old^imp ← store previously imputed matrix
    for s in k do
        Fit a random forest: y_obs^(s) ~ x_obs^(s)
        Predict y_mis^(s) using x_mis^(s)
        X_new^imp ← update imputed matrix, using predicted y_mis^(s)
    end
    update γ
end
return X_new^imp
```

To begin, missForest imputes all missing values using mean imputation, where all missing values are equal to the mean of the observed values, or other simple imputation method. Next, the variables are sorted by the proportion of missing values starting from the lowest proportion. For each variable $\mathbf{X}_s$, a random forest is fit with response $y_{\text{obs}}^{(s)}$ and predictors $x_{\text{obs}}^{(s)}$ and the trained random forest is used to predict $y_{\text{mis}}^{(s)}$ from $x_{\text{mis}}^{(s)}$. This imputation procedure is repeated until a stopping criterion is met. [Stekhoven and Bühlmann, 2011]

The stopping criterion is met when the difference between the new imputed matrix and the old one increases for the first time for both categorical and continuous variables. For continuous variables N, the difference is defined as

$$\Delta_N = \frac{\sum_{j \in \mathbf{N}} \left(\mathbf{X}_{\text{new}}^{\text{imp}} - \mathbf{X}_{\text{old}}^{\text{imp}} \right)^2}{\sum_{j \in \mathbf{N}} \left(\mathbf{X}_{\text{new}}^{\text{imp}} \right)^2},$$

and for the categorical variables F as

$$\Delta_N = \frac{\sum_{j \in \mathbf{F}} \sum_{i=1}^{n} \mathbf{I}_{\mathbf{X}_{\text{ncw}}^{\text{imp}} \neq \mathbf{X}_{\text{old}}^{\text{imp}}}}{\#\text{NA}},$$

where #NA is the number of missing values. [Stekhoven and Bühlmann, 2011]

MissForest uses out-of-bag estimation of error to assess the credibility of the imputation process. For continuous variables, the normalized root mean square error is used:

$$\text{NRMSE} = \sqrt{\frac{\text{mean}\left(\left(\mathbf{X}^{\text{true}} - \mathbf{X}^{\text{imp}}\right)^2\right)}{\text{var}\left(\mathbf{X}^{\text{true}}\right)}}$$

For categorical variables, proportion of falsely classified elements is used. For both metrics values close to 0 are desirable while values close to 1 signal poor performance. [Stekhoven and Bühlmann, 2011]

To summarize, random forest imputation has a high performance and many preferable characteristics that make it the choice for imputation method in this work.

5 Visualization techniques

5.1 Principal component analysis (PCA)

Principal component analysis (PCA) is a classic technique originally invented by Pearson in 1901 [Pearson, 1901] and later developed by Hotelling in 1933 [Hotelling, 1933] that is still popular today. While PCA has many applications, it is primarily used for dimensionality reduction: PCA aims to find linear combinations of the original variables called principal components that can explain the variance in the data in a low-dimensional space. The idea driving PCA and other dimensionality reduction methods is that a dataset with a high number of variables, such as a metabolomics dataset, probably contains numerous variables that contain little relevant information. [Mardia et al., 1979] In the case of metabolomics data, these variables can be metabolites that have very similar levels in each individual, or a group of metabolites that are strongly intercorrelated. In the latter case, a single linear combination of the closely intercorrelated metabolites can be sufficient to preserve nearly all information provided by the original metabolites. It should be noted that PCA is a linear method, and thus cannot represent nonlinear patterns in datasets. Figure 13 showcases the steps of PCA.

It is a common practice to mean-center and autoscale the data before applying PCA. In autoscaling all variables are divided by their standard deviation. As a result, all variables in the dataset have unit variance. Next, PCA aims to find the direction where the variance in the data is maximized (see Figure 13 B). This direction is called the first principal direction and the projection of the data points to that line is called the first principal component. The following principal directions are chosen so that they are orthogonal to all previous directions and have maximal variance (see Figure 13 C). This is the reason for autoscaling the variables: without scaling,

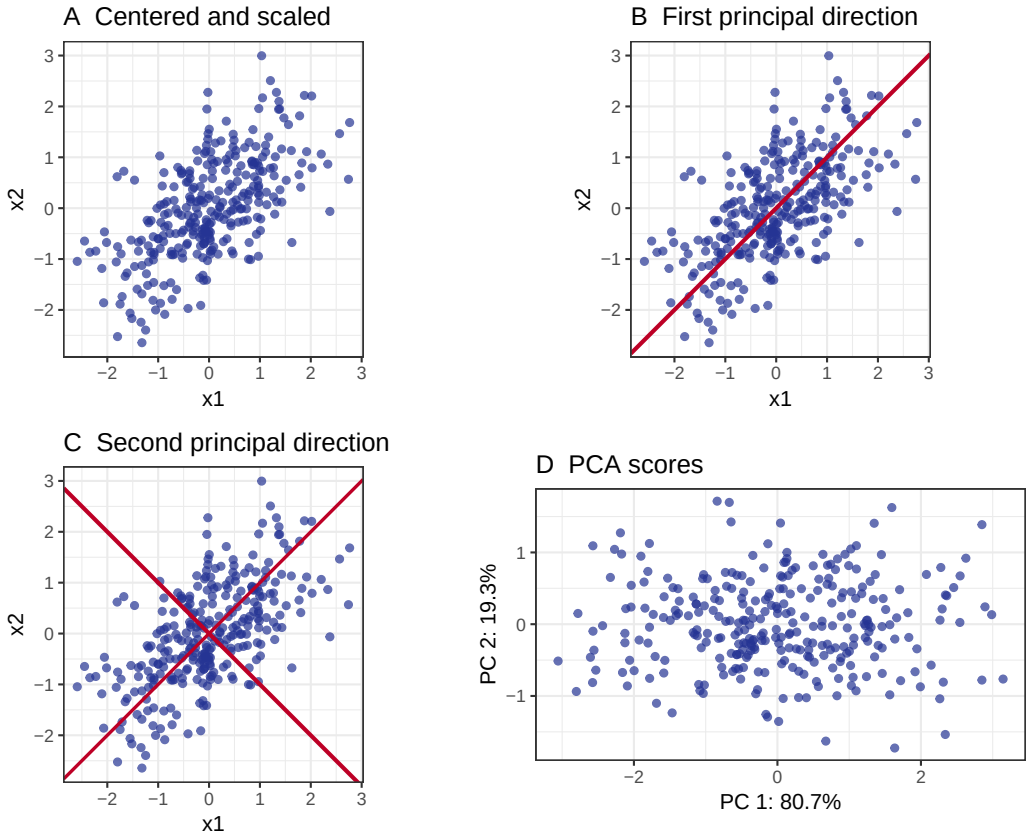


Figure 13: Phases of PCA

metabolites with larger concentrations, and thus larger variances, would dominate PCA, while metabolites with low concentrations would have little importance. The data points are then projected onto these directions, yielding the principal components represented in Figure 13 D. The coordinates of the data points in the principal component space are called scores. [Mardia et al., 1979]

Suppose we have a data matrix $\mathbf{X}$ of size $n \times p$, where n is the number of samples and p is the number of metabolic features. The principal components are the eigenvectors of the sample covariance matrix of $\mathbf{X}$. If $\mathbf{X}$ is centered (i.e. the column means of $\mathbf{X}$ are zero), the sample covariance matrix of $\mathbf{X}$ is simply equal to

$$\mathbf{S} = \mathbf{X^T X}. \tag{3}$$

We can then apply eigenvalue decomposition to $\mathbf{S}$ to get

$$\mathbf{S} = \mathbf{W \Lambda W^T}, \tag{4}$$

where $\mathbf{W}$ is a $p \times p$ orthogonal matrix with the eigenvectors of $\mathbf{S}$ as its columns and $\mathbf{\Lambda}$ is a $p \times p$ diagonal matrix containing the corresponding eigenvalues. $\mathbf{W}$ is generally referred to as the loading matrix. The eigenvalues are ordered largest to smallest, so that the first column of $\mathbf{W}$, w_1 corresponds to the largest eigenvalue λ_1. The eigenvectors in $\mathbf{W}$ are normalized so that $w_i^T w_i = 1$ for any eigenvector

$w_1, \ldots, w_p$. Then, the PCA transformation can be computed as

$$\mathbf{T} = \mathbf{XW}, \tag{5}$$

where $\mathbf{T}$ is an $n \times p$ matrix containing the computed scores on the principal components. It can be shown that using the normalized eigenvectors of the covariance matrix as the weights (or loadings) of the linear transformations yields principal components that explain a maximal amount of variance in the original data. [Mardia et al., 1979]

The proportion of variance explained by the i:th principal component can be calculated using the eigenvalues of the covariance matrix $\mathbf{S}$, and is given by

$$\frac{\lambda_i}{\sum_{k=1}^{p} \lambda_k}. \tag{6}$$

This property is a clear strength of PCA in visualization, since a viewer can assess the credibility of the projection by the proportion of variance explained by the first two or three components used in visualization. The proportion of variance explained by the principal components is commonly visualizes using a scree plot such as the one in Figure 14. In this example, PCA is applied to a metabolomics data set containing 269 samples and 405 metabolic features analyzed using a HILIC column and negative ionization. Even though 269 features would be needed to explain 100% of the variance in the data, the first 2 principal components alone explain approximately 33% of the total variance.

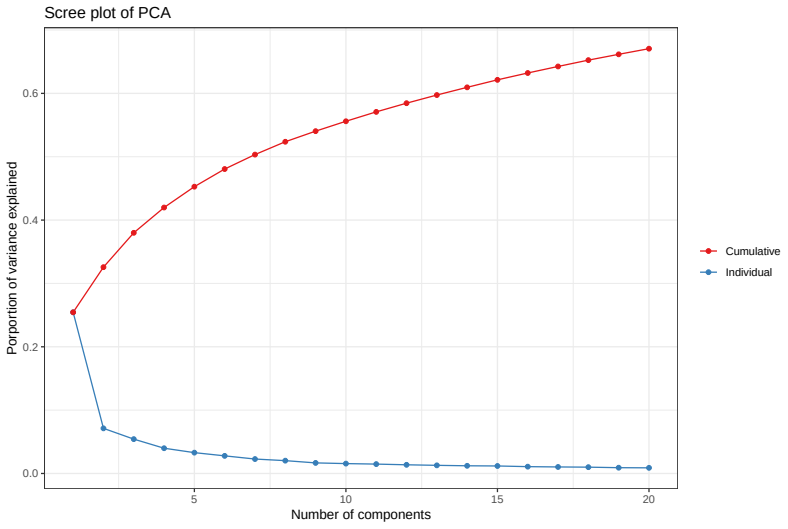


Figure 14: Proportion of variance explained by the first 20 principal components of PCA applied to a metabolomics dataset

As mentioned before, the major limitation of PCA is its inability to retain nonlinear patterns in datasets. Another limitation of the classic PCA is that the algorithm can not deal with missing values. Many variants of PCA have been proposed in the literature to tackle the problems with missing values, including SVDImpute [Botstein et al., 2001] and Bayesian PCA [Takemasa et al., 2003].

5.2 T-distributed stochastic neighbor embedding (t-SNE)

T-distributed stochastic neighbor embedding (t-SNE) is a nonlinear dimensionality reduction technique introduced by van der Maaten and Hinton that converts a high-dimensional dataset into a matrix of pairwise similarities and visualizes these similarities. The t-SNE algorithm is capable of capturing local structure in the dataset while also preserving global patterns. [van der Maaten and Hinton, 2008]

The goal of t-SNE is to construct similarities between datapoints in high-dimensional space, and a low-dimensional mapping that represents the similarities as faithfully as possible. In this section, the original, high-dimensional dataset is denoted $X = \{x_1, x_2, ..., x_n\}$ and the points x_i are referred to as datapoints where the low-dimensional counterpart is denoted $Y = \{y_1, y_2, ..., y_n\}$ and its elements are referred to as map points. [van der Maaten and Hinton, 2008]

First step of t-SNE is to convert high-dimensional Euclidean distances into conditional probabilities that represent similarities between data points. The similarity of datapoint x_i to another datapoint x_j, denoted $p_{j|i}$ is the probability that x_i picks x_j as its neighbor, where the neighbors are picked according to their relative probability density under a Gaussian distribution centered at x_i:

$$p_{j|i} = \frac{\exp\left(-\left\|x_i - x_j\right\|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-\left\|x_i - x_k\right\|^2 / 2\sigma_i^2\right)} \tag{7}$$

This probability is relatively high for nearby datapoints, but quickly tends towards zero for more distant datapoints. The similarity of a datapoint to itself, $p_{i|i}$ is set to zero, so the similarities with other datapoints sum up to 1. Note that the similarity metric defined above is not symmetric, as the variance σ_i^2 varies between datapoints. The method used for determining the variance is described later in this section. Next, t-SNE creates a final, symmetric similarity metric, defined as:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}. \tag{8}$$

A similar metric of similarity is defined for the map points in the low-dimensional mapping, but instead of a Gaussian distribution, a Student t-distribution with one degree of freedom is used. The similarity between two low-dimensional data points y_i and y_j, denoted q_{ij}, is then defined as:

$$q_{ij} = \frac{\left(1 + \left\|y_i - y_j\right\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \left\|y_k - y_l\right\|^2\right)^{-1}}, \tag{9}$$

where q_{ii} is set to zero as before. Collectively, the similarities p_{ij} and q_{ij} can be used to define two joint probability distributions: P in the high-dimensional space and Q in the low-dimensional space (since both p_{ij} and q_{ij} sum up to 1). The objective of function of t-SNE, denoted C, is the Kullback-Leibler divergence between P and Q, defined as:

$$C = KL(P\|Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

T-SNE minimizes the objective function using gradient descent. The gradient of the objective function is of the following form:

$$\frac{\delta C}{\delta y_i} = 4 \sum_j (p_{ij} - q_{ij}) (y_i - y_j) \left(1 + \|y_i - y_j\|^2\right)^{-1}$$

The optimization problem of t-SNE can be regarded as finding stable positions for points in the low-dimensional mapping that are all interconnected by springs. If two map points are pressed too close together, i.e. $q_{ij} > p_{ij}$, the string between them will exert a repulsive force. Similarly, if the two map points are too far from each other i.e. $q_{ij} < p_{ij}$, the string will exert an attractive force. Note that the Kullback-Leibler divergence is not symmetric, which means that different directions of error in the low-dimensional mapping are weighed unequally. Representing nearby datapoints by distant map points has a high cost, while representing distant data points with nearby map points only has a small cost. In practice this means that t-SNE accurately represents local structure such as clusters, while distances between clusters are rather meaningless. In addition, t-SNE deals with local relative distances, which means that two clusters that have a similar radius in the high-dimensional space can have different radiuses in the low-dimensional space. [van der Maaten and Hinton, 2008]

At this point, a careful reader might have two important questions: First, why is the variance of the Gaussians in the high-dimensional space allowed to vary, when replacing σ_i^2 by a common variance σ in Equation 7 would lead to symmetric probabilities and allow t-SNE to skip Equation 8? The answer is that the varying variance σ_i^2 allows the algorithm to adapt better to both dense and sparse regions in the high-dimensional data. For datapoints that lie in a dense region, the variance σ_i^2 is lower compared to that of datapoints located in sparse regions. Using a common variance is especially problematic in the presence of outliers: outlying points would score low for each pairwise similarity, as their distance to other datapoints are relatively large and thus have a negligible effect to the joint probability distribution P. The similarity metric defined in Equation 8, ensures that $\sum_j p_{ij} > \frac{1}{2n}$ for all datapoints, and thus all datapoints have an effect on the low-dimensional mapping. [van der Maaten and Hinton, 2008]

The second question is: why is a Student t-distribution used in the low-dimensional space instead of a Gaussian? In fact, t-SNE builds upon stochastic neighbor embedding (SNE) introduced by Hinton and Roweis [Hinton and Roweis, 2003] that uses a Gaussian distribution to represent similarities in both high- and low-dimensional spaces. The reason for choosing the t-distribution over a Gaussian is related to the heavier tails of the t-distribution. When modeling a high-dimensional dataset in lower dimensions, dimensionality reduction techniques run into what van der Maaten and Hinton call a "crowding problem". This problem is due to the fact that the space around a datapoint shrinks when moving to a lower dimension: a sphere with radius r in a space of 2000 metabolites has a volume proportional to r^{2000}, when a circle in a two-dimensional mapping used for visualization only has an area proportional to r^2. This means that datapoints that are moderately distant from x_i in the high-dimensional space need to be modeled by larger relative distances from y_i in the low-dimensional space, to make room for the neighbors of y_i. This in turn

leads to a situation where the map points in the centre of the map are surrounded by moderately distant map points that are all pulled closer to them, eventually crushing all the central map points together. Replacing the Gaussian by a t-distribution in the low-dimensional mapping alleviates the crowding problem, since the heavier tails mean that large distances result in higher q_{ij} under a t-distribution than under a Gaussian, allowing moderately distant map points to be modeled with relatively large distances in the lower-dimensional mapping. [van der Maaten and Hinton, 2008]

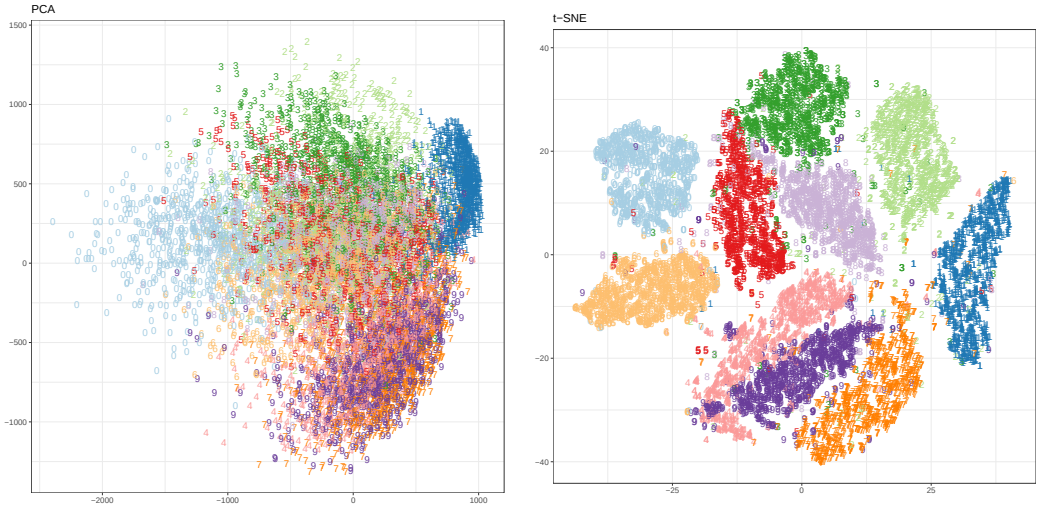
The variance of the Gaussian distribution P_i centered around a datapoint x_i is chosen so that the perplexity of the distribution is equal to a user specified value. Perplexity is defined as

$$\text{Perp}\left(P_i\right) = 2^{H(P_i)},$$

where $H(P_i)$ is the Shannon entropy of P_i measured in bits:

$$H\left(P_i\right) = -\sum_j p_{j|i} \log_2 p_{j|i}$$

The perplexity can be viewed as an effective number of neighbors. According to van der Maaten and Hinton, t-SNE is quite robust to changes in perplexity, while typical values range between 5 and 50. [van der Maaten and Hinton, 2008] However, there are examples where the perplexity parameter does greatly affect the ability of the algorithm to reveal global structure of the dataset. The need for a user-specified hyperparameter can be seen as a weak spot for t-SNE and trying multiple values for a dataset is advised. [Wattenberg et al., 2016]



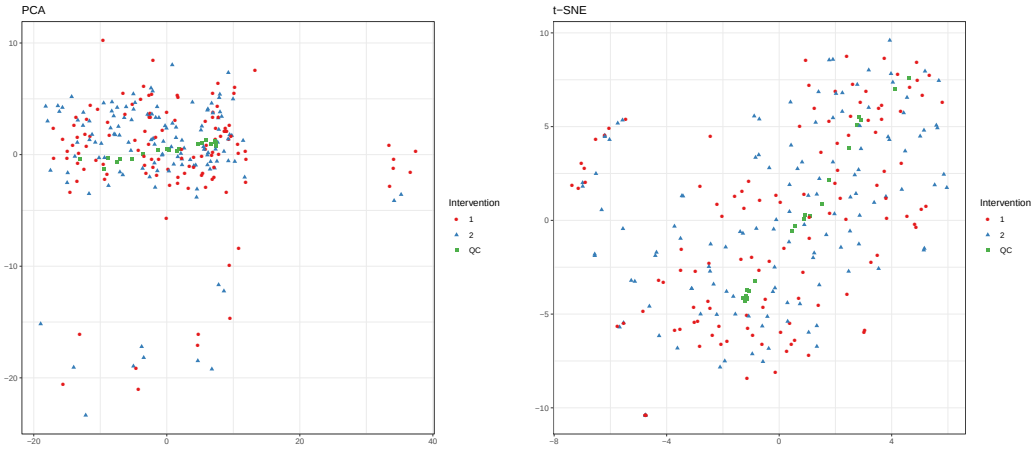
(a) PCA (b) t-SNE

Figure 15: Results of PCA and t-SNE applied to the MNIST dataset

Figure 15b showcases the performance of t-SNE on 10,000 observations from the MNIST-dataset [LeCun and Cortes, 2010], comprising of handwritten digits. Each observation in the dataset is a 28 by 28 pixel image, resulting in a 784 dimensional dataset. T-SNE is capable of separating the handwritten digits remarkable well.

As a comparison, Figure 15a shows how PCA applied to the same data fails to present the local structure of the data. The original publication features many more examples of the prowess of t-SNE, also comparing its performance to other nonlinear dimensionality reduction techniques [van der Maaten and Hinton, 2008].

In metabolomics experiments, the fact that t-SNE prioritizes local structure over correctly representing distances between clusters can make t-SNE a useful complementary tool to PCA. Figures 16a and 16b show the results of applying PCA and t-SNE, respectively, to a metabolomics dataset resulting from the HILIC negative mode of an intervention study. The dataset consists of 269 samples with 400 metabolites measured for each sample. While PCA represents the distance between the main cluster of points and the few outliers, t-SNE reveals the local structure inside the main cluster while also visualizing the global structure.



(a) PCA (b) t-SNE

Figure 16: Results of PCA and t-SNE applied to a metabolomics dataset

A potential weakness of t-SNE regarding future applications to metabolomics data is that the mapping of datapoints to the low-dimensional space is not explicit as in PCA, meaning that new observations cannot be projected directly into the low-dimensional map without rerunning t-SNE on the new, larger dataset. Luckily, this weakness has been addressed by introducing a parametric version of t-SNE, where the mapping is learned by a neural network [van der Maaten, 2009].

6 The R package

6.1 Requirements and implementation plan

All preprocessing steps not conducted by external software are implemented as a package for the R programming language [R Core Team, 2018]. The package is partly based on old analysis scripts written at the UEF. These R scripts contain many useful functions that are used in many of the projects. The old workflow for a data analysis project is as follows:

1. Copy analysis scripts from older projects
2. Modify the analysis scripts to fit the current project - possibly overwriting old functions with new, project-specific versions
3. Test the analysis flow manually
4. Repeat steps 2 and 3 until everything works as expected
5. Gather results and submit them to fellow researchers

Common reasons for redefining functions in step 2 are changing color scale or other parameters in visualization functions and changing functions conducting statistical analyses to allow more flexibility in terms of the model specified. The need to redefine functions for each project requires the user of the scripts to have substantial knowledge about the source code behind the functions, which makes the analysis workflow hard to adopt for new users. Another major issue with the old scripts is that the scripts have been written by multiple authors without clear guidelines. This leads to multiple issues, including arguments of similar functions having slightly

different meanings. For example, many visualization functions have a parameter that tells the function which column of a data frame should be used to color objects in the plot (for example, study group), but one function expects the column name as a string, while another very similar function expects that column as a vector. Nearly non-existent documentation of the old functions makes spotting these caveats near impossible for a new user. As a result of these problems, it was practically impossible for a new user to start using the analysis scripts without personal guidance or reading through the source code.

The goal of the new package is to fix many of the aforementioned problems. The main requirements are:

1. Automation of all the relevant analysis steps, from Excel spreadsheet back to Excel spreadsheet with results from statistical analyses
2. A suitable level of abstraction to allow both fast analysis and flexibility
3. Clear documentation
4. Sufficient unit testing to guarantee robustness of the package

Requirement 1 is quite obvious, but requirement 2 makes things slightly more complicated. The reason why the level of abstraction is mentioned explicitly here is that it can be tempting to develop high-level functions that would attempt to automate a multitude of tasks, for example a preprocessing function that would handle all the preprocessing steps in Sections 4.2 - 5.2 with relevant visualizations. Unfortunately, such approach would probably work in some projects, but fail terribly for others, leading the user back to redefining functions for each project. On the other hand, the primary goal of the project is to make analyzing LC-MS data sets faster, which means raising the level of abstraction from more primitive R-functions. It is thus crucial to find a suitable level of abstraction. Combined with a clear documentation, these steps should allow a new user to experiment with the functions and conduct analyses without excessive need of training. Careful unit testing makes sure that the package is robust to possible anomalies in the data, and makes future development significantly easier, as it is easy to test that new additions do not interfere with older functionality.

R was chosen as the programming language for the package because our research group already has a code base written in R, and most importantly there are many researchers who are starting to learn R for their studies. There are multiple reasons why the new version of the pipeline is implemented as an R package rather than individual scripts containing the functions. First, R packages offer unparalleled documentation tools: a function defined inside an R package is attached to a help page containing details of the function, as well as executable examples. Writing documentation that agrees with the standard format of R help pages is made extremely simple for the developer, thanks to R packages devtools [Wickham et al., 2018b], usethis [Wickham and Bryan, 2018] and roxygen2 [Wickham et al., 2018a] in conjunction with RStudio [RStudio Team, 2016]. Additionally, unit testing the functions

is simple and efficient with the testthat package [Wickham, 2011]. R packages can also be configured to automatically install all dependencies and provide example data delivered with the package, removing the need for separate installing or downloading processes.

Similar to any software project, when developing a data analysis pipeline, it is beneficial to follow fixed guidelines and style throughout the pipeline. This makes the code and the structure of the pipeline consistent, preventing errors and facilitating future development. The package developed for this thesis follows guidelines provided by Wickham [Wickham, 2015]

6.2 Features and workflow

The package allows the user to import and export all required data in a single Excel spreadsheet (.xlsx) format. The file contains:

- sample information such as study group and injection order
- feature information such as mass, retention time, as well as the corresponding LC column and ionization mode
- feature abundances across samples

This file format is the one used by researchers at UEF in other stages of analysis and was chosen for their convenience. The spreadsheet structure is illustrated in Figure 17. R package openxlsx is used to read and write .xlsx files [Walker, 2018].

All the data used in the analysis is recorded in an object of a custom class, MetaboSet, that is built on the ExpressionSet class provided in the Biobase package by Bioconductor [Huber et al., 2015]. The ExpressionSet class was originally designed for gene expression data, but the class is easily adaptable to metabolomics as the structure of datasets is very similar. Both datasets have information about samples, such as phenotype information, molecular features correspond to genes and the molecular feature abundances correspond to gene expression values. ExpressionSet implements fast and reliable ways for subsetting a dataset and other common operations. In addition, Bioconductor provides many tools that use ExpressionSet objects. This is one of the main reasons for choosing ExpressionSet as the parent class of MetaboSet.

The package implements easy-to-use functions for all the preprocessing steps in Sections 4.2 - 4.5. Network analysis algorithms from the igraph package [Csardi and Nepusz, 2006] are used to cluster the features as described in Section 4.2. Random forest imputation is conducted using the implementation in the missForest package [Stekhoven and Bühlmann, 2011]. For dimensionality reduction, the package offers PCA and t-SNE visualizations using pcaMethods package by Bioconductor [Stacklies et al., 2007] and Rtsne package [van der Maaten, 2014], respectively. The package also includes the algorithm for clustering molecular features originating from the same metabolite, presented in Section 4.2. In addition, the package implements a variety of visualizations for both data quality control and assessment of effects relevant to the study. All the visualizations of the package are built with ggplot2,

				Injection	1	2	3	4	5	6
				Group	A	A	A	B	B	B
				Visit	1	2	3	1	2	3
Compound	Mass	RT	Column	Ion Mode	Demo_1	Demo_2	Demo_3	Demo_4	Demo_5	Demo_6
1	514	14	RP	POS	1803	16000	28101	21107	11765	48253
2	300	10	RP	POS	20135	11722	32050	291288	23452	45092
3	140	15	RP	POS	21053	12052	30940	307293	31111	44221
4	213	7	RP	POS	18037	13751	31559	255064	44051	44754
5	321	6	RP	POS	21661	12733	31780	306093	29441	46499
6	312	11	RP	POS	19696	12021	32306	288656	43110	46823
7	489	9	RP	POS	20431	11327	29803	218196	14799	43809

Figure 17: Example of the data input format used by the package: a single spreadsheet containing feature information (green), sample information (blue) and feature abundances across samples (red).

the major R package for visualizations [Wickham, 2016] as well as cowplot package for combining multiple graphs [Wilke, 2019].

For the actual statistical analysis, the package offers a fast way of conducting multiple common univariate hypothesis tests, that can be applied to each feature in a MetaboSet object. All operations that are repeated for each feature, such as drift correction, quality metric computation and univariate hypothesis tests, can be parallelized across platforms thanks to the foreach [Microsoft and Weston, 2017] and doParallel [Microsoft Corporation and Weston, 2018] packages. The package also offers an interface for predicting a variable in the sample information (such as the study group of the sample) from the molecular features using a random forest. In its current state, the package implements all methods of statistical analyses commonly used at the research group. The package and completely replaces all old scripts and will become the main tool for data preprocessing and statistical analysis of the research group. The package is published under the open-source MIT license and is available at https://github.com/antonvsdata/amp/.

Code example The example below shows a piece of code from the package that performs common preprocessing steps on a dataset of bread samples. First, missing values need to be correctly marked as missing (NA in R), as peak picking software tend to record them as 0 or 1. Next, drift correction is performed (see Section 4.3), after which low quality features are flagged (flagged features are ignored by most of the functions). Only features with RSD < 0.2 and D-ratio < 0.4 are kept (see Section 4.4). QC samples are then removed before missing values are imputed using random

forest imputation (see Section 4.5) and the dataset is visualized using PCA (see Section 5.1) and t-SNE (see Section 5.2). For statistical analysis, we are interested in features that differ between bread varieties. These features can be found by first applying one-way analysis of variance to each feature, testing whether feature levels differ between bread varieties. For a multivariate approach, a classification random forest is fit to predict the bread variety using all the good-quality features, and feature importance in random forest prediction is assessed.

```
# Mark missing values as NA
marked <- mark_nas(bread_data, value = 0)

# Drift correction with plots
corrected <- correct_drift(marked, plotting = TRUE,
                           file = "drift_corr_plots.pdf")
# Flag low-quality features
flagged <- flag_quality(corrected,
                        condition = "RSD < 0.2 & D_ratio < 0.4")

# Remove QC samples
no_qc <- drop_qcs(flagged)
# Imputation
imputed <- impute_rf(no_qc)

# Visualizations
plot_tsne(imputed)
plot_tsne(imputed, color = "Time")
plot_pca(imputed, color = "Bread", shape = "Time")

# Statistics
# Feature-wise one-way anova
anova_results <- perform_oneway_anova(imputed,
                                      formula_char = "Feature ~ Bread")
# Multivariate: Random forest
rf <- fit_rf(imputed, response = "Bread")
bread_importance <- importance_rf(rf)
```

7 Conclusion

The R package is functional and ready to be installed by anyone. The package has already been tested in practice by running analyses for two separate projects without issues. The projects provided excellent test cases for the package. The first project included extra features that were analyzed separately from the rest of the dataset: in addition to the standard four modes (RP and HILIC columns and positive and negative ionization), the dataset included features from a smaller mass-to-charge ratio range. Unfortunately, due to an error in the LC-MS run, there were no quality control samples present when these features were measured. Thus, the extra features had to be merged to the dataset after drift correction and removal of low-quality compounds. The merge was further complicated by the fact that the extra features lacked information that was provided for the original features. Thanks to the suitable level of abstraction, the merge proved to be simple to conduct. Based on the experience of these projects, the new package makes running analysis significantly faster and smoother. The package is also robust against missing values and other anomalies often encountered in LC-MS data.

In each step of analysis of the projects, the package documentation was consulted to ensure that it contains all the information required to conduct the analyses. The documentation proved to be clear enough to provide information quickly. In addition, the fact that all functions in the package follow a similar logic makes the package feel intuitive to use. The package was also tested by a researcher with little experience in R and programming in general, and they were able to use the package with minor guidance. This shows that the package is built in a way that a person who is familiar with the workflow of an LC-MS experiment can use the package relying on the documentation. While more testers are needed, the level of abstraction seems adequate, and documentation of the package is written clearly enough to that even

beginners in R programming can use the package. It can thus be concluded, that the package satisfies all the requirements listed in the previous section.

The package will be used in all the future projects at UEF. The package will also be developed further after this thesis. Next steps include integration of more sophisticated multivariate statistical methods for assessing patterns between metabolite levels and other variables in a given study, such as health conditions. In addition, a future project will integrate a method to deal with a so-called batch effect, which arises when a study has too many samples to be included in a single LC-MS run, but needs to be distributed among several analytical batches. In such case, varying factors such as humidity, heat and instrument condition between the runs induce systematic bias in LC-MS data. The package will be published as part of a wider protocol paper that is being prepared by our research group.

Unfortunately, the results from feature clustering were contradictory, where the performance was acceptable for the cereal dataset but proved inadequate for the human plasma samples. Indeed, the large variation in the correlation structure of LC-MS datasets poses a major challenge to any feature clustering technique based on the correlations. The reliability of the results is further hampered by the fact that the identifications given to the features may not be completely accurate. Thus, the algorithm requires more extensive testing before it can be recommended as a standard procedure. However, it must be recognized that the problem of finding features originating from the same compound is a hard one to solve. A broad variety of methods have been developed to tackle the problem, by both instrument vendors and academic research groups worldwide. Still, after all these efforts, many such features are left in the dark.

As a final conclusion, despite the issues with the feature clustering algorithm, the goals of the thesis were met. The R package provides all the intended functionality, and the feature clustering algorithm has potential to be useful in the future. Once the R package is officially published, it has potential to benefit the scientific community beyond UEF.

References

[Aava Medical Centre, 2018] Aava Medical Centre (2018). Virta360 website. https://www.virtavalmennus.fi/en/virta360/virta360-for-me, Accessed on 11.01.2019.

[Airaksinen et al., 2018] Airaksinen, K., Jokkala, J., Ahonen, I., Auriola, S., Kolehmainen, M., Hanhineva, K., and Tiihonen, K. (2018). High-fat diet, betaine, and polydextrose induce changes in adipose tissue inflammation and metabolism in c57bl/6j mice. *Molecular Nutrition & Food Research*, 62(23):1800455.

[Aretz and Meierhofer, 2016] Aretz, I. and Meierhofer, D. (2016). Advantages and pitfalls of mass spectrometry based metabolome profiling in systems biology. *International Journal of Molecular Sciences*, 17(5).

[Bach et al., 2018] Bach, E., Szedmak, S., Brouard, C., Böcker, S., and Rousu, J. (2018). Liquid-chromatography retention order prediction for metabolite identification. *Bioinformatics*, 34(17):i875–i883.

[Batista and Monard, 2003] Batista, G. and Monard, M. C. (2003). A study of k-nearest neighbour as an imputation method. In *In HIS*.

[Berg et al., 2013] Berg, M., Vanaerschot, M., Jankevics, A., Cuypers, B., Breitling, R., and Dujardin, J.-C. (2013). Lc-ms metabolomics from study design to data-analysis – using a versatile pathogen as a test case. *Computational and Structural Biotechnology Journal*, 4(5):e201301002.

[Bird, 1989] Bird, I. M. (1989). High performance liquid chromatography: principles and clinical applications. *BMJ*, 299(6702):783–787.

[Botstein et al., 2001] Botstein, D., Sherlock, G., Cantor, M., Troyanskaya, O., Brown, P., Tibshirani, R., Altman, R. B., and Hastie, T. (2001). Missing value estimation methods for DNA microarrays . *Bioinformatics*, 17(6):520–525.

[Breiman, 2001] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

[Broadhurst et al., 2018] Broadhurst, D., Goodacre, R., Reinke, S. N., Kuligowski, J., Wilson, I. D., Lewis, M. R., and Dunn, W. B. (2018). Guidelines and considerations for the use of system suitability and quality control samples in mass spectrometry assays applied in untargeted clinical metabolomic studies. *Metabolomics*, 14(6):72.

[Brunius et al., 2016] Brunius, C., Shi, L., and Landberg, R. (2016). Large-scale untargeted lc-ms metabolomics data correction using between-batch feature alignment and cluster-based within-batch signal intensity drift correction. *Metabolomics*, 12(11):173.

[Crick, 1970] Crick, F. (1970). Central dogma of molecular biology. *Nature*, 227:561.

[Csardi and Nepusz, 2006] Csardi, G. and Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal*, Complex Systems:1695.

[Debnath et al., 2010] Debnath, M., Prasad, G., and Bisen, P. (2010). *Omics Technology*, pages 11–31. Springer.

[Dunn et al., 2011] Dunn, W. B., Broadhurst, D., Begley, P., Zelena, E., Francis-McIntyre, S., Anderson, N., Brown, M., Knowles, J. D., Halsall, A., Haselden, J. N., Nicholls, A. W., Wilson, I. D., Kell, D. B., Goodacre, R., and Consortium, T. H. S. M. H. (2011). Procedures for large-scale metabolic profiling of serum and plasma using gas chromatography and liquid chromatography coupled to mass spectrometry. *Nature Protocols*, 6:1060.

[Fratello and Tagliaferri, 2019] Fratello, M. and Tagliaferri, R. (2019). Decision trees and random forests. In Ranganathan, S., Gribskov, M., Nakai, K., and Schönbach, C., editors, *Encyclopedia of Bioinformatics and Computational Biology*, pages 374 – 383. Academic Press, Oxford.

[Guijas et al., 2018] Guijas, C., Montenegro-Burke, J. R., Domingo-Almenara, X., Palermo, A., Warth, B., Hermann, G., Koellensperger, G., Huan, T., Uritboonthai, W., Aisporna, A. E., Wolan, D. W., Spilker, M. E., Benton, H. P., and Siuzdak, G. (2018). Metlin: A technology platform for identifying knowns and unknowns. *Analytical Chemistry*, 90(5):3156–3164.

[Hastie et al., 1999] Hastie, T., Tibshirani, R., Sherlock, G., Eisen, M., Brown, P., and Botstein, D. (1999). Imputing missing data for gene expression arrays. Technical report, Division of Biostatistics, Stanford University.

[Hemström and Irgum, 2006] Hemström, P. and Irgum, K. (2006). Hydrophilic interaction chromatography. *Journal of Separation Science*, 29(12):1784–1821.

[Hinton and Roweis, 2003] Hinton, G. E. and Roweis, S. T. (2003). Stochastic neighbor embedding. In Becker, S., Thrun, S., and Obermayer, K., editors, *Advances in Neural Information Processing Systems 15*, pages 857–864. MIT Press.

[Hotelling, 1933] Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441.

[Huber et al., 2015] Huber, W., Carey, V. J., Gentleman, R., Anders, S., Carlson, M., Carvalho, B. S., Bravo, H. C., Davis, S., Gatto, L., Girke, T., Gottardo, R., Hahne, F., Hansen, K. D., Irizarry, R. A., Lawrence, M., Love, M. I., MacDonald, J., Obenchain, V., Ole's, A. K., Pag'es, H., Reyes, A., Shannon, P., Smyth, G. K., Tenenbaum, D., Waldron, L., and Morgan, M. (2015). Orchestrating high-throughput genomic analysis with Bioconductor. *Nature Methods*, 12(2):115–121.

[Hubert and Arabie, 1985] Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2(1):193–218.

[Jang et al., 2018] Jang, C., Chen, L., and Rabinowitz, J. D. (2018). Metabolomics and isotope tracing. *Cell*, 173(4):822 – 837.

[Johnson et al., 2016] Johnson, C. H., Ivanisevic, J., and Siuzdak, G. (2016). Metabolomics: beyond biomarkers and towards mechanisms. *Nature Reviews Molecular Cell Biology*, 17:451.

[Karczewski and Snyder, 2018] Karczewski, K. J. and Snyder, M. P. (2018). Integrative omics for health and disease. *Nature Reviews Genetics*, 19:299.

[Kårlund et al., 2015] Kårlund, A., Hanhineva, K., Lehtonen, M., Karjalainen, R. O., and Sandell, M. (2015). Nontargeted metabolite profiles and sensory properties of strawberry cultivars grown both organically and conventionally. *Journal of Agricultural and Food Chemistry*, 63(3):1010–1019.

[Kirwan et al., 2013] Kirwan, J. A., Broadhurst, D. I., Davidson, R. L., and Viant, M. R. (2013). Characterising and correcting batch variation in an automated direct infusion mass spectrometry (dims) metabolomics workflow. *Analytical and Bioanalytical Chemistry*, 405(15):5147–5157.

[Koeth et al., 2013] Koeth, R. A., Wang, Z., Levison, B. S., Buffa, J. A., Org, E., Sheehy, B. T., Britt, E. B., Fu, X., Wu, Y., Li, L., Smith, J. D., DiDonato, J. A., Chen, J., Li, H., Wu, G. D., Lewis, J. D., Warrier, M., Brown, J. M., Krauss, R. M., Tang, W. H. W., Bushman, F. D., Lusis, A. J., and Hazen, S. L. (2013). Intestinal microbiota metabolism of l-carnitine, a nutrient in red meat, promotes atherosclerosis. *Nature Medicine*, 19:576.

[Koistinen et al., 2018] Koistinen, V. M., Mattila, O., Katina, K., Poutanen, K., Aura, A.-M., and Hanhineva, K. (2018). Metabolic profiling of sourdough fermented wheat and rye bread. *Scientific Reports*, 8(1):5684.

[Kokla et al., 2019] Kokla, M., Virtanen, J., Kolehmainen, M., Paananen, J., and Hanhineva, K. (2019). Random forest -based imputation outperforms other methods for imputing lc-ms metabolomics data : a comparative study. unpublished.

[Kuligowski et al., 2015] Kuligowski, J., Sánchez-Illana, , Sanjuán-Herráez, D., Vento, M., and Quintás, G. (2015). Intra-batch effect correction in liquid chromatography-mass spectrometry using quality control samples and support vector regression (qc-svrc). *Analyst*, 140:7810–7817.

[LeCun and Cortes, 2010] LeCun, Y. and Cortes, C. (2010). MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/.

[Mardia et al., 1979] Mardia, K., Kent, J., and Bibby, J. (1979). *Multivariate analysis*, pages 213–228. Probability and mathematical statistics. Academic Press.

[Markley et al., 2017] Markley, J. L., Brüschweiler, R., Edison, A. S., Eghbalnia, H. R., Powers, R., Raftery, D., and Wishart, D. S. (2017). The future of nmr-based metabolomics. *Current Opinion in Biotechnology*, 43:34 – 40.

[Marshall and Powers, 2017] Marshall, D. D. and Powers, R. (2017). Beyond the paradigm: Combining mass spectrometry and nuclear magnetic resonance for metabolomics. *Progress in Nuclear Magnetic Resonance Spectroscopy*, 100:1 – 16.

[Microsoft and Weston, 2017] Microsoft and Weston, S. (2017). *foreach: Provides Foreach Looping Construct for R.* R package version 1.4.4.

[Microsoft Corporation and Weston, 2018] Microsoft Corporation and Weston, S. (2018). *doParallel: Foreach Parallel Adaptor for the 'parallel' Package.* R package version 1.0.14.

[Myers et al., 2017] Myers, O. D., Sumner, S. J., Li, S., Barnes, S., and Du, X. (2017). Detailed investigation and comparison of the xcms and mzmine 2 chromatogram construction and chromatographic peak detection methods for preprocessing mass spectrometry metabolomics data. *Analytical Chemistry*, 89(17):8689–8695.

[Nakahata et al., 2008] Nakahata, Y., Kaluzova, M., Grimaldi, B., Sahar, S., Hirayama, J., Chen, D., Guarente, L. P., and Sassone-Corsi, P. (2008). The nad+-dependent deacetylase sirt1 modulates clock-mediated chromatin remodeling and circadian control. *Cell*, 134(2):329–340.

[Nightingale Health, 2018] Nightingale Health (2018). Virta 360 blood test - scientific background. https://www.virtavalmennus.fi/NG_Whitepaper_Virta360-bloodtest.pdf.

[Nikolic et al., 2018] Nikolic, D., Shahaf, N., Schymanski, E., and Neumann, S. (2018). Critical assessment of small molecule identification.

[Noerman et al., 2018] Noerman, S., Kärkkäinen, O., Mattsson, A., Paananen, J., Lehtonen, M., Nurmi, T., Tuomainen, T.-P., Voutilainen, S., Hanhineva, K., and Virtanen, J. K. (2018). Metabolic profiling of high egg consumption and the associated lower risk of type 2 diabetes in middle-aged finnish men. *Molecular Nutrition & Food Research*, 0(0):1800605.

[Nyamundanda et al., 2010] Nyamundanda, G., Brennan, L., and Gormley, I. C. (2010). Probabilistic principal component analysis for metabolomic data. *BMC Bioinformatics*, 11(1):571.

[Pannkuk et al., 2015] Pannkuk, E. L., Laiakis, E. C., Authier, S., Wong, K., and Fornace, A. J. J. (2015). Global metabolomic identification of long-term dose-dependent urinary biomarkers in nonhuman primates exposed to ionizing radiation. *Radiat Res*, 184(2):121–133.

[Patti et al., 2012] Patti, G. J., Yanes, O., and Siuzdak, G. (2012). Metabolomics: the apogee of the omics trilogy. *Nature Reviews Molecular Cell Biology*, 13:263.

[Pearson, 1901] Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.

[Pitt, 2009] Pitt, J. J. (2009). Principles and applications of liquid chromatography-mass spectrometry in clinical biochemistry. *Clin Biochem Rev*, 30(1):19–34.

[Pluskal et al., 2010] Pluskal, T., Castillo, S., Villar-Briones, A., and Orešič, M. (2010). Mzmine 2: Modular framework for processing, visualizing, and analyzing mass spectrometry-based molecular profile data. *BMC Bioinformatics*, 11(1):395.

[R Core Team, 2018] R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/.

[Ripley and Maechler, 2019] Ripley, B. D. and Maechler, M. (2019). R: Fit a smoothing spline. https://stat.ethz.ch/R-manual/R-devel/library/stats/html/smooth.spline.html.

[RStudio Team, 2016] RStudio Team (2016). *RStudio: Integrated Development Environment for R*. RStudio, Inc., Boston, MA.

[Scalbert et al., 2014] Scalbert, A., Brennan, L., Manach, C., Andres-Lacueva, C., Dragsted, L. O., Draper, J., Rappaport, S. M., van der Hooft, J. J., and Wishart, D. S. (2014). The food metabolome: a window over dietary exposure. *The American Journal of Clinical Nutrition*, 99(6):1286–1308.

[Shi et al., 2018] Shi, L., Brunius, C., Johansson, I., Bergdahl, I. A., Lindahl, B., Hanhineva, K., and Landberg, R. (2018). Plasma metabolites associated with healthy Nordic dietary indexes and risk of type 2 diabetes—a nested case-control study in a Swedish population. *The American Journal of Clinical Nutrition*, 108(3):564–575.

[Smith et al., 2006] Smith, C. A., Want, E. J., O'Maille, G., Abagyan, R., and Siuzdak, G. (2006). Xcms: processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching, and identification. *Analytical Chemistry*, 78(3):779–787.

[Sperber et al., 2015] Sperber, H., Mathieu, J., Wang, Y., Ferreccio, A., Hesson, J., Xu, Z., Fischer, K. A., Devi, A., Detraux, D., Gu, H., Battle, S. L., Showalter, M., Valensisi, C., Bielas, J. H., Ericson, N. G., Margaretha, L., Robitaille, A. M., Margineantu, D., Fiehn, O., Hockenbery, D., Blau, C. A., Raftery, D., Margolin, A. A., Hawkins, R. D., Moon, R. T., Ware, C. B., and Ruohola-Baker, H. (2015). The metabolome regulates the epigenetic landscape during naive-to-primed human embryonic stem cell transition. *Nature Cell Biology*, 17:1523.

[Stacklies et al., 2007] Stacklies, W., Redestig, H., Scholz, M., Walther, D., and Selbig, J. (2007). pcamethods – a bioconductor package providing pca methods for incomplete data. *Bioinformatics*, 23:1164–1167.

[Stekhoven and Bühlmann, 2011] Stekhoven, D. J. and Bühlmann, P. (2011). MissForest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118.

[Takemasa et al., 2003] Takemasa, I., Matsubara, K.-i., Sato, M.-a., Monden, M., Oba, S., and Ishii, S. (2003). A Bayesian missing value estimation method for gene expression profile data. *Bioinformatics*, 19(16):2088–2096.

[Tsugawa et al., 2015] Tsugawa, H., Cajka, T., Kind, T., Ma, Y., Higgins, B., Ikeda, K., Kanazawa, M., VanderGheynst, J., Fiehn, O., and Arita, M. (2015). Ms-dial: data-independent ms/ms deconvolution for comprehensive metabolome analysis. *Nature Methods*, 12:523.

[van der Kloet et al., 2009] van der Kloet, F. M., Bobeldijk, I., Verheij, E. R., and Jellema, R. H. (2009). Analytical error reduction using single point calibration for accurate and precise metabolomic phenotyping. *Journal of Proteome Research*, 8(11):5132–5141.

[van der Maaten, 2009] van der Maaten, L. (2009). Learning a parametric embedding by preserving local structure. In van Dyk, D. and Welling, M., editors, *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 384–391, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA. PMLR.

[van der Maaten, 2014] van der Maaten, L. (2014). Accelerating t-sne using tree-based algorithms. *Journal of Machine Learning Research*, 15:3221–3245.

[van der Maaten and Hinton, 2008] van der Maaten, L. and Hinton, G. (2008). Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605.

[Walker, 2018] Walker, A. (2018). *openxlsx: Read, Write and Edit XLSX Files*. R package version 4.1.0.

[Wattenberg et al., 2016] Wattenberg, M., Viégas, F., and Johnson, I. (2016). How to use t-sne effectively. *Distill*. http://distill.pub/2016/misread-tsne.

[Wellen et al., 2009] Wellen, K. E., Hatzivassiliou, G., Sachdeva, U. M., Bui, T. V., Cross, J. R., and Thompson, C. B. (2009). Atp-citrate lyase links cellular metabolism to histone acetylation. *Science*, 324(5930):1076–1080.

[Wickham, 2011] Wickham, H. (2011). testthat: Get started with testing. *The R Journal*, 3:5–10.

[Wickham, 2015] Wickham, H. (2015). *R Packages*. O'Reilly Media, Inc., 1st edition.

[Wickham, 2016] Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York.

[Wickham and Bryan, 2018] Wickham, H. and Bryan, J. (2018). *usethis: Automate Package and Project Setup*. R package version 1.4.0.

[Wickham et al., 2018a] Wickham, H., Danenberg, P., and Eugster, M. (2018a). *roxygen2: In-Line Documentation for R.* R package version 6.1.1.

[Wickham et al., 2018b] Wickham, H., Hester, J., and Chang, W. (2018b). *devtools: Tools to Make Developing R Packages Easier.* R package version 2.0.1.

[Wilke, 2019] Wilke, C. O. (2019). *cowplot: Streamlined Plot Theme and Plot Annotations for 'ggplot2'.* R package version 0.9.4.

[Wishart, 2016] Wishart, D. S. (2016). Emerging applications of metabolomics in drug discovery and precision medicine. *Nature Reviews Drug Discovery*, 15:473.

[Wishart et al., 2018] Wishart, D. S., Feunang, Y. D., Marcu, A., Guo, A. C., Liang, K., Vázquez-Fresno, R., Sajed, T., Johnson, D., Li, C., Karu, N., Sayeeda, Z., Lo, E., Assempour, N., Berjanskii, M., Singhal, S., Arndt, D., Liang, Y., Badran, H., Grant, J., Serra-Cayuela, A., Liu, Y., Mandal, R., Neveu, V., Pon, A., Knox, C., Wilson, M., Manach, C., and Scalbert, A. (2018). Hmdb 4.0: the human metabolome database for 2018. *Nucleic Acids Research*, 46(D1):D608–D617.

[Zhou et al., 2012] Zhou, B., Xiao, J. F., Tuli, L., and Ressom, H. W. (2012). Lc-ms-based metabolomics. *Mol Biosyst*, 8(2):470–481.