`

Developing a Model of Post-Mortem Changes in Aquatic Decomposition Chemistry for PMSI Estimation

by

Charlotte Mary Winter

23017153

A dissertation submitted for the award of

BSc (Hons) Forensic Investigation

Under the supervision of Duncan Parker

in the Department of Sport and Science

University of Staffordshire

May 2026

Word count = [13,633]

Abstract:

Throughout the years, determining the postmortem submergence interval (PMSI) of individuals discovered in aquatic environments has been an everlasting challenge for forensic practitioners. Traditional approaches continue to focus on physical signs to estimate time since death, but they are hindered by environmental and individual variability, which these methods do not account for. Since this realisation, research has accelerated on the application of Machine Learning (ML) regression models, which have the ability to incorporate complex, high dimensional metadata. These models have displayed the ability to improve the accuracy and reliability of PMSI estimation by reducing the margin of error and identifying stable biomarkers. While current postmortem interval (PMI) research has taken a data-driven approach integrating biochemical and microbial markers, aquatic decomposition chemistry remains relatively unexplored, with limited validation models using waterborne biochemical changes.

This study aimed to develop a data-driven model using random forest regression models on provided metabolomic data, where six murine models were submerged in water for a 41-day study period, to predict postmortem submergence interval (PMSI) and time since death. Following initial data processing, 1393 variables were reduced to 178 stable features for analysis. Patterns of missingness were assessed and addressed by splitting the data into early (days 0-7) and late (days 8-41) PMSI days, with imputing performed using QRILC and k-nearest neighbour (KNN), respectively. Principal Component Analysis (PCA) was subsequently performed to aid in the detection of underlying patterns as well as reducing the dimensionality for improved ML performance. Two RF models were developed using principal component inputs to predict PMSI. The model based on PC2-associated metabolites demonstrated moderate performance ($R^2 = 0.68$; RMSE = 6.23), whilst the model using all PCs (PC1-PC10) resulted in an improved $R^2 = 0.78$ and RMSE = 4.76. The generalisability of the latter model was tested using leave-one-out (LOO) validation.

The LOO model demonstrated a reduced generalisability compared to standard train-test performance, with an overall R^2 of 0.41 (RMSE = 7.74; MAE = 5). Model accuracy varied substantially between mice ($R^2 > 0.7$ to < 0.3), reflecting inter-individual metabolomic trajectories, limiting model transferability.

These findings demonstrate that aquatic decomposition chemistry contains valuable temporally informative signals that can support PMSI. Whilst generalisation may still be a challenge, the

use of PCA and RF reflected promising predictability, reinforcing its suitability for metabolomic forensic estimation.

Acknowledgments:

Firstly, I would like to give a massive thank you to Alison Davidson. You have had faith in me since the very start of this project, even when I first walked into the office having never touched code in my life. You have been there for me when I sent you continuous panicked emails, proceeding to send you another few after realising what I had done. Your quick responses and consistently making time for me has always been greatly appreciated. I am not sure where this project would be without you, and I thank you for your continued support and encouragement.

I would also love to stay thank you to my closest friends I have made during my time at Staffordshire University. Neve, I am so glad I dragged you out of your room in first year, even if iPad time was essential. You've now made Birmingham bearable because you exist. Cam, you will always be my volleyball partner, even if I can't dig straight or far enough, we have been partners since day 1. I don't know what I will do without an Italian peering over my shoulder when I'm making 'authentic' carbonara.

Emma, from the first time we met online where I then proceeded to not recognise you in real life, we have never been apart. You were such a blessing in my life, and I honestly don't know what I would have done without you. From the nights we spent laughing at TikTok's that are seriously not that funny, to getting through some of the toughest days together, the struggles seemed like nothing when I was with you. You have always been my no.1 supporter, and I can't wait till we get sexy red tickets.

Daniel, you have got me through some of the hardest times in my life and have been my rock through it all. You're my biggest supporter and have made me feel capable of everything and more. I will never know what I did to deserve you. I love you more than words can express.

Mum, I know I've been rocky these last three years, but you have constantly been there to steady me. Through it all you have always told me how strong and capable I am, and I don't know what I would have done without you. I love you lots like jelly tots.

Finally, my biggest thank you goes to you Sean. Throughout my whole time in education, you have been with me every step of the way. We started out at GCSE maths where we would sit and go through papers until I was questioning the person who even created maths. We moved onto biology A-level, where I was now questioning the people who made the mark schemes. And now here we are. Through tears, stress and exhaustion, you have been my constant.

Somehow, you've made every stressful situation fun, mostly by telling me I have a massive brain. You will always be my work dad <3.

I also want to say how proud I am of myself. These three years at university have proved to be a substantial change in my life that resulted in the uncovering of old wounds that I thought I was past. Through many tears and help from others, I was able to gather the strength to push past these issues. But it would have never happened if it wasn't for me. Before, I never truly appreciated how special I was, but these past few years have shown me the strength and courage I hold and the beautiful person I have become. I know past me is now so proud of the person I have become.

"…The cold never lasts my darling.

It just teaches the heart *how to burn"* ***– RAYE 2026***

Contents

Abstract: 2
Acknowledgments: 4
1.0 **List of Abbreviations:** 9
2.0 **List of figures:** 10
3.0 **Introduction:** 12
3.1 **Forensic Taphonomy** 12
3.2 **The chemical & biological breakdown of the body** 12
3.3 **Factors affecting decomposition** 14
3.3.1 **Environmental factors** 14
3.3.2 **Intrinsic factors** 14
3.4 **Water related deaths** 15
3.5 **Why a Data-Driven Model?** 16
3.6 **Principles of Liquid Chromatography-Mass Spectrometry (LC-MS)** 17
4.0 **Aims and objectives:** 18
5.0 **Method:** 19
5.1 **Original study** 19
5.2 **Data Curation** 19
5.2.1 **Data importing & Initial processing** 19
5.2.2 **Molecular feature extraction & Peak detection** 20
5.2.3 **Export & Analysis in Profinder** 21
5.3 **Data Preparation and Reconstructing** 22
5.4 **Missing Data Exploration** 23
5.5 **Assessment and Visualisation of Missing Data Patterns in Shortlisted Metabolites** 23
5.6 **Data Imputation** 24
5.7 **Imputed Data Visualisation and LOESS** 26
5.8 **Principal Component Analysis (PCA)** 26
5.9 **Random Forest (RF) Modelling** 26
6.0 **Identification of Metabolites Based on the Importance of Variables in RF** 28
7.0 **Results & Discussion:** 28
7.1 **Chromatographic Peak Analysis** 28
7.2 **Initial Variable Filtering** 33
7.3 **Missing Data Patterns and Distribution** 33
7.4 **Handling of Missing Data** 36

7.5 **Data Processing and Visualisation of Shortlisted Data** 37
7.6 **Imputation of Early and Late PMSI days** 40
7.6.1 **QRILC Imputation** 41
7.6.2 **KNN Imputation** 42
7.6.3 **Result of the Combined Imputation** 44
7.7 **Visualisation of Newly Imputed Dataset** 45
7.8 **Principal Component Analysis (PCA) of the Processed Dataset** 47
7.8.1 **Overview of PCA and Selection of Principle Components** 47
7.8.2 **Principal Component Analysis (PCA) of the Processed Dataset** 49
7.8.3 **Temporal Patterns and Variability** 49
7.8.4 **Outlier Analysis** 51
7.8.5 **Metabolites driving PC1 and PC2** 53
7.9 **Random Forest Modelling for PMSI Prediction** 55
7.9.1 **PMSI ~ PC2 Scores** 56
7.9.2 **PMSI ~ PCA** 57
7.9.3 **PMSI ~ LOO on PCA** 59
8.0 **Metabolite identification from RF** 61
9.0 **Further Discussion** 62
10.0 **Conclusion and further work:** 65
11. 0 **References** 67
12.0 **Appendices:** 84
Appendix A – R code for data preparation and reconstruction (section 7.3) 84
Appendix B – R code for missing data exploration / Figure 15 summary table (section 7.4) 86
Appendix C - R code for missing data exploration / Figure 16 heat map (section 7.4) 87
Appendix D - R code for missing data exploration / Figure 17 histogram with 25% threshold (section 7.4) 88
Appendix E – Rubins framework for imputation decision (section 7.4) 89
Appendix F – Assessment and Visualisation of missing data patterns / Figure 18 scatter plots for M1-M6 (Section 7.5) 90
Appendix G - Assessment and Visualisation of missing data patterns / Figure 19 all mice in one scatter plot (Section 7.5) 92
Appendix H - Assessment and Visualisation of missing data patterns / Figure 20 scatter plot with greyed out datapoints below set threshold 93
Appendix I – Data split and QRILC imputation (Section 7.6) 94
Appendix J – KNN Imputation (Section 7.6) 96

Appendix K – Visualisations of imputation using histograms and summary tables / Figures 21, 23, 25 (Section 7.6) 98

Appendix L – Imputed data visualisation and LOESS scatter plots / Figure 26 & 27 (Section 7.7) 102

Appendix M – Scree plot and Principal Component Analysis / Figures 28 & 29 (Section 7.8)..... 103

Appendix N – PCA temporal trend analysis by labelling PMSI days / Figure 30 (Section 7.8)..... 105

Appendix O – PCA outlier analysis by highlighting and labelling outliers / Figure 31 (Section 7.8) 106

Appendix P - Bar charts of top metabolite loadings for PC1 and PC2 / Figure 32 & 33 (Section 7.8) 110

Appendix Q – Random Forest model with PMSI ~ PC2 scores including an importance plot for the PCs / Figure 34 & 37 (Section 7.9) 112

Appendix R - Random Forest model with PMSI ~ PCA (PC1-PC10) including an importance plot for the PCs / Figure 35 (Section 7.9) 118

Appendix S – Leave-one-out validation using PCA model on all six mice / Figure 36 (Section 7.9) 124

Appendix T- PCA importance plot (Section 9.8)..... 131

Appendix U – Human metabolome database (HMDB) search for metabolite 964.705 (Section 9.9) 132

Appendix V – Human metabolome database (HMDB) search for metabolite 540.1128 (Section 9.9) 133

Appendix W – Human metabolome database (HMDB) search for metabolite 508.1185 (Section 9.9) 134

1.0 **List of Abbreviations:**

PMI – Post-mortem interval

BMI – Body mass index

PMSI – Post-mortem submergence interval

ADD- Accumulated degree-days

TADS – Total Aquatic Decomposition Score

LC-MS – Liquid Chromatography- Mass Spectrometry

ESI- Electrospray Ionisation

ToF- Time of Flight

MFE – Molecular feature extraction

RFE – Recursive feature extraction

RT – Retention time

LC-MS – Liquid Chromatography - Mass Spectrometry

LOD – Limit of Detection

PCA – Principal Component Analysis

PC- Principal Components

LOESS - Locally Weighted Scatterplot Smoothing

RF – Random Forest

LOO – Leave-One-Out

ML – Machine Learning

2.0 **List of figures:**

Figure 1 - Spreadsheet Column Names 20
Figure 2 - Eight steps workflow featured in batch molecular feature extraction (MFE) and recursive feature extraction (RFE) (Source: Agilent Technologies Inc. 2017) 21
Figure 3 - The main functional areas of Profinder as viewed before you begin a project (Source: Agilent Technologies Inc. 2017) 21
Figure 4 -Excel spreadsheets uploaded as objects 22
Figure 5-Examples of new ID compound names.....22
Figure 6-Newly transposed dataset format 23
Figure 7- Renamed shortlisted variables.....24
Figure 8– New separated data frames for early PMSI days (0-7) and late PMSI days (8-41) 25
Figure 9– Newly imputed data frames.....25
Figure 10– The combined dataset of early and late imputed PMSI days 26
Figure 11- The process of Random Forest (RF) regression models. A machine learning (ML) model that makes predictions by combining the results of many smaller models which are called decision trees. 27
Figure 12- Representative overlaid LC–MS chromatograms out of 1393 datapoints, displaying clear, asymmetrical peak profiles for selected compounds. 29
Figure 13- Representative overlaid LC-MS chromatograms out of 1393 datapoints, displaying poor, inconsistent peak profiles for selected compounds. 31
Figure 14- Representative overlaid LC-MS chromatograms out of 1393 datapoints, displaying co-eluting, inconsistent peak profiles for selected compounds. 32
Figure 15- Proportion of missing data across PMSI days.....34
Figure 16-Heatmap to visualise the missing data over the study period of 41 days.....35
Figure 17-Histograms showing the distribution of missing data proportions across retained variables after applying a 25% missingness threshold 36
Figure 18-Summed normalized scatter graphs showing the temporal compound changes over PMSI days for each mouse (M1-M6). 38
Figure 19-Summed normalized intensities across PMSI days for each mouse. Each point represents the summed normalized intensity across compounds for a given mouse on a specific PMSI day 39
Figure 20-Summed normalized intensities across PMSI days for each mouse. The addition of greyed out MAR datapoints to help visualise overall trends. 40
Figure 21 - Distribution of $log2$-transformed intensities before and after QRILC imputation, showing the introduction of low-intensity values corresponding to left-censored missing values. 41
Figure 22-Summary table displaying the count of missing values before and after QRILC imputation on the early PMSI dataset.....42
Figure 23-Distribution of log1p intensities before and after KNN imputation 43
Figure 25-Summary table displaying the count of missing values before and after KNN imputation on the late PMSI dataset 44
Figure 26- shows the overall distribution of measured intensity values before and after imputation 44
Figure 27- New imputed summed normalized scatter graphs showing the temporal compound changes over PMSI days for each mouse (M1-M6). 45

Figure 28-Smoothed trajectories (LOESS) of summed normalised metabolite intensities across PMSI days for M1-M6. 46
Figure 29-Scree plot showing the variance explained by each principal component 48
Figure 30- Principal Component Analysis (PCA) of metabolomics data from early and late PMSI samples. Points represent individual samples, coloured by PMSI group. 49
Figure 31 - PCA plot with PMSI days marked on each datapoint to identify group clustering 50
Figure 32 - PCA plot but with the addition of outliers, marked and labelled. 51
Figure 33- Bar chart showing the top metabolites contributing to the variance along PC1 ranked by absolute loading values 53
Figure 34- Bar chart showing the top metabolites contributing to variance along PC2 ranked by absolute loading values. 54
Figure 35 - Observed versus predicted PMSI values for the random forest model based on top PC2 metabolites. The dashed red line indicates perfect prediction. 56
Figure 36- Observed versus predicted PMSI values for the random forest model based on the first 10 PCs. The dashed red line indicates perfect prediction. 58
Figure 37 -Observed vs Predicted PMSI by Mouse Using PCA-Based Random Forest with Leave-One-Mouse-Out Validation 60
Figure 38- Importance plot showing the top metabolites in PC2 that drive PMSI prediction. 61

3.0 **Introduction:**

3.1 **Forensic Taphonomy**

Forensic Taphonomy has been widely defined as the scientific study of postmortem processes that occur in the human body, from the time of death till discovery and analysis (Pokines and Symes, 2013). The primary focus of this field is based on the destructive transformations that occur after death, such as autolysis and putrefaction, which have been extensively mentioned in recent research (Shedge *et al*, 2023). These processes are known to be highly variable, with their progression influenced by environmental and individual factors (Castro *et al*, 2023).

Traditional methods for estimating postmortem interval (PMI) include algor mortis (body cooling), rigor mortis (muscle stiffening), and liver mortis (blood pooling), and have long been utilised in forensic investigations during the first 24-72 hours (Shrestha *et al*, 2023). However, in cases of unknown deaths, as highlighted by Strete *et al*. (2025), traditional methods such as algor mortis, rigor mortis, and liver mortis diminish in accuracy within the first two or three days after death due to their reliance on physiological alterations. As a result, their applicability becomes limited in cases involving advanced decomposition or prolonged postmortem intervals (Ibrahim *et al*, 2025).

Recent research has increasingly focused on postmortem dynamics, defining the significant disruptions that occur in the body due to oxygen depletion, enzymatic reactions, tissue breakdown and immunological suppression, leading to extensive biological changes (Kalanjali & Isukapatla, 2015; Finley *et al*, 2014). These changes gave rise to measurable chemical markers that have proved to assist in postmortem analysis (Javan, 2024), allowing insight into key forensic questions such as identification, cause and manner of death as well as Post-Mortem Interval (PMI) (Shrestha *et al*, 2023; Ikpa *et al*, 2024).

Despite PMI estimation involving various physical, chemical and biological changes, its accuracy remains limited due to environmental variability and differences in decomposition rates, preventing the development of a universally reliable method suitable for forensic application (Wells, 2018; Weisensee & Atwell, 2024).

3.2 **The chemical & biological breakdown of the body**

The breakdown of the human body after death has been extensively characterised in forensic literature as a series of complex processes, primarily driven by autolysis and putrefaction (Schotsmans *et al*, 2017). These processes are commonly categorised into five stages: fresh,

bloat, active decay, advanced decay and skeletonization, with the boundaries between each stage often described as fluid rather than discrete (Tibbett & Carter, 2008).

The fresh stage (18-20 hours after death) has been widely associated with the onset of autolysis, with cells releasing acid hydrolases (digestive enzymes) that degrade cellular components, leading to the rupture of cell membranes and subsequent tissue breakdown (Abdulaziz *et al*, 2023; Osamura *et al,* 2023). In the early stages of decomposition, algor mortis, rigor mortis and liver mortis are also apparent and have been consistently mentioned in literature (Saukko, 2023). Algor mortis refers to the internal cooling of the body, where heat is transferred to the surrounding ambient temperatures due to the ceasing of thermoregulation (Eden & Thomas, 2018). The body also undergoes rigor mortis, where chemical changes in the body result in the stiffening of the muscles. This is caused by the formation of chemical bridges between actin and myosin in the muscle tissues, due to the depletion of ATP which is required to separate the muscle filaments (Rattenbury, 2018; Ralebitso-Senior, 2018). This process is further reinforced as lactic acid begins to accumulate within the muscles as a byproduct of anaerobic metabolism, resulting in the decline of pH in the muscle tissue, contributing to the permanent binding of muscle filaments (Mccorry *et al*, 2019). Finally, livor mortis, also known as postmortem lividity, is reported as the bluish-purple discolouration of the skin that occurs after death due to the blood settling in the lowest parts of the body by gravity (Cohen *et al*, 2025).

The transition to the bloat stage (2-6 days after death) has been marked by the rapid multiplication of bacteria (Abdulziz *et al*, 2023). Without oxygen, aerobic bacteria die off, allowing anaerobic bacteria, that originate from the gastrointestinal tract, to thrive and proliferate. As they deplete their usual food sources, they begin to digest internal tissue which releases numerous gases, including hydrogen sulphide, methane and carbon dioxide, causing the body to bloat and swell (Hentges, 2019; Goff, 2009).

Active decay, typically occurring 5-15 days after death has been described as the most dynamic phase of decomposition, characterised by the high rate of mass loss through the liquidation of soft tissues and organs (Abdulziz *et al,* 2023). Literature highlights the role of insect activity, particularly larvae colonisation, that accelerate tissue consumption, alongside the release of intense odours as gases expel from the body (Carter & Tibbett, 2009). This stage results in the reduction soft tissue, exposing skin, cartilage and bone.

This is followed by advanced decay, occurring between 10-25 days or several months, where insect activity depletes and soft tissue is largely gone, creating a ‘caved-in’ appearance as the

remaining skin dries out (Shrestha *et al*, 2023). The final stage, skeletonization, is characterized by the complete loss of soft tissue, leaving only skeletal remains (Shrestha *et al*, 2023).

As a result, the duration of these stages is highly variable and can last from a few weeks to several years, limiting the reliability of stage-based estimations in forensic contexts.

3.3 **Factors affecting decomposition**

3.3.1 **Environmental factors**

To minimize uncertainty in PMI calculations, it has been emphasised in literature to examine the different internal and external variables that influence the rate and pattern of decomposition (Kõrgesaar *et* al). Temperature has been identified to be one of the most significant factors due to its role in regulating various ecosystems, such as the rate of organic matter breakdown and the microbial communities involved (Martlin *et al*, 2022). Forensic studies conducted across each season and in different geographical areas have demonstrate that in higher temperatures during the summer, decomposition accelerates, inducing rapid bloating, whereas in colder conditions in the winter, remains can retain a fresh appearance for a considerably long time (Iancu *et al*, 2018; Galloway *et al*, 1989). This can be further explained by considering the degree of preservation or destruction of the corpse, the function of the surrounding decomposer population, the quality of the resources being decomposed, and the cultural and environmental modulators, all of which combine to create a unique and yet ephemeral decomposition environment (Carter et al, 2007). Similarly, moisture availability, soil composition and microclimatic conditions have been mentioned to further influence decomposition (Wang *et al*, 2010) due to being primary drivers in contributing to the congregation of microbial communities and fungal biomass (Chen *et al*, 2025).

3.3.2 **Intrinsic factors**

In recent studies, researchers have found that even under similar environmental conditions, individuals can exhibit varying decomposition patterns, highlighting the role that intrinsic factors have on the rate of decay (Mason *et al*, 2022). Characteristics inherent to the cadaver, such as body mass index (BMI), age, sex and cause of death, have been shown to affect biomolecule degradation and microbial responses (Garrett-Rickman, 2024). At a cellular level, death involves the complex process of programmed cell death and nuclear DNA degradation, with cells and tissue showing differential resilience, which has been put forward as to shape

the trajectory of post-mortem decay (Javan *et al*, 2024). Furthermore, BMI has been able to explain variation in soil pH and microbial community changes, with underweight individuals showing minimal changes in microbial groups compared to normal, overweight and obese categories which exhibit an abundance of microbial activity over time (Mason *et al*, 2022). Individuals with a higher BMI release larger amounts of decomposition bioproducts that alter the pH of the soil (Mason *et al*, 2020). This is further supported by Shedge *et al*. (2023), who emphasised that individual characteristics (BMI, age, physical condition etc.) significantly influenced the rate and presentation of decomposition changes.

3.4 **Water related deaths**

Human remains can be found in aquatic environments for several reasons, such as recreational accidents, natural disasters, suicides or attempted disposals of homicide victims and drownings, collectively accounting for more than 300,000 worldwide deaths annually (Palmer, 2020; WHO, 2024). Despite the cause of death being known, accurately estimating post-mortem submergence interval (PMSI) remains a significant problem for modern forensics (Martlin *et al*, 2022; Dalal *et al*, 2023). Alongside geographical challenges posed by underwater environments, research by Schotsmans *et al*. (2017) demonstrated that aquatic decomposition differs significantly from terrestrial due to factors such as lower temperatures, limited oxygen and the absence of terrestrial insects, prolonging the progression of postmortem changes. This is further commented on by Siva *et al*. (2024), who identified several aquatic-specific variables such as water salinity, dissolved oxygen, pH and flow that can influence the degradation and preservation of human remains.

Due to these conditions, unique postmortem features have been identified, including washerwomen's hands/feet (wrinkling of the skin due to prolonged exposure), skin sloughing (skin can detach or 'deglove') and adipocere formation (grave wax that forms on the body) (Caruso, 2016). Moreover, research has also considered the water type in which victims are found, with bodies located in freshwater displaying accelerated preliminary signs of decay, compared to those in saltwater that exhibit bloating and increased buoyancy (Jangid *et al*, 2025). Water currents can further alter the original position of the body, causing abrasions by pushing it against underwater objects (Caruso, 2016).

While these observations contributed to the identification of general patterns in aquatic decomposition, the study emphasised that such changes were highly variable to environmental

and biological factors, hindering its use for accurately estimating PMSI on its own. The limitation of observational methods was reinforced by Magni *et al.* (2021) who demonstrated that adipocere formation was highly dependent on factors such as ambient temperature, oxygen levels and water conditions. In particular, low oxygen and moist conditions could slow or halt decomposition altogether, leading to prolonged preservation of the body. Therefore, although these physical signs can provide useful contextual information, its presence introduces further uncertainty in PMSI prediction due to its unpredictable onset and progression.

3.5 **Why a Data-Driven Model?**

Due to the limited research on aquatic decomposition, current methods consist of using accumulated degree days (ADD) that measures the thermal energy (time and temperature) required for decomposition, alongside the Total Aquatic Decomposition Score (TADS). TADS is a scoring method that quantifies the degree of decay in submerged human remains by summing facial, body and limb decomposition points (Jangid, 2025). Recent research by Jangid *et al*. (2025) demonstrated a strong correlation between TADS and PMSI ($R^2 = 0.925$, $p < 0.001$) which allowed the creation of regression models linking ADD values to decomposition stages, allowing for a more objective estimation of PMSI. This method was further supported by Dalal *et al*. (2023) whose study analysed 53 real drowning cases which also demonstrated a strong correlation between TADS and ADD ($R^2 = 0.917$).

However, these studies have also demonstrated that whilst TADS and ADD show strong correlation, estimations based on the two can overpredict PMSI due to being heavily influenced by environmental variables (Dalal *et al*, 2023). This is highlighted by Cockle & Bell (2015), where under different climatic and environmental conditions, standard PMSI formulae failed to perform across regions, reinforcing the need for region-specific models to improve accuracy across diverse aquatic ecosystems and climates.

In response to this, recent research has been able to link postmortem metabolic changes with PMI extensions, providing a potential strategy for estimating PMSI using metabolome data (Zhang *et al*, 2024). As mentioned previously, death results in very extensive biological changes within the metabolite content of the corpse (Donaldson & Lamont, 2013). By analysing these metabolic shifts through advanced techniques like liquid chromatography - mass spectrometry (LC-MS/MS) and Machine Learning algorithms (ML), researchers have gained the advantage of examining multi-variant data patterns over time to establish an

effective ‘time fingerprint’ mathematical model (Wang *et al*, 2022). This has therefore served as a more objective and potentially more precise molecular framework for PMSI estimation compared to traditional methods.

However, despite the increased application, the integration of metabolomic data within aquatic decomposition remains relatively underexplored, particularly in relation to chemically driven models of PMSI estimation. Therefore, this study aims to build on existing metabolomic approaches by developing a data-driven model of postmortem changes in aquatic decomposition chemistry to improve PMSI estimation.

3.6 **Principles of Liquid Chromatography-Mass Spectrometry (LC-MS)**

Liquid chromatography-mass spectrometry (LC-MS) is a widely used analytical chemistry technique that combines the physical separation capabilities of liquid chromatography with the high-sensitive mass analysis of mass spectrometry (MS) (Agilent, 2024). Its application is oriented toward the separation, general detection, and potential identification of chemicals that are contained within complex mixtures (Ayhan *et al*, 2021). A typical LC-MS system consists of a solvent pump, an injector to introduce the sample, a column to facilitate separation of the analyte and a mass spectrometer detector (Houck, 2023).

The technique first begins with liquid chromatography (LC) which separates the mixture into its individual compounds. Compounds are separated based on their chemical affinity for a stationary phase (packed in a column) and a liquid mobile phase (solvent). This separation is essential to reduce sample complexity before it enters MS (Ardrey, 2003). As the liquid sample is injected and passed through the column, various compounds move through the column at different rates depending on their chemical characteristics such as polarity, size and chemical interactions (Worsfold *et al,* 2005). Components therefore exit the column at different times (retention time) based on how strongly they interact with the stationary phase (Shi *et al*, 2004).

Upon elution from the LC system, analytes enter the mass spectrometer, where they are ionised (given charge), typically using Electrospray Ionization (ESI) (Korfmacher et al, 2005). ESI is a soft ionisation technique in which high voltage is applied to a liquid sample, generating charge droplets that undergo solvent evaporation and coulomb fission which ultimately produces gas-phase ions for MS analysis (Wilm, 2011). This technique is well suited for biological molecules due to its ability to preserve molecular integrity as metabolites, proteins and peptides are often fragile and structurally complex (Ho *et al*, 2003).

Following ionisation, ions can be separated within the mass analyser, such as Time-of-Flight (ToF) or an ion trap, allowing the ions to be separated based on their mass-to-charge ratio (m/z), enabling the determination of molecular weight and structure (Morris & Langari, 2016). In ToF analysers, ions are accelerated through an electric field and travel along a flight tube, with lighter ions reaching the detector faster than heavier ones due to their higher velocities (Ferrer & Thurman, 2003) This allows m/z to be determined based on the time taken for ions to reach the detector. These systems have been widely recognised in literature for their high mass accuracy, rapid acquisition speed, and ability to detect a broad range of ions simultaneously (Zhang & Henion, 2001), making them suitable for complex and untargeted metabolomic analysis.

As the separated ions reach the detector, they generate electrical signals proportional to their abundance. These signals can then be translated and displayed on a mass spectrum, where the x-axis corresponds to m/z and the y-axis to signal intensity, while peak position indicates ion identity and peak intensity reflects relative abundance, allowing for compound identification and quantification (Griffiths *et al*, 2009).

4.0 **Aims and objectives:**

The aim of this project was to improve postmortem submergence interval (PMSI) prediction by building on existing metabolomic approaches to develop a data-driven model using aquatic decomposition chemistry. The objectives undertaken to achieve this aim were:

- To curate and extract LC-ToF-MS metabolomic data using Agilent MassHunter Profinder Software.
- Explore the missing data to assess the appropriate method of imputation (QRILC and KNN)
- Identify temporal metabolite trends across PMSI using data normalisation and visualisation techniques.
- Apply multivariant approaches (PCA) to visualise patterns and key metabolites associated to PMSI
- Implement machine learning approaches (Random Forest) to explore its potential for estimating PMSI based on the metabolomic data.

5.0 **Method:**

It is important to note that all code used for data processing, analysis and modelling is provided within Appendix A -S. References to specific scripts are indicated where relevant.

5.1 **Original study**

The preliminary research underpinning this work was conducted by Jenks (2022) at The University of Staffordshire and explored the application of thanatochemistry on the chemical and physiochemical changes in murine models submerged in artificial aquatic environments. The original study implemented the use of water quality parameters and the chemical processes of mammalian decomposition to develop a potential method for estimating post-mortem submergence interval (PMSI) (Jenks, 2022).

The main study involved the use of six mice submerged in ultra-pure water over a six-week period with water samples collected at regular intervals. Water quality parameters, PH and conductivity were analysed alongside the use of the total aquatic decomposition scoring method (TADS) (Jenks, 2022).

The study utilised six unspecified species of mice (three females and three males) weighing 18.94g – 23.71g and were ethically sourced from Northampton reptile centre. Time since death, age and manner of death were unknown, so standard assumptions were made (i.e. asphyxiation by carbon monoxide). Prior to experimentation, all mice were stored at -18ºC for approximately 22 months (Jenks, 2022).

Targeted LC-ToF-MS analysis indicated significant changes in water conditions and identified decomposition-related compounds with temporal trends that had the potential to accurately predict PMSI, but further research was required to confirm this hypothesis (Jenks, 2022). As a result, the purpose of this study is to determine if they are capable of estimating PMSI using machine learning methods (ML).

5.2 **Data Curation**

5.2.1 **Data importing & Initial processing**

LC-ToF-MS data from the six decomposing mice were provided by Alison Davidson at The University of Staffordshire. The primary data consisted of 1393 data points representing chemical compounds detected in the water samples collected between 24/11/2021 to

06/01/2022, across 16 post-mortem submergence intervals (PMSI), spanning from day 0 to day 41.

The excel spreadsheet was exported into Microsoft Excel for preprocessing and organising. Non-numeric metadata (e.g. molecular formula and CAS numbers) were removed to allow for quantitative analysis and reduce redundancy in the dataset. This step ensured only variables suitable for statistical comparison were retained. A working file was created with additional columns ("Yes or No" and "Reason"), to record inclusion decisions and justifications for each variable (Figure 1).

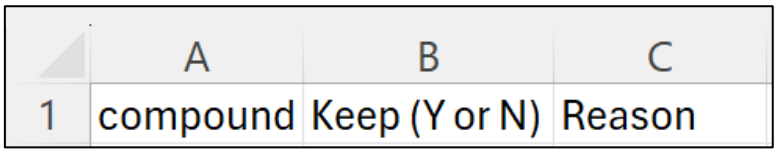


Figure 1 - Spreadsheet Column Names

5.2.2 **Molecular feature extraction & Peak detection**

LC-MS chromatograms were processed and analysed via Agilent MassHunter Profinder Software B.08.00 series (v13.0) to extract Retention Time (RT), mass-to-charge (m/z) ratios and peak areas. The workflow followed a multi-step pipeline involving feature detection, filtering, alignment and post-processing to systematically refine raw spectral data into reliable compound features (Figure 2). Batch Molecular Feature Extraction (MFE) was applied to identify co-eluting ions based on retention time (RT), charge state, accurate mass and isotopic patterns and was followed by Recursive Feature Extraction (RFE) to refine peak detection and recover low-intensity features, thereby improving overall sensitivity and feature consistency across samples.

Chromatographic alignment was performed using retention time (RT), to ensure consistent compound elution across samples. Peaks were matched based on RT, mass-to-charge ratio and peak shapes. A minimum abundance parameter of 2000 counts were applied to reduce background noise and exclude low-confidence features.

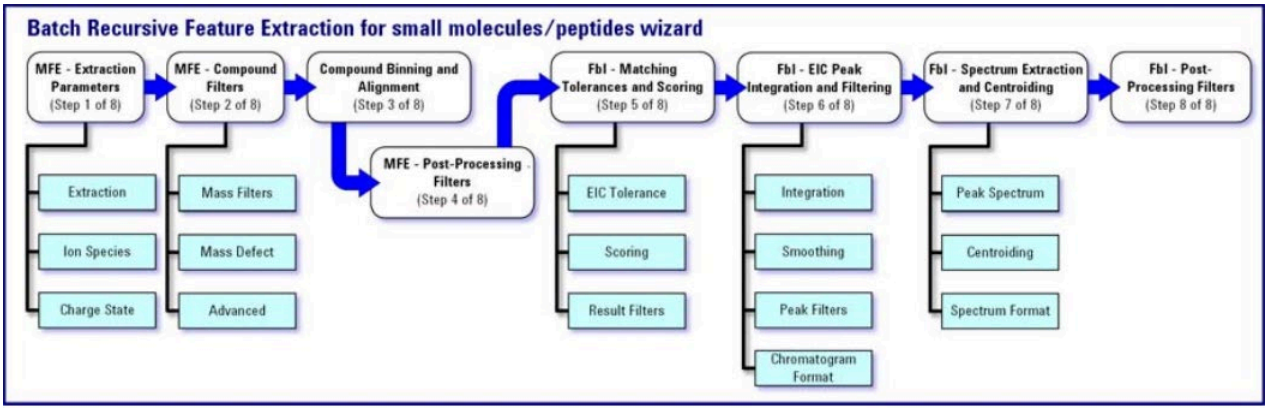


Figure 2 - Eight steps workflow featured in batch molecular feature extraction (MFE) and recursive feature extraction (RFE) (Source: Agilent Technologies Inc. 2017)

5.2.3 **Export & Analysis in Profinder**

Following processing, 1393 data points were obtained, with chromatographic and mass spectral data available for each compound. Peaks were selected based on symmetry, sharpness, and clear resolution, while those deviating from a gaussian profile were excluded. Anomalies were noted for traceability. Alignment of the peaks via the 'overlay' feature, was performed using the retention time (RT) of each ensuring that the same compound elutes at the same time across different data files (Figure 3).

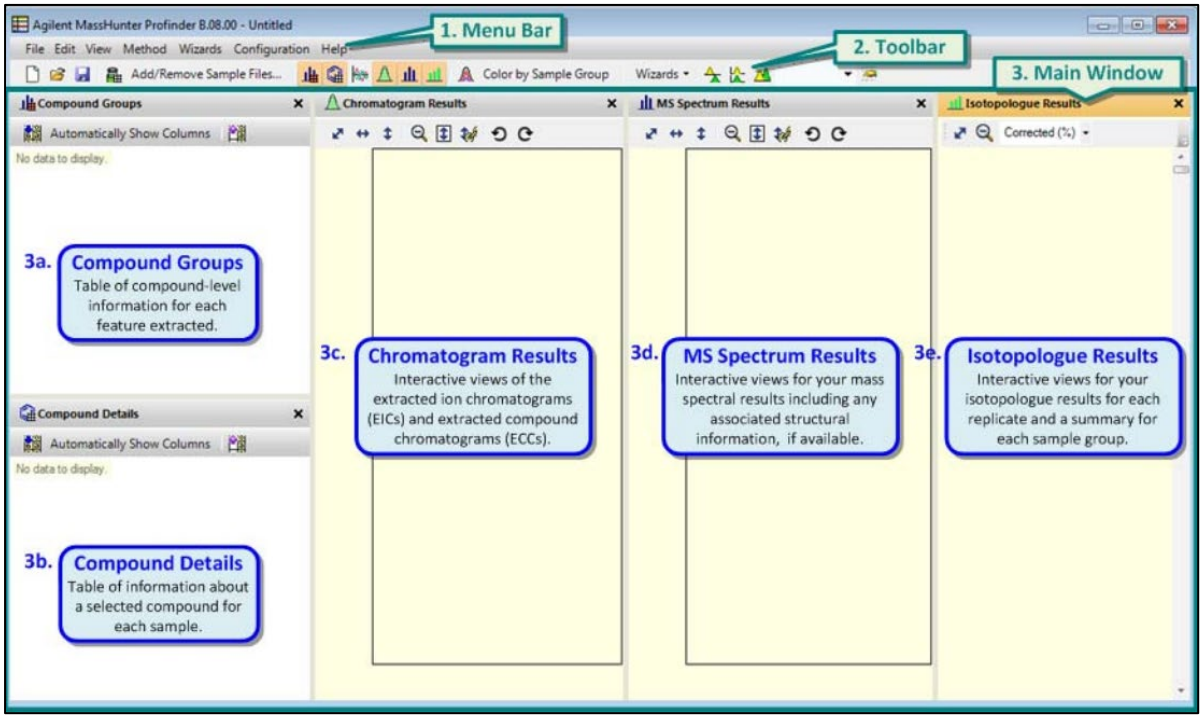


Figure 3 - The main functional areas of Profinder as viewed before you begin a project (Source: Agilent Technologies Inc. 2017)

Each compound was assigned an inclusion detail ("Yes or No") with the justification recorded in the working file. Following completion of data processing, all variables were prepared for downstream analysis in Rstudio (v.4.5.2), with Quarto used for integrated code and output generation.

5.3 **Data Preparation and Reconstructing**

The data was imported into Rstudio (v.4.5.2), where the simplified and detailed dataset was converted into data frames (Figure 4). The simplified dataset was filtered using an inclusion flag from the detailed file, retaining only variables marked 'Yes'.

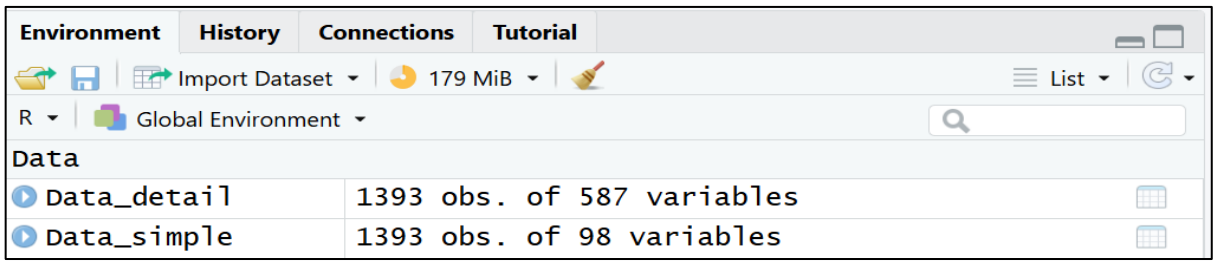


Figure 4 -Excel spreadsheets uploaded as objects

Unique compound identifiers were generated by combining mass-to-charge ratio and retention time (RT), with a standardised prefix applied to ensure compatibility within R (Figure 5). Redundant columns were removed.

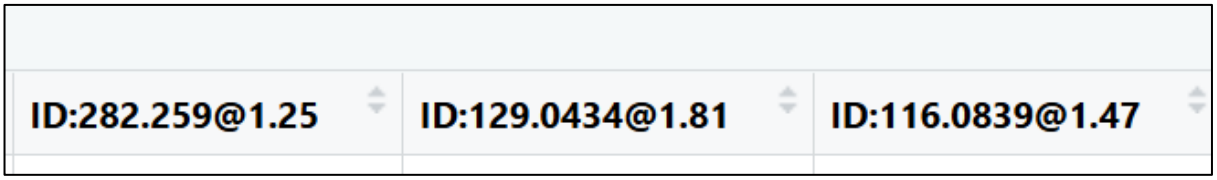


Figure 5-Examples of new ID compound names

The dataset was transposed using *data.table* (Barret *et al*, 2025) to a sample-by-variable format, with compound IDs as columns and samples as rows. Metadata was extracted from the file using *tidyr* (Wickham *et al*, 2025), including sample date and mouse ID, and then standardised using *lubridate* (Grolemund & Wickham, 2011), to ensure correct chronological ordering of PMSI days (Figure 6).

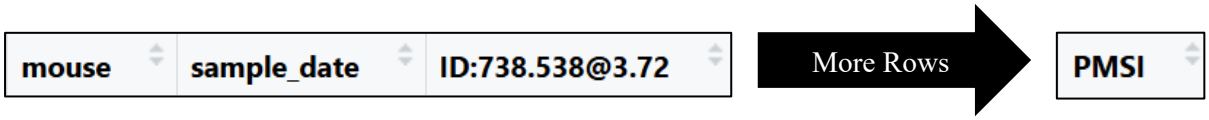


Figure 6-Newly transposed dataset format

(See full R script in Appendix A)

5.4 **Missing Data Exploration**

Following data reconstruction, missing data patterns were assessed across PMSI groups. Summary statistics were calculated per time point, and the proportion of missing values was determined as a percentage of total observation (Appendix B).

Visualisation techniques, including heat maps and histograms, were created using *naniar* (Tierney & Cook, 2023), *flextable* (Gohel & Skintzos, 2025) and *dplyr* (Wickham *et al*, 2023), and were used to examine the distribution and extent of missingness (Appendix C). Variables exceeding a 25% missingness threshold were excluded from the dataset (Appendix D). Further analysis was conducted to determine the type of missingness and inform the selection of appropriate imputation methods.

Missing data mechanisms were considered following Rubin's framework (Appendix E). Early PMSI days were treated as missing not at random (MNAR), consistent with left-censored metabolite measurements, while later-stage missingness was assumed to be missing at random (MAR).

Based on these patterns, early PMSI missing values were imputed using QRILC, while later-stage missingness was addressed using k-nearest neighbour (KNN) imputation.

5.5 **Assessment and Visualisation of Missing Data Patterns in Shortlisted Metabolites**

The remaining variables had only been printed to the console and not stored as a separate object. To formalise this shortlisted set for analysis, a new dataset was renamed and created in the environment (Figure 7).

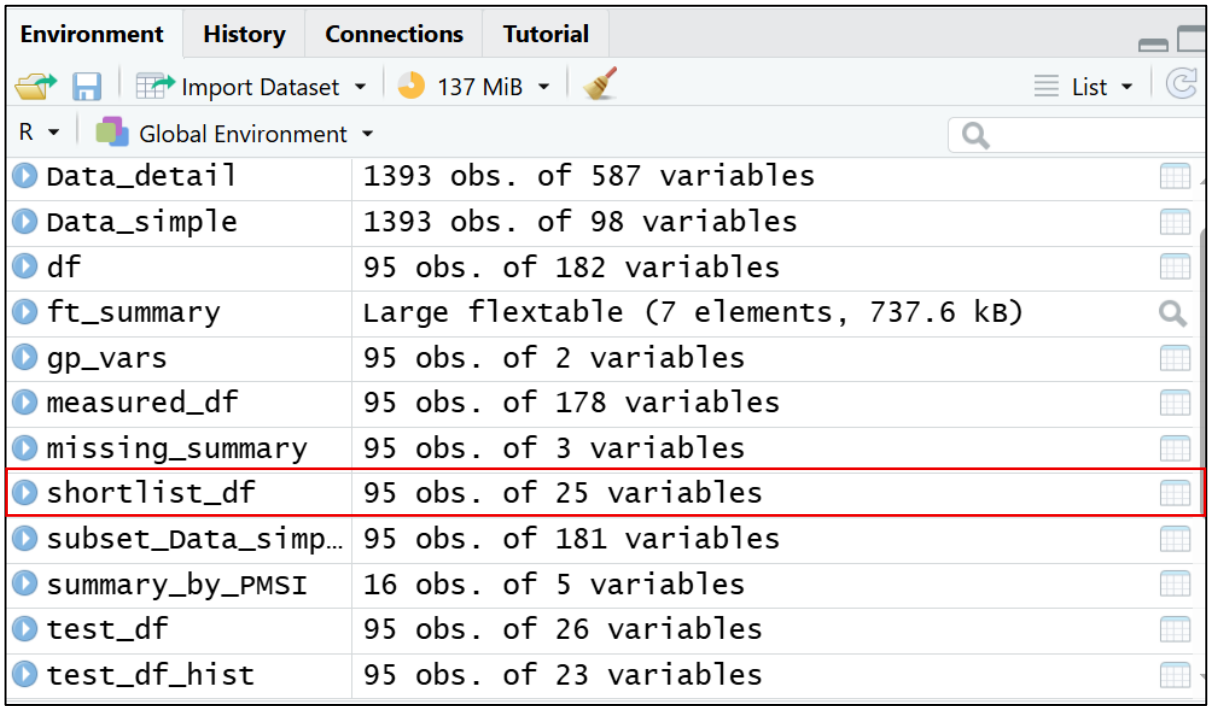


Figure 7- Renamed shortlisted variables

Changes in metabolite intensities were visualised by *dplyr* (Wickham et al, 2023) and *ggplot2* (Wickham, 2016), which generated summed and normalised scatter plots across PMSI days (Appendix F).

To account for variation between mice, compound intensities were min-max normalised within each mouse. For each sample, a summed normalised score was calculated to represent overall metabolite abundance.

These scores were plotted against PMSI days to show temporal compound changes during the process of decomposition. Scatter plots were generated to display the trajectories for each mouse individually, as well as combined (Appendix G). Low intensity datapoints were highlighted to aid the interpretation of patterns across the decomposition timeline (Appendix H).

5.6 **Data Imputation**

Based on observed missing data patterns, two imputation methods were applied: Quantile Regression of Left-Censored data (QRILC) for early PMSI and K Nearest Neighbour (KNN) for late PMSI.

The dataset that contained the 23 compounds was split at day 7 into early (days 0-7) and late (days 8-41) PMSI groups using *dplyr* (Wickham *et al*, 2023) (Figure 8).

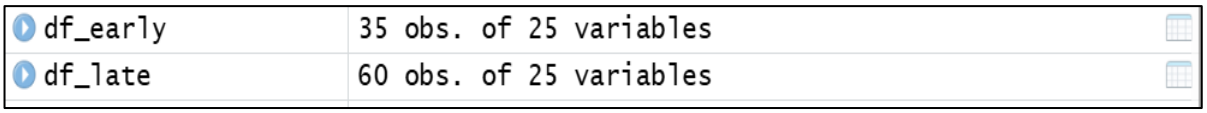


Figure 8– New separated data frames for early PMSI days (0-7) and late PMSI days (8-41)

For the early dataset, *BiocManager* (Morgan & Ramos, 2025) was used to install the Bioconductor *MSnbase* (Gatto & Lilley, 2011) and i*mputeLCMD* (Lazar & Burger, 2022). *MsCoreUtils* (Rainer *et al*, 2022) was also loaded to support matrix functions used in QRILC imputation.

For QRILC, feature data was isolated from metadata, log_2-transformed and imputed to account for left-censored (MNAR) values before returning to the original scale and recombined (Figure 9) (Appendix I)

For the late dataset, *VIM* (Kowarik & Templ, 2016) and *dplyr* (Wickham et al, 2023) were used to carry out KNN imputation. Metadata was excluded prior to imputation and logp1 transformed. Logp1 transformation applies a natural logarithm to (1 + x), providing a numerically stable way to handle skewed data, allowing zero values to be included in transformation. KNN imputation (k =5) was applied, with missing values estimated within each mouse to preserve temporal consistency. Data was then back transformed and recombined with the metadata (Figure 9) (Appendix J)

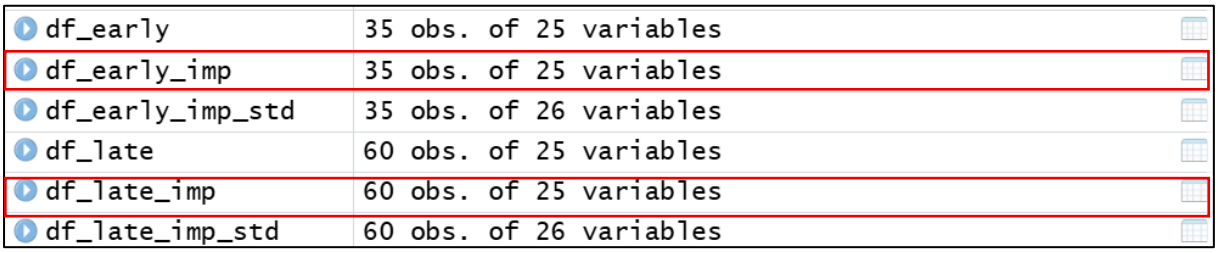


Figure 9– Newly imputed data frames

The datasets were to be recombined into a single data frame (Figure 10) using *dplyr* (Wickham et al, 2023), ordered by mouse ID and PMSI (Figure 10). Visualisation of data distributions before and after imputation was performed using *ggplot2* (Wickham, 2016) to confirm imputation effects (Appendix K).

df_combined	95 obs. of 26 variables

Figure 10– The combined dataset of early and late imputed PMSI days

5.7 **Imputed Data Visualisation and LOESS**

Using *dplyr* (Wickham *et al*, 2023) and *ggplot2* (Wickham, 2016), the same summed and normalised scatter plot method explained in section 5.5 was utilised with the newly imputed data, with no other modifications other than the use of the imputed set, and the addition of locally estimated scatterplot smoothing (LOESS).

(See full R code in Appendix L)

5.8 **Principal Component Analysis (PCA)**

Following imputation, Principe Component Analysis (PCA) was performed on the full dataset to identify underlying patterns in metabolite profiles. The analysis was conducted using *dplyr* (Wickham *et al*, 2023), *tidyr* (Wickham *et al*, 2025) and *ggplot2* (Wickham, 2016) for data manipulation and visualisation. Metabolite intensity variables were selected and *log1p* transformed to reduce skewness and stabilise variance

PCA was conducted on mean-centred and scaled data, with variance explained assessed using a scree plot. Using *ggrepel* (Slowikowski *et al*, 2026) principal component scores (PC1 and PC2) were combined with sample metadata to evaluate clustering by PMSI group (Appendix M), identify temporal trends (Appendix N) and analyse outliers (Appendix O). A confidence ellipse was created around each group to help show clustering tendencies and group separation in the reduced dimensional space.

Metabolite loadings were extracted to determine variables contributing most to the variation along PC1 and PC2, with key metabolites visualised based on magnitude and direction of contribution (Appendix P & Q)

5.9 **Random Forest (RF) Modelling**

Random Forest (RF) regression models were created to predict postmortem submergence intervals (PMSI) using the metabolome data. Prediction variables consisted of metabolite features, with PMSI treated as a continuous response variable. Random forest was selected due

its suitability for high-dimensional metabolomic data, robustness to outliers, and ability to identify important predictive variables.

RF models were implemented using *caret* (Kuhn, 2008) and *ranger* (Wright & Zeigler, 2017) and were trained using 500 trees and permutation-based variable importance. Hyperparameter tuning was performed using repeated k-fold cross validation (5-fold cross-validation repeated 5 times) (Figure 11) allowing the model to be trained and evaluated across multiple data partitions to improve robustness and reduce overfitting.

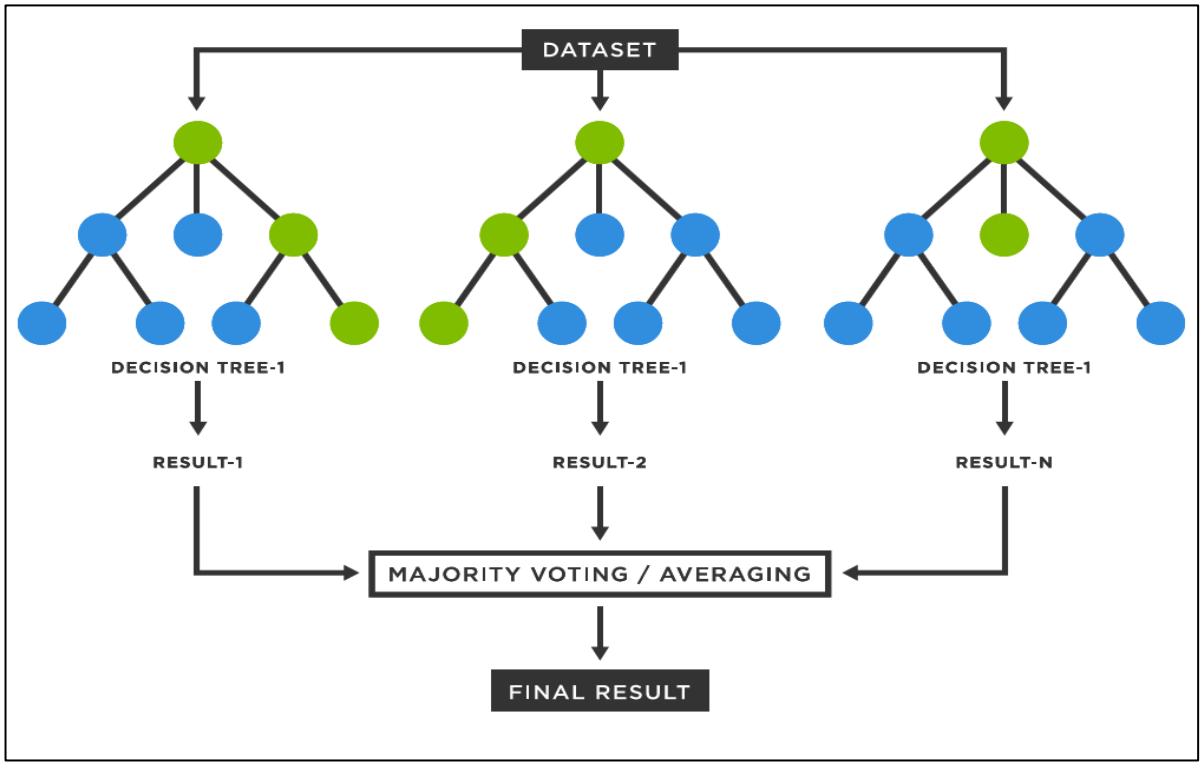


Figure 11- The process of Random Forest (RF) regression models. A machine learning (ML) model that makes predictions by combining the results of many smaller models which are called decision trees.

Two modelling approaches were evaluated: feature-selected model using top PC2 metabolites (Appendix Q) and a PCA based model using PC scores (PC1-PC10) (Appendix R)

Model performance was assessed using the metrics Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and Coefficient of determination (R2), based on an 80:20 train-test split. The RF model learns the data patterns from the training set (80%), while the test set (20%) will serve as an independent check to ensure the model generalises well and avoids overfitting. RMSE was used to quantify the overall magnitude prediction error, with greater weighting applied to larger deviations. MAE represents the absolute difference between observed and

predicted PMSI values, providing an interpretable measure of error in the same units as PMSI. R2 reflects the proportion of variance in PMSI explained by the model. Final model selection was based on minimising RMSE.

Observed versus predicted PMSI values were visualised using scatter plots with a 1:1 reference line to assess model agreement. Generalisability was further assessed using leave-one-out (LOO) validation on the PCA-based model Appendix S). For the PCA-based LOO models, PCA was recalculated within each training set, and the held-out mouse data was projected into the relevant PCA space before prediction. Performance was evaluated overall and on a per-mouse basis using *ggplot2* (Wickham, 2016).

(See full R code in Appendix H)

6.0 **Identification of Metabolites Based on the Importance of Variables in RF**

When generating the random forest (RF) regression models, an additional importance plot was created to identify the top metabolites contributing to PMSI. Permutation importance was used as it quantifies the contribution of each metabolite to model performance by measuring the decrease in predictive accuracy when its values are randomly shuffled. The mass spectral (m/z) features of these metabolites were input into the Human Metabolome Database (HMDB) as well as positive Ion mode, unknown adduct type, and a molecular weight tolerance of +-5 ppm (parts per million).

(See full R code in Appendix I)

7.0 **Results & Discussion:**

7.1 **Chromatographic Peak Analysis**

Extracted RT, mass-to-charge (m/z) ratios and peak area data was used to assess compound variation across PMSI. The peaks were aligned via the 'overlay' feature to ensure the same compounds elute at the same time across M1-M6. Although compounds may be identical, they can appear at different times during runs due to factors like column aging, temperature and flow rate variation. Overlay alignment enabled comparison across all mice (M1-M6) over the 41-day period, revealing temporal trends in compound profiles and identifying sample-specific anomalies (Agilent Technologies, 2017).

There was no specific order of the chromatograms, but due to the various jumps (day 23-day 0), it can be assumed that it was based on the order in which the raw files were originally imported.

Figure 12 presents representative extracted chromatograms, alongside corresponding mass spectra for a selected analyte across the PMSI days. The chromatograms (left panel) display signal intensity as a function of RT, while the mass spectra (right panel) show ion intensity across mass-to-charge (m/z) ratios for the same compound at each time point.

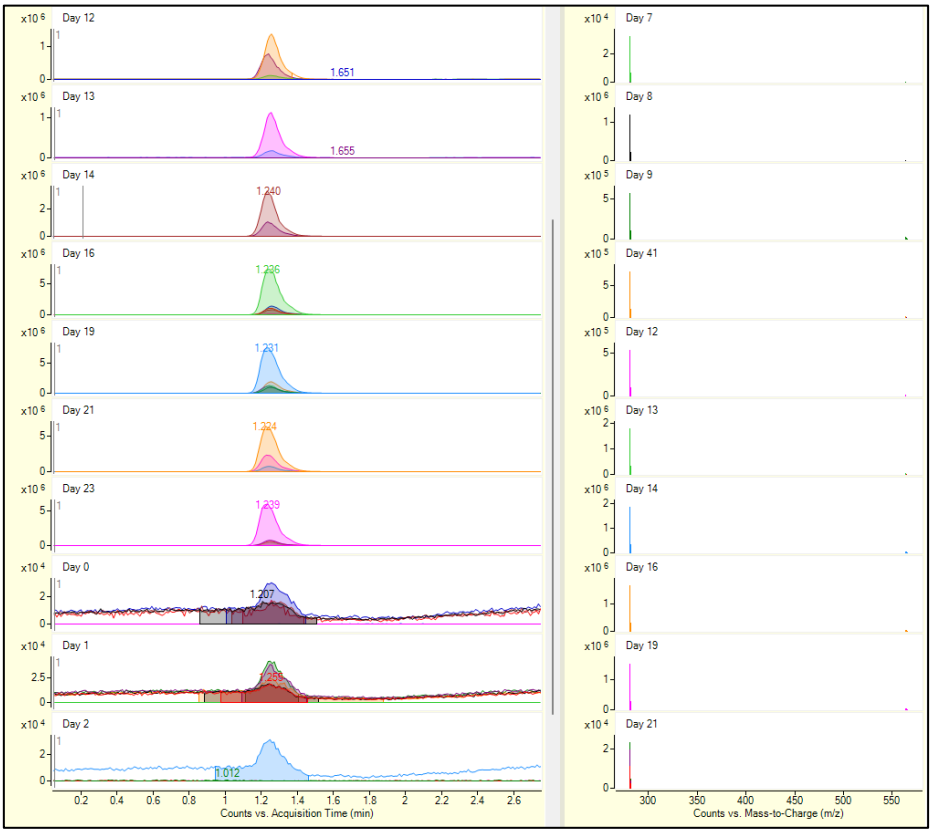


Figure 12- Representative overlaid LC–MS chromatograms out of 1393 datapoints, displaying clear, asymmetrical peak profiles for selected compounds.

Consistent and well-defined peaks were observed at approximately 1.24 ± 0.12 min, remaining stable across the sampling days, with a range of 1.01-1.66, indicating reproducible retention behaviour and suggesting the detection of a single, stable metabolite. Peaks were initially broad and noisy on Day 0 but became progressively sharper and more symmetrical over time. In later intervals, peaks exhibited slight tailing but a flat stable baseline, indicating efficient analyte separation with limited column dispersion, consistent with chromatographic principles (Pápai & Pap, 2002). This can suggest effective column efficiency and optimal instrument condition.

The corresponding mass spectra (m/z plots) further supports this trend, showing an increase in signal intensity and clarity with increasing PMSI. The absence of clear, identifiable signals in early PMSI samples likely reflects low compound abundance and signals falling below the detection limit, which is expected in the early stages of decomposition where compounds have yet to accumulate (Magnusson *et al*, 2026). However, in intermediate PMSI (days 7-21), a stronger signal of around 300 m/z was displayed, suggesting a higher abundance of the same compound over time. As not all PMSI days are presented in Figure 12, interpretation is restricted to only the spectra displayed.

Despite this, progressive improvement in peak clarity and clear m/z signals, made this analyte strong and reliable for future data modelling, as well as biomarker identification.

Figure 13 shows representative extracted chromatograms that display no trend or defiant peaks across the sampling days. In contrast to the previously discussed analyte, peak morphology in this datapoint was highly variable, with most graphs exhibiting substantial baseline noise, irregular signal fluctuations and poorly defined peak boundaries. When peaks were present, there inconsistent intensities, frequently showing broadening asymmetry, with some lacking clear distinguishable peaks altogether. No progressive change in RT was observed, with signals appearing sporadically rather than temporally structured.

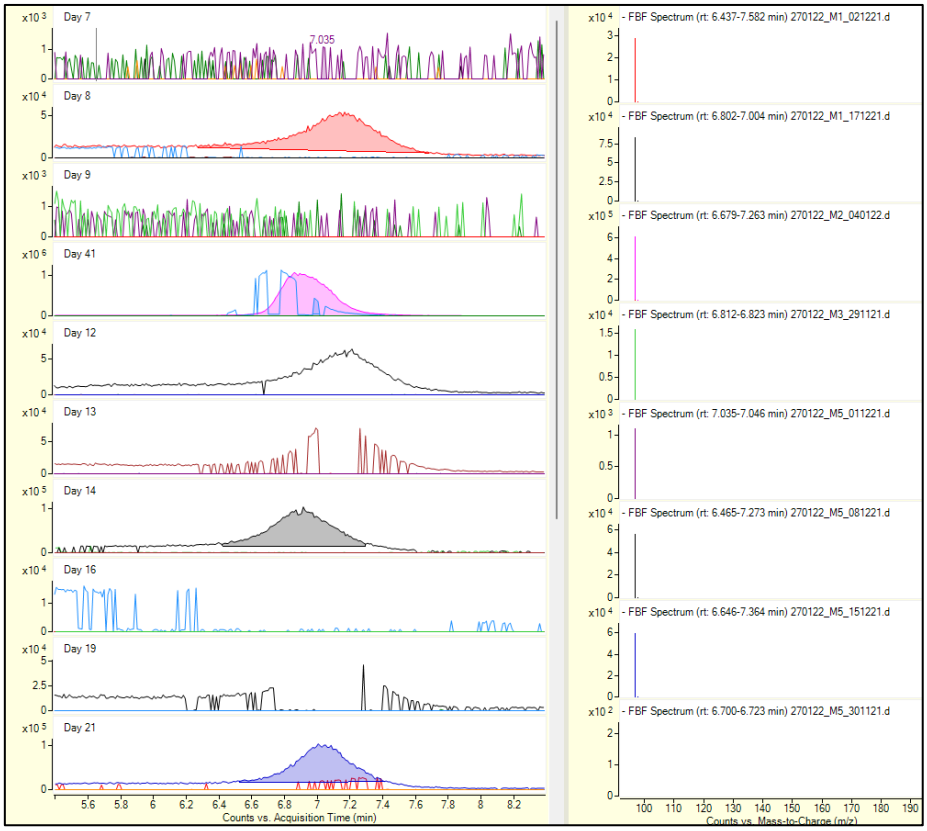


Figure 13- Representative overlaid LC-MS chromatograms out of 1393 datapoints, displaying poor, inconsistent peak profiles for selected compounds.

Irregular peak behaviour can stem from multiple interrelated factors, including sample preparation, instrumental parameters and data processing (McCalley, 2023). Effective sample preparation should aim to eliminate particulate matter that may cause column blockage in the instrument, which directly affects peaks shape, separation and RTs (Liu, 2022). It was noted in further work of Jenks (2022) that more QC samples should be taken to ensure more sufficient conditioning of the system. Therefore, potential column blockage could have occurred, resulting in poor peak shape, commonly displaying peak broadening, tailing, splitting, decreased sensitivity and inaccurate retention times (Dolan, 2015). Furthermore, insufficient instrument calibration and tuning can alter peak resolution and mass accuracy, thereby affecting reliable LC-MS interpretation (Cheng *et al*, 2022). Collectively, these observations indicate that this analyte lacks stability and reproducibility required for accurate PMSI modelling

However interestingly, the m/z data for these peaks showed to be relatively high and consistent throughout the PMSI days. At first look, this can seem contradictory to the poorly distributed chromatograms on the left, however, m/z spectra data only indicates what ions are present,

rather than the quality of their separation (Garg & Zubair, 2023). Therefore, consistent m/z signals can still be located within noisy, distorted peaks. But m/z signals alone are not sufficient when assessing feature reliability, therefore compounds that reflected Figure 13 were marked with ‘n’.

Finally, Figure 14 shows representative extracted chromatograms that displayed inconsistent and variable retention behaviour, with slight co-elution present across the sampling days.

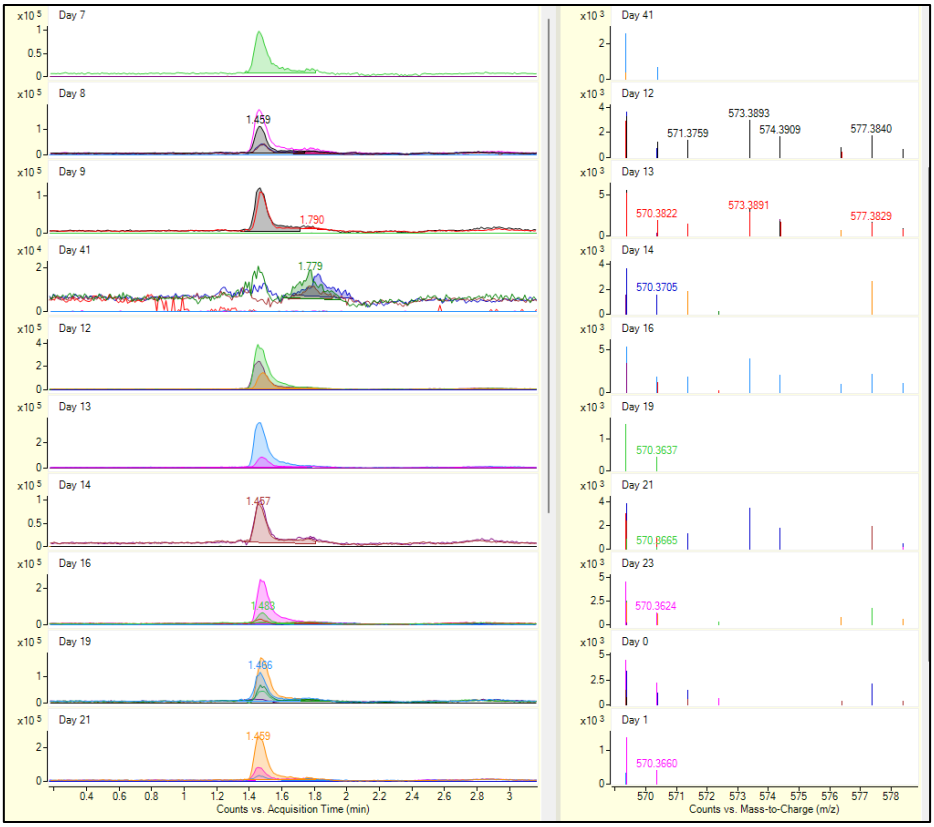


Figure 14- Representative overlaid LC-MS chromatograms out of 1393 datapoints, displaying co-eluting, inconsistent peak profiles for selected compounds.

A principal peak was identified within the RT window of 1.45-1.79 min, however, it was observed that there were notable shifts in RT, peak width, and signal intensity throughout. Several chromatographs exhibit peak broadening and asymmetry, as well as shouldering and co-elution. Chromatographic co-elution occurs when two or more compounds do not separate, appearing as a single, often distorted peak (Sawikowska *et al*, 2021). As displayed in later timepoints, some signal regions begin to overlap, distorting their boundaries, indicating matrix interference or the presence of another compound that was poorly separated. Corresponding

mass spectral data supports this inference, as variability in dominant m/z signals suggest inconsistent ion profiles and reduced chemical stability.

However, it can also be observed that there are well-defined ions distributed across the PSMI days that remain consistent within the 570-571 m/z range. Signal intensity is greatest during intermediate PMSI (days 12-21) with reduced clarity present in early days (day 1) and late days (day 23). The agreement between defined peaks and consistent m/z signals can indicate reliable compound detection, but it should be noted that clearer features should be considered first before using co-eluting, inconsistent compounds. Due to this consideration, compounds that reflected Figure 14 were marked with 'no' due to limited reproducibility and stability to be considered a robust PMSI marker suitable for modelling.

7.2 **Initial Variable Filtering**

Following data importation into R studio, variables marked with 'Yes' were retained for further analysis, resulting in a total of 178 variables.

7.3 **Missing Data Patterns and Distribution**

To assess the completeness of the dataset prior to imputation, missing data was summarised across PMSI categories. These results were presented in Figure 15. High variation in missingness was observed between PMSI groups.

Generally, missing data ranged from 14.09% (PMSI 19) to 76.34% (PMSI 2) with several days exhibiting particularly high levels of missingness including, PMSI 2 (76.34%), PMSI 0 (68.14%), PMSI 7 (63.76%) and PMSI 1 (56.91%). In contrast, PMSI 19 (14.09%), PMSI 16 (17.77%), PMSI 6 (22.01%) and PMSI 5 (23.66%) demonstrated lower levels of missingness. The mean number of missing values per row changed significantly across PMSI days, indicating variation in data completeness among groups.

PMSI	Number of Rows	Total Missing Values	Mean Missing per Row	% Missing
0	6	740	123.33	68.14%
1	6	618	103.00	56.91%
2	6	829	138.17	76.34%
5	6	257	42.83	23.66%
6	6	239	39.83	22.01%
7	5	577	115.40	63.76%
8	6	310	51.67	28.55%
9	6	581	96.83	53.50%
12	6	436	72.67	40.15%
13	6	554	92.33	51.01%
14	6	573	95.50	52.76%
16	6	193	32.17	17.77%
19	6	153	25.50	14.09%
21	6	443	73.83	40.79%
23	6	302	50.33	27.81%
41	6	434	72.33	39.96%

Figure 15- Proportion of missing data across PMSI days

A further visualisation of these results was presented in the form of a heatmap. These graphs are primarily used to represent matrix-like data and can be useful for revealing similar patterns shared by subsets of rows and columns, helping to detect hidden structures in complex datasets (Gu, 2022). In comparison to line and stacked graphs, the key advantage of heat maps is its ability to maintain visibility and interpretability even as the volume of data increases, as well as having a distinct colour-coded format (Endo & Hosobe, 2024).

As seen in Figure 16, light blue represents a low missing count, while dark red represents a high missing count. Each square reflects the missing count for each mouse (M1-M6) within one PMSI day. This visualisation supports Figure 15 by showing that missingness is not evenly distributed over PMSI. M1, M2 and M3 reveal significantly higher missing counts overall, while M4, M5 and M6 display fewer missing values, particularly at later PMSI days where lighter blue predominates. M6 also contains missing data on day 7. However, it was observed that there was a pattern in missingness, which appeared clustered on days 0-2.

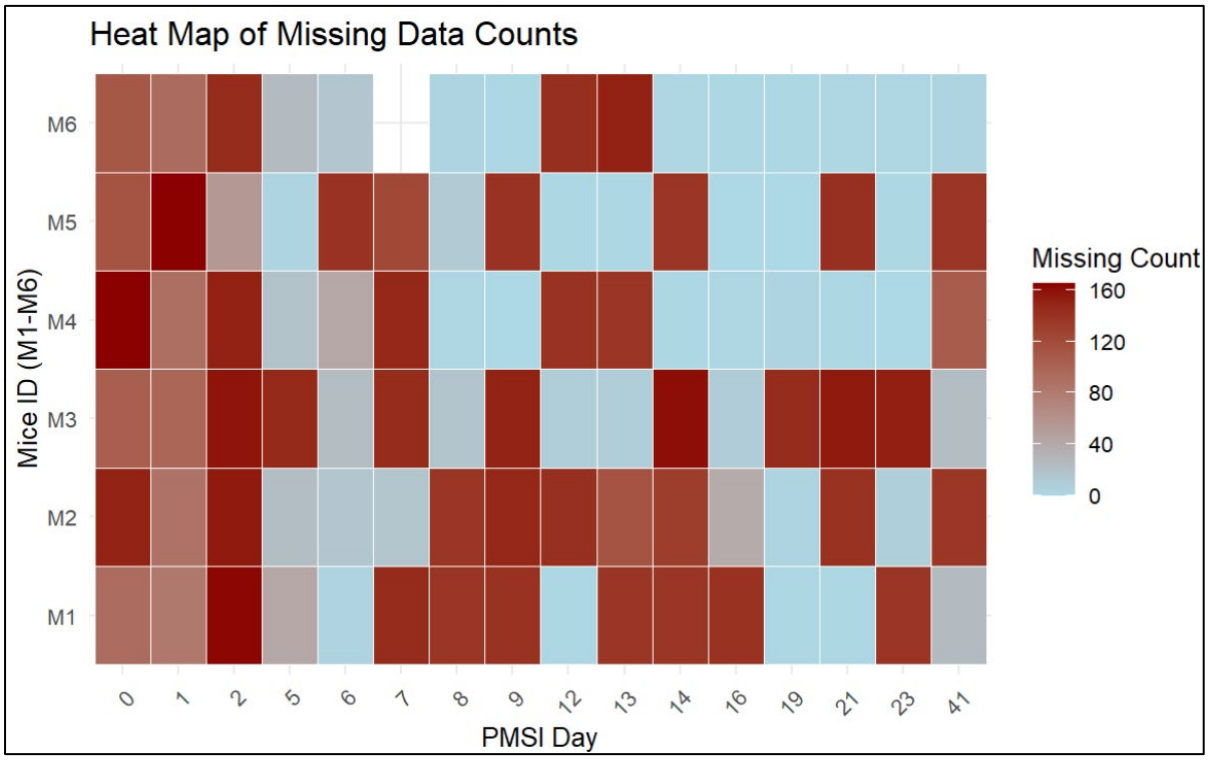


Figure 16-Heatmap to visualise the missing data over the study period of 41 days

This distinct pattern of missingness at the start is consistent with left-censored missing values, which commonly exists in targeted metabolomics datasets where compound intensities fall below the instruments detection limit and are therefore not detected (Wei *et al*, 2018). Therefore, they can be considered as missing not at random (MNAR) (Wei *et al*, 2018). In such cases, MNAR can lead to biased parameter estimation and jeopardizes following statistical analysis in different aspects, such as distorting sample distribution and impairing statistical power (Wei *et al*, 2018). It should also be noted that the M6 day 7 file was corrupted and therefore not reliable.

In contrast, there is also a considerable amount of missing data observed in later PMSI days. These clusters of missingness are unlikely to be MNAR and instead, are more consistent with missing at random (MAR), potentially reflecting differences in data processing and recording practices. This is because there is no evidence to suggest that values are missing because of the actual (unobserved) measurement itself. Instead, it appears associated with observable dataset characteristics such as Mouse ID and PMSI, making MAR more plausible than MNAR.

The impact of MAR on data analysis primarily concerned with the assumptions and procedures used to handle missingness, and the validity of the resulting inferences (Little, 2021). Modern techniques like multiple imputations and maximum likelihood methods are commonly applied

in MAR assumptions and are designed to use observed data to inform the imputation process (Goldberg *et al*, 2021). However, these approaches cannot fully eliminate bias when missingness could be caused by MNAR (Goldberg *et al*, 2021). As a result, utilising literature as confirmation, QRILC (Quantile Regression Imputation of Left-Censored data) was suggested for dealing with the MNAR missing data, whereas KNN (K nearest neighbour) was considered for MAR data (section 9.5).

7.4 **Handling of Missing Data**

As previously stated, missing data can reduce statistical power, produce bias and lead to invalid conclusions (Kang, 2013). Therefore, due to the extent of the missing data, a threshold of 0.25 was generated, where variables with more than 25% missing data were filtered out. The decision to apply a 25% threshold was based on the understanding that the proportion of missing data is directly related to the quality of statistical inference. However, there is not a universally established cut off regarding an acceptable percentage (Dong & Peng, 2013), therefore this threshold was able to remove problematic variables while maintaining data retention and quality. The results are shown in Figure 17.

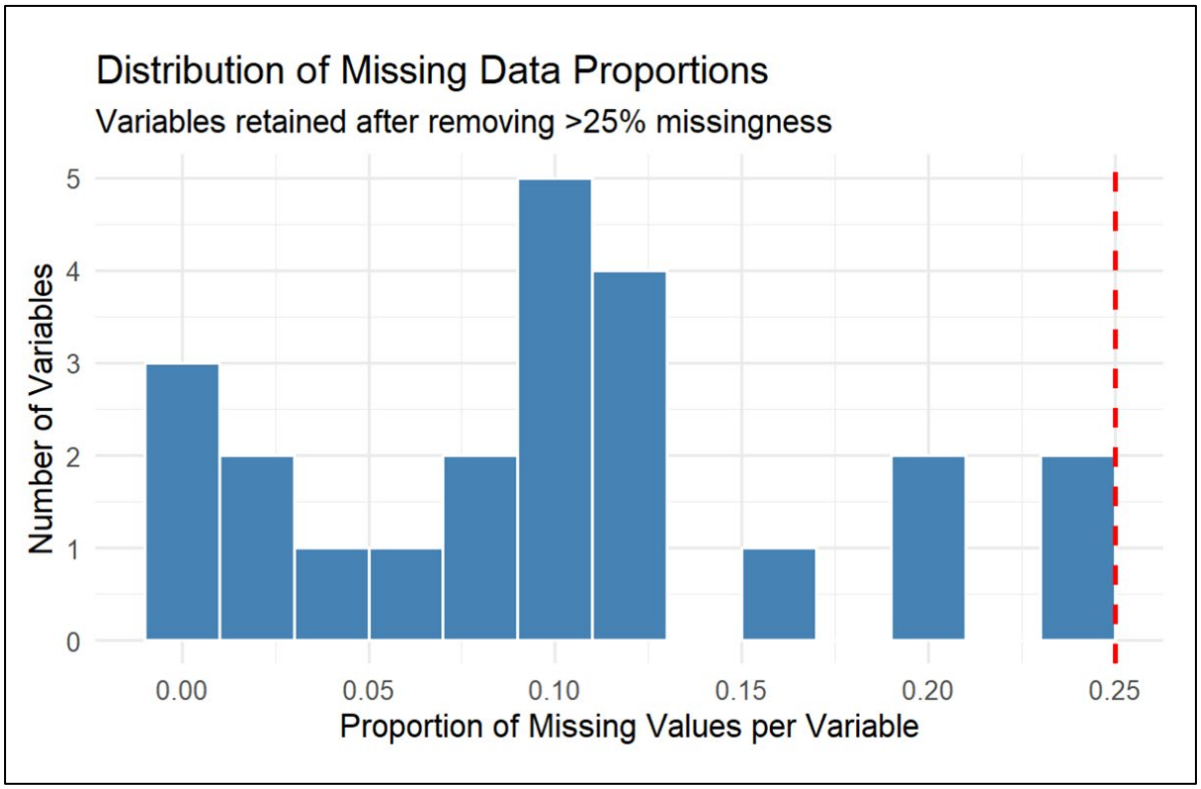


Figure 17-Histograms showing the distribution of missing data proportions across retained variables after applying a 25% missingness threshold

Following the exclusion of variables exceeding 25% missingness, variables exceeding the criterion were excluded, resulting in a shortlisted data set of 23 variables. The histograms therefore show the concentration of variables with low proportions of missing data, confirming that the filter was applied correctly. The distribution of the data appears to be right skewed, with fewer variables approaching the 25% threshold, suggesting minimal borderline retention. This implies that the filtering process removed substantially incomplete variables while preserving those with substantial information. Therefore, the retained values now provide a more stable foundation for imputation and predictive modelling.

7.5 **Data Processing and Visualisation of Shortlisted Data**

Six summed and normalized scatter plots were created for each mouse (M1-M6) to visualise the trajectory of the shortlisted variables across the 41-day period (Figure 18). These visualisations enabled the identification of gaps, outliers and low-quality data that could be imputed to improve the prediction of a PMSI model. Scatter plots were used due to their usefulness in early analysis where they can be generated to show correlations and patterns present in the data (Nguyen *et al*, 2020). The data was initially normalized using the mean score per mouse, however, the Sum proved to be more efficient. Sum normalization is considered a standard, effective method in metabolomics, particularly for reducing systematic variation between samples (Nam *et al*, 2020). This is accomplished by normalising each compounds intensity in a sample by the total sum of all detected peak intensities in that same sample, frequently scaling them to a total sum of 1 (Guida *et al*, 2016). As a result, it is easier to track cumulative metabolic change over time which is essential when attempting to predict PMSI.

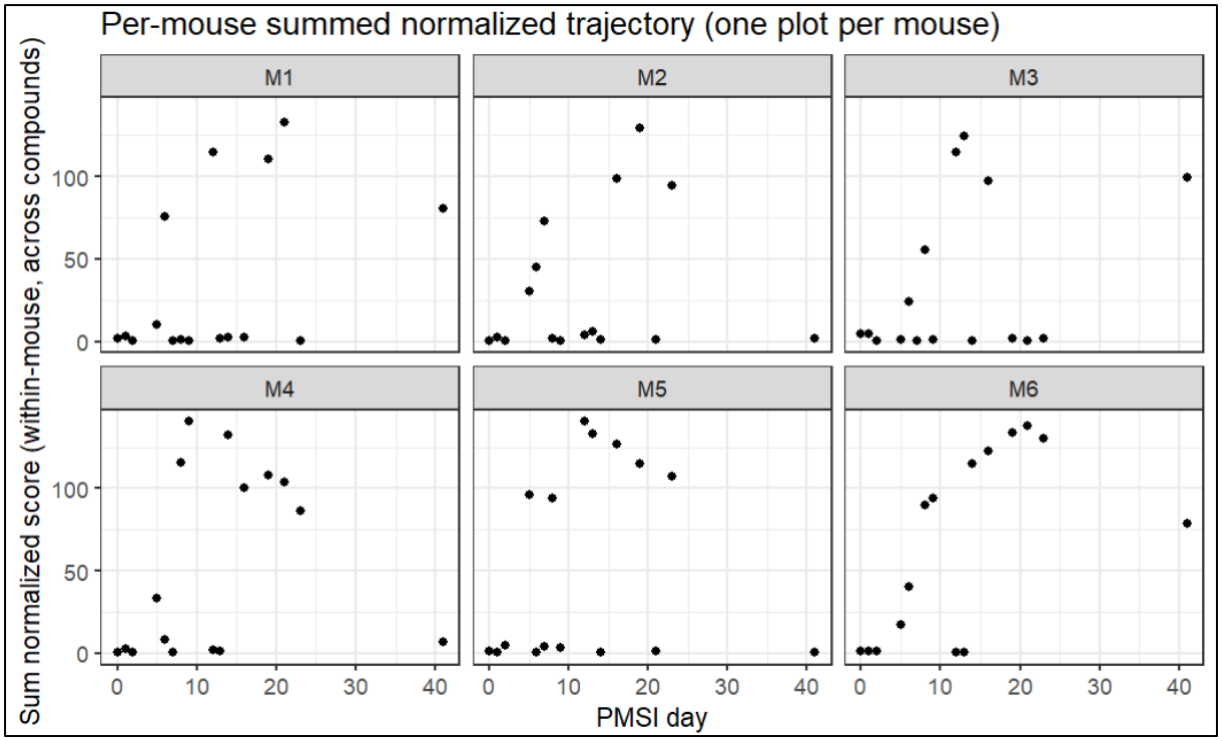


Figure 18-Summed normalized scatter graphs showing the temporal compound changes over PMSI days for each mouse (M1-M6).

The scatter plots for each mouse (M1-M6) revealed some temporal patterns across the 41-day period. At early PMSI days (0-3), the observed intensities remain consistently close to zero, indicating minimal compound signal. These low intensities could reflect the initial stage of decay, known as the fresh stage, which begins immediately upon death (days 1-6). During this stage, autolysis occurs as oxygen (O2) circulation stops and carbon dioxide (CO2) builds, increasing acidity and causing cell membranes to rupture (Almulhim & Menezes, 2023).

As cells breakdown, they release intracellular components such as Potassium (K+), Hypoxanthine (Hx) and Magnesium (Mg). (Grassi *et al*, 2025). This facilitates putrefaction where endogenous bacteria from the gut spread throughout the body, breaking down tissue and creating byproduct gases and chemicals (Shedge *et al*, 2023). This process could contribute to an influx of biological chemicals present in the surrounding water, which is reflected in the scatter graphs, where compound intensity rapidly grows between days 5-20. When visualised collectively across all mice (Figure 19), the highest intensities were observed between days 10 and 22.

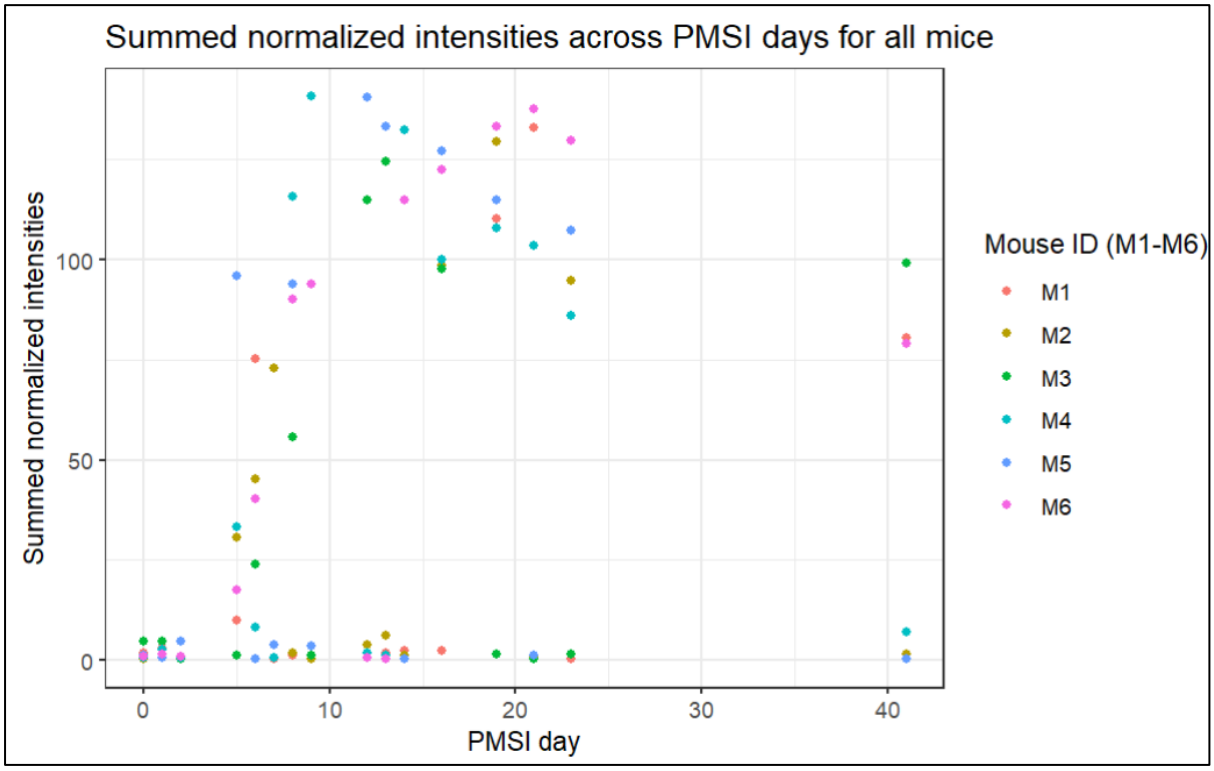


Figure 19-Summed normalized intensities across PMSI days for each mouse. Each point represents the summed normalized intensity across compounds for a given mouse on a specific PMSI day

For M1, M2 and M3, peak intensities ranged from approximately 120-140, whereas M4, M5 and M6 exhibited slightly higher peak intensities of roughly 140-150. This suggests a broadly similar temporal pattern across mice, with peak signals occurring during the mid-PMSI period. This rise could be related to increased metabolic and microbiological activity that occurs during active decomposition (days 10-20), where soft tissue liquefy, purge fluids are released and microbial infestations appear (Shedge *et al*, 2023). However, it is important to note that many datapoints across the mice remain close to near zero throughout the 41-day period, with clusters around days 5-10 and 10-15. This variability suggests that compound detection may be intermittent across sampling points, either reflecting fluctuations in breakdown processes or limitations in compound detectability at specific time intervals. Therefore, the apparent temporal trend is mostly inferred from the datapoints with clear identification.

After reaching peak intensity, the signals begin to gradually decrease across all mice, indicating a downward trend in compound intensity as PMSI days proceed. This decline remains consistent with advanced decomposition (approximately 25+ days after death), where microbial activity slows, and most soft tissue has been consumed or liquified, resulting in a drop of decomposition related compounds (Shedge *et al*, 2023).

After reviewing all the graphs, M6 demonstrated the greatest potential for having an identifiable temporal trend, with only two near zero data points located at around day 12-13. However, when viewing Figure 20, it can be observed that if near zero values are excluded, each mice follows a slight temporal trend in compound intensity.

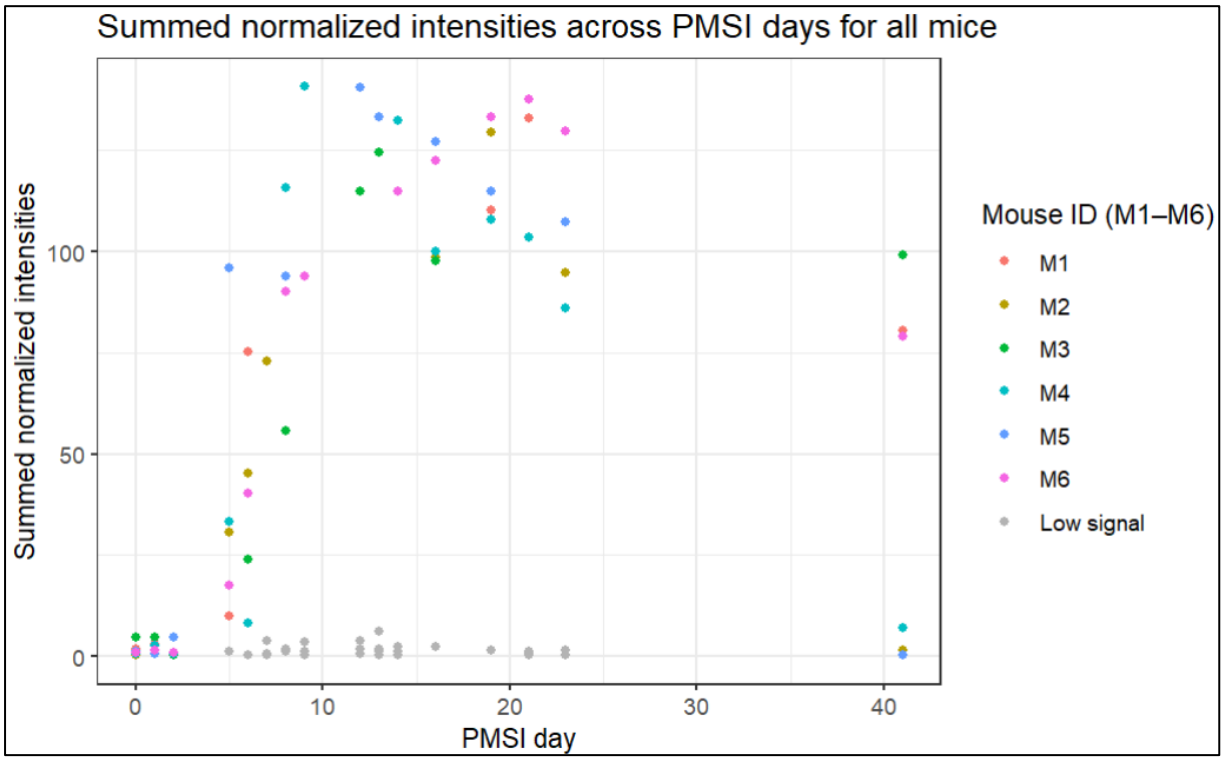


Figure 20-Summed normalized intensities across PMSI days for each mouse. The addition of greyed out MAR datapoints to help visualise overall trends.

Therefore, despite variability in signal detection, the overall pattern of low early intensities, increasing signals during mid-PMSI, peak levels between days 10-22 and a gradual decline during later PMSI, remain consistent with typical postmortem patterns. The next step was to impute the missing values and assess if it improved the mice's patterns further.

7.6 **Imputation of Early and Late PMSI days**

After visualising the data to assess current trends before imputation, it was decided to split the data into two separate datasets: early (0-7) and late PMSI days (8-41). This divide was intended to highlight potential temporal differences in the data, as patterns of missingness and variables behaviour varied between early and late PMSI days. By separating the dataset, imputation methods could be applied independently, allowing each process to better capture the underlying structure of the data within each time period.

7.6.1 **QRILC Imputation**

As previously noted, there was a high rate of missingness between days 0-7, which was consistent with left-censored missing values and indicated that values were missing not at random (MNAR). As a result, this pattern supported the use of QRILC imputation, a method that handles missing left-censored data, particularly values below the limit of detection (LOD) (Wei *et al*, 2018).

Instead of replacing missing values with a constant or randomly generated low value, QRILC estimates the distribution of low values and samples from it. Initially, the data is log-transformed to reduce variation and achieve a near-normal distribution (Feng et al, 2014). Using the observed values, a quantile regression model is then fitted to the lower tail of the distribution to help estimate where the detection limit is most likely to occurs. Missing values are then imputed by sampling from a left-truncated normal distribution that has been shifted towards the lower quantiles, based on the premise that missing values correspond to low abundance-measurements below the detection threshold (Wilson *et al*, 2022). Figure 21 exhibits the distribution of log_2-transformed intensities in the early PMSI dataset (days 0-7) before and after QRILC imputation.

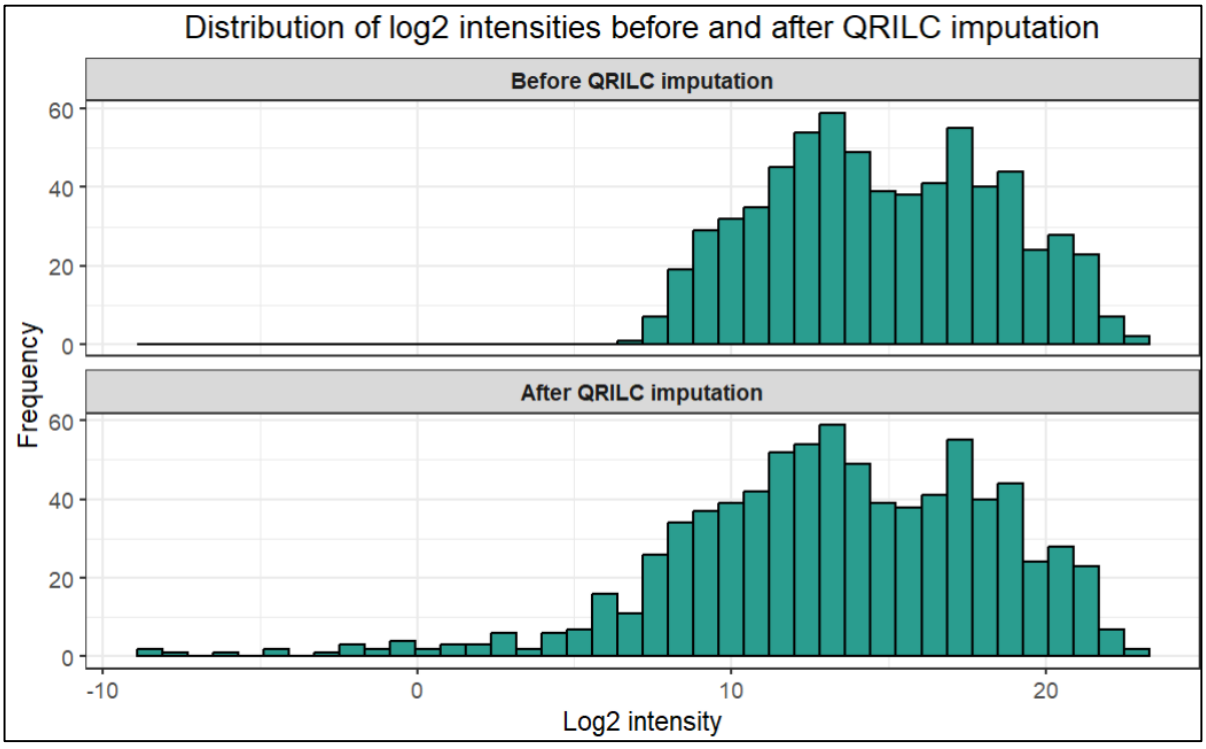


Figure 21 - Distribution of log_2-transformed intensities before and after QRILC imputation, showing the introduction of low-intensity values corresponding to left-censored missing values.

Prior to imputation, the distribution exclusively reflected observed values, with missing values absent from the lower intensity range. After imputation, additional values appear in the lower tail of the distribution, extending from approximately -10 to 10 on the log_2 scale, indicating that QRILC successfully imputed low-intensity values that correspond to left-censored missing data. These imputed values appear negative because on logged data, intensities below 1 become negative. The majority of intensities cluster around a central peak of 12-16 log2 values, which represents the main distribution of identified compounds in the dataset. It can be observed that even after imputation, the peaks are mostly unaltered, demonstrating that QRILC does not substantially alter the distribution of observed values.

To support the assumption that QRLIC successfully imputed all missing values, Figure 22 shows the missing value count of before and after.

Stage <chr>	Missing_values <int>
Before imputation	134
After imputation	0

2 rows

Figure 22-Summary table displaying the count of missing values before and after QRILC imputation on the early PMSI dataset

7.6.2 **KNN Imputation**

Although the missing data had now been resolved in early PMSI, the preceding days from 8-41 still displayed a considerable amount of missing data. This time, there was no apparent trend to the missingness, with mice such as M1 and M2 displaying higher counts than other mice (>50%). As a result, it could be suggested that the likelihood of missingness was not dependent on unobserved values themselves, and therefore the data was presumed to be missing at random (MAR). Therefore, this supported the use of k nearest neighbour (KNN) imputation on late PMSI days.

KNN is a popular non-parametric method for handling missing data in machine learning (ML) and data analysis. It involved replacing missing values using the datasets most similar observations (nearest neighbours) to identify a replacement for specific missing values (Khan *et al*, 2024). KNN was utilised within each mouse group to estimate missing values for a specific mouse based on the most comparable observations from the same mouse over time. A value of k = 5 was defined, so that each missing value was calculated using the five nearest

neighbouring observations. This threshold provided a balance between reducing noise and preserving logical structure to the data. Using a smaller number helped to ensure that imputed values were based on observations with similar properties, which prevented excessive smoothing. Figure 23 displays the distribution of logp1 transformed intensities in the late PMSI dataset (8-41) before and after KNN imputation.

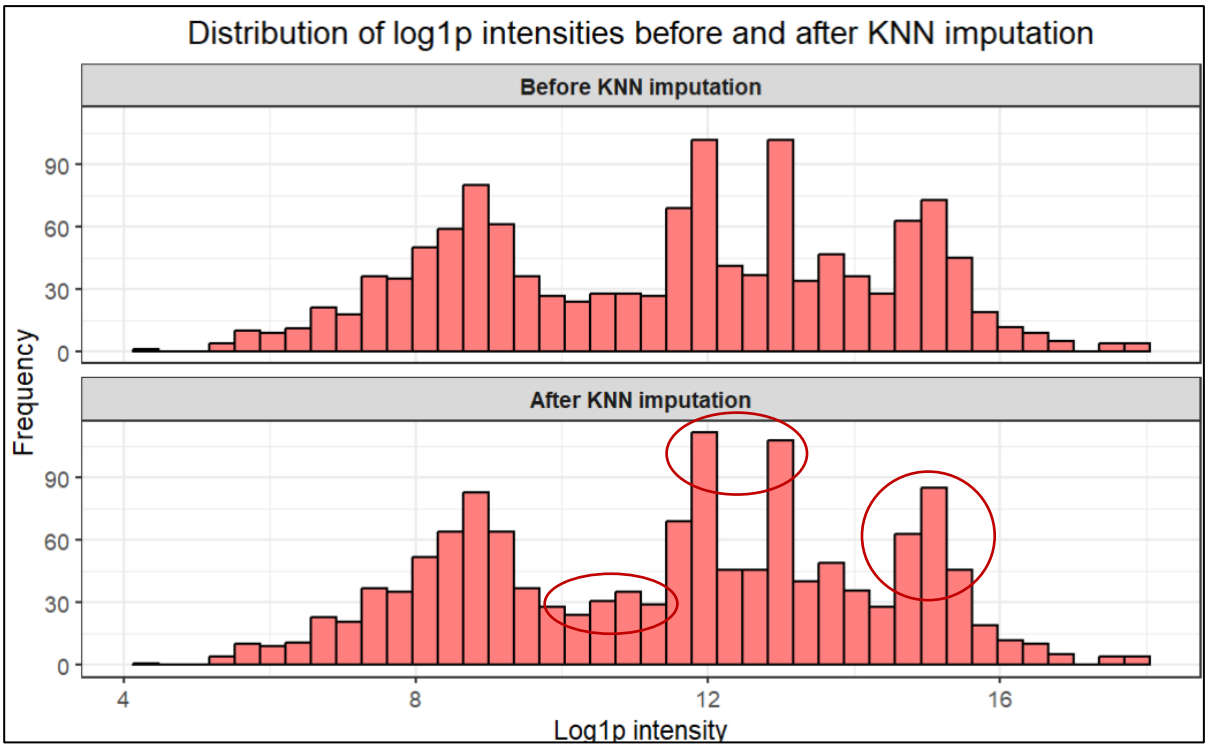


Figure 23-Distribution of log1p intensities before and after KNN imputation

Overall, the shape of the distribution remains relatively similar after imputation, which is common for KNN and does not imply that the method worked incorrectly. The similarity exists because as previously mentioned, KNN imputes missing values using neighbouring observations with similar properties, rather than inserting unrealistic arbitrary units. Since it was determined that the data was MAR, the neighbouring values used for imputation likely matched the true underlying values.

Several prominent peaks are present in the data and are located at around 9, 12, 13-14 and 15 log1p intensities. These peaks appear both before and after imputation and are only minorly increased, demonstrating that the main structure of the data is still preserved and underlying patterns did not get distorted. The success of this imputation was also supported by Figure 24 where a missingness count was produced for before and after KNN.

Stage <chr>	Missing_values <int>
Before imputation	85
After imputation	0

2 rows

Figure 24-Summary table displaying the count of missing values before and after KNN imputation on the late PMSI dataset

7.6.3 **Result of the Combined Imputation**

After successfully imputing the data, an additional graph was created to depict the newly combined data frames and highlight the overall impact of imputation. Shown in Figure 26, before imputation, the dataset contained 1,966 observations, while after imputation the number increased to 2,185, reflecting the replacement of previously missing values.

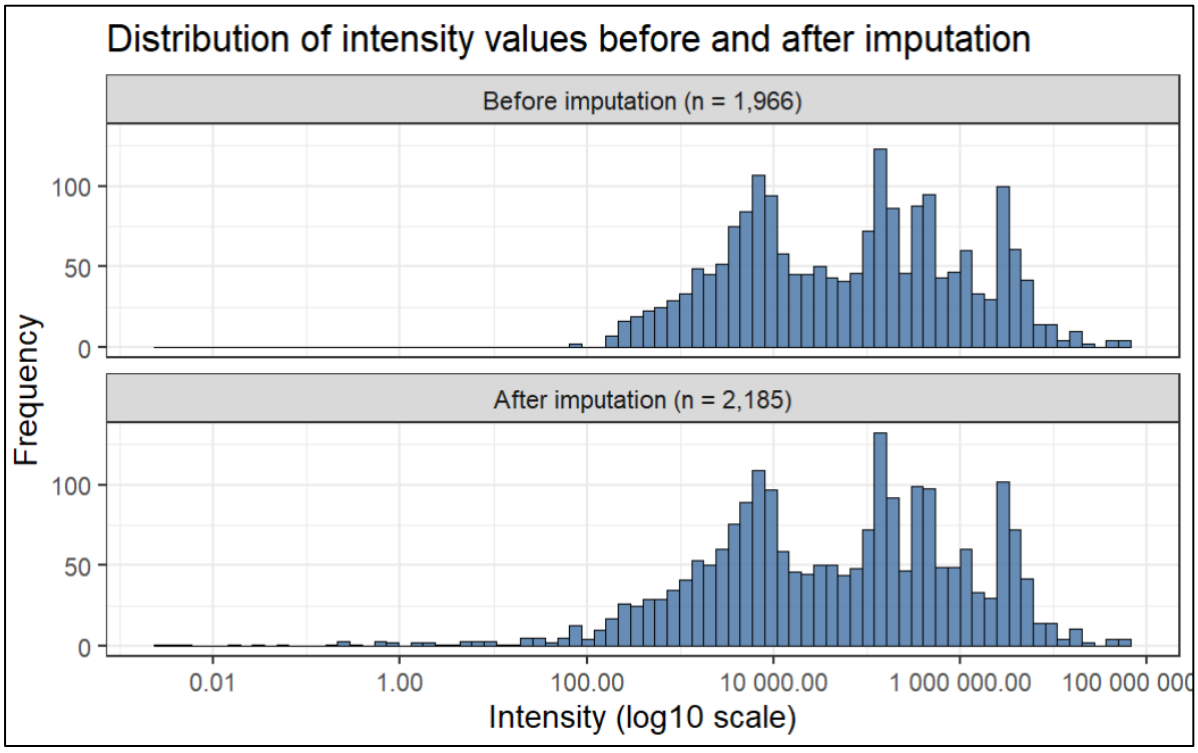


Figure 25- shows the overall distribution of measured intensity values before and after imputation

Therefore, the overall distribution remains largely similar following imputation, indicating that the methods of imputation increased the number of observations whilst maintaining the general intensity distribution. This approach can be supported by Wei *et al*. (2018), who analysed the performance of different imputation methods and discovered that KNN works best for randomly missing data and QRILC best on left-censored values. He also emphasised the need

of the method selection based on the missingness mechanisms, which supports our initial analysis of missingness (section 9.3)

7.7 **Visualisation of Newly Imputed Dataset**

The scatter plots created prior to imputation were recreated with the merged imputed dataset to examine whether any trends were more visible. Figure 27 depicts the individual mouse trajectories that demonstrate significant heterogeneity in summed normalised intensities throughout PMSI days.

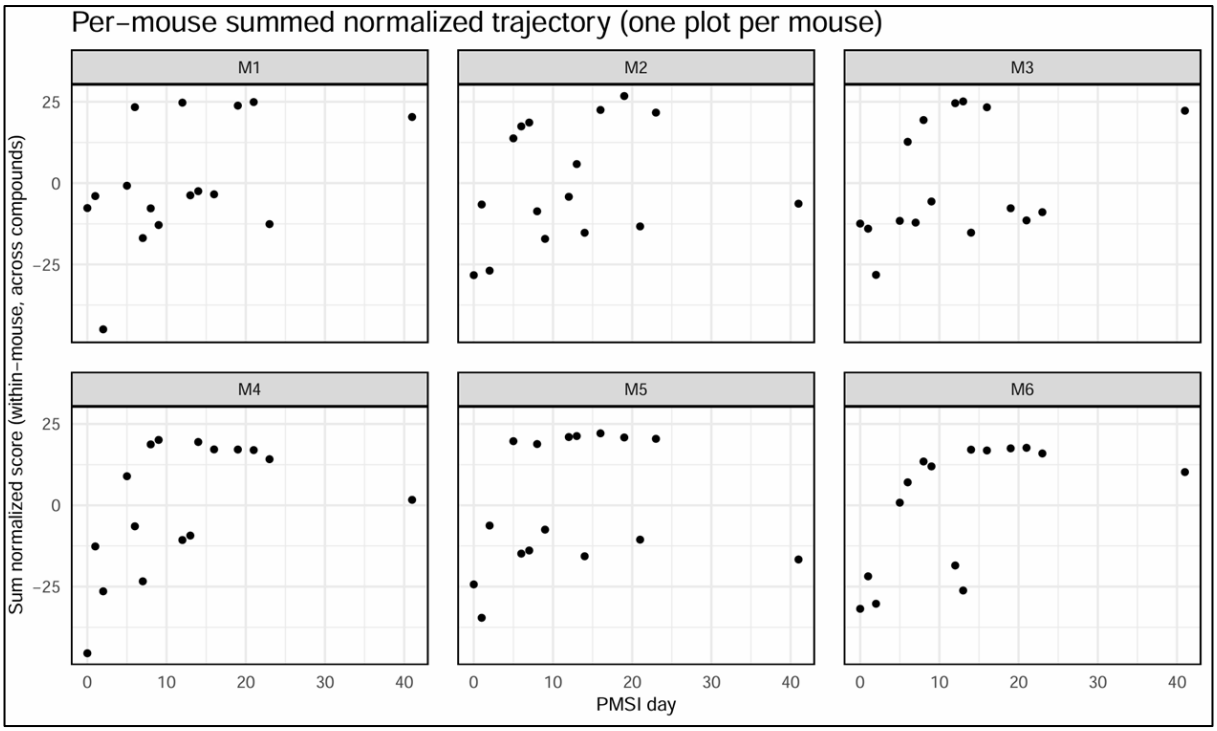


Figure 26- New imputed summed normalized scatter graphs showing the temporal compound changes over PMSI days for each mouse (M1-M6).

Even after imputation, data points are still unevenly distributed across the 41 days, with noticeable gaps still present at several PMSI intervals. Reduced data density is still more visible at early PMSI (days 0-5) where several mice exhibit similar observations, and at late PMSI intervals (approximately days 30-40), fewer measurements are consistently recorded among individuals. Although the early, near-zero points can be consistent with biological trends in post-mortem, the observed variability on selected days (around PMSI 12, 13, 15) suggests that missingness and detection limits continue to influence the dataset, even after preprocessing. However, if we exclude outliers and focus on key areas of significance, we could see how a

pattern might unfold. To visualises these potential underlying trends, smoothed trajectories were added (Figure 28).

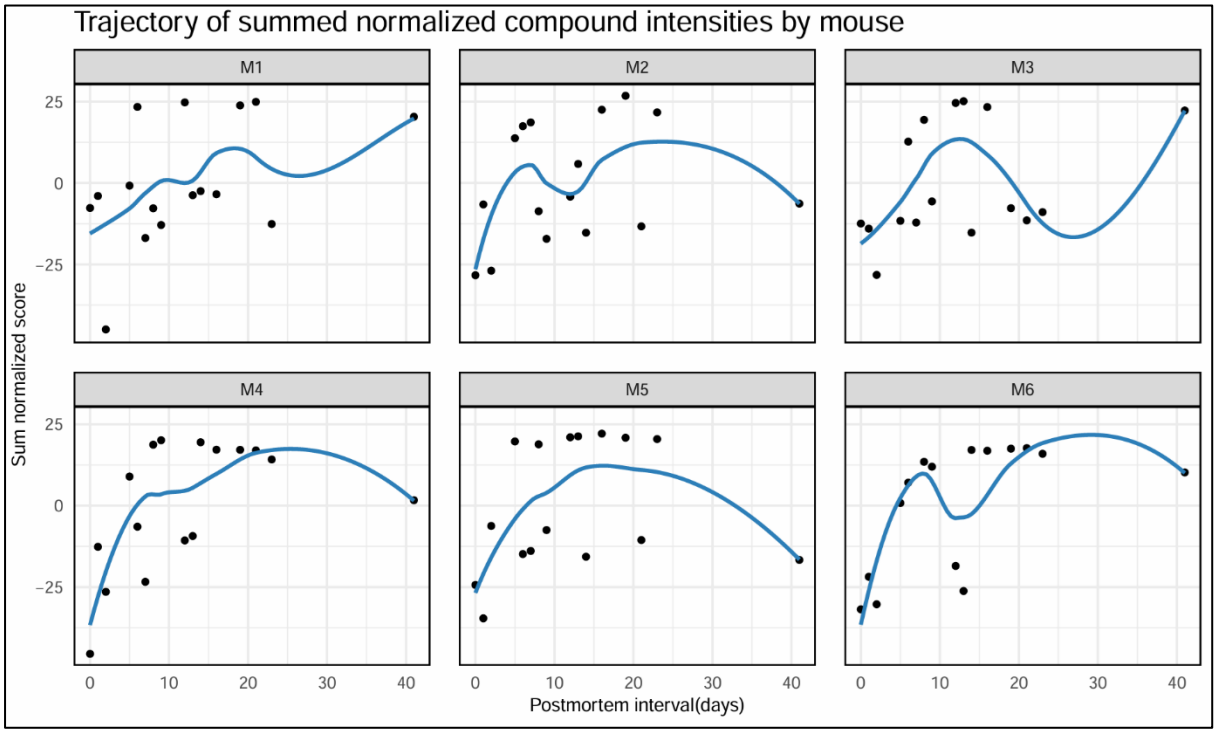


Figure 27-Smoothed trajectories (LOESS) of summed normalised metabolite intensities across PMSI days for M1-M6.

Because the data exhibited non-linear trends, locally weighted scatterplot smoothing (LOESS) was employed instead of a traditional regression line, as it allows smooth curves to be fitted to the data without assuming a parametric shape (Wanishsakpong & Notodiputro, 2017). Linear regression is often inadequate for modelling non-linear data because they presume a constant linear relationship which may indicate a false trend. This method fits a straight line to datapoints by minimising the sum of squared vertical distance. However, when applied to non-linear data, this approach might result in a poor fit and mislead interpretations since it cannot capture curvature or complex trends (Jones, 2024).

This is especially important and demonstrated in decomposition studies such as the one conducted by Magnusson *et al* (2026), where metabolic changes are expected to follow non-linear trajectories over time, such as an increase during active decay, followed by declines in later stages.

The clarity of temporal trends among the mice varies, but multiple demonstrate more distinct trends. Mouse 6 (M6) exhibited the most consistent and well-defined trajectory, only limited by a drop between days 12-13 (which could be excluded if used in future modelling). It is

characterised by a gradual increase from days 0-10, where it then peaks at around day 20, followed by an assumed gradual decline. Similarly, M2 and M4 displayed a relatively similar biological pattern, with a plausible, peak and decline. M3 and M5 show moderate trends, with M3 increasing on later days, and M5 declining. Finally, M1 displayed the least consistent pattern, with higher dispersion of datapoints and a less defined trajectory.

Overall, these findings suggest that there are temporal trends present across mice, but their strength varies between individuals. Therefore, this supports the use of machine learning algorithms, such as Random Forest (RF), that can accommodate for noisy, heterogenous data (Zhao *et al*, 2024). This means that if we used a more complete datasets (M6), we can see if RF can predict the trends. If this proves to be successful, the model can therefore be tested and applied to datasets with higher volumes of missing data.

However, to assess an alternative analytical approach, Principal Component Analysis (PCA) was performed on the entire dataset to determine whether metabolomic variance across samples show systematic changes with PSMI, as well as identifying key metabolites that drive the change.

7.8 **Principal Component Analysis (PCA) of the Processed Dataset**

7.8.1 **Overview of PCA and Selection of Principle Components**

Principal Component Analysis (PCA) was utilised to visually group similar PC scores, where the coordinates of original datapoints are projected onto new, lower-dimensional PC axes, to aid in the detection of patterns and clustering among the PMSI groups (Hansel *et al*, 2025). PCA scores and loading plots are fundamental multivariant statistical tools used to reduce the dimensionality of high-dimensional data, such as metabolite information from LC-MS, into 2D visual plots (Nyamundanda *et al*, 2010). Scores for the samples are represented as points, and those with similar metabolic profiles cluster together, while dissimilar samples appear further apart (Mayonu *et al,* 2025). This facilitates the rapid discovery of visual trends, patterns and relationships within biological data, as well as subsequent analysis of the metabolites causing the most change (Farooq *et al*, 2024). In alignment with other literature, due to a small sample size but considerable variability, an 85% confidence ellipse was used since it provided a more robust representation of the central trend, without being excessively influenced by extreme outliers (Walsh *et al*, 2006).

To evaluate how much each principal component contributed to the total variance, a scree plot was generated (Figure 29). Scree plots in PCA serve as a visual tool to assist in selecting the appropriate number of principle components (PCs) to retain. They display eigenvalues associated with each PC in descending order, allowing for the identification of where values begin to level off, often referred to as the 'elbow'. This point offers a natural cutoff. Additionally, the Kaiser Criterion has shown that components with eigenvalues greater than one should be retained for analysis (Truong, 2026).

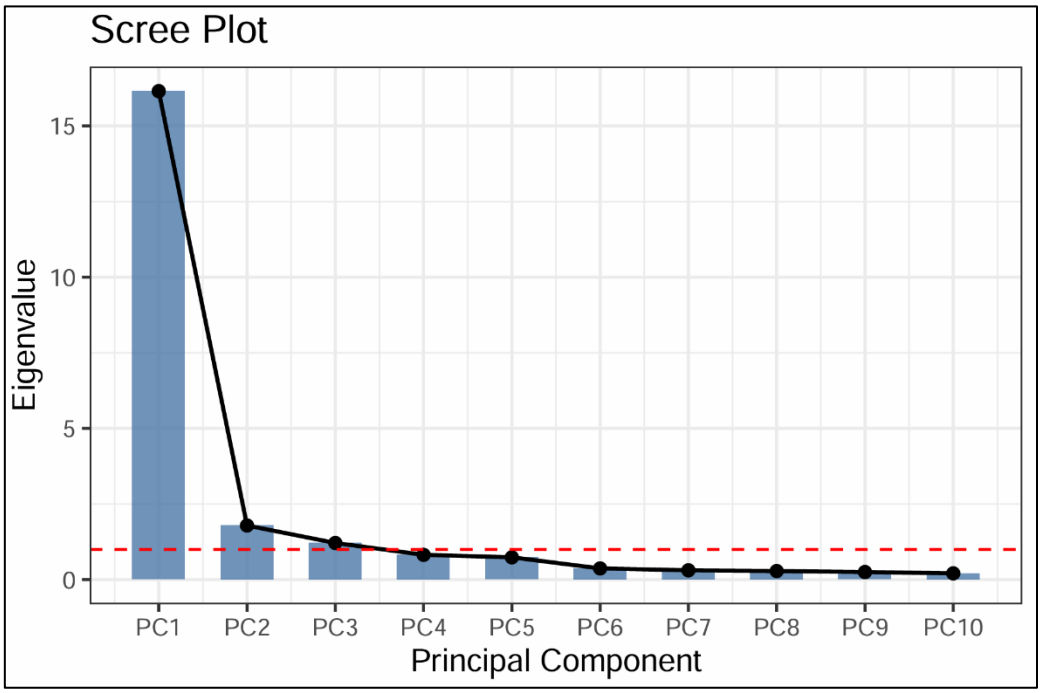


Figure 28-Scree plot showing the variance explained by each principal component

In the scree plot, PC1 dominates the other PCs, reaching an eigenvalue of approximately 16, signifying ~~that~~ it causes around 70% of the variance. Although PC2 accounts for a smaller proportion (approximately 7-8%), it is still above the 1 threshold, as shown by the red dotted line, meaning it will still be included in PCA analysis. PC3 marginally exceeds the threshold, however, based on the scree plot and point of inflection, it was decided not to include it in further analysis because its variance contribution was minimal.

Overall, this suggests that the underlying structure of the data can be captured through PC1 and PC2, allowing for a simpler more interpretable model when plotted on PCA.

7.8.2 **Principal Component Analysis (PCA) of the Processed Dataset**

As illustrated in Figure 30, the first two principal components accounted for 78.0% of the total variance, with PC1 accounting for 70.2% and PC2 accounting for 7.8%, indicating a strong dimensionality reduction. While there is no universally accepted percentage, research suggests that PC1 and PC2 explain around 70-90% of total variance in the data, however, it is subjective to the type and complexity of the data (Jolliffe & Cadima, 2016).

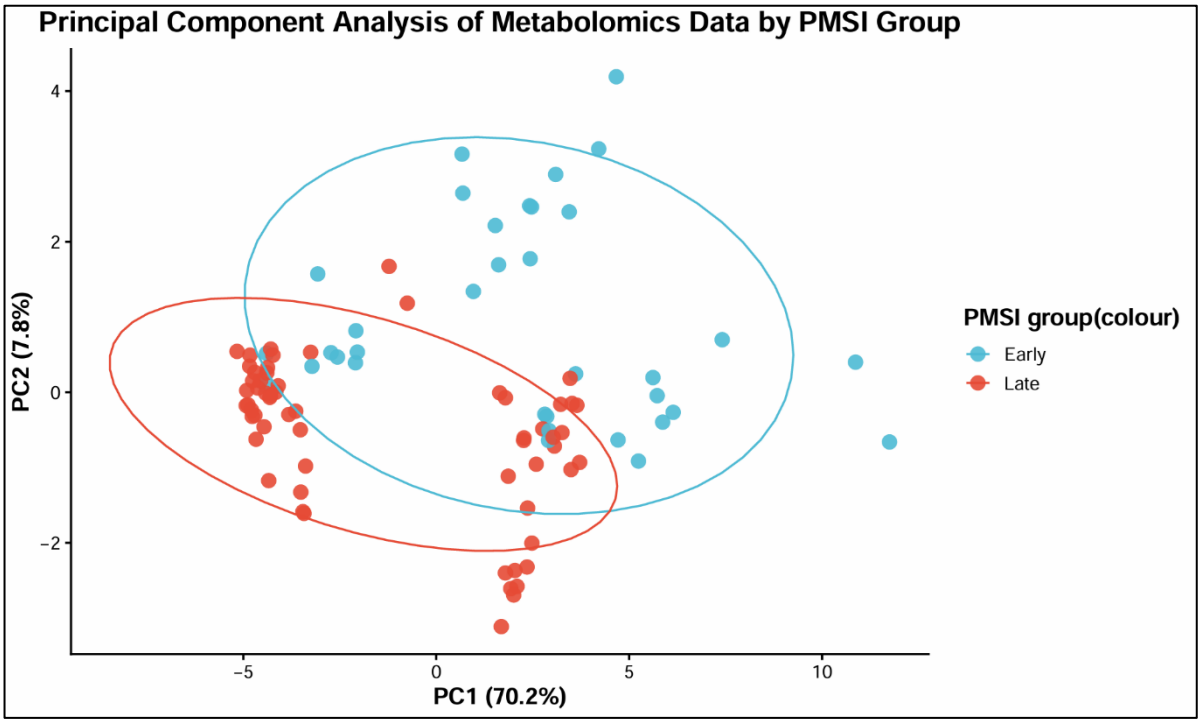


Figure 29- Principal Component Analysis (PCA) of metabolomics data from early and late PMSI samples. Points represent individual samples, coloured by PMSI group.

The PCA plot demonstrates no clear trend of separation between early and late PMSI samples, with significant overlap between groups. While early PMSI samples appear more frequently at positive PC1 values and late samples at negative PC1 values, this trend is not distinct, and any assumption could be overstating the findings. PC1 captures the majority of variance, but the observed distribution suggests only limited differentiation in metabolomic profiles over time.

7.8.3 **Temporal Patterns and Variability**

To identify if the samples were driven by a biological PMSI trend or technical issues, datapoints were labelled with their days (Figure 31).

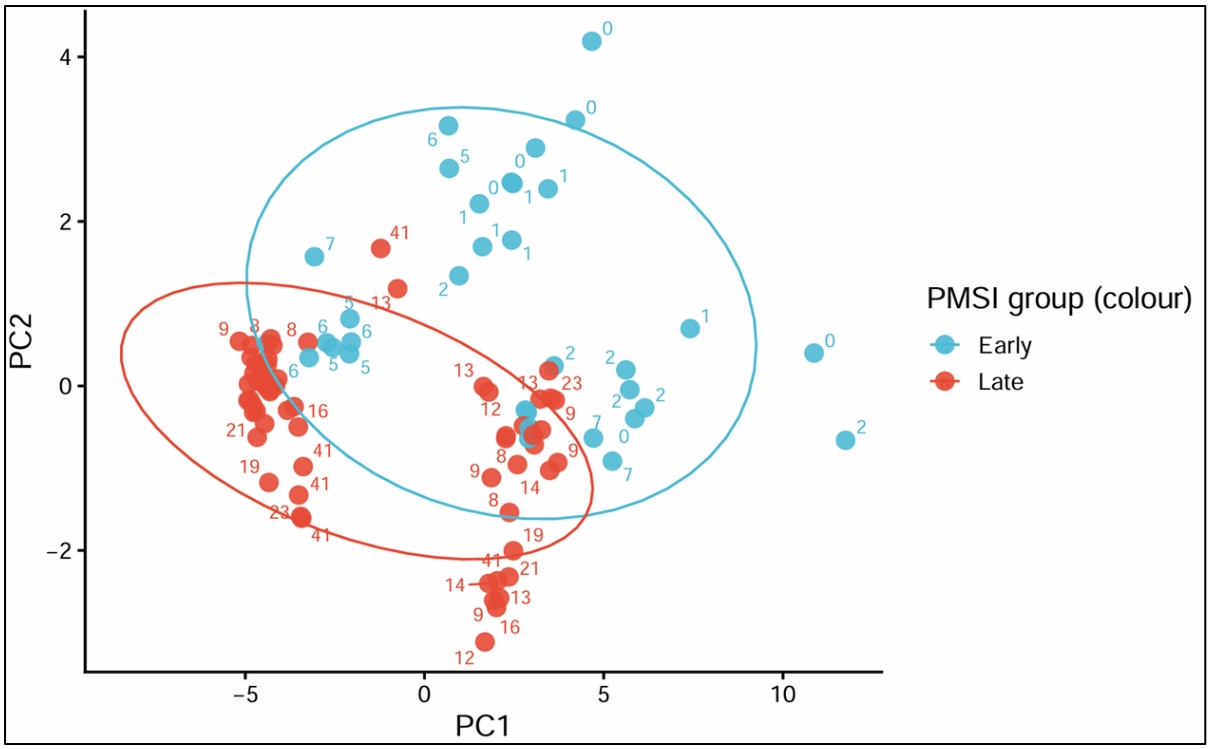


Figure 30 - PCA plot with PMSI days marked on each datapoint to identify group clustering

Figure 31 revealed high overlap across PMSI days, with no clear temporal gradient in the PCA space. While earlier days (0-2) were more frequently located at positive PC1 values and late days towards negative PC1 values, this pattern is not distinct. Intermediate PMSI days were distributed between both clusters. Overall, although some PMSI days showed slight clustering, there is no real gradual trend or directional pattern across the PCA space. Ideally, in metabolomic studies of PMSI, PCA plots often show distinct groupings for each day, reflecting how metabolic profiles evolve over time (Zhang *et al*, 2024). In this dataset, some days are situated closer together (e.g. days 1, 2, 5, 6 and 41), whilst others are more dispersed or appear as outliers. This could either reflect the variable nature of decomposition or technical factors (Zhang *et al*, 2024).

Overlap between early and late PMSI is also evident, suggesting that metabolic changes are not entirely distinct, but instead occur along a continuum, contributing to the observed overlap between groups in the PCA space (Werth *et al*, 2010).

Within-group variability can also be observed, with early PMSI samples showing greater dispersion. This could reflect biological heterogeneity which can indicate biological and technical diversity among samples such as, interindividual differences and technical noise (Marco-Ramell et al, 2018). Preprocessing steps, such as imputation, could also influence the

data due to PCA being highly sensitive to small artificial changes. The confidence ellipse further supports these observations, with early PMSI displaying a greater spread compared to the more compact distribution of late PMSI samples. Typical grouping patterns have been reported in metabolomic research investigating PMI, such as Locci *et al* (2019), where progressive metabolic changes that occur after death cause systematic change in metabolite concentrations over time. As decomposition progresses through biological processes like autolysis and microbial activity and cellular degradation, metabolic profiles rapidly change and are often reflected in multivariate techniques like PCA.

7.8.4 **Outlier Analysis**

Alongside these patterns, several outliers can be observed within the PCA plot; four located outside the ellipses and one located inside. Outliers are considered as samples that deviate from the majority of features and can contribute uniquely to the model variance (Walach *et al*, 2019). These are caused by technical errors, biological variation or data processing errors (Li *et al*, 2006). If outliers were to be determined biologically relevant rather than technical errors, it would often represent an extreme physiological state or unique biological response (Julian *et al*, 2024). Figure 32 helps to visualise this curiosity, with outliers labelled by their day and mouse.

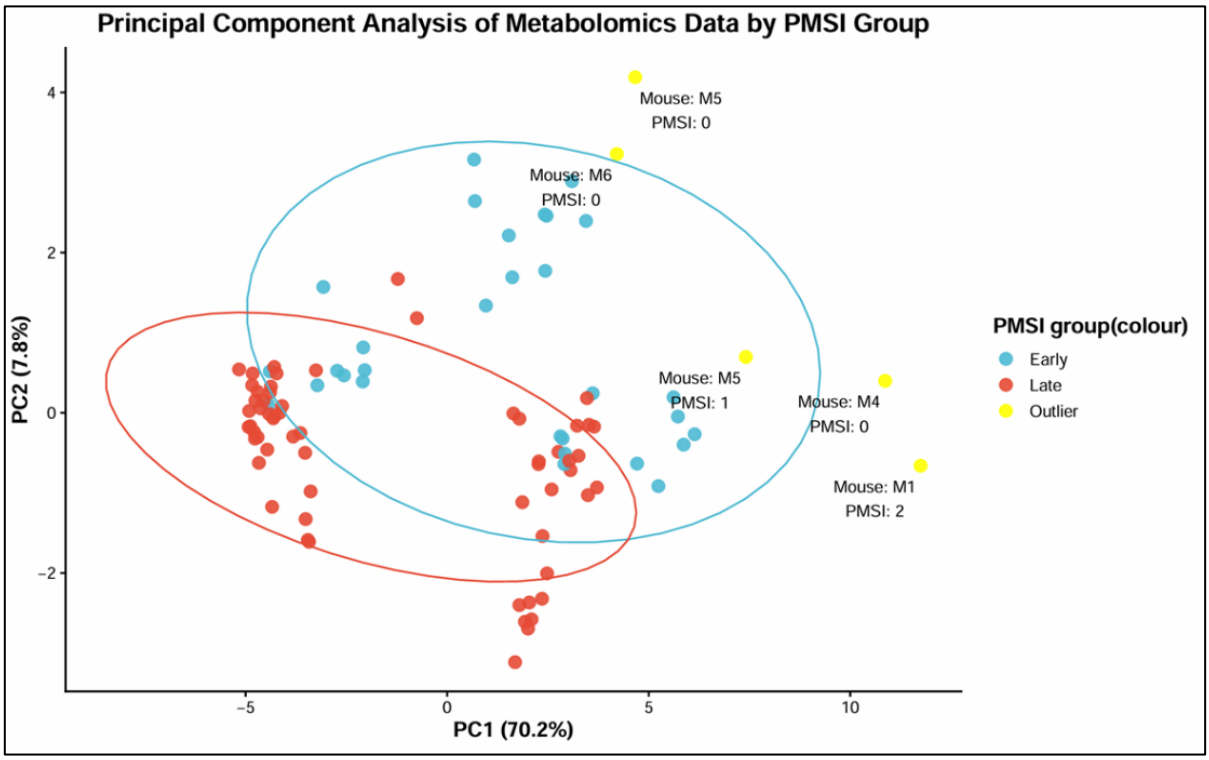


Figure 31 - PCA plot but with the addition of outliers, marked and labelled.

Several outliers can be observed in the PCA plot, with four samples located outside the confidence ellipses and one within (e.g., M1, M4, M5, M6). These points are primarily separated along PC1, indicating that they are driven by features contributing strongly to the dominant source of variance in the dataset. Notably, samples M4 (PMSI 0) and M1 (PMSI 2) show particularly high PC1 values, suggesting that specific metabolites with large intensities are influencing their position.

Reviewing the data matrix supports this interpretation. For example, M1 (PMSI 2) showed an unusually high intensity for metabolite 257.0712@1.78 (530,961.113) exceeding the majority values. This value was more than three times larger than the next highest value (166,566.625), while most were below 1,000. A similar pattern was observed in M4 (PMSI 0), with the same metabolite also reaching a higher value than the rest (663,666.353). The next largest value was 18,386.333, with other values falling below 1,000. These extreme values are likely to have a strong influence on PC1, which reflects overall variance across metabolites.

In contrast, M5 (PMSI 0) also demonstrated elevated intensities in metabolite 257.0712@1.78 (921,122.2), but it did not deviate as markedly as observed in previous mice. Its next largest value was of a comparable magnitude (633,885.3), suggesting a less pronounced influence on PCA positioning.

Identifying these outliers was important for assessing the suitability of the dataset for PMSI prediction. It highlighted the sensitivity of PCA to extreme values, as samples with unusually high metabolites can disproportionately influence the orientation of principle components. The presence of these extreme values suggests that the data are not normally distributed and may reflect the biological and environmental variability that is common in aquatic decomposition (Zhang *et al*, 2024). Such observations have the potential to influence model fitting and predictive accuracy if not carefully evaluated. However, it cannot be assumed that these outliers represent an error, as they could also be capturing genuine extremes in the decomp process. Consequently, outlier identification was not done to exclude extreme values, but to evaluate their potential effect on model robustness and ensure the final model remained statistically reliable and forensically meaningful.

7.8.5 **Metabolites driving PC1 and PC2**

Linking these observations to the PCA loadings (Figure 33 & Figure 34) provides further insight into which metabolites contribute most strongly to PC1 and therefore drive the separation of these outliers.

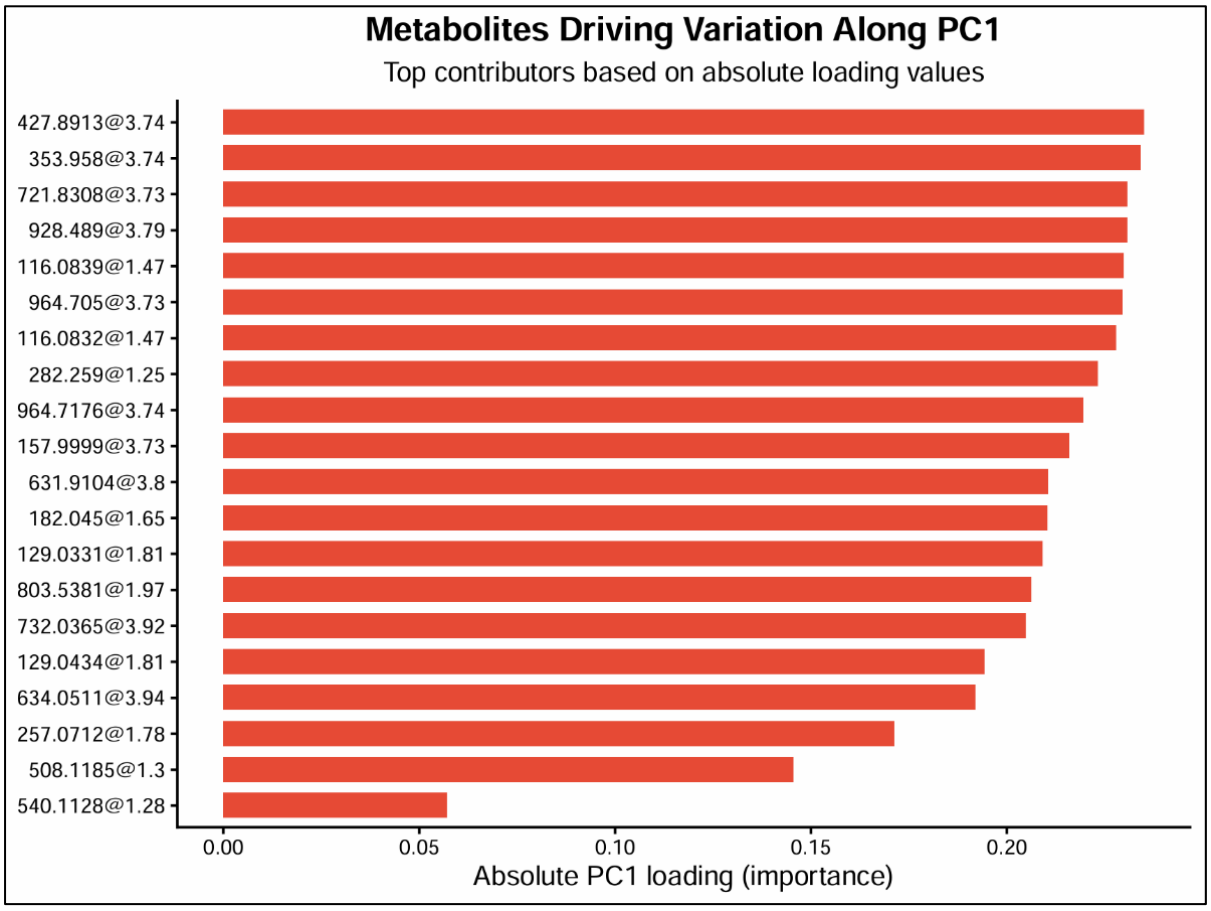


Figure 32- Bar chart showing the top metabolites contributing to the variance along PC1 ranked by absolute loading values

Examination of the PC1 loadings revealed that all metabolites contribute to negative loadings. This indicates that PC1 does not represent opposite metabolic trends but rather reflects a cumulative effect of metabolite abundance across the dataset, common in metabolomic datasets (Sitt *et al*, 2025). In this context, PC1 can be interpreted as a weighted combination of metabolite intensities, separating samples with higher overall abundance from those with lower abundance. As PCA is an unsupervised method, this separation arises from inherent variance in the data rather than predefined groupings. The scree plot supports this interpretation by showing that PC1 explains more variance than other components (e.g. PC6).

Therefore, the separation observed along PC1 in the scores plot likely reflects broad changes in metabolite abundance across PMSI, rather than variation driven by specific metabolites acting in opposite directions. This suggests that decomposition is associated with coordinated changes in overall metabolite abundance, rather than alterations confined to specific metabolic pathways.

However, in contrast to PC1 which reflects overall metabolite abundance, PC2 demonstrated a clear separation of metabolites with both positive and negative loadings (Figure 34). This indicates that PC2 captures opposing metabolic trends in the dataset, rather than cumulative intensity efforts.

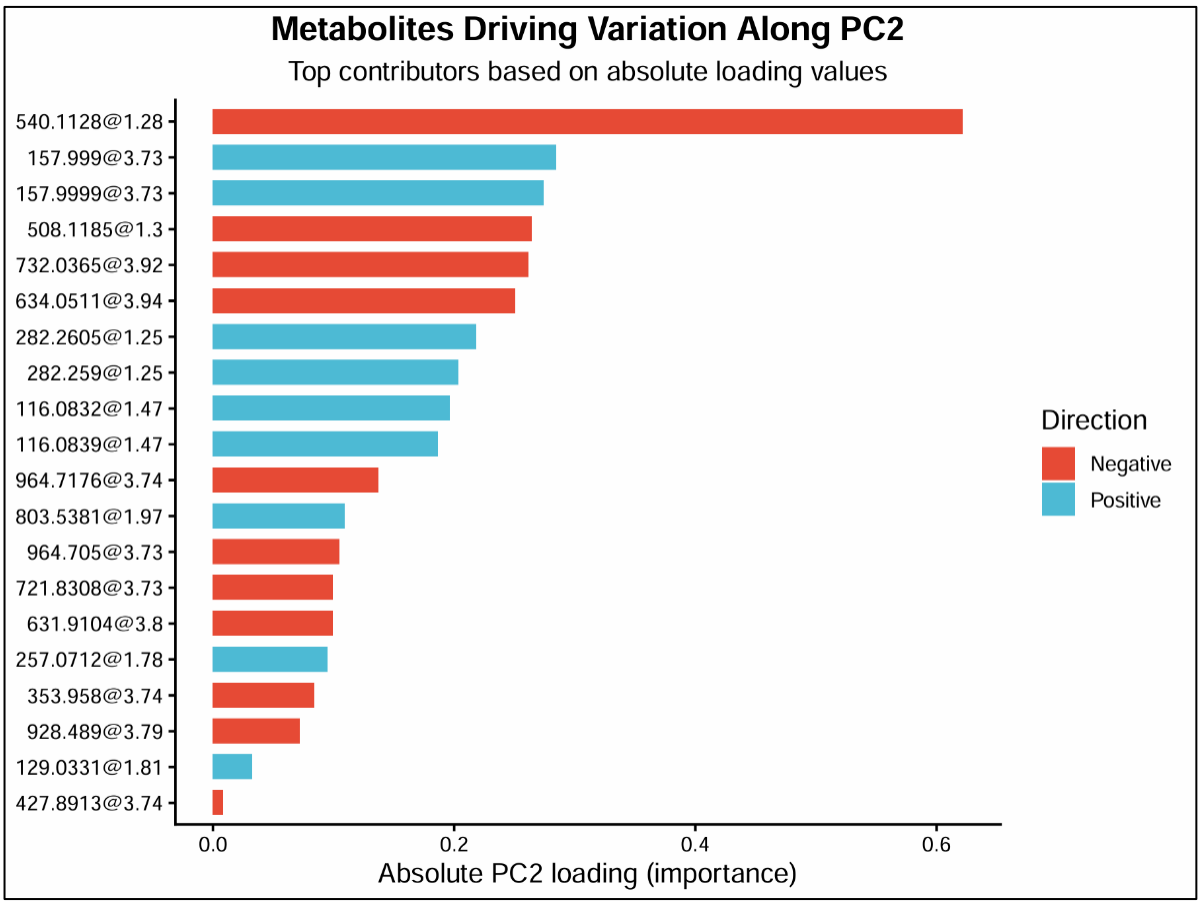


Figure 33- Bar chart showing the top metabolites contributing to variance along PC2 ranked by absolute loading values.

Metabolites with positive loadings are associated with samples on the positive side of PC2, whereas those with negative loadings are associated with samples on the opposite side, highlighting a contrast between metabolites that vary in opposing directions. This suggests that the variation captured by PC2 reflects differential regulation or presence of specific metabolites.

Metabolites with the highest absolute loadings (e.g. 540.1128@1.28, 157.9999@3.73 and 508.1185@1.3) are therefore key drivers in the separation. These metabolites could therefore represent potential biomarkers for distinguishing stages of PMSI, as they contribute to the separation of samples beyond general metabolic intensity. This can also be supported by referring to the data matrix which demonstrates that metabolite 540.1128@1.28 shows substantial variability across samples, with samples ranging from the low thousands to over a million. This wide range indicates a high degree of dispersion and a significant change across PMSI. Similar trends can also be observed in the other high metabolites.

Therefore, this could suggest that PC2 may hold greater relevance in predictive modelling, as it captures specific biochemical variation rather than overall signal magnitude.

7.9 **Random Forest Modelling for PMSI Prediction**

Following PCA analysis, Random Forest (RF) emerged as the most suitable approach to developing a predictive model for PMSI using the now reduced datasets. This technique constructs an ensemble of decision trees, each trained on random subsets of data, to generate predictions by aggregating the outcome of these trees (Salman *et al*, 2024). The application of machine learning technology in the prediction of PMSI from metabolomic data offers the capability to navigate through complex datasets, undercover hidden correlations that would otherwise go unseen, and robustly improve the accuracy of postmortem interval estimation (Aljeaid, 2024).

The selected datasets were divided into a training and test subset using an 80:20 split. This train-test split is a frequently used rule of thumb in RF that balances training enough data (80%) to develop a robust model with holding back enough data (20%) to adequately evaluate its ability to generalise to new, unknown data, hence reducing overfitting (Sivakumar *et al,* 2024). Model tuning was also carried out using a 5-fold cross-validation repeated five times, which is a machine learning (ML) process that divides the data into five equal parts, training the model on four folds and testing on the remaining one (Baturynska & Martinsen, 2020). This estimates the average performance, which improves model reliability and reduces overfitting.

Final model selection was based on the minimum root mean square error (RMSE), with predictive accuracy quantified using RMSE, mean absolute error (MAE) and coefficient of determination (R2). Together, these measures evaluate the performance of regression models.

Due to PC2 scores reflecting the most biologically significant results, the first regression model was built on the top PC2 metabolites (Figure 34).

7.9.1 **PMSI ~ PC2 Scores**

Analysis began by selecting the metabolite variables with the highest loadings in PC2. Initially, predictor variables were renamed as simplified model labels (M1, M2, M3 etc.) to prevent errors caused by special characters or long metabolite names during model training. A preserved independent mapping database allowed the labels to be traced back to their original metabolite. The predictive performance of the resulting random forest model is shown in Figure 35, where observed PMSI values are plotted against model predictions.

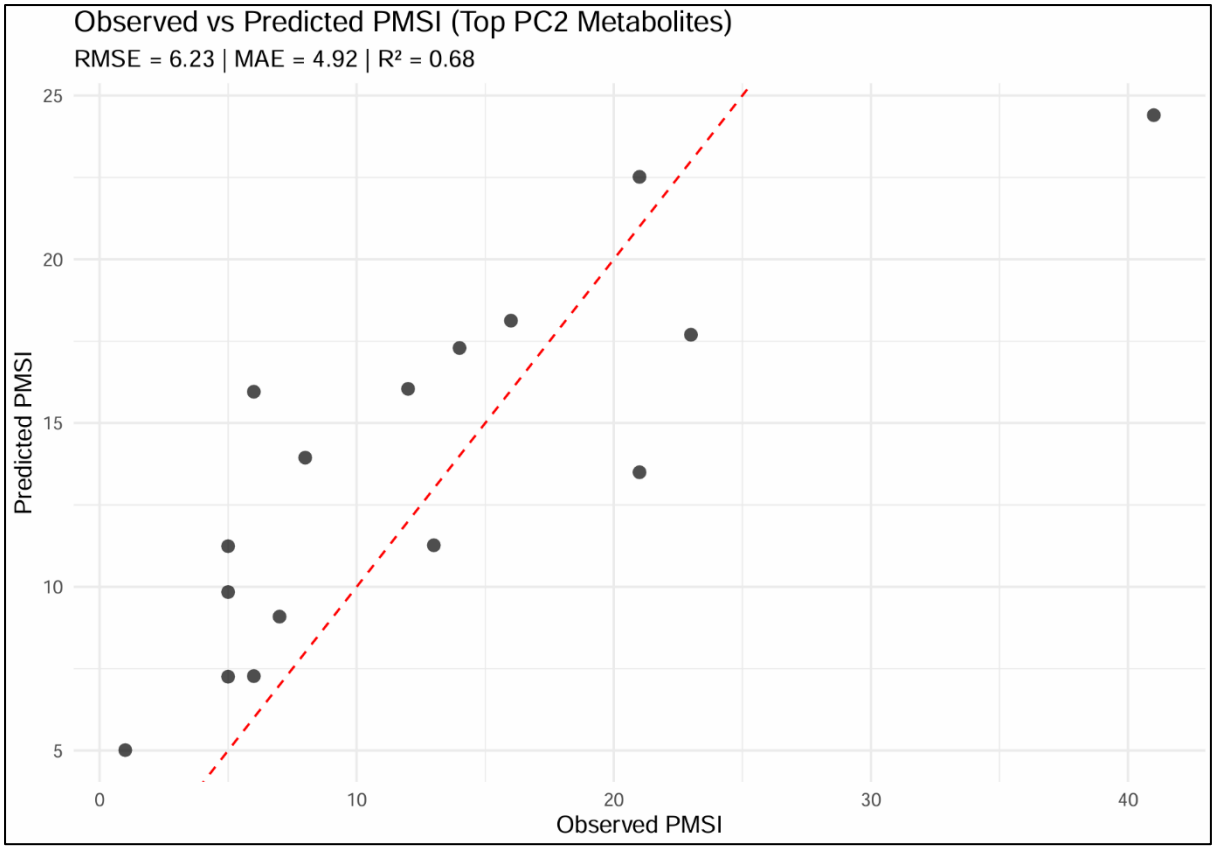


Figure 34 - Observed versus predicted PMSI values for the random forest model based on top PC2 metabolites. The dashed red line indicates perfect prediction.

The model demonstrated moderate predictive accuracy, with an RMSE of 6.23 and a MAE of 4.92, indicating that predicted PMSI values deviated from observed values by around 5-6 units on average. The coefficient of determination (R^2) suggests that 68% of variance in PMSI was explained by the selected metabolite features. When visually analysing the plot, a general positive relationship can be identified between predicted and observed PMSI values, with

points roughly distributed around the line of perfect agreement. However, it is evident that as PMSI days increase, so does the dispersion from the line (days 21, 24) indicating reduced prediction accuracy in this range. This is particularly evident for day 41, which lies outside the main distribution of the dataset.

Several points fall considerably below or above the line, indicating the model occasionally overestimates and underestimates. The apparent gap between days 25-40 could be due to discrete and uneven distribution of PSMI values in the test dataset, rather than missing observations. Each point represents an individual sample rather than time, with only the held-out test set visualised for clarity.

The moderate predictive performance observed suggests that metabolites associated with PC2 capture biologically relevant changes linked to PMSI, but do not fully account for all the variability within the dataset. The average R^2 value of 0.68 indicates a meaningful relationship between metabolite profiles and submergence interval, supporting past interpretations in PCA analysis, as well as its suitability as a feature selection strategy. However, the difference in RMSE (6.23) and MAE (4.92) values suggest the presence of occasional large prediction errors, which could reflect increased biological variability or noise within the data.

Despite these limitations, the models demonstrated that a subset of PC2-associated metabolites has a moderate predictive accuracy in PMSI estimation. The ability for RF to capture non-linear relationships likely contributed to its performance, highlighting its suitability in metabolomic studies. However, the observed predicted errors indicate the need for additional features, alternative components or larger datasets to improve model robustness and predictive accuracy. Therefore, the next regression model was applied to principal components (PC1-PC10) to see if a larger dataset had greater effect.

7.9.2 **PMSI ~ PCA**

The same method in section 8.7.1 was applied to this model, with the only difference being the dataset. Although only two PCs were analysed (PC1 and PC2), signals can spread across multiple components because PCA is designed to maximise variance explanation rather than isolate specific, independent biological processes (Yao *et al*, 2012). This means that different PCs could hold varied decomposition information such as metabolic shifts, cellular breakdown and protein degradation (Yao *et al*, 2012). The predictive performance of the first 10 principal components is shown in Figure 36.

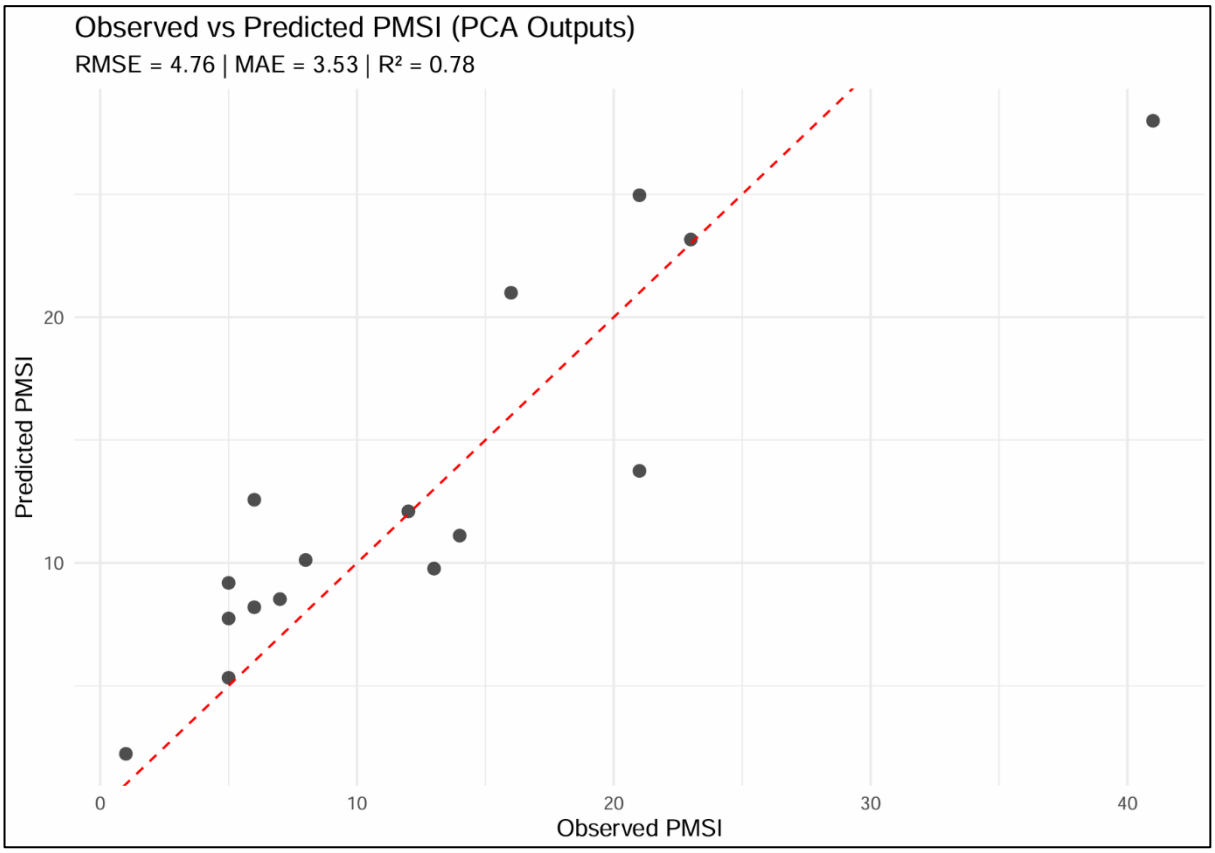


Figure 35- Observed versus predicted PMSI values for the random forest model based on the first 10 PCs. The dashed red line indicates perfect prediction.

The model demonstrated greater predictive accuracy, with an increased positive relationship between predicted and actual PSMI values, as evidenced by the clustering of points around the line of perfect fit. The model achieved an R^2 of 78% indicating that a substantial proportion of the variance in PMSI is explained by the RF model using the first 10 principal components.

It is still apparent that in late PMSI days (e.g. 17, 21), the model exhibits limitations by overestimating and underpredicting values, indicating reduced predictive accuracy in this range, with the highest PMSI value (day 41) again appearing as a clear outlier. In addition to a potential uneven distribution in the test dataset, the smaller number of samples could also create an imbalanced distribution that biases the model to more common, moderate values. This pattern is consistent with earlier PMSI days, where a greater number of observations results in improved predictive accuracy. Despite this, the relatively low RMSE (4.76) and MAE (3.53) indicate that the model's predictions are generally close to the observed PMSI days, with average prediction errors of approximately 3-5 PMSI days.

While the PCA-based RF model demonstrated improved predictive performance, this evaluation is based on a conventional test-train split, which may include samples from the same mouse in both training and test sets. To address this limitation and provide a more accurate assessment of the model's generalisability, a leave-one-out (LOO) validation approach was implemented.

7.9.3 **PMSI ~ LOO on PCA**

A leave-one-out (LOO) cross validation approach was applied where within each iteration, PCA was performed on the training dataset and the resulting principal components were used to train the random forest model. The same PCA transformation was applied to the excluded mouse prior to prediction.

This approach ensures that all samples from a single mouse are excluded during training and used solely for testing, therefore preventing data leakage arising from within-subject similarity. Given the biological variation between mice, LOO allows for evaluation of the model's ability to generalise to entirely unknown mice, providing a more realistic estimate of predictive performance. The performance of the PCA-based random forest model on mice 1-6 is presented in Figure 37.

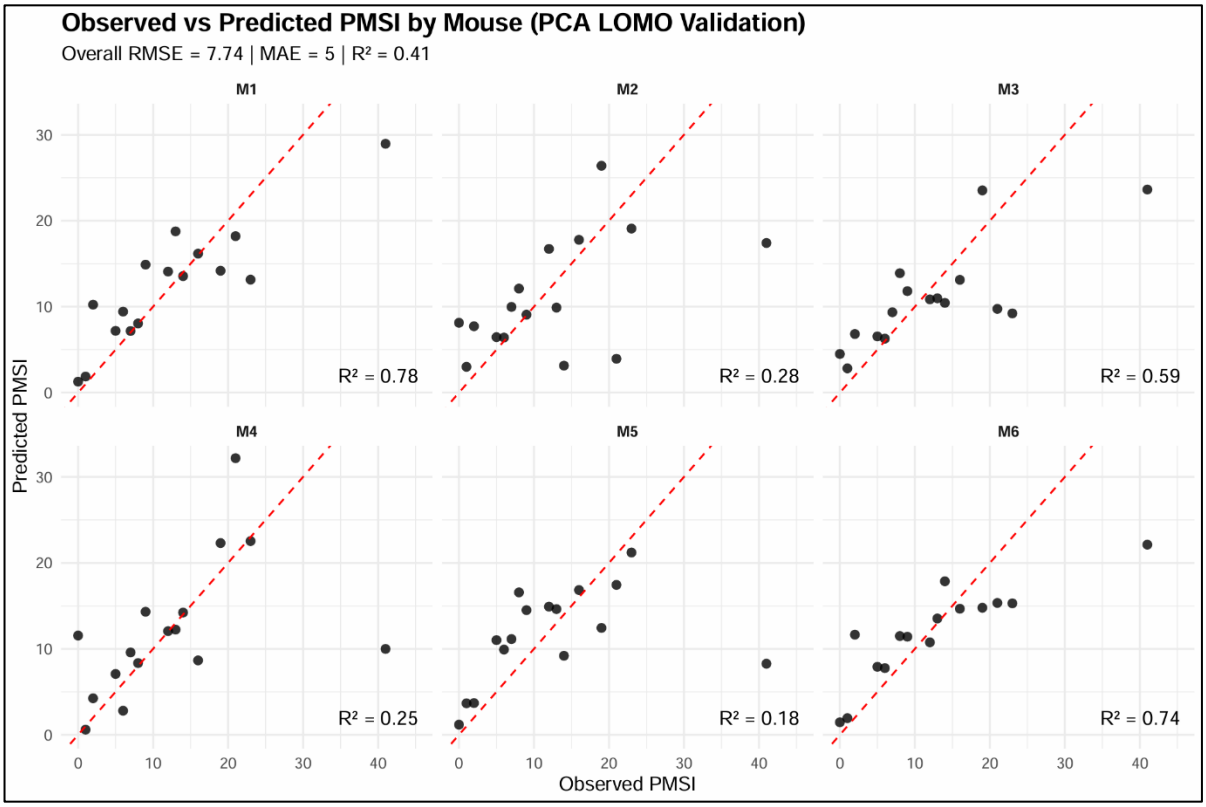


Figure 36 -Observed vs Predicted PMSI by Mouse Using PCA-Based Random Forest with Leave-One-Mouse-Out Validation

The LOO validation revealed a substantial reduction in model performance in comparison to the standard train-test split, with an overall R^2 of 0.41 as well as an increased prediction error (RMSE = 7.74, MAE = 5). The results show that the PCA-based model has limited ability when generalised to completely unknown mice. Day 41 again exerts a strong influence on model performance, resulting in systematic underprediction.

However, it can be observed that performance varied between mice, with M1 and M6 showing the highest predictive accuracy ($R^2 > 0.7$), while M4 and M5 displayed the poorest performance ($R^2 < 0.3$). The variability reflected can be linked back to earlier findings (section 8.6) that highlighted high inter-individual differences in metabolomic trajectories between mice. This can mean that patterns learned from one set of individuals is not always transferable to others, highlighting the importance of incorporating biological variability into predictive modelling frameworks for PMSI estimation.

Interestingly, M6 demonstrated a relatively strong predictive performance despite earlier observations showing it exhibited the most distinct metabolomic trajectory. This can suggest that a clearer, more defined internal pattern does not necessarily imply poor predictability. In

this case, the consistency of M6's trajectory may have enabled the model to capture its underlying structure more efficiently, even when trained on other mice. In contrast, mice M4 and M5 displayed greater variability and coherent trajectories, which likely contributed to their poor predictive performance under LOO validation. These findings could suggest that the consistency of metabolic patterns, rather than similarity alone, plays a vital role in model prediction.

8.0 **Metabolite identification from RF**

Within the code for developing a PCA-based random forest regression model, a principal component importance plot was generated that displayed which components contributed the most to the model's prediction (Appendix T). PC2 was identified as the main contributor and therefore an additional importance plot was created for the top metabolites that were directly relevant for predicting PMSI, including non-linear relationships not captured by principal components. These can be observed in Figure 38.

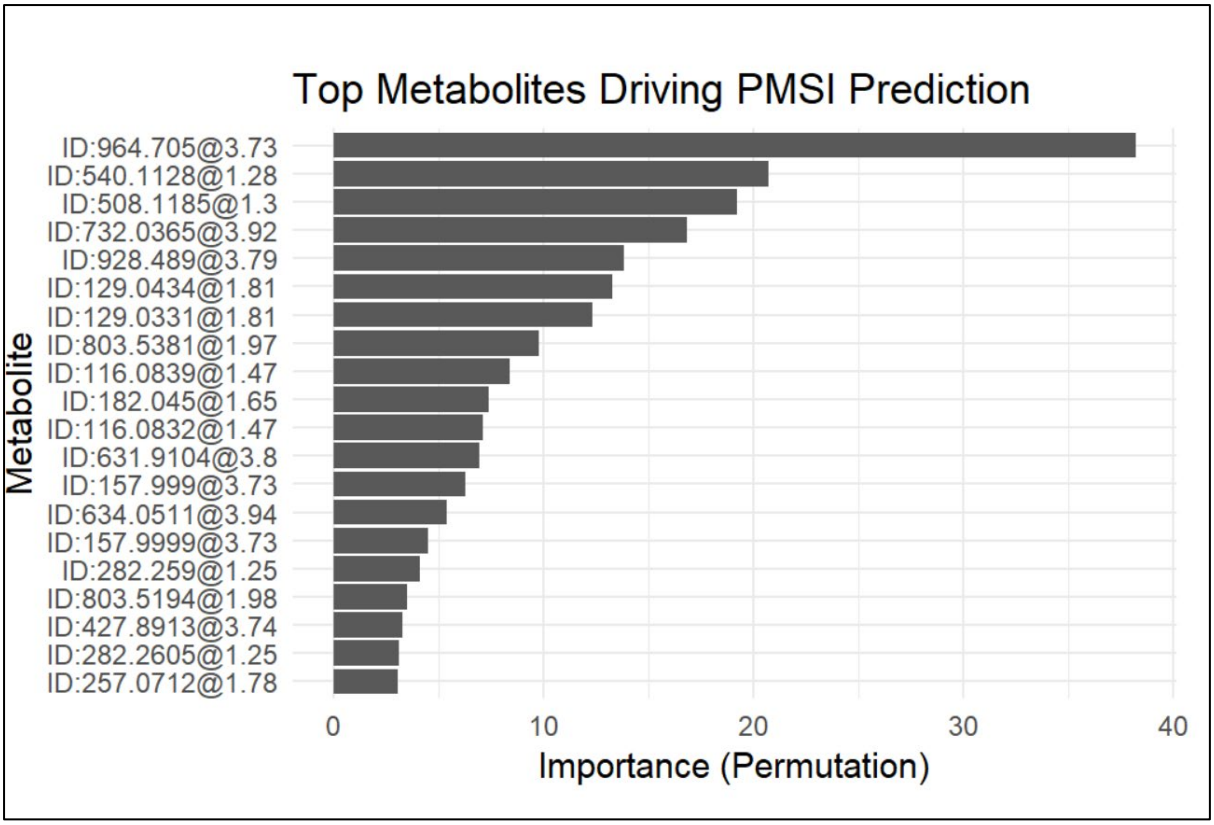


Figure 37- Importance plot showing the top metabolites in PC2 that drive PMSI prediction

From the results of the importance plot, the top three metabolites (ID: 964.705@3.73, 540.1128@1.28 and 508.1185@1.3) were searched within the Human Metabolome Database

(HMDB), an electronic database containing detailed information about metabolites found in the human body (HMDA, 2018). The observed m/z value (e.g. 964.705) was queried against the database using the LC-MS search function, specifying positive Ion mode, unknown adduct type, and a molecular weight tolerance of +-5 ppm (parts per million). This approach enabled the identification of candidate metabolites based on accurate mass matching.

However, due to a limited structural information available from the data alone and the unspecified adduct forms, the database search returned multiple potential matches for each feature, rather than a single definitive metabolite. Furthermore, none of the candidate metabolites showed clear or consistent association with previously reported biomarkers in PMSI-related studies (See Appendix U-W)

These findings highlight the limitations of relying solely on accurate mass for metabolite identification and indicate that additional confirmatory analyses, such as tandem mass spectrometry (MS/MS), are required to improve identification confidence.

9.0 **Further Discussion**

In this study, we were able to develop a machine learning (ML) model that has improved the capabilities in predicting postmortem submergence interval (PMSI) using the provided metabolomic data. Whilst the models were subjected to controlled experimental conditions, the metabolic and decomposition processes present in the mice, may not fully replicate those observed in the human body. Therefore, caution must be taken when generalising these findings to human PMSI estimation.

The first essential aim was to process and impute the data into a set that was complete enough to provide meaningful analysis and modelling. Initial chromatographic assessment revealed that retained variables displayed clear, asymmetrical peaks, with minimal tailing and rising trends. Early PMSI days exhibited a trend of increased noise, which gradually stabilised as the days progressed, resulting in more defined peak patterns. However, out of 1393 variables, only 178 variables were retained due to a large amount of poor and inconsistent peaks. This significant reduction clearly impacted the end model. This is cautioned in literature surrounding RF, with studies by Luan *et al.* (2020) identifying that while smaller sample sizes can sometimes reduce overfitting, they also reduce model performance, reliability and generalisation due to RF models having difficulties in capturing complex patterns and increased sensitivity to noisy data (Luan *et al,* 2020; Han *et al,* 2021).

The reduced dataset also amplified the presence of missing data, with high levels of missingness present within multiple mice over the 41-day period. This is cautioned by Little & Rubin (2019) that although a small dataset does not create missing data, it can amplify its impact, making gaps appear more significant than they are, thereby increasing variability and potentially reducing the reliability of later analysis.

Despite these limitations, the chosen method to split the data into early and late days and apply separate imputation methods (QRILC and KNN), proved to be effective while maintaining the general structure and underlying trends of the data. QRILC successfully estimated the distribution of low values by sampling them from the lower quartile, thereby reflecting low-abundance measurements that were below detection level. This approach avoided overestimation and instead provided more accurate imputed values. Similarly, KNN ($k = 5$) enabled a balance between reducing noise whilst preserving logical structure of the data, ensuring that imputed values were based on observations with similar properties without excessive smoothing. Therefore, it can be argued that if these methods were applied to a larger and more complete dataset, this combined imputation approach would have be reliable and robust, with the strong potential to enhance the accuracy of missing value estimation and support downstream modelling of PMSI.

Principal component analysis (PCA) was utilised due to it being a widely recognised explanatory tool in metabolomic data (Nyamundanda *et al*, 2010) and proved to be an effective explanatory framework for assessing variation within the metabolomic dataset. This can be supported through multiple studies using PCA plots in metabolomics and PMI estimation, where distinct groups of metabolites and days have successfully been identified (Zhang *et al*, 2024; Bonicelli *et al*, 2022; Davidson (2025).

Additionally, the use of scree, score and loading plots allowed for both global and feature-level interpretation, with both biological (PMSI progression and decomposition processes) and technical (batch effects, preprocessing, and instrument variation) sources of variation critically considered and addressed. Groupings and outliers were also investigated ensuring that extreme values were assessed rather than removed. This multi-level and critical interpretation enhanced the robustness and reliability of PCA findings.

However, PCA is inherently an unsupervised technique that ranks principal components based on variance explained rather than their relevance to PMSI outcomes. While it was observed that PC2 captured greater biochemical variation, an importance plot generated alongside RF

identified PC6 to also include high importance for PMSI prediction. This supports previous findings that biologically meaningful variation may not be captured in the first principal components (Lenz *et al*, 2016). For example, *Magnusson et al*. (2026) proved that PCA can fail to capture biologically meaningful variation in the early PCs and instead it appears in later components. This highlights the need for multi-level interpretation beyond leading principal components.

When assessing the results of the PCA plots, it is important to consider that its disordered appearance could potentially be due to biological factors. Although PMSI provides a temporal framework, Weisensee & Atwell. (2024) noted that decomposition does not progress uniformly due to its complex biological and chemical processes, governed by highly variable environmental factors, substrate quality, and microbial activity. Therefore, decomposition processes, as described in the introduction, may accelerate or decelerate metabolic changes (Almulhim & Menezes, 2023). Additionally, Yu *et al* (2021) mentions that tissue-specific composition also affects rate of decay, with soft tissues decaying faster than hard and fatty tissues. In the original study, three female and three male mice were used, weighing 18.94g – 23.71g. Differences in fat distribution, hormonal profiles, and metabolic rate between male and female mice can influence both the rate and pathway of decomposition. Similarity, variation in body weight may affect tissue composition and moisture content. These intrinsic biological differences could also influence microbial colonisation and activity, and when combined with stochastic variation in microbial community dynamics, can result in differences between samples at equivalent time points (Campobasso *et al*, 2001).

However, technical factors such as sampling handling and analytical variation can also cause group separation. Hann *et al*. (2024) highlighted that issues such as batch effects, temperature fluctuations or calibration errors often cause structure variance because PCA ranks components by variance. These often appear in PC1, which therefore drive the separation (Razifar *et al*, 2009). This is also apparent in recent work by Magnusson *et al*. (2026), who observed that large sources of variation (e.g. environmental effects, technical noise, system-wide changes) can dominate the data structure. Collectively, these factors contribute to the trends reflected in the dataset and must be considered when interpreting results.

Random forest modelling was finally applied to selected features and demonstrated that metabolomic data can be used to predict PMSI with moderate accuracy, depending on feature selection and validation strategy. The PC2 model demonstrated reasonable performance

(R^2=0.68), whilst the model incorporating all principal components (PC1-PC10), improved predictive accuracy (R^2 = 0.78). This is a considerably successful outcome, with work done previously around this topic by Davidson (2025) only having an R^2 = 0.548. This also supports the hypothesis that important information could have been distributed across higher PCs

The reduced predictive accuracy observed in later PMSI days could be attributed to greater heterogeneity in metabolomic profiles as decomposition progresses. As discussed by Almulhim & Menezes (2023) in advanced decomposition, there are significant biochemical changes that are occurring in the body such as microbial accumulation and tissue breakdown which can be highly affected by environmental conditions. This can introduce variability that is not able to be fully captured by the reduced selection of metabolite features, as the smaller number of samples available at late PMSI days may limit the model's ability to learn these patterns effectively. This can be supported by James *et al*. (2013) as random forest models are unable to extrapolate beyond the range of the trained dataset making predictions for extreme PMSI values biased towards the central distribution.

However, leave-one-out (LOO) validation revealed reduced performance (R^2 = 0.41), highlighting limited generalisability and the impact of inter-individual variability. This is reflected in recent work by M Løber *et al*. (2025) where LOO validation is effectively used in metabolome data and states it is an inherent feature of biological data and would reflect the same result in real world application. However, the limitation of LOO is also highlighted, as it may introduce distributional bias, which could lead to distorted performance estimates and affect model evaluation (M Løber *et al,* 2025).

10.0 **Conclusion and further work:**

The overall aim of this study was to develop a machine learning (ML) model capable of predicting postmortem submergence interval (PMSI) using the provided metabolomic data. This approach was intended to enhance traditional methods surrounding PMSI estimation, which are consistently limited by the environmental factors in marine settings. The first aim was curate and extract LC-ToF-MS metabolomic data using Agilent MassHunter Profinder Software. 1393 features were detected across the samples, however due to a large number of inconsistent and poor peaks, this was reduced to 178 for the next stage of analysis. After exploring the missing data, a cut-off threshold of 0.25 was applied, where a shortlist of 23 features was retained, whose missing data was then imputed separately with QRILC and KNN.

Despite imputation, when visualising the temporal metabolite trends each mouse (M1-M6), the dataset remained heterogeneous with uneven distribution and persistent gaps, particularly in later PMSI days. However, locally weighted scatterplot smoothing (LOESS) revealed underlying non-linear temporal trends consistent with the decomposition process. Variation between individual mice suggested that biologically meaningful patterns are present, but are obscured by noise and missing data, which supported the use of machine learning algorithms, such as random forests (RF), to model these complex relationships.

To further assess any trends, PCA analysis was utilised and showed that a clear combination of biological and technical factors influenced the unstructured trends present in the plots, which showed a non-linear trend to PMSI, even after imputation. PC2 demonstrated to have the highest biological importance, with PC1 only reflecting global trends. However, it was established that later PCs could have provided more important biological data, with PC6 ranking high alongside PC2.

Two random forest models were generated as a result of PC findings. The model using PC2-associated metabolites showed moderate predictive performance ($R^2 = 0.68$; RMSE = 6.23), capturing general PMSI trends but with reduced accuracy at later PMSI. Building on the limitation of this smaller model, the second model used all principal components (PC1-PC10) and achieved an improved $R^2 = 0.78$ and RMSE = 4.76, which was then cross validated via leave-one-out (LOO) validation.

LOO validation revealed a reduced model performance ($R^2 = 0.41$; RMSE = 7.74), indicating limited generalisability to unseen data. Performance varied between individuals, reflecting high biological variability, with more consistent metabolic pathways improving predictability, highlighting the importance of accounting for inter-individual differences.

Despite the reduced model performance observed in LOO validation, both RF models demonstrated strong predictive capability, capturing key non-linear relationships within the metabolomic data and producing comparatively high R^2 values. This indicates that whilst generalisability remains a challenge, random forest has proved to be highly effective in modelling the complex nature of PMSI, reinforcing its suitability as a powerful tool for metabolomics-based forensic estimation. Therefore, using PCA and RF is highly recommended.

11. 0 **References**

Abdulkreem Abdullah AlJuhani, Rodan Mahmoud Desoky, Binshalhoub, A.A., Alzahrani, M.J., Mofareh Shubban Alraythi and Alzahrani, F.F. (2025). Advances in postmortem interval estimation: A systematic review of machine learning and metabolomics across various tissue types. *Forensic Science Medicine and Pathology*. doi:https://doi.org/10.1007/s12024-025-01026-3.

Agilent (2024). *Basics of LC/MS - a primer | Agilent*. [online] Agilent.com. Available at: https://www.agilent.com/en/product/liquid-chromatography-mass-spectrometry-lc-ms/lcms-fundamentals [Accessed 31 Mar. 2026].

Agilent Technologies, Inc. (2017). *Agilent G3835AA MassHunter Mass Profiler Professional Software: Application Guide*. [online] Available at: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.agilent.com/cs/library/usermanuals/public/G3835-90028_MassProfilerPro_Application.pdf [Accessed 19 Jan. 2026].

Allaire, JJ, Yihui Xie, Christophe Dervieux, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, et al. 2025. *Rmarkdown: Dynamic Documents for r*. https://github.com/rstudio/rmarkdown

Al-Juhani, A.A., Gaber, A.M., Desoky, R.M., Binshalhoub, A.A., Alzahrani, M.J., Alraythi, M.S., Showail, S. and Aseeri, A.A. (2025). From microbial data to forensic insights: systematic review of machine learning models for PMI estimation. *Forensic science, medicine, and pathology*, [online] pp.10.1007/s12024-02501002-x. doi:https://doi.org/10.1007/s12024-025-01002-x.

Aljeaid, R. (2024). Application of Metabolomics and Machine Learning for the Prediction of Postmortem Interval. *Cureus*. doi:https://doi.org/10.7759/cureus.74161.

Almulhim, A.M. and Menezes, R.G. (2023). *Evaluation of Postmortem Changes*. [online] PubMed. Available at: https://www.ncbi.nlm.nih.gov/books/NBK554464/ [Accessed 6 Mar. 2026].

Ardrey, R.E. (2003). *Liquid Chromatography - Mass Spectrometry*. John Wiley & Sons.

Ayhan, K., Coşansu, S., Orhan-Yanıkan, E. and Gülseren, G. (2021). Advance methods for the qualitative and quantitative determination of microorganisms. *Microchemical Journal*, [online] 166, p.106188. doi:https://doi.org/10.1016/j.microc.2021.106188.

B. Donald, R. (1976). *Inference and Missing Data* . [online] Available at: http://qwone.com/~jason/trg/papers/rubin-missing-76.pdf [Accessed 16 Feb. 2026].

Barrett, Tyson, Matt Dowle, Arun Srinivasan, Jan Gorecki, Michael Chirico, Toby Hocking, Benjamin Schwendinger, and Ivan Krylov. 2025. *Data.table: Extension of 'Data.frame'*. https://doi.org/10.32614/CRAN.package.data.table.

Baturynska, I. and Martinsen, K. (2020). Prediction of geometry deviations in additive manufactured parts: comparison of linear regression with machine learning algorithms. *Journal of Intelligent Manufacturing*. doi:https://doi.org/10.1007/s10845-020-01567-0.

Belk, A.D., Deel, H.L., Burcham, Z.M., Knight, R., Carter, D.O. and Metcalf, J.L. (2018). Animal models for understanding microbial decomposition of human remains. *Drug Discovery Today: Disease Models*, 28, pp.117–125. doi:https://doi.org/10.1016/j.ddmod.2019.08.013.

Berg, R.A. van den, Hoefsloot, H.C., Westerhuis, J.A., Smilde, A.K. and van der Werf, M.J. (2006). Centering, scaling, and transformations: improving the biological information content of metabolomics data. *BMC Genomics*, [online] 7, p.142. doi:https://doi.org/10.1186/1471-2164-7-142.

Bonicelli, A., Mickleburgh, H.L., Chighine, A., Locci, E., Wescott, D.J. and Procopio, N. (2022). The 'ForensOMICS' approach for postmortem interval estimation from human bone by integrating metabolomics, lipidomics, and proteomics. *eLife*, 11. doi:https://doi.org/10.7554/elife.83658.

Campobasso, C.P., Di Vella, G. and Introna, F. (2001). Factors affecting decomposition and Diptera colonization. *Forensic Science International*, [online] 120(1-2), pp.18–27. doi:https://doi.org/10.1016/s0379-0738(01)00411-x.

Carter, D.O. and Tibbett, M. (2009). *Cadaver decomposition and soil: processes*. [online] ResearchGate. Available at: https://www.researchgate.net/publication/313107271_Cadaver_decomposition_and_soil_processes [Accessed 31 Mar. 2026].

Caruso, J.L. (2016). Decomposition changes in bodies recovered from water. *Academic Forensic Pathology*, [online] 6(1), pp.19–27. doi:https://doi.org/10.23907/2016.003.

Castro, G., Araujo, B., Araújo, S., Santos, J., José, S., Lucas, S. and Maciel, P. (2023). *MEDICINA LEGAL E TAFONOMIA FORENSE*. [online] Perspectivas em Medicina Legal e

Perícia Médica. Available at: https://www.perspectivas.med.br/2023/04/medicina-legal-e-tafonomia-forensee/.

Chen, B., Jiang, L., Liu, J., Gu, X., Hong, Y., Zhu, D., Li, W., Xu, D., Kuang, K. and He, Z. (2025). What control home-field advantage of foliar litter decomposition along an elevational gradient in subtropical forests? *Plant and Soil*, 512(1-2), pp.1493–1508. doi:https://doi.org/10.1007/s11104-024-07165-w.

Chen, T., Cao, Y., Zhang, Y., Liu, J., Bao, Y., Wang, C., Jia, W. and Zhao, A. (2013). Random Forest in Clinical Metabolomics for Phenotypic Discrimination and Biomarker Selection. *Evidence-Based Complementary and Alternative Medicine*, [online] 2013, p.e298183. doi:https://doi.org/10.1155/2013/298183.

Cheng, W.L., Markus, C., Lim, C.Y., Tan, R.Z., Sethi, S.K. and Loh, T.P. (2022). Calibration Practices in Clinical Mass Spectrometry: Review and Recommendations. *Annals of Laboratory Medicine*, [online] 43(1), pp.5–18. doi:https://doi.org/10.3343/alm.2023.43.1.5.

Cockle, D.L. and Bell, L.S. (2015). Human decomposition and the reliability of a 'Universal' model for post mortem interval estimations. *Forensic Science International*, 253, pp.136.e1–136.e9. doi:https://doi.org/10.1016/j.forsciint.2015.05.018.

Cohen, P.R., Moss, R.J. and Prahlow, J.A. (2025). Livor Mortis and Forensic Dermatology: A Review of Death-Related Gravity-Dependent Lividity and Postmortem Hypostasis. *Cureus*. [online] doi:https://doi.org/10.7759/cureus.90760.

Dalal, J., Sharma, S., Bhardwaj, T. and Dhattarwal, S.K. (2023). Assessment of post-mortem submersion interval using total aquatic decomposition scores of drowned human cadavers. *Journal of Forensic Sciences*. doi:https://doi.org/10.1111/1556-4029.15220.

Dekermanjian, J.P., Shaddox, E., Nandy, D., Ghosh, D. and Katerina Kechris (2022). Mechanism-aware imputation: a two-step approach in handling missing values in metabolomics. *BMC bioinformatics*, 23(1). doi:https://doi.org/10.1186/s12859-022-04659-1.

Do, K.T., Wahl, S., Raffler, J., Molnos, S., Laimighofer, M., Adamski, J., Suhre, K., Strauch, K., Peters, A., Gieger, C., Langenberg, C., Stewart, I.D., Theis, F.J., Grallert, H., Kastenmüller, G. and Krumsiek, J. (2018). Characterization of missing values in untargeted MS-based metabolomics data and evaluation of missing data handling strategies. *Metabolomics*, 14(10). doi:https://doi.org/10.1007/s11306-018-1420-2.

Dolan, J. (2015). Detective Work, Part II: Physical Problems with the Column. *LCGC North America*, [online] 33(12), pp.894–899. Available at: https://www.chromatographyonline.com/view/detective-work-part-ii-physical-problems-column [Accessed 13 Apr. 2026].

Donaldson, A.E. and Lamont, I.L. (2013). Biochemistry Changes That Occur after Death: Potential Markers for Determining Post-Mortem Interval. *PLoS ONE*, 8(11), p.e82011. doi:https://doi.org/10.1371/journal.pone.0082011.

Dong, Y. and Peng, C.-Y.J. (2013). Principled Missing Data Methods for Researchers. *SpringerPlus*, [online] 2(1). Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3701793/ [Accessed 24 Feb. 2026].

Eden, R.E. and Thomas, B. (2018). *Algor Mortis*. [online] Nih.gov. Available at: https://www.ncbi.nlm.nih.gov/books/NBK534875/ [Accessed 23 Feb. 2026].

Endo, R. and Hosobe, H. (2024). Visualization of Time Series Data Using Clustered Heatmaps and Line Graphs. *Proceedings of the 17th International Symposium on Visual Information Communication and Interaction*, pp.1–8. doi:https://doi.org/10.1145/3678698.3678705.

Farooq, A., Sajad Majeed Zargar, Parvaze Ahmad Sofi, Sudan, J., Uneeb Urwat and Hussain, K. (2024). *Concepts and Techniques in OMICS and System Biology*. Elsevier.

Feng, C., Wang, H., Lu, N., Chen, T., He, H., Lu, Y. and Tu, X.M. (2014). Log-transformation and its implications for data analysis. *Shanghai archives of psychiatry*, [online] 26(2), pp.105–109. doi:https://doi.org/10.3969/j.issn.1002-0829.2014.02.009.

Ferrer, I. and Thurman, E.M. (2003). Liquid chromatography/time-of-flight/mass spectrometry (LC/TOF/MS) for the analysis of emerging contaminants. *TrAC Trends in Analytical Chemistry*, [online] 22(10), pp.750–756. doi:https://doi.org/10.1016/S0165-9936(03)01013-6.

Finley, S.J., Benbow, M.E. and Javan, G.T. (2014). Microbial communities associated with human decomposition and their potential use as postmortem clocks. *International Journal of Legal Medicine*, 129(3), pp.623–632. doi:https://doi.org/10.1007/s00414-014-1059-0.

Galloway, A., Birkby, W.H., Jones, A.M., Henry, T.E. and Parks, B.O. (1989). Decay rates of human remains in an arid environment. *Journal of Forensic Sciences*, [online] 34(3), pp.607–616. Available at: https://pubmed.ncbi.nlm.nih.gov/2738563/.

Garg, E. and Zubair, M. (2023). *Mass Spectrometer*. [online] PubMed. Available at: https://www.ncbi.nlm.nih.gov/books/NBK589702/ [Accessed 17 Mar. 2026].

Garrett-Rickman, S.H. (2024). Assessment of taphonomic effects on biomolecule degradation for the estimation of post-mortem interval of human remains. *Uts.edu.au*. [online] doi:http://hdl.handle.net/10453/179444.

Gatto, L. and Lilley, K.S. (2011). MSnbase-an R/Bioconductor package for isobaric tagged mass spectrometry data visualization, processing and quantitation. *Bioinformatics*, 28(2), pp.288–289. doi:https://doi.org/10.1093/bioinformatics/btr645.

Goff, M.L. (2009). Early post-mortem changes and stages of decomposition in exposed cadavers. *Experimental and Applied Acarology*, 49(1-2), pp.21–36. doi:https://doi.org/10.1007/s10493-009-9284-9.

Gohel, D. and Skintzos, P. (2025). *Using the flextable R package*. [online] *ardata-fr.github.io*. Available at: https://ardata-fr.github.io/flextable-book/ [Accessed 16 Apr. 2026].

Grolemund, Garrett, and Hadley Wickham. 2011. "Dates and Times Made Easy with lubridate." *Journal of Statistical Software* 40 (3): 1–25. https://www.jstatsoft.org/v40/i03/.

Gohel, David, and Panagiotis Skintzos. 2025. *Flextable: Functions for Tabular Reporting*. https://doi.org/10.32614/CRAN.package.flextable.

Goldberg, S.B., Bolt, D.M. and Davidson, R.J. (2021). Data Missing Not at Random in Mobile Health Research: Assessment of the Problem and a Case for Sensitivity Analyses. *Journal of Medical Internet Research*, [online] 23(6), p.e26749. doi:https://doi.org/10.2196/26749.

Grassi, V.M., Ciasca, G., Vetrugno, G., Urbani, A., Pascali, V.L. and De-Giorgio, F. (2025). Exploring the post-mortem interval through blood biochemistry: a preliminary case series study and review of the literature. *International Journal of Legal Medicine*. doi:https://doi.org/10.1007/s00414-025-03576-1.

Griffiths, W.J. and Wang, Y. (2009). Mass spectrometry: from proteomics to metabolomics and lipidomics. *Chemical Society Reviews*, 38(7), p.1882. doi:https://doi.org/10.1039/b618553n.

Grolemund, G. and Wickham, H. (2011). Dates and Times Made Easy withlubridate. *Journal of Statistical Software*, 40(3). doi:https://doi.org/10.18637/jss.v040.i03.

Gu, Z. (2022). Complex heatmap visualization. *iMeta*, 1(3). doi:https://doi.org/10.1002/imt2.43.

Guida, R.D., Engel, J., Allwood, J.W., Weber, R.J.M., Jones, M.R., Sommer, U., Viant, M.R. and Dunn, W.B. (2016). Non-targeted UHPLC-MS metabolomic data processing methods: a comparative investigation of normalisation, missing value imputation, transformation and scaling. *Metabolomics*, [online] 12(5). doi:https://doi.org/10.1007/s11306-016-1030-9.

Haglund, W.D. and Sorg, M.H. (2002). *Advances in forensic taphonomy : method, theory, and archaeological perspectives*. Boca Raton, Fla.: Crc Press.

Han, S., Williamson, B.D. and Fong, Y. (2021). Improving random forest predictions in small datasets from two-phase sampling designs. *BMC Medical Informatics and Decision Making*, [online] 21, p.322. doi:https://doi.org/10.1186/s12911-021-01688-3.

Hann, W., Kong, W. and Wen, W. (2024). Thinking points for effective batch correction on biomedical data. *Briefings in Bioinformatics*, [online] 25(6). doi:https://doi.org/10.1093/bib/bbae515.

Hansel, V., Karunarathne, P., Borelli, T.C., Quinn, R. and da Silva, R.R. (2025). ClusterApp to visualize, organize, and navigate metabolomics data. doi:https://doi.org/10.1101/2025.02.12.637912.

Hayman, J. and Oxenham, M. (2016). *Human body decomposition*. London, Uk: Academic Press Is An Imprint Of Elsevier.

Hentges, D.J. (2019). *Anaerobes: General Characteristics*. [online] Nih.gov. Available at: https://www.ncbi.nlm.nih.gov/books/NBK7638/ [Accessed 31 Mar. 2026].

Hmdb.ca. (2018). *Human Metabolome Database*. [online] Available at: https://www.hmdb.ca/ [Accessed 6 Apr. 2026].

Ho, C., Lam, C., Chan, M., Cheung, R., Law, L., Lit, L., Ng, K., Suen, M. and Tai, H. (2003). Electrospray Ionisation Mass Spectrometry: Principles and Clinical Applications. *The Clinical Biochemist Reviews*, [online] 24(1), p.3. Available at: https://pmc.ncbi.nlm.nih.gov/articles/PMC1853331/ [Accessed 31 Mar. 2026].

Houck, M.M. ed., (2023). *Encyclopedia of Forensic Sciences*. Third Edition ed. Elsevier.

Iancu, L., E Dean, D. and Purcarea, C. (2018). *Temperature Influence on Prevailing Necrophagous Diptera and Bacterial Taxa With Forensic Implications for Postmortem Interval Estimation: A Review*. [online] proquest.com. Available at: https://www.proquest.com/docview/2363966155?pq-origsite=primo&sourcetype=Scholarly%20Journals [Accessed 15 Jan. 2026].

Ibrahim, S.M., Salih, A.M., Ibrahim, Z.A., Mohammed, D.A. and Ibrahim, H.K. (2025). Comparison of traditional and modern postmortem interval estimation techniques in forensic investigations. *Iraqi Journal of Bioscience and Biomedical*, [online] 2(1), pp.117–126. Available at: https://www.researchgate.net/publication/394437344_Comparison_of_traditional_and_modern_postmortem_interval_estimation_techniques_in_forensic_investigations.

Iqbal, M.A., Ueland, M. and Forbes, S.L. (2018). Recent advances in the estimation of post-mortem interval in forensic taphonomy. *Australian Journal of Forensic Sciences*, 52(1), pp.1–17. doi:https://doi.org/10.1080/00450618.2018.1459840.

J.O. Ikpa, Umana, U.E., J.A. Timbuak, C.O. Obun, E.J. Ema and M.E. Omuh (2024). The concept of forensic taphonomy: understanding the postmortem processes of dead remains. *Journal of Experimental and Clinical Anatomy*, 21(2), pp.409–417. doi:https://doi.org/10.4314/jeca.v21i2.36.

James, G., Witten, D., Hastie, T. and Tibshirani, R. (2013). *An Introduction to Statistical Learning : with Applications in R*. Springer.

Jangid, C., Dalal, J. and Malik, K.K. (2025). Estimation of postmortem submersion interval using total aquatic decomposition scores of human cadavers from Punjab. *Journal of Forensic Sciences*. doi:https://doi.org/10.1111/1556-4029.70040.

Javan, G.T., Singh, K., Finley, S.J., Green, R.L. and Sen, C.K. (2024). Complexity of human death: its physiological, transcriptomic, and microbiological implications. *Frontiers in Microbiology*, 14. doi:https://doi.org/10.3389/fmicb.2023.1345633.

Jin, Z., Kang, J. and Yu, T. (2017). Missing value imputation for LC-MS metabolomics data by incorporating metabolic network and adduct ion relations. *Bioinformatics*, 34(9), pp.1555–1561. doi:https://doi.org/10.1093/bioinformatics/btx816.

Jolliffe, I.T. and Cadima, J. (2016). Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, [online] 374(2065), p.20150202. doi:https://doi.org/10.1098/rsta.2015.0202.

Jones, A. (2024). *Best Fit Lines & Curves*. Routledge.

Julián, C., Villadangos, S., Jené, L., Pasques, O., Pintó-Marijuan, M. and Munné-Bosch, S. (2024). Biological outliers: essential elements to understand the causes and consequences of reductions in maximum photochemical efficiency of PSII in plants. *Planta*, [online] 260(1), p.32. doi:https://doi.org/10.1007/s00425-024-04466-3.

Kang, H. (2013). The Prevention and Handling of the Missing Data. *Korean Journal of Anesthesiology*, [online] 64(5), pp.402–406. doi:https://doi.org/10.4097/kjae.2013.64.5.402.

Khan, Y., Shah, S.F. and Asim, S.M. (2024). A novel ranked *k* -nearest neighbors algorithm for missing data imputation. *Journal of Applied Statistics*, pp.1–25. doi:https://doi.org/10.1080/02664763.2024.2414357.

Khoshvaght, H., Permala, R.R., Razmjou, A. and Khiadani, M. (2025). A critical review on selecting performance evaluation metrics for supervised machine learning models in wastewater quality prediction. *Journal of environmental chemical engineering*, 13(6), pp.119675–119675. doi:https://doi.org/10.1016/j.jece.2025.119675.

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. *BMC Bioinformatics*, 20(1). doi:https://doi.org/10.1186/s12859-019-3110-0.

Korfmacher, W.A. (2005). Foundation review: Principles and applications of LC-MS in new drug discovery. *Drug Discovery Today*, 10(20), pp.1357–1367. doi:https://doi.org/10.1016/s1359-6446(05)03620-2.

Kõrgesaar, K., Jordana, X., Gallego, G., Defez, J. and Galtés, I. (2022). Taphonomic model of decomposition. *Legal Medicine*, 56, p.102031. doi:https://doi.org/10.1016/j.legalmed.2022.102031.

Kowarik, A. and Templ, M. (2016). Imputation with the R Package VIM. *Journal of Statistical Software*, 74(7). doi:https://doi.org/10.18637/jss.v074.i07.

Kuhn, M. (2008). Building predictive models in R using the caret package. *Journal of Statistical Software*, [online] 28(5). doi:https://doi.org/10.18637/jss.v028.i05.

Lanzinger, N., Verhoff, M.A., Birngruber, C.G. and Lutz, L. (2026). Factors influencing the progression of post-mortem changes between scene and autopsy. *Scientific Reports*. [online] doi:https://doi.org/10.1038/s41598-026-35786-x.

Lazar, C. and Burger, T. (2022). A Collection of Methods for Left-Censored Missing Data Imputation [R package imputeLCMD version 2.1]. *R-project.org*. [online] doi:https://cran.r-project.org/package=imputeLCMD.

Lendoiro, E., Cordeiro, C., Rodríguez-Calvo, M.S., Vieira, D.N., Suárez-Peñaranda, J.M., López-Rivadulla, M. and Muñoz-Barús, J.I. (2012). Applications of Tandem Mass Spectrometry (LC–MSMS) in estimating the post-mortem interval using the biochemistry of the vitreous humour. *Forensic Science International*, 223(1-3), pp.160–164. doi:https://doi.org/10.1016/j.forsciint.2012.08.022.

Lenz, M., Müller, F.-J., Zenke, M. and Schuppert, A. (2016). Principal components analysis and the reported low intrinsic dimensionality of gene expression microarray data. *Scientific Reports*, [online] 6(1), p.25696. doi:https://doi.org/10.1038/srep25696.

Li, J., Wu, X.-J., Liu, C.-X. and Yuan, Y.-J. (2006). Application of new method for data processing in metabonomic studies. *Yao xue xue bao = Acta pharmaceutica Sinica*, [online] 41(1), pp.47–53. Available at: https://pubmed.ncbi.nlm.nih.gov/16683527/ [Accessed 19 Mar. 2026].

Li, Y., Zhang, Y., Yu, M. and Li, X. (2018). Drawing and studying on histogram. *Cluster Computing*, 22(S2), pp.3999–4006. doi:https://doi.org/10.1007/s10586-018-2606-0.

Little, R.J. (2021). Missing Data Assumptions. *Annual Review of Statistics and Its Application*, 8(1), pp.89–107. doi:https://doi.org/10.1146/annurev-statistics-040720-031104.

Little, R.J.A. and Rubin, D.B. (2019). *Statistical Analysis with Missing Data*. 3rd ed. Hoboken, Nj: John Wiley & Sons, Inc.

Liu, J.-J. (2022). Preparation of Sample for Mass Spectrometry. *Indian Journal of Clinical Biochemistry*, 34(S1). doi:https://go-gale-com.ezproxy.staffs.ac.uk/ps/i.do?p=AONE&u=staffordshire&id=GALE%7CA739813734&v=2.1&it=r.

Locci, E., Stocchero, M., Noto, A., Chighine, A., Natali, L., Napoli, P.E., Caria, R., De-Giorgio, F., Nioi, M. and d'Aloja, E. (2019). A 1H NMR metabolomic approach for the estimation of the time since death using aqueous humour: an animal model. *Metabolomics*, 15(5). doi:https://doi.org/10.1007/s11306-019-1533-2.

Luan, J., Zhang, C., Xu, B., Xue, Y. and Ren, Y. (2020). The predictive performances of random forest models with limited sample size and different species traits. *Fisheries Research*, 227, p.105534. doi:https://doi.org/10.1016/j.fishres.2020.105534.

M Løber, I.M., Hedemann, M.S., Villesen, P. and Nielsen, K.L. (2025). Untangling the Postmortem Metabolome: A Machine Learning Approach for Accurate PMI Estimation. *Analytical chemistry*, [online] 97(30), pp.16123–16132. doi:https://doi.org/10.1021/acs.analchem.4c05796.

Magni, P.A., Lawn, J. and Guareschi, E.E. (2021). A practical review of adipocere: Key findings, case studies and operational considerations from crime scene to autopsy. *Journal of Forensic and Legal Medicine*, [online] 78, p.102109. doi:https://doi.org/10.1016/j.jflm.2020.102109.

Magnusson, R., Söderberg, C., Ward, L.J., Arpe, J., Kugelberg, F.C., Elmsjö, A., Green, H. and Nyman, E. (2026). The human metabolome and machine learning improves predictions of the post-mortem interval. *Nature Communications*, [online] 17(1). doi:https://doi.org/10.1038/s41467-026-69158-w.

Marco-Ramell, A., Palau-Rodriguez, M., Alay, A., Tulipani, S., Urpi-Sarda, M., Sanchez-Pla, A. and Andres-Lacueva, C. (2018). Evaluation and comparison of bioinformatic tools for the enrichment analysis of metabolomics data. *BMC Bioinformatics*, 19(1). doi:https://doi.org/10.1186/s12859-017-2006-0.

Martínez, A., Larrañaga, A., Pérez, J., Descals, E. and Pozo, J. (2013). Temperature affects leaf litter decomposition in low-order forest streams: field and microcosm approaches. *FEMS Microbiology Ecology*, 87(1), pp.257–267. doi:https://doi.org/10.1111/1574-6941.12221.

Martlin, B.A., Anderson, G.S. and Bell, L.S. (2022). A Review of Human Decomposition in Marine Environments. *Canadian Society of Forensic Science Journal*, 56(2), pp.1–30. doi:https://doi.org/10.1080/00085030.2022.2135741.

Mason, A.R., McKee-Zech, H.S., Hoeland, K.M., Davis, M.C., Campagna, S.R., Steadman, D.W. and DeBruyn, J.M. (2022). Body Mass Index (BMI) Impacts Soil Chemical and Microbial Response to Human Decomposition. *mSphere*, [online] 0(0), pp.e00325-22. doi:https://doi.org/10.1128/msphere.00325-22.

Mayonu, M., Saeedeh Babaee, Jiang, L., Wilson, D. and Wang, B. (2025). Evaluation of a New Approach for Principal Component Analysis Application in Metabolomics Studies. *Analytical Letters*, pp.1–14. doi:https://doi.org/10.1080/00032719.2025.2513548.

McCalley, D.V. (2023). Understanding and managing peak shape for basic solutes in reversed-phase high performance liquid chromatography. *Chemical Communications*, [online] (Issue 51). Available at: https://pubs-rsc-org.ezproxy.staffs.ac.uk/en/content/articlelanding/2023/cc/d3cc01535a [Accessed 23 Feb. 2026].

Mccorry, L.K., Gonnella, C.Y. and Zdanowicz, M.M. (2019). *Essentials of human physiology and pathophysiology for pharmacy and allied health*. New York, Ny: Routledge.

Morgan, M. and Ramos, M. (2025). *Access the Bioconductor Project Package Repository*. [online] bioconductor.github.io. Available at: https://bioconductor.github.io/BiocManager/ [Accessed 16 Apr. 2026].

Morris, A.S. and Langari, R. (2016). *Measurement and instrumentation : theory and application*. 2nd ed. Amsterdam: Elsevier Academic Press.

Nam, S.L., de la Mata, A.P., Dias, R.P. and Harynuk, J.J. (2020). Towards Standardization of Data Normalization Strategies to Improve Urinary Metabolomics Studies by GC×GC-TOFMS. *Metabolites*, 10(9), p.376. doi:https://doi.org/10.3390/metabo10090376.

Nguyen, V.Q., Huang, M. and Simoff, S. (2020). Enhancing Scatter-plots with Start-plots for Visualising Multi-dimensional Data. *2020 24th International Conference Information Visualisation (IV)*. [online] doi:https://doi.org/10.1109/IV51561.2020.00023.

Nyamundanda, G., Brennan, L. and Gormley, I. (2010). Probabilistic principal component analysis for metabolomic data. *BMC Bioinformatics*, 11(1), p.571. doi:https://doi.org/10.1186/1471-2105-11-571.

Osamura, T., Takahashi, F., Endo, K., Okuda, M. and Takimura, Y. (2023). Autolysis-induced extracellular production of intracellular carboxylesterase EstGtA2 using multiple-protease-

deficient Bacillus subtilis strains. *Biochemical Engineering Journal*, 198, p.108996. doi:https://doi.org/10.1016/j.bej.2023.108996.

Oza, V., Aicher, J. and Reed, L. (2018). Random Forest Analysis of Untargeted Metabolomics Data Suggests Increased Use of Omega Fatty Acid Oxidation Pathway in Drosophila Melanogaster Larvae Fed a Medium Chain Fatty Acid Rich High-Fat Diet. *Metabolites*, 9(1), p.5. doi:https://doi.org/10.3390/metabo9010005.

Palmer, C. (2020). Estimating the Impact of Laminar Flow on the Pattern and Rate of Decomposition in Aquatic Environments—Is There a Better Way of Modeling Decomposition? *Journal of Forensic Sciences*, 65(5), pp.1601–1609. doi:https://doi.org/10.1111/1556-4029.14441.

Pápai, Z. and Pap, T.L. (2002). Analysis of peak asymmetry in chromatography. *Journal of Chromatography. A*, [online] 953(1-2), pp.31–38. doi:https://doi.org/10.1016/s0021-9673(02)00121-8.

Pokines, J. and Symes, S.A. eds., (2013). *Manual of Forensic Taphonomy*. CRC Press. doi:https://doi.org/10.1201/b15424.

quarto.org. (n.d.). *Quarto*. [online] Available at: https://quarto.org/ [Accessed 17 Feb. 2026].

Rainer, J. (2022). *Core Utils for Mass Spectrometry Data*. [online] Github.io. Available at: https://rformassspectrometry.github.io/MsCoreUtils/ [Accessed 16 Apr. 2026].

Ralebitso-Senior, K. (2018). *Forensic ecogenomics : the application of microbial ecology analyses in forensic contexts*. London, United Kingdom ; San Diego, Ca: Academic Press.

Rattenbury, A. (2018). *Rigor Mortis - an overview | ScienceDirect Topics*. [online] www.sciencedirect.com. Available at: https://www.sciencedirect.com/topics/medicine-and-dentistry/rigor-mortis [Accessed 23 Feb. 2026].

Razifar, P., Muhammed, H.H., Engbrant, F., Svensson, P.-E., Olsson, J., Bengtsson, E., Långström, B. and Bergström, M. (2009). Performance of Principal Component Analysis and Independent Component Analysis with Respect to Signal Extraction from Noisy Positron Emission Tomography Data - a Study on Computer Simulated Images. *The Open Neuroimaging Journal*, 3, pp.1–16. doi:https://doi.org/10.2174/1874440000903010001.

Salman, H.A., Kalakech, A. and Steiti, A. (2024). Random Forest Algorithm Overview. *Babylonian Journal of Machine Learning*, [online] 2024, pp.69–79. doi:https://doi.org/10.58496/bjml/2024/007.

Saukko, B. (2023). *Knight's Forensic Pathology.* S.L.: Crc Press.

Sawikowska, A., Piasecka, A., Kachlicki, P. and Krajewski, P. (2021). Separation of Chromatographic Co-Eluted Compounds by Clustering and by Functional Data Analysis. *Metabolites*, [online] 11(4), p.214. doi:https://doi.org/10.3390/metabo11040214.

Schmitt, P., Mandel, J. and Guedj, M. (2015). *A Comparison of Six Methods for Missing Data Imputation*. [online] Journal of Biometrics & Biostatistics . Available at: https://www.hilarispublisher.com/open-access/a-comparison-of-six-methods-for-missing-data-imputation-2155-6180-1000224.pdf.

Schotsmans, E.M., Márquez-Grant, N. and Forbes, S. (2017). *Taphonomy of Human Remains: Forensic Analysis of the Dead and the Depositional Environment*. Chichester, West Sussex: John Wiley & Sons.

Shedge, R., Krishan, K., Warrier, V. and Kanchan, T. (2023). *Postmortem Changes*. [online] PubMed. Available at: https://www.ncbi.nlm.nih.gov/books/NBK539741/.

Shi, Y., Xiang, R., Horváth, C. and Wilkins, J.A. (2004). The role of liquid chromatography in proteomics. *Journal of Chromatography A*, [online] 1053(1), pp.27–36. doi:https://doi.org/10.1016/j.chroma.2004.07.044.

Shrestha, R., Tanuj Kanchan and Krishan, K. (2023). *Methods of Estimation of Time Since Death*. [online] Nih.gov. Available at: https://www.ncbi.nlm.nih.gov/books/NBK549867/?utm_source=chatgpt.com.

Siva, T., Nur Airie Zainudin, Nurul Shahida Redzuan, Edlic Sathiamurthy, Mohd, B., Mohd. and Ismail, S.S. (2024). Effect of different water salinity ecosystems on the decompositions of partially submerged buried cadavers in tropical regions. *Australian Journal of Forensic Sciences*, pp.1–17. doi:https://doi.org/10.1080/00450618.2024.2434488.

Slowikowski, K., Schep, A., Hughes, S., Dang, T.K., Lukauskas, S., Irisson, J.-O., Kamvar, Z.N., Ryan, T., Christophe, D., Hiroaki, Y., Gramme, P., Abdol, A.M., Barrett, M., Cannoodt, R., Krassowski, M., Chirico, M. and Aphalo, P. (2023). *ggrepel: Automatically Position Non-*

Overlapping Text Labels with 'ggplot2'. [online] R-Packages. Available at: https://cran.r-project.org/web/packages/ggrepel/index.html [Accessed 16 Apr. 2026].

Sivakumar, M., Parthasarathy, S. and Thiyagarajan Padmapriya (2024). Trade-off between training and testing ratio in machine learning for medical image processing. *PeerJ Computer Science*, [online] 10, pp.e2245–e2245. doi:https://doi.org/10.7717/peerj-cs.2245.

Stanimirova, I., Daszykowski, M. and Walczak, B. (2007). Dealing with missing values and outliers in principal component analysis. *Talanta*, 72(1), pp.172–178. doi:https://doi.org/10.1016/j.talanta.2006.10.011.

Stekhoven, D.J. and Buhlmann, P. (2012). *Missforest-Non-parametric missing value imputation for mixed-type data*. [online] Scopus.com. Available at: https://www.scopus.com/pages/publications/84855177476?inward= [Accessed 13 Nov. 2025].

Stitt, M., Luca Borghi, G. and Arrivault, S. (2021). Targeted metabolite profiling as a top-down approach to uncover interspecies diversity and identify key conserved operational features in the Calvin–Benson cycle. *Journal of Experimental Botany*, [online] 72(17), pp.5961–5986. doi:https://doi.org/10.1093/jxb/erab291.

Strete, G., Sălcudean, A., Cozma, A.-A. and Radu, C.-C. (2025). Current Understanding and Future Research Direction for Estimating the Postmortem Interval: A Systematic Review. *Diagnostics*, 15(15), p.1954. doi:https://doi.org/10.3390/diagnostics15151954.

Tibbett, M. and Carter, D.O. (2008). *Soil Analysis in Forensic Taphonomy*. CRC Press.

Tierney, Nicholas, and Dianne Cook. 2023. "Expanding Tidy Data Principles to Facilitate Missing Data Exploration, Visualization and Assessment of Imputations." *Journal of Statistical Software* 105 (7): 1–31. https://doi.org/10.18637/jss.v105.i07.

Trautner, T. and Bruckner, S. (2021). Line Weaver: Importance-Driven Order Enhanced Rendering of Dense Line Charts. *Computer Graphics Forum*, 40(3), pp.399–410. doi:https://doi.org/10.1111/cgf.14316.

Truong, D. (2026). *Data Science and Machine Learning for Non-Programmers: Principal Component Analysis*. CRC Press.

V. Akila, A. Vasuki, J.Anita Christaline, R. Sathiya, Rishi, P. and Edward, A.Shirly. (2023). Enhancing Software Testing with Machine Learning Techniques. doi:https://doi.org/10.1109/icscds56580.2023.10105028.

Walach, J., Filzmoser, P., Štěpán Kouřil, Friedecký, D. and Adam, T. (2019). Cellwise outlier detection and biomarker identification in metabolomics based on pairwise log ratios. *Journal of Chemometrics*, 34(1). doi:https://doi.org/10.1002/cem.3182.

Walsh, M.C., Brennan, L., Malthouse, J.P.G., Roche, H.M. and Gibney, M.J. (2006). Effect of acute dietary standardization on the urinary, plasma, and salivary metabolomic profiles of healthy humans. *The American Journal of Clinical Nutrition*, 84(3), pp.531–539. doi:https://doi.org/10.1093/ajcn/84.3.531.

Wang, S., Ruan, H. and Han, Y. (2010). Effects of microclimate, litter type, and mesh size on leaf litter decomposition along an elevation gradient in the Wuyi Mountains, China. *Ecological Research*, 25(6), pp.1113–1120. doi:https://doi.org/10.1007/s11284-010-0736-9.

Wang, Z., Zhang, F., Wang, L., Yuan, H., Guan, D. and Zhao, R. (2022). Advances in artificial intelligence-based microbiome for PMI estimation. *Frontiers in Microbiology*, 13. doi:https://doi.org/10.3389/fmicb.2022.1034051.

Wanishsakpong, W. and Notodiputro, K.A. (2017). Locally weighted scatter-plot smoothing for analysing temperature changes and patterns in Australia. *Meteorological Applications*, 25(3), pp.357–364. doi:https://doi.org/10.1002/met.1702.

Wei, R., Wang, J., Jia, E., Chen, T., Ni, Y. and Jia, W. (2018a). GSimp: A Gibbs sampler based left-censored missing value imputation approach for metabolomics studies. *PLOS Computational Biology*, 14(1), p.e1005973. doi:https://doi.org/10.1371/journal.pcbi.1005973.

Wei, R., Wang, J., Su, M., Jia, E., Chen, S., Chen, T. and Ni, Y. (2018b). Missing Value Imputation Approach for Mass Spectrometry-based Metabolomics Data. *Scientific Reports*, 8(1). doi:https://doi.org/10.1038/s41598-017-19120-0.

Weisensee, K.E. and Atwell, M.M. (2024). Human Decomposition and Time Since Death: Persistent Challenges and Future Directions of Postmortem Interval Estimation in Forensic Anthropology. *American Journal of Physical Anthropology*, 186(S78). doi:https://doi.org/10.1002/ajpa.70011.

Wells, J.D. (2018). A Forensic Entomological Analysis Can Yield an Estimate of Postmortem Interval, and Not Just a Minimum Postmortem Interval: An Explanation and Illustration Using a Case. *Journal of Forensic Sciences*, 64(2), pp.634–637. doi:https://doi.org/10.1111/1556-4029.13912.

Werth, M.T., Halouska, S., Shortridge, M.D., Zhang, B. and Powers, R. (2010). Analysis of metabolomic PCA data using tree diagrams. *Analytical Biochemistry*, [online] 399(1), pp.58–63. doi:https://doi.org/10.1016/j.ab.2009.12.022.

Wickham, H., François, R., Henry, L., Müller, K. and Vaughan, D. (2019). *Dplyr: a Grammar of Data Manipulation*. [online] Tidyverse.org. Available at: https://dplyr.tidyverse.org/ [Accessed 7 Apr. 2026].

Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org.

Wickham, Hadley, Davis Vaughan, and Maximilian Girlich. 2024. *Tidyr: Tidy Messy Data*. https://doi.org/10.32614/CRAN.package.tidyr.

Wilm, M. (2011). Principles of Electrospray Ionization. *Molecular & Cellular Proteomics*, 10(7), p.M111.009407. doi:https://doi.org/10.1074/mcp.m111.009407.

Wilson, M., Ponzini, M.D., Taylor, S.L. and Kim, K. (2022). Imputation of Missing Values for Multi-Biospecimen Metabolomics Studies: Bias and Effects on Statistical Validity. *Metabolites*, 12(7), pp.671–671. doi:https://doi.org/10.3390/metabo12070671.

World Health Organization (2024). *Drowning*. [online] Who.int. Available at: https://www.who.int/news-room/fact-sheets/detail/drowning.

Worsfold, P., Townshend , A. and Poole, C. eds., (2005). *Encyclopedia of analytical science* . Second Edition ed. Amsterdam ; Heidelberg: Elsevier Academic Press.

Wright, M.N. and Ziegler, A. (2017). ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R. *Journal of Statistical Software*, 77(1). doi:https://doi.org/10.18637/jss.v077.i01.

Y, K. and Isukapatla, A.R. (2025). Postmortem microbiome dynamics: Review of forensic microbial clock. *Journal of Forensic and Legal Medicine*, [online] 117, p.103024. doi:https://doi.org/10.1016/j.jflm.2025.103024.

Yao, F., Coquery, J. and Lê Cao, K.-A. (2012). Independent Principal Component Analysis for biologically meaningful dimension reduction of large biological data sets. *BMC Bioinformatics*, 13(1), p.24. doi:https://doi.org/10.1186/1471-2105-13-24.

Yu, T., Zhu, J., Li, D. and Zhu, D. (2021). Physical and chemical mechanisms of tissue optical clearing. *iScience*, [online] 24(3), p.102178. doi:https://doi.org/10.1016/j.isci.2021.102178.

Zhang, F.-Y., Wang, L.-L., Dong, W.-W., Zhang, M., Tash, D., Li, X.-J., Du, S.-K., Yuan, H.-M., Zhao, R. and Guan, D.-W. (2022). A preliminary study on early postmortem submersion interval (PMSI) estimation and cause-of-death discrimination based on nontargeted metabolomics and machine learning algorithms. *International Journal of Legal Medicine*, 136(3), pp.941–954. doi:https://doi.org/10.1007/s00414-022-02783-4.

Zhang, F.-Y., Wang, L.-L., Zeng, K., Dong, W.-W., Yuan, H.-Y., Ma, X.-Y., Wang, Z.-W., Zhao, Y., Zhao, R. and Guan, D.-W. (2024a). A fundamental study on postmortem submersion interval estimation by metabolomics analyzing of gastrocnemius muscle from submersed rat models in freshwater. *International Journal of Legal Medicine*, 138(5), pp.2037–2047. doi:https://doi.org/10.1007/s00414-024-03258-4.

Zhang, H. and Henion, J. (2001). Comparison between liquid chromatography–time-of-flight mass spectrometry and selected reaction monitoring liquid chromatography–mass spectrometry for quantitative determination of idoxifene in human plasma. *Journal of Chromatography B: Biomedical Sciences and Applications*, 757(1), pp.151–159. doi:https://doi.org/10.1016/s0378-4347(01)00132-3.

Zhang, N., Casasent, T.D., Casasent, A.K., Kumar, S.V., Wakefield, C., Broom, B.M., Weinstein, J.N. and Akbani, R. (2024b). PCA-Plus: Enhanced principal component analysis with illustrative applications to batch effects and their quantitation. *BioRxiv*, [online] p.2024.01.02.573793. doi:https://doi.org/10.1101/2024.01.02.573793.

Zhao, C., Su, K.-J., Wu, C., Cao, X., Sha, Q., Li, W., Luo, Z., Qing, T., Qiu, C., Zhao, L.J., Liu, A., Jiang, L., Zhang, X., Shen, H., Zhou, W. and Deng, H.-W. (2024a). Multi-scale variational autoencoder for imputation of missing values in untargeted metabolomics using whole-genome sequencing data. *Computers in Biology and Medicine*, [online] 179, p.108813. doi:https://doi.org/10.1016/j.compbiomed.2024.108813.

Zhao, X., Yang, F., Yang, F., Nie, H., Hu, S., Gui, P., Guo, Y. and Zhang, C. (2024b). Seasonal mouse cadaver microbial study: rupture time and postmortem interval estimation model construction. *PeerJ*, [online] 12, pp.e17932–e17932. doi:https://doi.org/10.7717/peerj.17932.

12.0 **Appendices:**

Appendix A – R code for data preparation and reconstruction (section 7.3)

```
library(readxl)

Data_detail <- read_excel("Raw data/Profinder Export complete.csv.xlsx")
Data_simple <- read.csv("Raw data/Profinder simple export.csv")

# Assuming both data frames have the same number of rows and align by row
subset_Data_simple <- Data_simple[Data_detail$'Keep (Y or N)' == "y", ]

# a)    Combine mass and RT columns to produce a ‘compound_ID’ column to be used a
s a unique compound identifier.
subset_Data_simple$ID <- paste(subset_Data_simple$Mass, subset_Data_simple$RT, sep
= "@")
# Having column names start with numbers can cause problems so add a prefix
subset_Data_simple$ID <- paste("ID", subset_Data_simple$ID, sep = ":")

# Then delete the mass and RT columns.
subset_Data_simple <- subset_Data_simple[,-(c(1,2,3))]

# b)    Transpose the data frame so the compound_ID’s are now the new column names
#   and the file names are the row names.
#install.packages("data.table")
library(data.table)
#citation("data.table")
subset_Data_simple <- transpose(subset_Data_simple, keep.names = "file_name", make
.names = "ID")
```

```
# c)    Using the file names, extract into new columns the sample date and mouse I
D
# using separate() from the tidyr package.
#install.packages("tidyr")
library("tidyr")
citation("tidyr")

subset_Data_simple <- separate(subset_Data_simple, "file_name", into = c(NA, "mous
e", "sample_date"), sep = "_")
```

```
# d)    Make a new column for the PMSI in days, either by calculating from the dat
e
#  or, having already worked out the days from part 1, use an IF function.
#install.packages("lubridate")
library("lubridate")

citation("lubridate")

subset_Data_simple$sample_date <- as.numeric(subset_Data_simple$sample_date)
# convert to numeric

subset_Data_simple$sample_date <- dmy(subset_Data_simple$sample_date)
# convert to date format
```

```
# show difference from day 0 which is 24/11/21
subset_Data_simple$PMSI <- as.numeric((interval("2021-11-24", subset_Data_simple$s
ample_date)) / ddays(1))
```

Appendix B – R code for missing data exploration / Figure 15 summary table (section 7.4)

```
library(dplyr)

#citation("dplyr")
# Compute row-wise missing counts
df <- subset_Data_simple %>%
  mutate(n_miss_row_all = n_miss_row(subset_Data_simple))

# Total number of columns in the original data frame
n_cols_all <- ncol(subset_Data_simple)

# Summarise by PMSI /summarise by the days that samples were collected
summary_by_PMSI <- df %>%
  group_by(PMSI) %>%
  summarise(
    rows = n(),
    total_missing = sum(n_miss_row_all),
    mean_missing_per_row = round((mean(n_miss_row_all)),2),
    pct_missing = round(((total_missing / (rows * n_cols_all)) * 100),2),  # % mis
sing across all cells for this PMSI
    .groups = "drop"
  )

# Build flextable for Word/ so that it can look presentable in word
ft_summary <- summary_by_PMSI %>%
  flextable() %>%
  set_header_labels(
    PMSI = "PMSI",
    rows = "Number of Rows",
    total_missing = "Total Missing Values",
    mean_missing_per_row = "Mean Missing per Row",
    pct_missing = "% Missing"
  ) %>%
  colformat_num(j = c("total_missing", "mean_missing_per_row"), digits = 2) %>%
  colformat_num(j = "pct_missing", digits = 2, suffix = "%") %>%
  autofit() %>%
  theme_booktabs() %>%
  bold(part = "header") %>%
  align(align = "center", part = "all")

ft_summary
```

Appendix C - R code for missing data exploration / Figure 16 heat map (section 7.4)

```r
# Select measured columns (3:2002)
measured_df <- subset_Data_simple[, 3:180]

# Add mouse and PMSI for grouping
gp_vars <- subset_Data_simple %>%
  select(mouse, PMSI)

# Compute missing counts for each combination of mouse and PMSI
missing_summary <- gp_vars %>%
  mutate(n_missing = n_miss_row(measured_df)) %>%
  group_by(mouse, PMSI) %>%
  summarise(total_missing = sum(n_missing), .groups = "drop")

# Convert to factors for categorical axes
missing_summary <- missing_summary %>%
  mutate(
    mouse = as.factor(mouse),
    PMSI = as.factor(PMSI)
  )

library(ggplot2)
# Create heat map
ggplot(missing_summary, aes(x = PMSI, y = mouse, fill = total_missing)) +
  geom_tile(color = "white") +
  scale_fill_gradient(low = "lightblue", high = "darkred") +
  labs(
    title = "Heat Map of Missing Data Counts",
    x = "PMSI Category",
    y = "Mouse Category",
    fill = "Missing Count"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Appendix D - R code for missing data exploration / Figure 17 histogram with 25% threshold (section 7.4)

```
#This code will tell me which columns remian after filtering out those wit
h >25%NAs
#How much missing data is still present in the remaining columns
#If the histogram shows most values near 0, then that means my data is nea
rly compelte
threshold <- 0.25
test_df <- subset_Data_simple[, colMeans(is.na(subset_Data_simple)) <=thre
shold]
test_df_hist <- test_df[3:(length(test_df)-1)]
cat("columns with less than", threshold, "% NA values", (colnames(test_df_
hist)))
```

```
columns with less than 0.25 % NA values ID:282.259@1.25 ID:129.0434@1.81 I
D:116.0839@1.47 ID:157.9999@3.73 ID:257.0712@1.78 ID:928.489@3.79 ID:540.1
128@1.28 ID:508.1185@1.3 ID:116.0832@1.47 ID:282.2605@1.25 ID:353.958@3.74
ID:182.045@1.65 ID:803.5194@1.98 ID:964.705@3.73 ID:964.7176@3.74 ID:129.0
331@1.81 ID:803.5381@1.97 ID:634.0511@3.94 ID:157.999@3.73 ID:427.8913@3.7
4 ID:631.9104@3.8 ID:721.8308@3.73 ID:732.0365@3.92
```

```
hist(colMeans(is.na(test_df_hist)))
```

```
missing_prop <- colMeans(is.na(test_df_hist))

library(ggplot2)

missing_df <- data.frame(Missing_Proportion = missing_prop)

ggplot(missing_df, aes(x = Missing_Proportion)) +
  geom_histogram(binwidth = 0.02, fill = "steelblue", color = "white") +
  geom_vline(xintercept = 0.25, linetype = "dashed", color = "red", size =
1) +
  labs(
    title = "Distribution of Missing Data Proportions",
    subtitle = "Variables retained after removing >25% missingness",
    x = "Proportion of Missing Values per Variable",
    y = "Number of Variables"
  ) +
  theme_minimal(base_size = 14)
```

Appendix E – Rubins framework for imputation decision (section 7.4)

Types of Missing data

Identifying the nature of the missing data is essential in selecting the appropriate data imputation method. Following 'Rubin's Framework', missing data when using LC-MS/MS can be classified into three mechanisms:

1. **Missing Completely at Random (MCAR)**

   Missing values are completely unrelated to any observed or unobserved data. The missingness mostly occurs randomly due to instrument failure. This data is typically removed so it doesn't bias the sample.

2. **Missing at Random (MAR)**

   The probability of missingness is related to only observed variables, not the missing values themselves. For example, certain measurements may be missing more frequently for older samples, but this pattern can be observed from the data.

3. **Missing Not at Random (MNAR)**

   The missingness depends on the unobserved values themselves. This can occur in samples with extreme or unusual characteristics that are more likely to be unrecorded.

Appendix F – Assessment and Visualisation of missing data patterns / Figure 18 scatter plots for M1-M6 (Section 7.5)

```
#Attempt SUM with last code here.
# ========================
# Full workflow (SUM score per mouse over PMSI days)
# ========================

library(dplyr)
#citation("dplyr")
library(ggplot2)
#citation("ggplot2")

# 1) Normalize each compound (ID:*) within EACH mouse (min–max 0–1),
#    then compute ONE summary value per row: the SUM across normalized com
pounds.
normalized_with_sum <- df %>%
  group_by(mouse) %>%
  mutate(
    across(
      starts_with("ID:"),
      ~ {
        x <- .

        # if the whole vector for this mouse+compound is NA, keep NA
        if (all(is.na(x))) return(x)

        mn <- min(x, na.rm = TRUE)
        mx <- max(x, na.rm = TRUE)
        rng <- mx - mn

        # avoid divide-by-zero
        if (is.na(rng) || rng == 0) return(rep(NA_real_, length(x)))

        (x - mn) / rng
      }
    )
  ) %>%
  ungroup() %>%
  rowwise() %>%
  mutate(
    # SUM across ALL normalized compounds for that sample (mouse x PMSI)
    norm_score_sum = sum(c_across(starts_with("ID:")), na.rm = TRUE),

    # convert 0 (from all NA) to NA
    norm_score_sum = ifelse(all(is.na(c_across(starts_with("ID:")))), NA_r
eal_, norm_score_sum)
  ) %>%
  ungroup()

# 2) Keep only the columns needed for plotting
plot_df <- normalized_with_sum %>%
```

```
  select(mouse, PMSI, norm_score_sum) %>%
  arrange(mouse, PMSI)


# 3B) Plot: one panel per mouse
p_facet <- ggplot(plot_df, aes(x = PMSI, y = norm_score_sum)) +
  geom_point() +
  facet_wrap(~mouse) +
  theme_bw() +
  labs(
    x = "PMSI day",
    y = "Sum normalized score (within-mouse, across compounds)",
    title = "Per-mouse summed normalized trajectory (one plot per mouse)"
  )

print(p_facet)
```

Appendix G - Assessment and Visualisation of missing data patterns / Figure 19 all mice in one scatter plot (Section 7.5)

```
# 3A) Plot: all mice in one figure/connected to last code section
p_all <- ggplot(plot_df, aes(x = PMSI, y = norm_score_sum, color = mouse,
group = mouse)) +
  geom_point() +
  theme_bw() +
  labs(
    x = "PMSI day",
    y = "Summed normalized intensities",
    color = "Mouse ID (M1-M6)",
    title = "Summed normalized intensities across PMSI days for all mice"
  )

print(p_all)

# 4) Optional: save outputs
# write.csv(plot_df, "mouse_sum_normalized_scores.csv", row.names = FALSE)
# write.csv(normalized_with_sum, "normalized_compounds_with_sum_score.csv"
, row.names = FALSE)
```

Appendix H - Assessment and Visualisation of missing data patterns / Figure 20 scatter plot with greyed out datapoints below set threshold

```
#Attempting to grey out datapoints that are below a threshold to demontrat
e a trend
#This worked really well and is now able to be uploaded into my diss.

p_all <- ggplot(plot_df, aes(PMSI, norm_score_sum, group = mouse)) +

  geom_point(aes(color = ifelse(
    (norm_score_sum <= 5 & PMSI >= 5 & PMSI <= 40) |
    (mouse == "M2" & PMSI >= 12 & PMSI <= 14),
    "low",
    mouse
  ))) +

  scale_color_manual(
    values = c(
      "low" = "grey70",
      "M1" = "#F8766D",
      "M2" = "#B79F00",
      "M3" = "#00BA38",
      "M4" = "#00BFC4",
      "M5" = "#619CFF",
      "M6" = "#F564E3"
    ),
    breaks = c("M1","M2","M3","M4","M5","M6","low"),
    labels = c(
      "M1",
      "M2",
      "M3",
      "M4",
      "M5",
      "M6",
      "Low signal"
    )
  ) +

  theme_bw() +

  labs(
    x = "PMSI day",
    y = "Summed normalized intensities",
    color = "Mouse ID (M1-M6)",
    title = "Summed normalized intensities across PMSI days for all mice"
  )

p_all
```

Appendix I – Data split and QRILC imputation (Section 7.6)

```
library(dplyr)
#citation("dplyr")
cut_day <- 7   # change this to whatever “half” means for you

df_early <- shortlist_df %>% filter(PMSI <= cut_day)
df_late  <- shortlist_df %>% filter(PMSI >  cut_day)

# This is QRLIC Impuation for df_early data.

# Install once if needed
#install.packages("BiocManager")
#BiocManager::install(c("MSnbase", "imputeLCMD"))
#citation("BiocManager")
#citation("MSnbase")
#citation("imputeLCMD")

library(MsCoreUtils)  # provides impute_matrix


Attaching package: 'MsCoreUtils'

The following object is masked from 'package:dplyr':

    between

The following object is masked from 'package:naniar':

    impute_zero

The following objects are masked from 'package:data.table':

    %between%, between

The following object is masked from 'package:stats':

    smooth

#citation("MsCoreUtils")

meta_cols <- c("mouse", "PMSI")
feat_cols <- setdiff(names(df_early), meta_cols)

# 1) Pull feature block + coerce to numeric safely
feat_df <- df_early[, feat_cols, drop = FALSE]
feat_df[] <- lapply(feat_df, function(v) {
  if (is.factor(v)) v <- as.character(v)
  v[v %in% c("NA","NaN","","  ")] <- NA
  as.numeric(v)
})

X <- as.matrix(feat_df)   # rows = samples, cols = features
```

```r
# 2) (Optional) handle zeros if they mean "below detection"
# If zeros are real values in your data, comment this out.
X[X <= 0] <- NA

# Count missing values before imputation
na_before <- sum(is.na(X))

# 3) Log transform (pick ONE)
X_log <- log2(X)   # common for intensities
# X_log <- log(X)  # natural log (like your earlier code)

# 4) MsCoreUtils expects features in rows for many workflows,
# so transpose: rows = features, cols = samples
X_log_t <- t(X_log)

# 5) QRILC imputation
set.seed(123)
X_log_t_imp <- impute_matrix(X_log_t, method = "QRILC")
```

```
Loading required namespace: imputeLCMD
```

```
Imputing along margin 2 (samples/columns).
```

```r
# 6) Transpose back and inverse-log
X_log_imp <- t(X_log_t_imp)
X_imp <- 2^X_log_imp  # if you used log2
# X_imp <- exp(X_log_imp)  # if you used natural log

# 7) Put back into original data.frame
df_early_imp <- df_early
df_early_imp[, feat_cols] <- X_imp

X_imp <- 2^X_log_imp

# Count missing values after imputation
na_after <- sum(is.na(X_imp))

# Display results
cat("Missing values before imputation:", na_before, "\n")
```

```
Missing values before imputation: 134
```

```r
cat("Missing values after imputation:", na_after, "\n")
```

```
Missing values after imputation: 0
```

```r
data.frame(
  Stage = c("Before imputation", "After imputation"),
  Missing_values = c(na_before, na_after)
)

#Count of before and after imputation
```

```
              Stage Missing_values
1 Before imputation            134
2  After imputation              0
```

Appendix J – KNN Imputation (Section 7.6)

```
#This is KNN imputation for the late dataset

#install.packages("VIM")
library(VIM)

Loading required package: colorspace

Loading required package: grid

VIM is ready to use.

Suggestions and bug-reports can be submitted at: https://github.com/statis
tikat/VIM/issues


Attaching package: 'VIM'

The following object is masked from 'package:datasets':

    sleep

library(dplyr)
#citation("VIM")


# Identify feature columns
id_cols <- grep("^ID:", names(df_late), value = TRUE)

# Count missing values BEFORE imputation
na_before <- sum(is.na(df_late[id_cols]))

# Log transform for better distance calculation
df_late_log <- df_late %>%
  mutate(across(all_of(id_cols), log1p))

# KNN imputation within each mouse
df_late_imp <- df_late_log %>%
  group_by(mouse) %>%
  group_modify(~ kNN(.x,
                     variable = id_cols,
                     dist_var = id_cols,
                     k = 5,
                     imp_var = FALSE)) %>%
  ungroup()

# Convert back to original scale
df_late_imp <- df_late_imp %>%
  mutate(across(all_of(id_cols), expm1))

# Count missing values AFTER imputation
na_after <- sum(is.na(df_late_imp[id_cols]))
```

```r
# Print comparison
cat("Missing values before imputation:", na_before, "\n")
```

```
Missing values before imputation: 85
```

```r
cat("Missing values after imputation:", na_after, "\n")
```

```
Missing values after imputation: 0
```

```r
# Create summary table
data.frame(
  Stage = c("Before imputation", "After imputation"),
  Missing_values = c(na_before, na_after)
)

#Count for missing values before and after
```

```
              Stage Missing_values
1 Before imputation             85
2  After imputation              0
```

Appendix K – Visualisations of imputation using histograms and summary tables / Figures 21, 23, 25 (Section 7.6)

```
# Separate chunk: Before vs After QRILC histograms

library(ggplot2)

before_vals <- as.vector(X_log)
after_vals  <- as.vector(X_log_imp)

before_vals <- before_vals[!is.na(before_vals)]
after_vals  <- after_vals[!is.na(after_vals)]

plot_df <- data.frame(
  value = c(before_vals, after_vals),
  Stage = factor(
    c(rep("Before QRILC imputation", length(before_vals)),
      rep("After QRILC imputation", length(after_vals))),
    levels = c("Before QRILC imputation", "After QRILC imputation")
  )
)

ggplot(plot_df, aes(x = value)) +
  geom_histogram(bins = 40, fill = "#2A9D8F", color = "black") +
  facet_wrap(~Stage, ncol = 1) +
  labs(
    title = "Distribution of log2 intensities before and after QRILC imput
ation",
    x = "Log2 intensity",
    y = "Frequency"
  ) +
  theme_bw() +
  theme(
    plot.title = element_text(hjust = 0.5),
    strip.text = element_text(face = "bold")
  )
```

```
# Separate chunk: Before vs After KNN histograms

library(ggplot2)

# Before imputation = logged data before KNN
before_vals <- as.vector(as.matrix(df_late_log[, id_cols]))

# After imputation = logged data after KNN
after_vals <- as.vector(as.matrix(log1p(df_late_imp[, id_cols])))

# Remove missing values
before_vals <- before_vals[!is.na(before_vals)]
after_vals  <- after_vals[!is.na(after_vals)]
```

```
# Combine into plotting data frame
plot_df <- data.frame(
  value = c(before_vals, after_vals),
  Stage = factor(
    c(rep("Before KNN imputation", length(before_vals)),
      rep("After KNN imputation", length(after_vals))),
    levels = c("Before KNN imputation", "After KNN imputation")
  )
)

# Plot
ggplot(plot_df, aes(x = value)) +
  geom_histogram(bins = 40, fill = "#FF7F7F", color = "black") +
  facet_wrap(~Stage, ncol = 1) +
  labs(
    title = "Distribution of log1p intensities before and after KNN imputa
tion",
    x = "Log1p intensity",
    y = "Frequency"
  ) +
  theme_bw() +
  theme(
    plot.title = element_text(hjust = 0.5),
    strip.text = element_text(face = "bold")
  )
```

```
#combined histogram to show the overall change

library(dplyr)

# Identify all metabolite feature columns
id_cols_all <- union(
  grep("^ID:", names(df_early_imp), value = TRUE),
  grep("^ID:", names(df_late_imp), value = TRUE)
)

# Function to standardize columns
standardize_cols <- function(dat, id_cols_all) {

  missing_cols <- setdiff(id_cols_all, names(dat))

  if(length(missing_cols) > 0){
    dat[missing_cols] <- NA_real_
  }

  dat %>%
    mutate(
      mouse = as.factor(mouse),
      PMSI = as.numeric(as.character(PMSI))
    ) %>%
    select(mouse, PMSI, all_of(id_cols_all))
```

```
}

# Standardize both datasets
df_early_imp_std <- standardize_cols(df_early_imp, id_cols_all) %>%
  mutate(imputation = "QRILC")

df_late_imp_std <- standardize_cols(df_late_imp, id_cols_all) %>%
  mutate(imputation = "KNN")

# Combine rows
df_combined <- bind_rows(df_early_imp_std, df_late_imp_std) %>%
  arrange(mouse, PMSI)

library(ggplot2)
library(tidyr)
library(dplyr)
library(scales)

id_cols <- grep("^ID:", names(df_combined), value = TRUE)

# Long format: before imputation
before_long <- shortlist_df %>%
  pivot_longer(
    cols = all_of(id_cols),
    names_to = "feature",
    values_to = "value"
  ) %>%
  mutate(stage = "Before imputation")

# Long format: after imputation
after_long <- df_combined %>%
  pivot_longer(
    cols = all_of(id_cols),
    names_to = "feature",
    values_to = "value"
  ) %>%
  mutate(stage = "After imputation")

# Combine and clean
combined_long <- bind_rows(before_long, after_long) %>%
  filter(is.finite(value), value > 0) %>%
  mutate(
    stage = factor(stage, levels = c("Before imputation", "After imputatio
n"))
  )

# Counts for facet labels
counts <- combined_long %>%
  group_by(stage) %>%
  summarise(n = n(), .groups = "drop")

facet_labels <- setNames(
  paste0(counts$stage, " (n = ", scales::comma(counts$n), ")"),
  counts$stage
```

```
)

# Plot
ggplot(combined_long, aes(x = value)) +
  geom_histogram(
    bins = 80,
    fill = "#4C78A8",
    color = "black",
    alpha = 0.85,
    linewidth = 0.2
  ) +
  scale_x_log10(
    breaks = scales::trans_breaks("log10", function(x) 10^x),
    labels = scales::label_number()
  ) +
  facet_wrap(
    ~stage,
    ncol = 1,
    strip.position = "top",
    labeller = labeller(stage = facet_labels)
  ) +
  labs(
    title = "Distribution of intensity values before and after imputation"
,
    x = "Intensity (log10 scale)",
    y = "Frequency"
  ) +
  theme_bw(base_size = 13)
```

Appendix L – Imputed data visualisation and LOESS scatter plots / Figure 26 & 27 (Section 7.7)

```
library(dplyr)
library(ggplot2)

id_cols <- grep("^ID:", names(df_combined), value = TRUE)

zscore <- function(x) {
  s <- sd(x, na.rm = TRUE)
  if (is.na(s) || s == 0) return(rep(0, length(x)))
  (x - mean(x, na.rm = TRUE)) / s
}

df_mouse_score <- df_combined %>%
  mutate(PMSI = as.numeric(PMSI)) %>%
  group_by(mouse) %>%
  mutate(across(all_of(id_cols), ~ zscore(log1p(.)))) %>%
  rowwise() %>%
  mutate(sum_z_score = sum(c_across(all_of(id_cols)), na.rm = TRUE)) %>%
  ungroup() %>%
  select(mouse, PMSI, sum_z_score)

ggplot(df_mouse_score, aes(x = PMSI, y = sum_z_score)) +
  geom_point(size = 1.4) +
  geom_smooth(method = "loess", se = FALSE, linewidth = 1.1, color = "#2C7
FB8") +
  facet_wrap(~ mouse, ncol = 3) +
  theme_minimal(base_size = 14) +
  theme(
    panel.spacing = unit(1.2, "lines"),
    panel.border = element_rect(color = "black", fill = NA, linewidth = 0.
8),
    strip.background = element_rect(fill = "grey85", color = "black", line
width = 0.8),
    strip.text = element_text(size = 10),  # no bold
    axis.text = element_text(size = 10),
    axis.title = element_text(size = 11)
  ) +
  labs(
    title = "Trajectory of summed normalized compound intensities by mouse
",
    x = "Postmortem interval (days)",
    y = "Sum normalized score"
  )

`geom_smooth()` using formula = 'y ~ x'
```

Appendix M – Scree plot and Principal Component Analysis / Figures 28 & 29 (Section 7.8)

```
citation("dplyr")

# 1) Feature columns
id_cols <- grep("^ID:", names(df_combined), value = TRUE)

# 2) Log-transform feature data
df_log <- df_combined
df_log[id_cols] <- log1p(df_log[id_cols])

# 3) Create early/late grouping from PMSI
df_log <- df_log %>%
  mutate(
    time_group = ifelse(PMSI <= 7, "Early", "Late")
  )

# 4) Run PCA on the FULL dataset
pca <- prcomp(df_log[id_cols], center = TRUE, scale. = TRUE)
#-----------------------------------------------------------
# Scree plot
# Variance explained
eigenvalues <- pca$sdev^2

scree_df <- data.frame(
  PC_num = seq_along(eigenvalues),
  eigenvalue = eigenvalues
)

ggplot(scree_df %>% slice(1:10), aes(x = PC_num, y = eigenvalue)) +
  geom_col(width = 0.6, fill = "#4C78A8", alpha = 0.8) +
  geom_line(group = 1, linewidth = 0.8, color = "black") +
  geom_point(size = 2, color = "black") +
  geom_hline(yintercept = 1, linetype = "dashed", color = "red", linewidth
= 0.6) +
  scale_x_continuous(
    breaks = 1:10,
    labels = paste0("PC", 1:10)
  ) +
  labs(
    title = "Scree Plot",
    x = "Principal Component",
    y = "Eigenvalue"
  ) +
  theme_bw(base_size = 13)

#-----------------------------------------------------------
#plot(pca, type = "l", main = "Scree Plot")

# 5) Build PCA plotting dataframe from full PCA results
pca_df <- data.frame(
```

```
  PC1 = pca$x[, 1],
  PC2 = pca$x[, 2],
  mouse = df_log$mouse,
  PMSI = df_log$PMSI,
  imputation = df_log$imputation,
  time_group = df_log$time_group
)

# 6) Filter ONLY for plotting:
#    Early = QRILC
#    Late  = KNN
pca_df_plot <- pca_df %>%
  filter(
    (time_group == "Early" & imputation == "QRILC") |
    (time_group == "Late"  & imputation == "KNN")
  ) %>%
  mutate(
    time_group = factor(time_group, levels = c("Early", "Late")),
    imputation = factor(imputation, levels = c("QRILC", "KNN"))
  )

# 7) % variance explained
var_exp <- (pca$sdev^2) / sum(pca$sdev^2)
pc1_lab <- paste0("PC1 (", round(100 * var_exp[1], 1), "%)")
pc2_lab <- paste0("PC2 (", round(100 * var_exp[2], 1), "%)")

# 8) PCA plot
ggplot(pca_df_plot, aes(x = PC1, y = PC2, color = time_group)) +
  geom_point(size = 3.5, alpha = 0.9) +
  stat_ellipse(aes(group = time_group), level = 0.85, linewidth = 0.6) +
  scale_color_manual(values = c(
    "Early" = "#4DBBD5",  # blue
    "Late"  = "#E64B35"   # red
  )) +
  labs(
    title = "Principal Component Analysis of Metabolomics Data by PMSI Gro
up",
    x = pc1_lab,
    y = pc2_lab,
    color = "PMSI group(colour)"
  ) +
  theme_classic(base_size = 14) +
  theme(
    plot.title = element_text(face = "bold", hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.5),
    legend.title = element_text(face = "bold"),
    axis.title = element_text(face = "bold"),
    axis.text = element_text(color = "black")
  )

ggsave("intensity_distribution.pdf", width = 10, height = 6)
ggsave("scree_plot.pdf", width = 6, height = 4)
```

Appendix N – PCA temporal trend analysis by labelling PMSI days / Figure 30 (Section 7.8)

```
library(ggrepel)
citation("ggrepel")

ggplot(pca_df_plot, aes(PC1, PC2, color = time_group, label = PMSI)) +
  geom_point(size = 3.5, alpha = 0.9) +
  geom_text_repel(size = 3) +
  stat_ellipse(aes(group = time_group), level = 0.85, linewidth = 0.6) +
  scale_color_manual(
    name = "PMSI group (colour)",
    values = c("Early" = "#4DBBD5", "Late" = "#E64B35")
  ) +
  theme_classic(base_size = 14)

ggsave("PCA_plot.pdf", width = 8, height = 5)
```

Appendix O – PCA outlier analysis by highlighting and labelling outliers / Figure 31 (Section 7.8)

```
# ================================
# Complete PCA plot + outlier identification
# with solid-colour outliers in legend
# ================================

library(dplyr)
library(tidyr)
library(ggplot2)
library(ggrepel)

# 1) Feature columns
id_cols <- grep("^ID:", names(df_combined), value = TRUE)

# 2) Log-transform feature data
df_log <- df_combined
df_log[id_cols] <- log1p(df_log[id_cols])

# 3) Create early/late grouping from PMSI
df_log <- df_log %>%
  mutate(
    time_group = ifelse(PMSI <= 7, "Early", "Late")
  )

# 4) Run PCA on the full dataset
pca <- prcomp(df_log[id_cols], center = TRUE, scale. = TRUE)

# ================================
# PCA dataframe
# ================================
pca_df <- data.frame(
  row_id = seq_len(nrow(df_log)),   # unique row identifier
  PC1 = pca$x[, 1],
  PC2 = pca$x[, 2],
  mouse = df_log$mouse,
  PMSI = df_log$PMSI,
  imputation = df_log$imputation,
  time_group = df_log$time_group
)

# 5) Filter only points shown in your PCA plot
#    Early = QRILC
#    Late  = KNN
pca_df_plot <- pca_df %>%
  filter(
    (time_group == "Early" & imputation == "QRILC") |
    (time_group == "Late"  & imputation == "KNN")
  ) %>%
  mutate(
    time_group = factor(time_group, levels = c("Early", "Late")),
```

```
    imputation = factor(imputation, levels = c("QRILC", "KNN"))
  )

# ================================
# Find outliers
# ================================
center_PC1 <- mean(pca_df_plot$PC1, na.rm = TRUE)
center_PC2 <- mean(pca_df_plot$PC2, na.rm = TRUE)

pca_df_plot <- pca_df_plot %>%
  mutate(
    dist_center = sqrt((PC1 - center_PC1)^2 + (PC2 - center_PC2)^2)
  )

# Top 4 most distant points
outliers <- pca_df_plot %>%
  arrange(desc(dist_center)) %>%
  slice(1:4) %>%
  mutate(
    label = paste0("Mouse: ", mouse, "\nPMSI: ", PMSI)
  )

# ================================
# Manually choose ONE extra point to highlight
# Best method: use approximate PCA coordinates
# Replace 4.2 and 3.2 with the point you actually want
# ================================
extra_highlight <- pca_df_plot %>%
  filter(abs(PC1 - 4.2) < 0.25,
         abs(PC2 - 3.2) < 0.25)

# Check what got selected
print(extra_highlight %>% select(row_id, mouse, PMSI, PC1, PC2))

  row_id mouse PMSI      PC1      PC2
1     81    M6    0 4.211643 3.231606

# Combine top 4 outliers + extra highlighted point
highlight_points <- bind_rows(outliers, extra_highlight) %>%
  distinct(row_id, .keep_all = TRUE) %>%
  mutate(
    label = paste0("Mouse: ", mouse, "\nPMSI: ", PMSI)
  )

# Create one plotting group so points stay solid-coloured
pca_df_plot <- pca_df_plot %>%
  mutate(
    plot_group = ifelse(row_id %in% highlight_points$row_id,
                        "Outlier",
                        as.character(time_group)),
    plot_group = factor(plot_group, levels = c("Early", "Late", "Outlier")
)
  )
```

```r
# Print highlighted table
cat("\nTop 4 PCA outliers plus extra highlighted point:\n")
```

```
Top 4 PCA outliers plus extra highlighted point:
```

```r
print(highlight_points)
```

```
  row_id       PC1        PC2 mouse PMSI imputation time_group dist_center
1      3 11.746767 -0.6604096    M1    2      QRILC      Early   11.765317
2     49 10.866485  0.3994622    M4    0      QRILC      Early   10.873824
3     66  7.411209  0.6973933    M5    1      QRILC      Early    7.443949
4     65  4.666603  4.1895751    M5    0      QRILC      Early    6.271342
5     81  4.211643  3.2316063    M6    0      QRILC      Early    5.308598
               label
1 Mouse: M1\nPMSI: 2
2 Mouse: M4\nPMSI: 0
3 Mouse: M5\nPMSI: 1
4 Mouse: M5\nPMSI: 0
5 Mouse: M6\nPMSI: 0
```

```r
# Save highlighted table
write.csv(highlight_points, "PCA_outliers_top4_plus_extra.csv", row.names
= FALSE)

# ================================
# Variance explained labels
# ================================
var_exp <- (pca$sdev^2) / sum(pca$sdev^2)
pc1_lab <- paste0("PC1 (", round(100 * var_exp[1], 1), "%)")
pc2_lab <- paste0("PC2 (", round(100 * var_exp[2], 1), "%)")

# ================================
# PCA plot
# ================================
p_pca <- ggplot(pca_df_plot, aes(x = PC1, y = PC2)) +

  geom_point(
    aes(color = plot_group),
    size = 3.5,
    alpha = 0.9
  ) +

  stat_ellipse(
    aes(color = time_group, group = time_group),
    level = 0.85,
    linewidth = 0.6,
    show.legend = FALSE
  ) +

  geom_text_repel(
    data = highlight_points,
    aes(x = PC1, y = PC2, label = label),
    inherit.aes = FALSE,
    size = 4,
```

```
    color = "black",
    box.padding = 0.6,
    point.padding = 0.4,
    segment.color = "grey40",
    max.overlaps = Inf
  ) +

  scale_color_manual(
    values = c(
      "Early"   = "#4DBBD5",
      "Late"    = "#E64B35",
      "Outlier" = "yellow"
    ),
    name = "PMSI group(colour)"
  ) +

  labs(
    title = "Principal Component Analysis of Metabolomics Data by PMSI Gro
up",
    x = pc1_lab,
    y = pc2_lab
  ) +

  coord_cartesian(clip = "off") +
  theme_classic(base_size = 14) +
  theme(
    plot.margin = margin(10, 140, 10, 10),
    plot.title = element_text(face = "bold", hjust = 0.5),
    legend.title = element_text(face = "bold"),
    axis.title = element_text(face = "bold"),
    axis.text = element_text(color = "black")
  )

print(p_pca)

ggsave("PCA_plot_with_outliers.pdf", plot = p_pca, width = 13, height = 7)
```

Appendix P - Bar charts of top metabolite loadings for PC1 and PC2 / Figure 32 & 33 (Section 7.8)

```
top_pc1 <- bind_rows(
  pc1_loadings %>% arrange(desc(PC1)) %>% slice(1:10),
  pc1_loadings %>% arrange(PC1) %>% slice(1:10)
) %>%
  distinct() %>%
  arrange(PC1) %>%
  mutate(
    direction = ifelse(PC1 > 0, "Positive", "Negative"),
    Metabolite = gsub("ID:", "", Metabolite)
  )

p_final <- ggplot(top_pc1, aes(x = abs(PC1), y = reorder(Metabolite, abs(P
C1)), fill = direction)) +
  geom_col(width = 0.7) +
  scale_fill_manual(values = c("Positive" = "#4DBBD5", "Negative" = "#E64B
35")) +
  labs(
    title = "Metabolites Driving Variation Along PC1",
    subtitle = "Top contributors based on absolute loading values",
    x = "Absolute PC1 loading (importance)",
    y = NULL,
    fill = "Direction"
  ) +
  theme_classic(base_size = 13) +
  theme(
    plot.title = element_text(face = "bold", hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.5),
    legend.position = "none"
  )

p_final

ggsave(
  filename = "pc1_loadings_plot.pdf",
  plot = p_final,
  width = 8,
  height = 6
)
```

```
loadings <- as.data.frame(pca$rotation) %>%
  tibble::rownames_to_column("Metabolite")

top_pc2 <- bind_rows(
  loadings %>% arrange(desc(PC2)) %>% slice(1:10),
  loadings %>% arrange(PC2) %>% slice(1:10)
) %>%
  distinct() %>%
  arrange(PC2) %>%
```

```
  mutate(
    direction = ifelse(PC2 > 0, "Positive", "Negative"),
    Metabolite = gsub("ID:", "", Metabolite)
  )

p_final <- ggplot(top_pc2, aes(x = abs(PC2), y = reorder(Metabolite, abs(P
C2)), fill = direction)) +
  geom_col(width = 0.7) +
  scale_fill_manual(values = c("Positive" = "#4DBBD5", "Negative" = "#E64B
35")) +
  labs(
    title = "Metabolites Driving Variation Along PC2",
    subtitle = "Top contributors based on absolute loading values",
    x = "Absolute PC2 loading (importance)",
    y = NULL,
    fill = "Direction"
  ) +
  theme_classic(base_size = 13) +
  theme(
    plot.title = element_text(face = "bold", hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.5),
    legend.position = "right"
  )

p_final

ggsave(
  filename = "pc2_loadings_plot.pdf",
  plot = p_final,
  width = 8,
  height = 6
)
```

Appendix Q – Random Forest model with PMSI ~ PC2 scores including an importance plot for the PCs / Figure 34 & 37 (Section 7.9)

```
selected_features <- top_pc2$feature

selected_features[selected_features %in% colnames(df_combined)]

setdiff(selected_features, colnames(df_combined))

character(0)

library(dplyr)
library(ranger)
#citation("ranger")
library(caret)

Warning: package 'caret' was built under R version 4.5.3

Loading required package: lattice


Attaching package: 'caret'

The following object is masked from 'package:purrr':

    lift

citation("caret")

To cite caret in publications use:

  Kuhn, M. (2008). Building Predictive Models in R Using the caret
  Package. Journal of Statistical Software, 28(5), 1-26.
  https://doi.org/10.18637/jss.v028.i05

A BibTeX entry for LaTeX users is

  @Article{,
    title = {Building Predictive Models in R Using the caret Package},
    volume = {28},
    url = {https://www.jstatsoft.org/index.php/jss/article/view/v028i05},
    doi = {10.18637/jss.v028.i05},
    number = {5},
    journal = {Journal of Statistical Software},
    author = {{Kuhn} and {Max}},
    year = {2008},
    pages = {1-26},
  }

library(ggplot2)

set.seed(123)

# Select top PC2 metabolite features
```

```r
selected_features <- top_pc2$feature

# Build random forest dataset
rf_data <- df_combined %>%
  dplyr::select(all_of(selected_features), PMSI) %>%
  na.omit()

# Rename predictor columns to safe names
old_names <- colnames(rf_data)
new_names <- old_names
new_names[new_names != "PMSI"] <- paste0("M", seq_len(sum(old_names != "PM
SI")))
colnames(rf_data) <- new_names

# Save mapping between model names and original metabolite names
name_map <- data.frame(
  original_name = old_names[old_names != "PMSI"],
  model_name = new_names[new_names != "PMSI"]
)

print(name_map)
```

```
      original_name model_name
1  ID:540.1128@1.28         M1
2   ID:508.1185@1.3         M2
3  ID:732.0365@3.92         M3
4  ID:634.0511@3.94         M4
5  ID:964.7176@3.74         M5
6   ID:964.705@3.73         M6
7  ID:721.8308@3.73         M7
8   ID:631.9104@3.8         M8
9   ID:353.958@3.74         M9
10  ID:928.489@3.79        M10
11 ID:427.8913@3.74        M11
12 ID:129.0331@1.81        M12
13 ID:257.0712@1.78        M13
14 ID:803.5381@1.97        M14
15 ID:116.0839@1.47        M15
16 ID:116.0832@1.47        M16
17  ID:282.259@1.25        M17
18 ID:282.2605@1.25        M18
19 ID:157.9999@3.73        M19
20  ID:157.999@3.73        M20
```

```r
# Split into training and test sets
train_index <- createDataPartition(rf_data$PMSI, p = 0.8, list = FALSE)
train_data <- rf_data[train_index, ]
test_data  <- rf_data[-train_index, ]

# Cross-validation settings
train_control <- trainControl(
  method = "repeatedcv",
  number = 5,
  repeats = 5
```

```r
)

# Tuning grid
tune_grid <- expand.grid(
  mtry = c(2, 4, 6, 8),
  splitrule = "variance",
  min.node.size = c(1, 3, 5, 10)
)

# Train random forest model
rf_pc2_cv <- train(
  PMSI ~ .,
  data = train_data,
  method = "ranger",
  trControl = train_control,
  tuneGrid = tune_grid,
  importance = "permutation",
  num.trees = 500,
  metric = "RMSE"
)

# View model summary
print(rf_pc2_cv)
```

```
Random Forest

79 samples
20 predictors

No pre-processing
Resampling: Cross-Validated (5 fold, repeated 5 times)
Summary of sample sizes: 63, 64, 63, 63, 63, 64, ...
Resampling results across tuning parameters:

  mtry  min.node.size  RMSE      Rsquared   MAE
  2      1             7.276962  0.5433839  4.959856
  2      3             7.241546  0.5528441  4.962395
  2      5             7.316783  0.5451235  5.031852
  2     10             7.467159  0.5323949  5.182412
  4      1             7.106599  0.5632359  4.905531
  4      3             7.151980  0.5572953  4.957352
  4      5             7.178451  0.5553487  4.998207
  4     10             7.308285  0.5465237  5.129364
  6      1             7.057084  0.5663129  4.890302
  6      3             7.107544  0.5627133  4.903687
  6      5             7.107941  0.5617862  4.963907
  6     10             7.227398  0.5521088  5.089227
  8      1             7.078253  0.5613120  4.906971
  8      3             7.134185  0.5547086  4.957729
  8      5             7.124480  0.5590833  4.972123
  8     10             7.192373  0.5531473  5.072542

Tuning parameter 'splitrule' was held constant at a value of variance
RMSE was used to select the optimal model using the smallest value.
```

```
The final values used for the model were mtry = 6, splitrule = variance
 and min.node.size = 1.
```

```
print(rf_pc2_cv$bestTune)
```

```
  mtry splitrule min.node.size
9    6  variance             1
```

```
print(rf_pc2_cv$results)
```

```
   mtry splitrule min.node.size     RMSE  Rsquared      MAE   RMSESD Rsqua
redSD
1     2  variance             1 7.276962 0.5433839 4.959856 2.483419  0.23
91547
2     2  variance             3 7.241546 0.5528441 4.962395 2.484341  0.23
98611
3     2  variance             5 7.316783 0.5451235 5.031852 2.460067  0.23
93561
4     2  variance            10 7.467159 0.5323949 5.182412 2.425321  0.24
01785
5     4  variance             1 7.106599 0.5632359 4.905531 2.442505  0.24
12510
6     4  variance             3 7.151980 0.5572953 4.957352 2.504626  0.24
80868
7     4  variance             5 7.178451 0.5553487 4.998207 2.473021  0.24
46802
8     4  variance            10 7.308285 0.5465237 5.129364 2.444949  0.24
48880
9     6  variance             1 7.057084 0.5663129 4.890302 2.515818  0.25
16438
10    6  variance             3 7.107544 0.5627133 4.903687 2.541288  0.25
42543
11    6  variance             5 7.107941 0.5617862 4.963907 2.481874  0.24
88985
12    6  variance            10 7.227398 0.5521088 5.089227 2.481498  0.25
00777
13    8  variance             1 7.078253 0.5613120 4.906971 2.545870  0.25
54456
14    8  variance             3 7.134185 0.5547086 4.957729 2.551947  0.25
74099
15    8  variance             5 7.124480 0.5590833 4.972123 2.525212  0.25
61716
16    8  variance            10 7.192373 0.5531473 5.072542 2.496075  0.25
59465
      MAESD
1  1.324692
2  1.351787
3  1.310884
4  1.304920
5  1.360718
6  1.408941
7  1.372007
8  1.364739
9  1.449131
10 1.453371
```

```
11 1.430899
12 1.424065
13 1.514977
14 1.516655
15 1.482691
16 1.470304
```

```r
# Predict on held-out test data
pred_pc2 <- predict(rf_pc2_cv, newdata = test_data)

# Calculate performance metrics
rmse_val <- RMSE(pred_pc2, test_data$PMSI)
mae_val  <- MAE(pred_pc2, test_data$PMSI)
r2_val   <- cor(test_data$PMSI, pred_pc2)^2

cat("RMSE:", rmse_val, "\n")
```

```
RMSE: 6.223345
```

```r
cat("MAE :", mae_val, "\n")
```

```
MAE : 5.034625
```

```r
cat("R2  :", r2_val, "\n")
```

```
R2  : 0.660829
```

```r
# Create plotting dataframe
plot_df_pc2 <- data.frame(
  Observed = test_data$PMSI,
  Predicted = pred_pc2
)

# Plot observed vs predicted
p <- ggplot(plot_df_pc2, aes(x = Observed, y = Predicted)) +
  geom_point(size = 3, alpha = 0.7) +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", colour = "red
") +
  labs(
    title = "Observed vs Predicted PMSI (Top PC2 Metabolites)",
    subtitle = paste0(
      "RMSE = ", round(rmse_val, 2),
      " | MAE = ", round(mae_val, 2),
      " | R² = ", round(r2_val, 2)
    ),
    x = "Observed PMSI",
    y = "Predicted PMSI"
  ) +
  theme_minimal(base_size = 14)

print(p)

ggsave("pc2_model_plot.pdf", plot = p, width = 10, height = 7)

# Extract variable importance
var_imp <- varImp(rf_pc2_cv, scale = FALSE)
```

```r
# Convert to dataframe
var_imp_df <- var_imp$importance
var_imp_df$model_name <- rownames(var_imp_df)

# Merge with original metabolite names
var_imp_df <- merge(
  var_imp_df,
  name_map,
  by = "model_name",
  all.x = TRUE
)
ggsave("pc2_model_predicitions.pdf", width = 10, height = 7)

# Order by importance
var_imp_df <- var_imp_df[order(var_imp_df$Overall, decreasing = TRUE), ]

# Take top 20 metabolites (adjust if needed)
top_imp <- head(var_imp_df, 20)

# Plot with real metabolite names
ggplot(top_imp, aes(
  x = reorder(original_name, Overall),
  y = Overall
)) +
  geom_col() +
  coord_flip() +
  labs(
    title = "Top Metabolites Driving PMSI Prediction",
    x = "Metabolite",
    y = "Importance (Permutation)"
  ) +
  theme_minimal(base_size = 14)

ggsave("pc2_variable_importance.pdf", width = 10, height = 7)
```

Appendix R - Random Forest model with PMSI ~ PCA (PC1-PC10) including an importance plot for the PCs / Figure 35 (Section 7.9)

```
library(dplyr)
library(ranger)
library(caret)
library(ggplot2)
library(stringr)

set.seed(123)

# Extract metabolite-only data (exclude PMSI)
metabolite_data <- df_combined %>%
  dplyr::select(where(is.numeric)) %>%
  dplyr::select(-PMSI)

# Keep PMSI separately
pmsi_values <- df_combined$PMSI

# Remove rows with missing values before PCA
complete_index <- complete.cases(metabolite_data, pmsi_values)
metabolite_data_complete <- metabolite_data[complete_index, ]
pmsi_values_complete <- pmsi_values[complete_index]

# Run PCA on scaled metabolite data
pca_model <- prcomp(metabolite_data_complete, scale. = TRUE)

# Optional: inspect variance explained
pca_variance <- summary(pca_model)
print(pca_variance)
```

```
Importance of components:
                          PC1     PC2     PC3     PC4     PC5     PC6
PC7
Standard deviation     3.9439 1.36846 1.19825 1.11610 0.90330 0.73167 0.57
677
Proportion of Variance 0.6763 0.08142 0.06243 0.05416 0.03548 0.02328 0.01
446
Cumulative Proportion  0.6763 0.75768 0.82011 0.87427 0.90975 0.93302 0.94
749
                           PC8     PC9    PC10    PC11    PC12    PC13
PC14
Standard deviation     0.54666 0.49970 0.42913 0.39943 0.35313 0.24874 0.2
1681
Proportion of Variance 0.01299 0.01086 0.00801 0.00694 0.00542 0.00269 0.0
0204
Cumulative Proportion  0.96048 0.97134 0.97934 0.98628 0.99170 0.99439 0.9
9643
                          PC15    PC16    PC17    PC18    PC19    PC20
PC21
Standard deviation     0.16075 0.13346 0.11280 0.10343 0.08378 0.06524 0.0
5267
```

```
Proportion of Variance 0.00112 0.00077 0.00055 0.00047 0.00031 0.00019 0.0
0012
Cumulative Proportion  0.99756 0.99833 0.99889 0.99935 0.99966 0.99984 0.9
9996
                          PC22    PC23
Standard deviation     0.02780 0.01058
Proportion of Variance 0.00003 0.00000
Cumulative Proportion  1.00000 1.00000

# Convert PCA scores to dataframe
pca_scores <- as.data.frame(pca_model$x)

# Add PMSI back in
pca_scores$PMSI <- pmsi_values_complete

# Select first 10 PCs + PMSI
# Adjust number of PCs here if needed
pca_data <- pca_scores %>%
  dplyr::select(PC1:PC10, PMSI)

# Split into training and test sets
train_index_pca <- createDataPartition(pca_data$PMSI, p = 0.8, list = FALS
E)
train_data_pca <- pca_data[train_index_pca, ]
test_data_pca  <- pca_data[-train_index_pca, ]

# Cross-validation settings
train_control <- trainControl(
  method = "repeatedcv",
  number = 5,
  repeats = 5
)

# Tuning grid
tune_grid_pca <- expand.grid(
  mtry = c(2, 3, 5, 7),
  splitrule = "variance",
  min.node.size = c(1, 3, 5)
)

# Train random forest model on PCA outputs
rf_pca_cv <- train(
  PMSI ~ .,
  data = train_data_pca,
  method = "ranger",
  trControl = train_control,
  tuneGrid = tune_grid_pca,
  importance = "permutation",
  num.trees = 500,
  metric = "RMSE"
)

# View model summary
print(rf_pca_cv)
```

```
Random Forest

79 samples
10 predictors

No pre-processing
Resampling: Cross-Validated (5 fold, repeated 5 times)
Summary of sample sizes: 63, 64, 63, 63, 63, 64, ...
Resampling results across tuning parameters:

  mtry  min.node.size  RMSE      Rsquared   MAE
  2     1              8.033148  0.4178481  5.628194
  2     3              8.072825  0.4090389  5.650257
  2     5              8.066824  0.4149797  5.653194
  3     1              7.998494  0.4223543  5.601067
  3     3              7.998095  0.4200286  5.607073
  3     5              7.969574  0.4260160  5.556415
  5     1              7.888534  0.4393739  5.483942
  5     3              7.885295  0.4403167  5.493428
  5     5              7.957000  0.4296507  5.560717
  7     1              7.899310  0.4418494  5.475944
  7     3              7.906227  0.4385425  5.482448
  7     5              7.881397  0.4420144  5.488611

Tuning parameter 'splitrule' was held constant at a value of variance
RMSE was used to select the optimal model using the smallest value.
The final values used for the model were mtry = 7, splitrule = variance
 and min.node.size = 5.
```

```
print(rf_pca_cv$bestTune)
```

```
   mtry splitrule min.node.size
12    7  variance             5
```

```
print(rf_pca_cv$results)
```

```
   mtry splitrule min.node.size     RMSE  Rsquared      MAE   RMSESD Rsqua
redSD
1     2  variance             1 8.033148 0.4178481 5.628194 2.188100  0.20
67388
2     2  variance             3 8.072825 0.4090389 5.650257 2.182068  0.20
15463
3     2  variance             5 8.066824 0.4149797 5.653194 2.203394  0.20
76026
4     3  variance             1 7.998494 0.4223543 5.601067 2.214552  0.20
34978
5     3  variance             3 7.998095 0.4200286 5.607073 2.218266  0.20
32421
6     3  variance             5 7.969574 0.4260160 5.556415 2.194484  0.19
96402
7     5  variance             1 7.888534 0.4393739 5.483942 2.265089  0.20
34955
8     5  variance             3 7.885295 0.4403167 5.493428 2.242071  0.20
41279
9     5  variance             5 7.957000 0.4296507 5.560717 2.190878  0.20
```

```
39166
10    7  variance             1 7.899310 0.4418494 5.475944 2.196643  0.19
94676
11    7  variance             3 7.906227 0.4385425 5.482448 2.245260  0.20
40982
12    7  variance             5 7.881397 0.4420144 5.488611 2.228992  0.19
79651
      MAESD
1  1.054919
2  1.067668
3  1.073249
4  1.126394
5  1.121973
6  1.101237
7  1.217837
8  1.172905
9  1.149845
10 1.178958
11 1.212132
12 1.181624
```

```r
# Predict on held-out test data
pred_pca <- predict(rf_pca_cv, newdata = test_data_pca)

# Calculate performance metrics
rmse_pca <- RMSE(pred_pca, test_data_pca$PMSI)
mae_pca  <- MAE(pred_pca, test_data_pca$PMSI)
r2_pca   <- cor(test_data_pca$PMSI, pred_pca)^2

cat("PCA MODEL\n")
```

```
PCA MODEL
```

```r
cat("RMSE:", rmse_pca, "\n")
```

```
RMSE: 4.761432
```

```r
cat("MAE :", mae_pca, "\n")
```

```
MAE : 3.530875
```

```r
cat("R2  :", r2_pca, "\n")
```

```
R2  : 0.7841895
```

```r
# Create plotting dataframe
plot_df_pca <- data.frame(
  Observed = test_data_pca$PMSI,
  Predicted = pred_pca
)

# Plot observed vs predicted
p_pca <- ggplot(plot_df_pca, aes(x = Observed, y = Predicted)) +
  geom_point(size = 3, alpha = 0.7) +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", colour = "red
") +
```

```
  labs(
    title = "Observed vs Predicted PMSI (PCA Outputs)",
    subtitle = paste0(
      "RMSE = ", round(rmse_pca, 2),
      " | MAE = ", round(mae_pca, 2),
      " | R² = ", round(r2_pca, 2)
    ),
    x = "Observed PMSI",
    y = "Predicted PMSI"
  ) +
  theme_minimal(base_size = 14)

print(p_pca)
```

```
# Save observed vs predicted plot as PDF
ggsave("pca_model_plot.pdf", plot = p_pca, width = 10, height = 7)

# Extract variable importance for PCs
var_imp_pca <- varImp(rf_pca_cv, scale = FALSE)

# Convert to dataframe
var_imp_df_pca <- var_imp_pca$importance
var_imp_df_pca$PC <- rownames(var_imp_df_pca)

# Order by importance
var_imp_df_pca <- var_imp_df_pca[order(var_imp_df_pca$Overall, decreasing
= TRUE), ]

# Plot PC importance
p_imp_pca <- ggplot(var_imp_df_pca, aes(
  x = reorder(PC, Overall),
  y = Overall
)) +
  geom_col() +
  coord_flip() +
  labs(
    title = "Importance of PCA Outputs for PMSI Prediction",
    x = "Principal Component",
    y = "Importance (Permutation)"
  ) +
  theme_minimal(base_size = 14)

print(p_imp_pca)

# Save PC importance plot as PDF
ggsave("pca_variable_importance.pdf", plot = p_imp_pca, width = 10, height
= 7)

# Save predictions to CSV
write.csv(plot_df_pca, "pca_model_predictions.csv", row.names = FALSE)

# Save PCA importance table to CSV
```

```r
pdf("pca_all_rf.pdf", width = 10, height = 7)
print(p_pca)
print(p_imp_pca)
dev.off()
```

```
png
  2
```

Appendix S – Leave-one-out validation using PCA model on all six mice / Figure 36 (Section 7.9)

```
library(dplyr)
library(ranger)
library(caret)
library(ggplot2)

set.seed(123)

# ----------------------------
# 1. Define mouse ID column
# ----------------------------
mouse_id_col <- "mouse"   # change if needed

if (!mouse_id_col %in% colnames(df_combined)) {
  stop(paste("Column not found:", mouse_id_col))
}

mouse_ids <- unique(df_combined[[mouse_id_col]])
mouse_ids <- mouse_ids[!is.na(mouse_ids)]

cat("Mice found:\n")
```

```
Mice found:
```

```
print(mouse_ids)
```

```
[1] M1 M2 M3 M4 M5 M6
Levels: M1 M2 M3 M4 M5 M6
```

```
# ----------------------------
# 2. Storage objects
# ----------------------------
lomo_pca_results <- data.frame(
  Mouse = character(),
  n_test = integer(),
  RMSE = numeric(),
  MAE = numeric(),
  R2 = numeric(),
  stringsAsFactors = FALSE
)

all_pca_predictions <- data.frame(
  Mouse = character(),
  Observed = numeric(),
  Predicted = numeric(),
  stringsAsFactors = FALSE
)

# ----------------------------
# 3. Loop through mice
# ----------------------------
```

```r
for (test_mouse in mouse_ids) {

  cat("\n============================\n")
  cat("Running PCA LOMO for:", test_mouse, "\n")
  cat("============================\n")

  # Split data
  train_df <- df_combined %>%
    filter(.data[[mouse_id_col]] != test_mouse)

  test_df <- df_combined %>%
    filter(.data[[mouse_id_col]] == test_mouse)

  # Keep only numeric predictors (excluding PMSI)
  train_metabolites <- train_df %>%
    dplyr::select(where(is.numeric)) %>%
    dplyr::select(-matches("^PMSI$"))

  test_metabolites <- test_df %>%
    dplyr::select(where(is.numeric)) %>%
    dplyr::select(-matches("^PMSI$"))

  # Keep PMSI separately
  train_pmsi <- train_df$PMSI
  test_pmsi  <- test_df$PMSI

  # Remove incomplete rows
  train_complete <- complete.cases(train_metabolites, train_pmsi)
  test_complete  <- complete.cases(test_metabolites, test_pmsi)

  train_metabolites <- train_metabolites[train_complete, , drop = FALSE]
  test_metabolites  <- test_metabolites[test_complete, , drop = FALSE]
  train_pmsi <- train_pmsi[train_complete]
  test_pmsi  <- test_pmsi[test_complete]

  # Skip if too few test samples
  if (nrow(test_metabolites) < 2) {
    cat("Skipping", test_mouse, "- too few complete test samples\n")
    next
  }

  # Make sure train and test have the same predictor columns
  common_cols <- intersect(colnames(train_metabolites), colnames(test_meta
bolites))
  train_metabolites <- train_metabolites[, common_cols, drop = FALSE]
  test_metabolites  <- test_metabolites[, common_cols, drop = FALSE]

  # PCA on training data only
  pca_model <- prcomp(train_metabolites, scale. = TRUE)

  # Project train and test into PCA space
  train_scores <- as.data.frame(pca_model$x)
  test_scores  <- as.data.frame(predict(pca_model, newdata = test_metaboli
tes))
```

```
  # Choose number of PCs safely
  n_pcs <- min(10, ncol(train_scores), ncol(test_scores))

  train_data <- train_scores[, 1:n_pcs, drop = FALSE]
  train_data$PMSI <- train_pmsi

  test_data <- test_scores[, 1:n_pcs, drop = FALSE]
  test_data$PMSI <- test_pmsi

  # Tune grid based on number of PCs
  p <- ncol(train_data) - 1
  mtry_vals <- unique(pmax(1, pmin(p, c(2, 3, 5, 7))))

  tune_grid <- expand.grid(
    mtry = mtry_vals,
    splitrule = "variance",
    min.node.size = c(1, 3, 5)
  )

  # Cross-validation settings
  train_control <- trainControl(
    method = "repeatedcv",
    number = 5,
    repeats = 5
  )

  # Train RF on PCA outputs
  rf_pca_lomo <- train(
    PMSI ~ .,
    data = train_data,
    method = "ranger",
    trControl = train_control,
    tuneGrid = tune_grid,
    importance = "permutation",
    num.trees = 500,
    metric = "RMSE"
  )

  # Predict on left-out mouse
  preds <- predict(rf_pca_lomo, newdata = test_data)

  # Metrics
  rmse <- RMSE(preds, test_data$PMSI)
  mae  <- MAE(preds, test_data$PMSI)
  r2   <- if (length(unique(test_data$PMSI)) > 1) cor(test_data$PMSI, pred
s)^2 else NA_real_

  # Store per-mouse results
  lomo_pca_results <- rbind(
    lomo_pca_results,
    data.frame(
      Mouse = as.character(test_mouse),
      n_test = as.integer(nrow(test_data)),
```

```
      RMSE = as.numeric(rmse),
      MAE = as.numeric(mae),
      R2 = as.numeric(r2),
      stringsAsFactors = FALSE
    )
  )

  # Store predictions
  all_pca_predictions <- rbind(
    all_pca_predictions,
    data.frame(
      Mouse = as.character(test_mouse),
      Observed = as.numeric(test_data$PMSI),
      Predicted = as.numeric(preds),
      stringsAsFactors = FALSE
    )
  )

  cat("n_test:", nrow(test_data), "\n")
  cat("RMSE:", round(rmse, 2), "\n")
  cat("MAE :", round(mae, 2), "\n")
  cat("R2  :", round(r2, 2), "\n")
}


============================
Running PCA LOMO for: M1
============================
n_test: 16
RMSE: 5.2
MAE : 3.76
R2  : 0.78

============================
Running PCA LOMO for: M2
============================
n_test: 16
RMSE: 8.66
MAE : 6.08
R2  : 0.28

============================
Running PCA LOMO for: M3
============================
n_test: 16
RMSE: 6.9
MAE : 5.04
R2  : 0.59

============================
Running PCA LOMO for: M4
============================
n_test: 16
RMSE: 9.15
```

```
MAE : 5.13
R2  : 0.25

============================
Running PCA LOMO for: M5
============================
n_test: 16
RMSE: 9.18
MAE : 5.53
R2  : 0.18

============================
Running PCA LOMO for: M6
============================
n_test: 15
RMSE: 6.38
MAE : 4.4
R2  : 0.74
```

```
# ----------------------------
# 4. Force numeric columns
# ----------------------------
lomo_pca_results$n_test <- as.integer(lomo_pca_results$n_test)
lomo_pca_results$RMSE   <- as.numeric(lomo_pca_results$RMSE)
lomo_pca_results$MAE    <- as.numeric(lomo_pca_results$MAE)
lomo_pca_results$R2     <- as.numeric(lomo_pca_results$R2)

# ----------------------------
# 5. View and save results
# ----------------------------
print(lomo_pca_results)
```

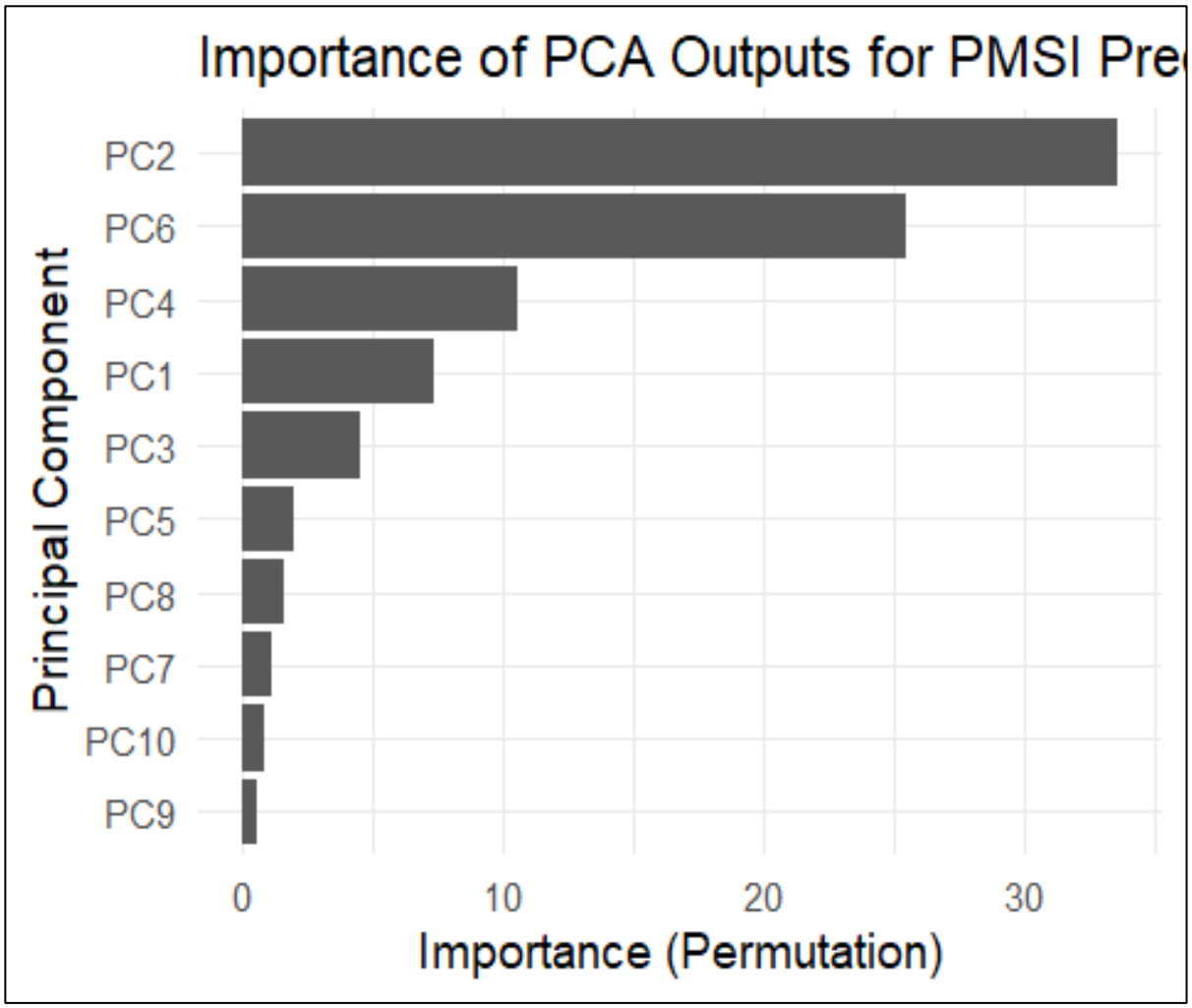
```
  Mouse n_test     RMSE      MAE        R2
1    M1     16 5.199109 3.756750 0.7830781
2    M2     16 8.664879 6.082000 0.2820598
3    M3     16 6.903718 5.035875 0.5916413
4    M4     16 9.153557 5.130187 0.2464767
5    M5     16 9.181453 5.532125 0.1796454
6    M6     15 6.375818 4.403600 0.7355488
```

```
write.csv(lomo_pca_results, "LOMO_PCA_results_all_mice.csv", row.names = F
ALSE)
write.csv(all_pca_predictions, "LOMO_PCA_predictions_all_mice.csv", row.na
mes = FALSE)

# ----------------------------
# 6. Summary metrics across mice
# ----------------------------
cat("\n============================\n")


============================

cat("PCA LOMO SUMMARY ACROSS MICE\n")
```

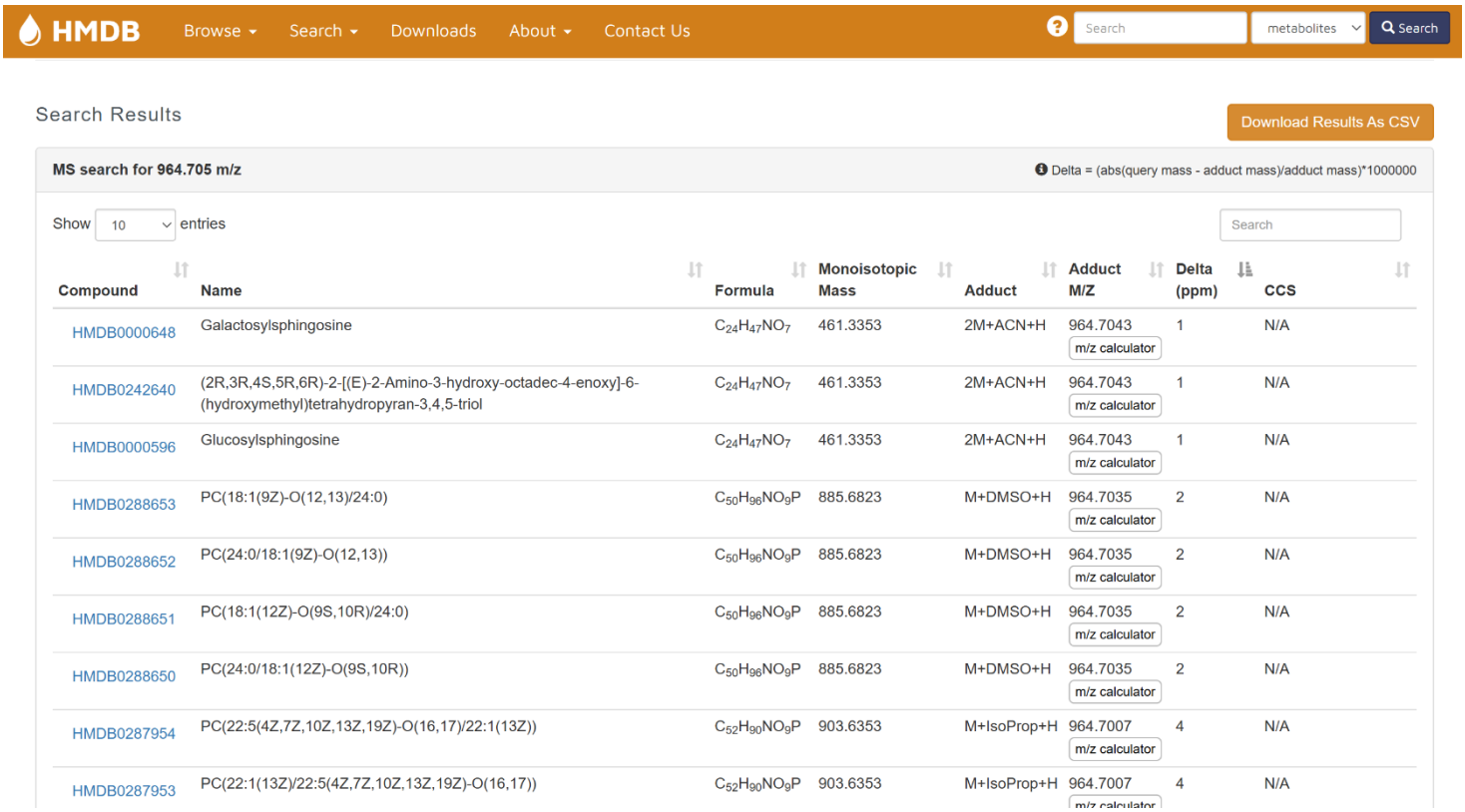
```
PCA LOMO SUMMARY ACROSS MICE

cat("============================\n")

============================

cat("Mean RMSE:", round(mean(lomo_pca_results$RMSE, na.rm = TRUE), 2), "\n
")

Mean RMSE: 7.58

cat("Mean MAE :", round(mean(lomo_pca_results$MAE, na.rm = TRUE), 2), "\n"
)

Mean MAE : 4.99

cat("Mean R2  :", round(mean(lomo_pca_results$R2, na.rm = TRUE), 2), "\n")

Mean R2  : 0.47

# ----------------------------
# 7. Overall observed vs predicted
# ----------------------------
overall_rmse_pca <- RMSE(all_pca_predictions$Predicted, all_pca_prediction
s$Observed)
overall_mae_pca  <- MAE(all_pca_predictions$Predicted, all_pca_predictions
$Observed)
overall_r2_pca   <- cor(all_pca_predictions$Observed, all_pca_predictions$
Predicted)^2

p_overall_pca <- ggplot(all_pca_predictions, aes(x = Observed, y = Predict
ed, shape = Mouse)) +
  geom_point(size = 3, alpha = 0.8) +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", colour = "red
") +
  labs(
    title = "Observed vs Predicted PMSI Across All PCA LOMO Folds",
    subtitle = paste0(
      "Overall RMSE = ", round(overall_rmse_pca, 2),
      " | MAE = ", round(overall_mae_pca, 2),
      " | R² = ", round(overall_r2_pca, 2)
    ),
    x = "Observed PMSI",
    y = "Predicted PMSI"
  ) +
  theme_minimal(base_size = 14)

print(p_overall_pca)

ggsave("LOMO_PCA_overall_observed_vs_predicted.pdf", plot = p_overall_pca,
width = 9, height = 7)

# ----------------------------
# 8. Faceted plot by mouse
# ----------------------------
mouse_stats_pca <- all_pca_predictions %>%
```

```
  group_by(Mouse) %>%
  summarise(
    R2 = if (length(unique(Observed)) > 1) cor(Observed, Predicted)^2 else
NA_real_,
    .groups = "drop"
  )

p_facet_pca <- ggplot(all_pca_predictions, aes(x = Observed, y = Predicted
)) +
  geom_point(size = 2.5, alpha = 0.8) +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", colour = "red
") +
  facet_wrap(~ Mouse, ncol = 3) +
  geom_text(
    data = mouse_stats_pca,
    aes(label = paste0("R² = ", round(R2, 2))),
    x = 40, y = 2,
    inherit.aes = FALSE
  ) +
  coord_cartesian(xlim = c(0, 45), ylim = c(0, 32)) +
  labs(
    title = "Observed vs Predicted PMSI by Mouse (PCA LOMO Validation)",
    subtitle = paste0(
      "Overall RMSE = ", round(overall_rmse_pca, 2),
      " | MAE = ", round(overall_mae_pca, 2),
      " | R² = ", round(overall_r2_pca, 2)
    ),
    x = "Observed PMSI",
    y = "Predicted PMSI"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    strip.text = element_text(face = "bold"),
    plot.title = element_text(size = 18, face = "bold")
  )

print(p_facet_pca)

ggsave("LOMO_PCA_facet_by_mouse.pdf", plot = p_facet_pca, width = 12, heig
ht = 8)
```

Appendix T- PCA importance plot (Section 9.8)



Appendix U – Human metabolome database (HMDB) search for metabolite 964.705 (Section 9.9)



HMDB Browse Search Downloads About Contact Us Search metabolites Search

Search Results

Download Results As CSV

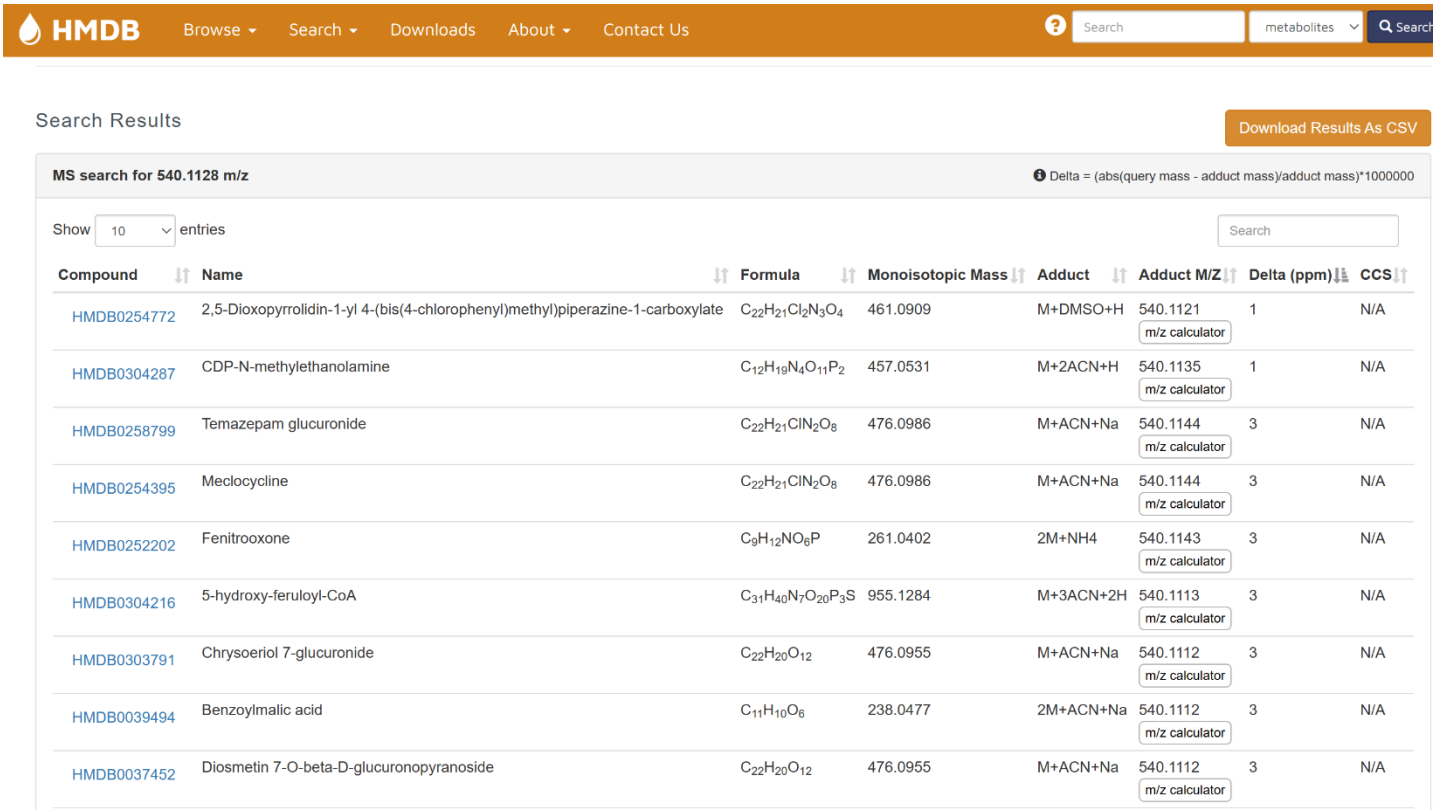
MS search for 964.705 m/z

Delta = (abs(query mass - adduct mass)/adduct mass)*1000000

Show 10 entries Search

Compound	Name	Formula	Monoisotopic Mass	Adduct	Adduct M/Z	Delta (ppm)	CCS
HMDB0000648	Galactosylsphingosine	$C_{24}H_{47}NO_7$	461.3353	2M+ACN+H	964.7043 m/z calculator	1	N/A
HMDB0242640	(2R,3R,4S,5R,6R)-2-[(E)-2-Amino-3-hydroxy-octadec-4-enoxy]-6-(hydroxymethyl)tetrahydropyran-3,4,5-triol	$C_{24}H_{47}NO_7$	461.3353	2M+ACN+H	964.7043 m/z calculator	1	N/A
HMDB0000596	Glucosylsphingosine	$C_{24}H_{47}NO_7$	461.3353	2M+ACN+H	964.7043 m/z calculator	1	N/A
HMDB0288653	PC(18:1(9Z)-O(12,13)/24:0)	$C_{50}H_{96}NO_9P$	885.6823	M+DMSO+H	964.7035 m/z calculator	2	N/A
HMDB0288652	PC(24:0/18:1(9Z)-O(12,13))	$C_{50}H_{96}NO_9P$	885.6823	M+DMSO+H	964.7035 m/z calculator	2	N/A
HMDB0288651	PC(18:1(12Z)-O(9S,10R)/24:0)	$C_{50}H_{96}NO_9P$	885.6823	M+DMSO+H	964.7035 m/z calculator	2	N/A
HMDB0288650	PC(24:0/18:1(12Z)-O(9S,10R))	$C_{50}H_{96}NO_9P$	885.6823	M+DMSO+H	964.7035 m/z calculator	2	N/A
HMDB0287954	PC(22:5(4Z,7Z,10Z,13Z,19Z)-O(16,17)/22:1(13Z))	$C_{52}H_{90}NO_9P$	903.6353	M+IsoProp+H	964.7007 m/z calculator	4	N/A
HMDB0287953	PC(22:1(13Z)/22:5(4Z,7Z,10Z,13Z,19Z)-O(16,17))	$C_{52}H_{90}NO_9P$	903.6353	M+IsoProp+H	964.7007 m/z calculator	4	N/A

Appendix V – Human metabolome database (HMDB) search for metabolite 540.1128 (Section 9.9)



Search Results

Download Results As CSV

MS search for 540.1128 m/z

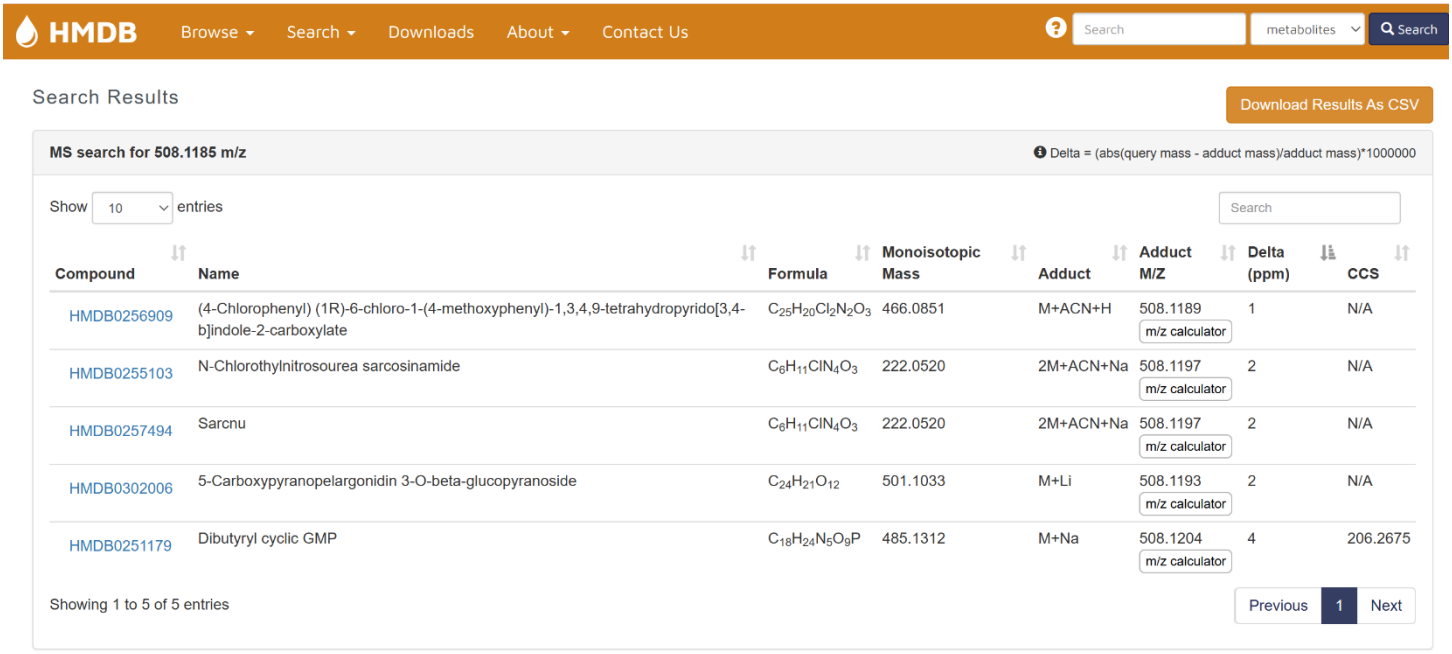
Delta = (abs(query mass - adduct mass)/adduct mass)*1000000

Show 10 entries

Search

Compound	Name	Formula	Monoisotopic Mass	Adduct	Adduct M/Z	Delta (ppm)	CCS
HMDB0254772	2,5-Dioxopyrrolidin-1-yl 4-(bis(4-chlorophenyl)methyl)piperazine-1-carboxylate	$C_{22}H_{21}Cl_2N_3O_4$	461.0909	M+DMSO+H	540.1121 m/z calculator	1	N/A
HMDB0304287	CDP-N-methylethanolamine	$C_{12}H_{19}N_4O_{11}P_2$	457.0531	M+2ACN+H	540.1135 m/z calculator	1	N/A
HMDB0258799	Temazepam glucuronide	$C_{22}H_{21}ClN_2O_8$	476.0986	M+ACN+Na	540.1144 m/z calculator	3	N/A
HMDB0254395	Meclocycline	$C_{22}H_{21}ClN_2O_8$	476.0986	M+ACN+Na	540.1144 m/z calculator	3	N/A
HMDB0252202	Fenitrooxone	$C_9H_{12}NO_6P$	261.0402	2M+NH4	540.1143 m/z calculator	3	N/A
HMDB0304216	5-hydroxy-feruloyl-CoA	$C_{31}H_{40}N_7O_{20}P_3S$	955.1284	M+3ACN+2H	540.1113 m/z calculator	3	N/A
HMDB0303791	Chrysoeriol 7-glucuronide	$C_{22}H_{20}O_{12}$	476.0955	M+ACN+Na	540.1112 m/z calculator	3	N/A
HMDB0039494	Benzoylmalic acid	$C_{11}H_{10}O_6$	238.0477	2M+ACN+Na	540.1112 m/z calculator	3	N/A
HMDB0037452	Diosmetin 7-O-beta-D-glucuronopyranoside	$C_{22}H_{20}O_{12}$	476.0955	M+ACN+Na	540.1112 m/z calculator	3	N/A

Appendix W – Human metabolome database (HMDB) search for metabolite 508.1185 (Section 9.9)



HMDB Browse Search Downloads About Contact Us Search metabolites Search

Search Results

Download Results As CSV

MS search for 508.1185 m/z

Delta = (abs(query mass - adduct mass)/adduct mass)*1000000

Show 10 entries Search

Compound	Name	Formula	Monoisotopic Mass	Adduct	Adduct M/Z	Delta (ppm)	CCS
HMDB0256909	(4-Chlorophenyl) (1R)-6-chloro-1-(4-methoxyphenyl)-1,3,4,9-tetrahydropyrido[3,4-b]indole-2-carboxylate	$C_{25}H_{20}Cl_2N_2O_3$	466.0851	M+ACN+H	508.1189 m/z calculator	1	N/A
HMDB0255103	N-Chlorothylnitrosourea sarcosinamide	$C_6H_{11}ClN_4O_3$	222.0520	2M+ACN+Na	508.1197 m/z calculator	2	N/A
HMDB0257494	Sarcnu	$C_6H_{11}ClN_4O_3$	222.0520	2M+ACN+Na	508.1197 m/z calculator	2	N/A
HMDB0302006	5-Carboxypyranopelargonidin 3-O-beta-glucopyranoside	$C_{24}H_{21}O_{12}$	501.1033	M+Li	508.1193 m/z calculator	2	N/A
HMDB0251179	Dibutyryl cyclic GMP	$C_{18}H_{24}N_5O_9P$	485.1312	M+Na	508.1204 m/z calculator	4	206.2675

Showing 1 to 5 of 5 entries Previous 1 Next