Enhancing multi-modal metabolome and microbiome analysis with modern data containers

Vilhelm Suksi, 41856

[vsuksi@abo.fi](mailto:vsuksi@abo.fi)

Biochemistry major, bioscience program at Åbo Akademi

Supervisor: Leo Lahti, Turun Yliopisto

2023



Abstract

Insight in how the microbiome relates to human biology is increasingly data-driven, where new data science methodologies need to account for hierarchical, heterogenous and multi-modal data. The planned contribution presented herein is aimed at improving multi-modal metabolome and microbiome analysis using the modern R/Bioconductor TreeSummarizedExperiment data container and the metabolomics workflow suggested by the Notame R package. The work spans implementation of the Notame workflow in the TreeSummarizedExperiment context, integration with microbial abundance data and a multi-modal analysis for contrast to earlier work. This will extend the functional scope of the R/Bioconductor ecosystem to encompass methods for extracting insight from multi-modal metabolomic and microbiome data in a user-friendly, reproducible workflow. Using the workflow, substantive research efforts will be better equipped to tackle questions relating the microbiome to physiology and pathology alike.

Table of Contents

Abbreviations:

MS – mass spectrometry

# Introduction

With the advent of new experimental techniques, insight in life science has become increasingly reliant on sophisticated data science methodologies. This also holds true for microbiome analysis, where new data science methodologies need to account for hierarchical, heterogenous and multi-modal data. The manipulation, analysis and reproducible reporting of such data is well developed in the R/Bioconductor ecosystem, focused on high-quality open research software for life science. The R/Bioconductor ecosystem can be conceptualized as consisting of data containers, R packages and a community of users and developers, who contribute to the ecosystem in an interoperable and modular fashion. The Bioconductor package repository delivers releases consisting of a set of compatible R package versions intended for compatibility only with a certain version of R, allowing for rigorous and reproducible analysis.

The notion of data containers arises from the need to organize biological data including assay matrices and meta-data such as sample descriptions and feature annotations into a single instance, facilitating the development and usage of complex analysis workflows. For example, it is possible to exclude a sample from both the meta-data and assay data in one operation, keeping the meta-data and assay data synchronized. In Bioconductor, the SummarizedExperiment family of classes provides data container solutions for various research needs. In microbiome research, the TreeSummarizedExperiment derivative of SummarizedExperiment allows for storing taxonomical information as a hierarchical tree structure. TreeSummarizedExperiment and other SummarizedExperiment derivatives also come with functions for making efficient use of the data structure for the research at hand.

Orchestration of microbiome research in the R/Bioconductor ecosystem using TreeSummarizedExperiment lineage of containers has been explored thoroughly, including basic data manipulation, transformation, exploration and quality control, taxonomic-focused tasks and machine learning. Many of the microbiome analysis tools are implemented by the mia R package, but metabolomics support is underdeveloped, as evidenced by the lack of metabolomics packages that support the TreeSummarizedExperiment lineage of containers. The Notame package, detailed in "the Metabolomics Data Processing and Data Analysis—Current Best Practices" special issue of the Metabolites journal presents a contemporary analysis workflow for LC-MS research. Few of the analysis steps detailed therein are available in the Bioconductor/R ecosystem. The maplet package, which has been included on/off in the Bioconductor repository, supports the TreeSummarizedExperiment lineage of containers and provides some relevant functionalities. Shortcomings include MS data quality control, drift correction, retention time-incorporating metabolite clustering, some univariate methods and multivariate models for different downstream analyses. As such, maplet is not sufficient to implement the Notame workflow. An overview of multi-omics microbiome research reflects this lack of metabolomics support in R/Bioconductor, showing extensive use of ad hoc scripting to manage all aspects of the analyses.

The proposed work aims to streamline multi-modal metabolome and microbiome analyses by implementing metabolomics functionalities as per the Notame package using the modern TreeSummarizedExperiment container. The Notame-inspired workflow is showcased in a multi-modal metabolome and microbiome analysis. Finally, the multi-modal analysis workflow is reviewed and contrasted with alternative data analysis approaches. The workflow is expected to benefit microbiome research efforts, ultimately knowledge of biological functions at large.

*Bakgrund/Introduktion:*Beskrivning av forskningsområdet samt det specifika området som handlas i projektet. Beskriv även de brister i information som ditt projekt syftar till att lösa.

# Research objectives

The planned work presented herein aims to overcome the above shortcomings in R/Bioconductor metabolomics support by using the modern TreeSummarizedExperiment container to implement the Notame workflow. Capitalizing on recent advances in R/Bioconductor data analysis, can complex multi-modal microbiome research be more streamlined? How does using the modern TreeSummarizedExperiment container compare in terms of reproducibility and reportability? What about the margin for user error? And execution time?

Much of what matters here can be summed up in terms of friction at the level of an individual researcher and science collective enterprise. For the individual researcher, friction can be reduced by minimizing the need for temporary variables, performing operations in a single action across assays, and allowing for changes in the analysis workflow without having to rebuild the code around the change. Such simplification of code further reduces friction in the form of easier developing, commenting and documentation. The shorter execution time also reduces friction; a researcher is more likely to re-run the code with different pseudo-random number generator seeds to make sure the findings are robust, for example.

From the perspective of science as a collective enterprise, reducing friction makes the work more accessible. For the research project, this makes for a less arduous process for collaborators and peer-reviewers alike. After publishing, more people grasping and re-running the code results in verification and error-correction at scale. Perhaps a seed was cherry picked, resulting in further wasted research efforts?

In other words, minimizing friction in code is not only in line with good practice, but translates to reproducibility and the advancement of open science at large. This will be made apparent by the planned contribution.

*Målsättningar:*Frågeställningar i projektet, de specifika frågor man söker svar på i detta project

# Research plan

*Forskningsplan:*En utförlig beskrivning av de **material och metoder** som används för att uppnå målsättningarna

## Metabolomics pipeline

### 3.1.1 Data collection

MS-driven metabolomics research requires expertise in analytical chemistry, biochemistry, bioinformatics and data analysis. Sensitive biomolecules are best analysed using LC-MS/MS, where analytes in the liquid phase are separated in a chromatography column and ionized for detection in a tandem MS system, where analytes are further fragmented and ionized for detection with improved specificity. MS instrumentation and analysis conditions can differ substantially among laboratories and experiments, necessitating careful data collection to achieve reproducible results. There are several approaches to data collection, but the general picture is as follows. First, the m/z ratio of ionized analytes is detected, and the resulting raw data undergoes algorithmic peak-picking to sort out the signal from each analyte. Second, the retention time, or the time an analyte takes to pass through the chromatography column, is calculated for each analyte based on peak areas or the raw data from the non-fragmented spectra. Finally, since the retention time for each analyte varies between samples, retention times are aligned to identify corresponding analytes across samples. These data collection steps will be implemented using the MS-DIAL software, (version 3.70), using the parameters detailed in Notame.

The data is now ready for preprocessing, annotation and analysis, where the areas of the peaks reflect the relative abundancies of analytes in the sample, while the identities of the peaks are teased apart by the mass-to-charge ratio spectrum (m/z) of ionized components. Retention time is also included in analyte identification as per the most robust identification level as specified by the Metabolomics Standards Initiative.

### 3.1.2 Drift correction and flagging of low-quality features

LC-MS measurements suffer from systematic intensity drift, the removal of which increases the quality of the data by minimizing variance introduced by the experimental methodology while conserving the biological variance of interest. Quality control (QC) samples included at regular intervals in the LC-MS measurement, consisting of aliquots from each sample, are used to remove the drift by subtracting values according to a smoothed cubic spline fit to QC samples by injection order. Features are temporarily log-transformed to better meet the assumptions of the smoothed cubic spline model. Low-quality features, as identified by detection in less than 70% of the QC samples or having an RSD < 0.2 and D-ratio < 0.4 (Broadhurst), are flagged and monitored by visualizations before and after drift correction. The D-ratio compares the standard deviation of the QC samples to relative to the standard deviation of the samples for each feature. Features identified as low-quality after drift correction are removed for downstream analysis. These steps will be implemented as per the Notame package.

### 3.1.2 Quality control

Linear models relating each feature to injection order are fit to visualize the effect of drift correction on individual features by drawing histograms of the p-values for the regression coefficient of the models. The p-values should follow a uniform distribution represented by a horizontal line under the null hypothesis, namely that p-values across features are normally distributed. This would indicate that systematic drift has been removed. Such histograms are also drawn for QC samples and biological samples. To visualize the effect of drift correction, these histograms are drawn before and after drift correction, as is the case with subsequent quality control visualizations. For example, a comparison of histograms for QC samples shows how the p-values of the features tend towards zero before drift correction, whereas after drift correction the p-values tend towards one. This is because before drift correction, the intensities of the features do not follow a normal distribution because of systemic drift. In other words, we can reject the null hypothesis, namely that sample injection order has no effect on the features’ intensities. After drift correction, the within-group variance of features in QC samples is very small because the predictor, injection order, is not associated with the response, intensity. This indicates that drift correction was successful.

To visualize systematic drift in global feature intensities across samples, boxplots representing the distribution of feature intensities in each sample are drawn. The median is represented by a line, with the interquartile range as boxes and whiskers at values max 1.5x of the interquartile range. Drawing these boxplots before and after drift correction typically shows a systematic decrease in signal intensity as a function of injection order.

Features are then normalized by subtraction of mean and division by standard deviation. This allows for visualization of Euclidean pairwise distances between features across samples using a density plot. Drawing such density plots before and after drift correction hopefully shows how features’ intensities are more similar after drift correction, especially the for the QC samples which should group independently of the biological samples.

A dimensionality reduction technique such as PCA or UMAP is then applied to visualize patterns in the data according to study group and QC sample membership. Trends in the biological samples may not be apparent before or after drift correction, but the QC samples should group tightly after drift correction. Samples can also be colored by gradient according to injection order, where after drift correction, any trends should be dissipated. If too many samples make the dimensionality reduction plot hard to interpret, hexbin versions are drawn such that each hexagon is colored by the mean of the injection orders of the points in the hexagon.

Finally, hierarchical clustering using Ward’s criterion on Euclidean distances between samples is used to visualize the sample clusters in a dendrogram, where the QC-samples should cluster together early after drift correction. Using the same clustering methodology, a heatmap is drawn to represent pairwise distances between sample clusters on the axes.

All of the above quality control steps will be implemented as per the Notame package.

### 3.1.3 Metaboset to TreeSummarizedExperiment

Abstraction, in general, is the process of paring something down to a set of essential elements for the work at hand. Data containers meet this description in simplifying the representation of data, while hiding its complexities and associated operations. For example, instead of storing meta-data in a separate table and accessed by extensive scripting, meta-data is stored in the same instance and accessed by user-friendly operations.

Bioinformatics projects typically require a data structure with a count matrix, sample descriptions and annotation functionality. The MetaboSet container used for the above parts of the analysis workflow is Bioconductor base package ExpressionSet container derivative. ExpressionSet was designed for array-based experiments and can store only one count matrix per object. This results in back-and-forth or creation of new MetaboSet objects to handle transformations and other data manipulation steps, complicating the workflow.

Although many packages interface with ExpressionSet and are largely compatible with MetaboSet, new developments in the fast-moving field of reproducible computation in the R/Bioconductor ecosystem increasingly leans on the modern SummarizedExperiment family of containers. SummarizedExperiment is also based on the ExpressionSet class, but is more flexible, highly optimized and can store multiple count matrices in a single instance. The TreeSummarizedExperiment derivative adds on the functionality of SummarizedExperiment by facilitating storage of hierarchical structure of the data as per the phyloseq package for exploring microbiome profiles. Although microbiome analysis is a downstream concern, storing the hierarchical structure of data can also be used to track clustering of samples and features in metabolomics analysis. Moreover, since TreeSummarizedExperiment is derived from SummarizedExperiment via RangedSummarizedExperiment and SingleCellExperiment, it also includes functionality for representing ranges, adding low-dimensional representations, adding alternative features sets, storing data pairings and addition of further metadata fields. To top it off the TreeSummarizedExperiment is compatible with packages that interface with any of the above containers in the SummarizedExperiment family, further extending the functionality and relevance of TreeSummarizedExperiment into the future.

An overview of alternative data containers for metabolomics analysis, chiefly a suite of data containers included in the RForMassSpectrometry initiative, show promise but do not match the flexibility of TreeSummarizedExperiment. The RForMassSpectrometry suite of containers does not support chromatographic data, used for clustering of features and identification of metabolites as per the most robust identification level as specified by the Metabolomics Standards Initiative. Another concern is that the RForMassSpectroMetry suite of containers rely on the legacy eSet data container.

Regarding multi-omics support, the MSexperiment integrative data container from the RForMassSpectrometry initiative shows promise in being based on MultiAssayExperiment, but only one slot is available for storage of data from other modalities. Thus, the MSexperiment container is rejected in favor of MultiAssayExperiment, which allows for differing numbers of samples and features for data from different modalities.

To this end, the MetaboSet instance used in the above parts of the analysis is converted into a TreeSummarizedExperiment instance, which is included in a MultiAssayExperiment instance for downstream multimodal analysis. The converter will be based on the makeSummarizedExperimentFromExpressionSet() function from the SummarizedExperiment package.

### 3.1.4 Imputation, transformation, normalization and scaling

Values missing due to being below an instrument’s LOD are often referred to as “missing not at random”, or MNAR. Missing values caused by processing errors are often referred to as “missing completely at random”, or MCAR, because they are uniformly distributed across the dataset and are not missing directly due to any property of the metabolite or measurement itself. The Notame workflow deals with MNAR and MCAR values by filtering out values that are not detected in > 50% of the samples and > 70% of QC samples. This is a heuristic method that should remove all MCARs but doesn't explicitly deal with the detection threshold and probably removes features that are barely detectable. The starting point for imputation in the Notame workflow thus involves features that are detected in over > 50% of samples, which probably removes features that are at the detection threshold of the instrument. This means that any zero values from below the detection threshold don't impact the imputation.

Imputation is performed using the MAI package, which features a two-step approach; first missing values are classified as MCAR or MNAR, after which random forest imputation is applied separately to predict MCAR and MNAR values. A smaller α parameter value from the model indicates that MCAR values are predominant, as can be expected because of the filtering of values that are not detected in >50% of the samples.

estimates the ratio of

Alpha parameter

Random forest-based imputation outperforms other methods for imputing LC-MS metabolomics data: a comparative study

https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-019-3110-0

Data visualization will inevitably shape interpretation and motivate the next steps of the analysis.

annotation includes counting elements in chemical formulas, converting element counts to chemical formulas, removing elements from chemical formulas, adding elements to chemical formulas, standardizing formulas according to Hill notation, calculating mass from formulas, converting between masses and m/z values, grouping of isotopologue peaks, matching numerical values accepting differences, calculating a normalized dot product, calculating a normalized Euclidian distance, calculating normalized absolute values distance and calculating normalized spectra angle

\*\*How is redundant representation of the same metabolite due to unpredictable adduct behavior and neutral loss formation avoided\*\*

Features are clustered and combined anew based on correlated feature pairs using a specified retention time window and correlation threshold. After this, the dataset is ready for analysis.

## Integration with microbiome analysis

Start with how the microbial data has been analyzed.

\*\*Where does microbiome analysis start in OMA?\*\*

After processing Illumina-sequenced paired-end fastq files that have been split (or “demultiplexed”) by sample. The resulting amplicon sequence variant (ASV) table is a higher-resolution analogue of the traditional OTU table, which records the number of times each exact amplicon sequence variant was observed in each sample.

\*\*What is host trait information in microbiome research?\*\*

Any information pertaining to the subjects (hosts) biological functioning, for example body mass index; in other words, any information that is not from bacteria.

Storing multiple data tables in microbiome analysis is preferrably done as follows:

- Assays are used within the data container for data transformations

- the altExp slot in TreeSE if for alternative versions of the data with the same number of samples, arising from differing taxonomic agglomeration or alternative profiling technologies, for example

- MultiAssayExperiment for multi-omic experiments, allowing for differing numbers of samples and features

- Taxonomic-focused tasks, including assignment and access of taxonomic information, agglomeration by taxonomic rank, data transformation, and picking specific elements for analysis

- Community diversity, including estimation and visualization

- Community similarity, including explained variance, community comparisons by beta diversity analysis, other ordination methods, visualization of the most dominant genus from PCoA

- Community composition, including visualizing taxonomic composition

- Community clustering, including hierarchical clustering, K-means clustering, Dirichlet multinomial mixtures, community detection and biclustering

- Differential abundance, including differential abundance analysis, confounding variables and tree-based methods

- Machine learning, including supervised machine learning and unsupervised machine learning

- Multi-assay analyses, including cross-correlation analysis and multi-omics factor analysis

- Visualization of pre-analysis exploration, diversity estimation and statistical analyses

**What steps were performed for the multi-assay analysis?**

- Cross-correlation analysis, for elucidating the relationship between individual species of bacteria and metabolite concentrations with heatmap visualization.

- Multi-omics factor analysis, which could be seen as a generalization of PCA, for unsupervised multi-omics integration to latent space. The factor loadings (variance explained per factor and assay) was visualized with bar plots

## 3.3 Comparison with earlier multi-modal metabolome and microbiome analyses

# Research schedule

Table and text.

*Tidtabell:*Ett detaljerat schema över projektets tidtabell. Inkludera även det skriftliga avhandlingsarbetet.

# Research synopsis

*Sammanfattning:*En kort beskrivning av de förväntade resultaten samt betydelsen av dessa.

# References

*Litteraturförteckning:*En komplett lista över publikationer som citerats

Textdelen av forskningsplanen skall vara skriven med 1,5 radavstånd, med 12 pt skrift (Times New Roman eller motsvarande), den vänstra marginalen skall vara 4 cm bred och den högra marginalen 2,54 cm och sidorna justerade. Figurtexter kan med fördel skrivas med mindre font så som 10 pt (Times New Roman eller motsvarande). En lämplig längd på forskningsplan kan vara 10-15 sidor, men det finns inga specifika krav på längden; kom ihåg att en välskriven forskningsplan är till stor hjälp vid skrivandet av självaste pro gradu-avhandlingen.