UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE QUÍMICA E BIOQUÍMICA


# Binary Similarity Measures and Mass-Difference Network Analysis as Effective Tools in Metabolomics Data Analysis

Francisco Maria Reis Ventura Rosado Traquete

**Mestrado em Bioquímica**
Bioquímica Médica

Dissertação orientada por:
Professor Doutor António Eduardo do Nascimento Ferreira
Doutora Marta Filomena de Sousa Silva Ferreira

2020

# Acknowledgments

First, I would like to start by thanking both my supervisors, Professors António Ferreira and Marta Sousa Silva for guiding and helping me at every turn throughout this year even amidst the Covid-19 pandemic situation. Moreover, I want to thank Professor António Ferreira for all his help and patience in teaching me all things coding and project management despite my inexperience in the field. I would also like to extend this thanks to all other members of the FT-ICR-MS-Lisboa laboratory group, where I worked for the past year, especially to Marisa Maia that provided me the grapevine MS datasets (two of the three main datasets of this work) and helped me publish my first paper and to group coordinator Professor Carlos Cordeiro and Dr. Ana Marques. This work was supported by 3 different funded projects. Thus, I would like to thank Fundação para a Ciência e a Tecnologia (project FCT PTDC/BAA-MOL/28675/2017), the Portuguese Mass Spectrometry Network (LISBOA-01-0145-FEDER-022125) and the Europe and Union's Horizon 2020 research and innovation programme under grant agreement number 731077 (Project EU_FT-ICR_MS) for funding and making this work possible.

To all my fellow masters' students that were, like me, in the FT-ICR-MS-Lisboa laboratory group and were sharing the same path in the same group throughout the year despite our different works, especially, João Luz for the yeast mass spectrometry data which was one of the three main datasets I used for my work and Margarida Isidro who was my work bench neighbour for the 1$^{st}$ part of the year and helped me run a formula assignment algorithm.

To my friends from my Biochemistry courses, primary and high school (especially João Ferreira, Luis Palmares and Carlota Cardoso) for sharing the highs and lows of each other's situations (and putting up with me while I was writing this work) and Mário Nascimento for helping me with coding advice and computer problems in installing and updating a lot of stuff that made my work easier. And I would like to thank everyone that will read this dissertation for giving my work the time of day.

Finally, to my family for their never ending support across my life and enduring me in the half-year plus of work from home due to the pandemic. During this period, I, my father and my mother had to coordinate ourselves to split the only room in the house suited to work from home and online meetings by respecting and minimizing noise whenever a meeting from any of us three was happening and for that, and much much more I'm grateful. Also, I want to thank my brother for his input and helping me many times with English and sentence construction while I was writing this work.

# Abstract

Metabolomics is an emerging field in systems biology that aims to perform a comprehensive analysis of a biological system's metabolome by identifying and quantifying all its metabolites. Due to their high diversity in concentration, structure and chemical characteristics, this is an extremely complex task which requires high resolution methodologies such as mass spectrometry (MS) or nuclear magnetic resonance (NMR) to provide an approximated overview of the metabolome. These analyses also generate complex data, which, in turn, requires first suitable pre-processing and then pre-treatment to be properly analysed – crucial steps in the workflow that must be pondered and carefully applied. Since there are many factors that significantly affect the metabolome, metabolomics data obtained from different sources and conditions has successfully been used to discriminate samples of biological systems and to find key metabolites supporting that discrimination. The pre-processing of the data generates a 2D-dataset with features (usually *m/z* peaks for MS analysis) on one axis and samples on the other. Subsequent data analysis aims to extract and highlight the significant biological variation between samples over the background variation in the data. Traditional data analysis in metabolomics focuses primarily on the comparison of intensity of the features in the samples rather than on information such as their presence/absence in each sample. However, a major problem of this analysis is the high variability of the intensity data between different samples (even of the same biological system) when analysed in different experimental batches, instruments, pre-processed with different methods or parameters, etc., which leads to a low level of reproducibility. Another bottleneck is the unambiguous structural identification of the metabolites that can be key in discriminating between the studied systems.

The aim of this work was to develop two new approaches for the computational analysis of metabolomics data, in the context of profiling and discrimination of biological samples. As part of this development, a systematic evaluation of their performance when compared to more established methods for selected high-resolution MS datasets was also a major goal.

The first approach is based on the concept of considering only the occurrence of spectral features to construct a binary sample vector encoding feature presence as 1 and absence as 0. The use of such data encoding, followed by the adoption of binary metrics of sample distance, can be used as a pre-treatment method to transform data before the application of unsupervised and supervised methods related to profiling and classification. While using such pre-treatment, called Binary Similarity (BinSim) effectively discards information contained in the metabolite signal intensities, the resulting data has less variability than intensity data and more consistent results on the discrimination of biological systems can be obtained. Furthermore, BinSim greatly simplifies the analysis by skipping most of the peak filtering, and the choice of the missing value imputation, normalization and scaling methods to use. The performance of statistical methods in discriminating the datasets transformed with BinSim was consistently as good as or slightly better than datasets treated with different combinations of traditional, intensity-based, pre-treatments. In the former, features that appeared in one (biomarker-like) or a few of the groups were the most important to build discriminant classifiers, which was markedly different from those computed from datasets treated in traditional ways, emphasizing the new perspective that BinSim offers.

The second approach is based on the construction of a Mass-Difference Network (MDiN) for each sample, using masses as nodes and a set of mass differences derived from common biochemical reactions to establish edges. The information in the network is the possible transformations between the identified metabolites that could happen in a biological context. Results from different network analysis on sample MDiNs were compared using statistical methods to discriminate the samples into

their respective groups. Analysis that focused on node centrality measures, especially their degree, allowed a better discrimination of the samples compared to analysis focused on global network characteristics and was on par with the discrimination achieved in the same datasets treated with more established intensity-based methods, while offering the versatility of other network analysis methods on the sample MDiNs to complement the discrimination.

**Keywords:** Metabolomics; Data Analysis; Data Pre-Treatment; Statistical Analysis; Network Analysis.

# Resumo

A metabolómica é um campo emergente na biologia de sistemas que visa realizar uma análise global do metaboloma de um sistema biológico ao identificar e quantificar todos os seus metabolitos. Devido à alta diversidade na concentração, estrutura e caraterísticas químicas dos metabolitos, esta é uma tarefa complexa que requer a utilização de metodologias de alta resolução como espetrometria de massa (MS, *Mass Spectrometry*) ou ressonância magnética nuclear (NMR, *Nuclear Magnetic Resonance*). Apesar destes métodos não identificarem todos os metabolitos presentes num sistema (devido a limitações na gama dinâmica dos instrumentos utilizados e a preferência de cada abordagem para certos tipos de metabolitos), estes oferecem uma visão aproximada do metaboloma completo. A complexidade dos dados obtidos requerem primeiro um pré-processamento e depois um pré-tratamento adequados para extrair a informação presente. Assim, ambas estas etapas são cruciais no fluxo normal de trabalho em metabolómica e, como tal, devem ser ponderados e escolhidos cuidadosamente. Sendo que muitos factores afectam significativamente o metaboloma de um sistema biológico, dados de metabolómica têm sido usados com sucesso na discriminação de amostras de diferentes sistemas e para a identificação de metabolitos chave que suportam esta discriminação, através de variados métodos estatísticos. O pré-processamento gera um conjunto de dados 2D com caraterísticas (normalmente picos *m/z* em análise MS) num eixo e amostras no outro. Na formação destes dados surgem valores em falta – amostras que não têm caraterísticas presentes noutras amostras. Sendo que diversos métodos estatísticos não suportam a existência de valores em falta, são aplicados métodos de filtração de picos para reduzir o número destes; seguidos da aplicação de um método de imputação dos valores em falta que restam após filtração. A análise de dados procede com a aplicação de pré-tratamentos que podem ser divididos em três sub-categorias – normalizações (incluído às vezes no pré-processamento), transformações e *scaling*. Uma combinação de métodos destas categorias é utilizado para extrair e destacar a variação biológica significativa entre as amostras. Contudo, todos estes métodos tradicionais destacam os padrões de intensidades entre as caraterísticas em detrimento de outras informações importantes no contexto da metabolómica como a presença e ausência destas nas amostras. Um possível problema desta utilização para a análise de dados de metabolómica é a intensidade ter uma variabilidade elevada mesmo entre amostras do mesmo grupo. Esta variabilidade aumenta ainda mais quando analisadas em lotes experimentais diferentes, instrumentos diferentes com preparação de amostras diferentes, métodos ou parâmetros de pré-processamento diferentes, entre outros, originando uma baixa reprodutibilidade dos dados. A dificuldade da identificação estrutural inequívoca dos metabolitos chave na discriminação de grupos coloca-se como outro problema na análise de dados.

O objetivo deste trabalho foi desenvolver duas novas abordagens para a análise computacional de dados de metabolómica, no contexto da caraterização e discriminação de amostras biológicas. Estes tratamentos descartam a informação de sinais da intensidade predominantemente utilizada pelos métodos de tratamento estabelecidos, de forma a evitar a elevada variabilidade desta, concentrando-se noutros aspectos dos dados, o que deve oferecer uma nova perspetiva sobre estes. Como parte deste desenvolvimento, uma avaliação sistemática da performance destes tratamentos para um set seleccionado de conjuntos de dados de MS de alta resolução foi outro objetivo principal do trabalho. Três combinações de métodos de pré-tratamento tradicionais foram comparadas na análise de resultados: 1) Pareto *scaling*; 2) Normalização por uma caraterística de referência e Pareto *scaling*; 3) Normalização, transformação logarítmica generalizada e Pareto *scaling*. Foram utilizados dois conjuntos de dados metabolómica de videira (*Vitis*) contendo 3 réplicas de 11 variedades cada – um obtido por *electrospray* em modo negativo de ionização ($ESI^-$) e outro em modo positivo de ionização

($ESI^+$) – e um conjunto de dados de 3 réplicas de 5 estirpes de leveduras, utilizando ou a lista de picos *m/z* ou fórmulas atribuídas aos picos (quando possível) como caraterísticas.

Semelhança binária (BinSim, *Binary Similarity*) é a primeira abordagem desenvolvida, sendo baseada no conceito de considerar exclusivamente a ocorrência de características espectrais. A ideia é que o conjunto de metabolitos identificados por métodos de alta resolução é caraterístico dos diferentes sistemas e pode ser utilizado para os discriminar, conseguindo obter resultados mais consistentes devido à menor variabilidade da identificação de metabolitos em relação à informação dos sinais de intensidade (descartada). Este método consiste na construção de um vector binário para cada amostra que codifica a presença de uma caraterística como 1 e ausência como 0 que pode ser usado para transformar os dados antes da aplicação de métodos estatísticos para caraterizar e classificar amostras. A simplicidade deste método encontra-se no facto de que necessita (e até prefere) pouca filtração de picos e de que salta a escolha dos métodos de imputação de valores em falta e combinação de normalizações, transformações e *scaling* a usar, acelerando a análise de dados. Utilizando métodos de agrupamento de amostras (não supervisionados) e modelos de classificação (supervisionados), a qualidade da discriminação das amostras nos seus respetivos grupos em dados transformados com BinSim foi consistentemente semelhante ou ligeiramente melhor do que quando tratados com tratamentos baseados em intensidade, levando, quase sempre, à melhor ou segunda melhor discriminação (dos 4 tratamentos comparados). Uma discriminação perfeita foi atingida nos dados da levedura em todos os métodos estatísticos usados; nos dados da videira, métodos não supervisionados agruparam corretamente cerca de metade dos grupos e os métodos de classificação supervisionados (*Random Forest* e *Partial Least Squares - Discrimination* Analysis, PLS-DA) previram com cerca de 80% de precisão os grupos das amostras. Para observar se esta discriminação era obtida por informação menos usada pelos métodos tradicionais, retirou-se os 2% de caraterísticas consideradas mais importantes para construir os modelos de classificação de *Random Forest* e de PLS-DA dos dados tratados das diferentes formas. Este conjunto de caraterísticas importantes nos dados tratados com o BinSim é muito distinto, tendo um grande número de caraterísticas apenas presentes neste (73,5% em média) em comparação com os conjuntos obtidos dos modelos construídos de dados tratados de forma diferente. Além disso, estas apareciam num pequeno número de grupos (em comparação com os restantes casos), ou seja, eram caraterísticas com muitos valores em falta e que, por isso, são muitas vezes filtradas. Nas caraterísticas importantes para construir modelos *Random Forest* nos dados da levedura, esta tendência foi mais acentuada com características importantes a aparecerem predominantemente apenas num grupo, ou seja, a atuarem como biomarcadores desse grupo nos dados estudados. Conclui-se, então, que a informação obtida por este tratamento é distinta em relação aos outros tratamentos baseados em intensidade no fluxo de trabalho da metabolómica.

A segunda abordagem consiste em construir uma rede de diferença de massas (MDiN, *Mass-Difference Network*) para cada amostra de um conjunto de dados e discriminar estas pela comparação das suas caraterísticas. MDiN foi um conceito originalmente desenvolvido por Breitling et al. que usa a lista de massas de dados de metabolómica como vértices/nós na rede e um conjunto de diferença de massas que estabelece arestas entre os vértices com diferenças que se enquadram nesse conjunto. Cada diferença de massa (MDB, *Mass-Difference-based Building block*) corresponde a uma diferença na fórmula elementar de um metabolito após a ocorrência de uma reação bioquímica comum (enzimática ou não enzimática). Assim, para cada amostra, forma-se uma rede semelhante, conceptualmente, às redes metabólicas mas gerada apenas pela informação do conjunto de dados. Cada rede tem a informação das possíveis transformações biologicamente significativas entre os metabolitos presentes que podem ocorrer num contexto biológico, enfatizando, a presença destas interações sobre a intensidade de cada caraterística. Apesar da complexidade, as redes construídas podem ser analisadas

e comparadas de inúmeras formas diferentes, mostrando ter uma grande versatilidade no modo como podem ser usadas, sendo esta a principal vantagem do método. As redes construídas foram analisadas por diferentes métodos de análise de redes: focadas na centralidade dos nós (grau, intermediação e proximidade), ou nas caraterísticas globais das redes como no número de vezes que cada MDB foi usada para estabelecer arestas e na topologia da rede (usando o GCD-11, *Graphlet Correlation Distance using 11 graphlet orbits*). Comparando os resultados das análises por variados métodos estatísticos, a análise da centralidade dos nós, especificamente do grau, permitiu a melhor discriminação das amostras nos seus grupos. Resultados indicaram que a análise de cada nó pelas suas possíveis interações permite uma discriminação dos grupos semelhante à alcançada quando os dados são tratados com os tratamentos tradicionais mencionados anteriormente. Contudo, a análise das caraterísticas globais das redes deu indicações que poderá demonstrar diferenças importantes e biologicamente significativas gerais do metabolismo ao nível da proeminência de diferentes tipos de reações no sistema.

Conclui-se, então, que ambas as abordagens são viáveis na análise de dados de metabolómica, extraindo informação que pode ser utilizada para discriminar as amostras dos conjuntos de dados. A sua diferente perspetiva também permite que sejam usados numa análise que complemente a de outros tratamentos. Ainda mais, como estes tratamentos enfatizam informação com menos variabilidade do que a intensidade, têm um grande potencial na análise de múltiplos conjuntos de dados obtidos com diferentes instrumentos, laboratórios, entre outras hipóteses dos mesmos grupos biológicos, abrindo portas para estudos futuros que se possam focar na viabilidade destas estratégias neste contexto.

**Palavras-Chave:** Metabolómica; Análise de Dados; Tratamento de Dados; Análise Estatística; Análise de Redes.

# Index

# List of Figures

# List of Tables

# List of Abbreviations

In alphabetical order:

**ANOVA** – Analysis of Variance

**BinSim** – Binary Similarity pre-treatment

**CID** – Collision Induced Dissociation

**CV** – Cross-Validation

**DD** – Discrimination Distance

**DR** – Decision Rule

**ECD** – Electron Capture Dissociation

**ESI** – Electrospray Ionization

**FDR** – False Discovery Rate

**FT-ICR-MS** – Fourier Transform Ion Cyclotron Resonance Mass Spectrometry

**G** – Generalized logarithmic transformation

**GC** – Gas Chromatography

**GCD-11** – Graphlet Correlation Distance using 11 graphlet orbits

**GCM** – Graphlet Correlation Matrix

**GD** – Grapevine Datasets

**Glog** – Generalized Logarithmic Transformation

**HCA** – Hierarchical Clustering Analysis

**HILIC** – Hydrophilic Interaction Liquid Chromatography

**HMDB** – Human Metabolome Database

**HPLC** – High Performance/Pressure Liquid Chromatography

**KMD** – Kendrick Mass Defect

**kNN** – k-Nearest Neighbours

**LC** – Liquid Chromatography

**LOOCV** – Leave-One-Out Cross-Validation

**LV** – Latent Variable

***m/z*** – Mass over Charge

**MAR** – Missed At Random

**MCAR** – Missed Completely At Random

**MDB** – Mass-Difference based Building blocks

**MDB Inf. –** MDB (Mass-Difference-based Building block) Influence

**MDiN** – Mass-Difference Networks

**MNAR** – Missed Not At Random

**MS** – Mass Spectrometry

**N** – Normalization by leucine encephalin feature pre-treatment

**NGP** – Normalization by leucine enkephalin, Generalized logarithmic transformation and Pareto scaling pre-treatment

**NIPALS –** Nonlinear Iterative Partial Least Squares (or Projection to Latent Structures)

**NMR** – Nuclear Magnetic Resonance

**NP** – Normalization by leucine encephalin followed by Pareto scaling pre-treatment

**P** – Pareto scaling

**PC** – Principal Component

**PCA** – Principal Component Analysis

**PLS** – Partial Least Squares (or Projection to Latent Structures)

**PLS-DA** – Partial Least Squares (or Projection to Latent Structures) – Discriminant Analysis

**PQN** – Probabilistic Quotient Normalization

**PRESS** – Predictive Residual Sum of Squares

**QRILC** – Quantile Regression Imputation of Left-Censored data

**RF** – Random Forest

**RP** – Reverse Phase

**S/N** – Signal-to-Noise ratio

**SS** – residual Sum of Squares

**T-ReX** – Time aligned Region complete eXtraction

**UPGMA** – Unweighted Pair Group Method with Arithmetic mean

**UPLC** – Ultra Performance Liquid Chromatography

**VIVC** – *Vitis* International Variety Catalogue

**VIP** – Variable Importance/Influence in Projection

**YD** – Yeast Dataset

**YFD** – Yeast Formula Dataset

**YMDB** – Yeast Metabolome Database

$\mathbf{y_{pred}}$ – Predicted response variable of a test sample from PLS Regression

# 1. Introduction

Metabolomics is an emerging field in systems biology which can be defined as a comprehensive analysis aiming to identify and quantify all the metabolites of a biological system [1,2,3]. Metabolites are endogenous and exogenous small molecules (<1500 Da) whose ensemble constitutes the system's metabolome. They are the end-product of all cellular processes and are, consequently, informative of the biochemical activity of the system. Moreover, they can be quite diverse at a structural and physical-chemical level and include peptides, amino acids, nucleic acids, inorganic species, cofactors, and hormones, among others [4–8]. Therefore, specific metabolites can be representative of different phenotypes of a biological system and, as such, the metabolome can be a source of phenotypic biomarkers [5,9,10]. This leads to the diverse applications of metabolomics – from studying human diseases [11], plants [12] and bacteria [13] to drug discovery [14] and others [15].

Metabolomics experiments can be divided into two categories: *targeted* and *untargeted* metabolomics. In *targeted* metabolomics, the focus is put on a particular set of characterized and annotated metabolites, classes of metabolites, or involved in specific metabolic pathways in a hypothesis-driven experiment. In *untargeted* metabolomics, the focus is on getting a global picture of a system with the ultimate ambitious objective of identifying and characterizing all metabolites [2,16]. This endeavour leads to very complex metabolomics data, which requires robust and scalable computational and statistical tools to treat and extract meaningful information from them, as will be explained later in more detail.

## 1.1 Mass Spectrometry Techniques for Data Acquisition

Due to the desired holistic analysis of a complex system, the two main analytical methodologies used in metabolomics experiments are nuclear magnetic resonance (NMR) and mass spectrometry (MS). Although less sensitive, NMR is a non-invasive analytical technique that allows the identification and quantification of metabolites, as well as the determination of their chemical structures, conformations and absolute stereochemistry. MS has a superior sensitivity and dynamic range as well as a high throughput, allowing the detection of hundreds to tens of thousands of compounds from one single biological sample (reviewed in [17]).

In this work, the focus will be on mass spectrometry-based metabolomics data. This is a technique that detects and measures the relative quantities (coded as intensities) of ionized molecules, separated based on their *m/z* (mass to charge ratio), in a biological sample. A sample must be previously ionized (different ionization methods such as electrospray can be used) in a mass spectrometer [4,18,19]. $MS^n$ is the process through which multiple (*n*) stages of MS are carried out in succession with selection and fragmentation of ions from the previous MS stage occurring in-between them. Common fragmentation methods are collision induced dissociation (CID) and electron capture dissociation (ECD). The pattern of ion fragmentation helps the elucidation of the molecular formula and structure of the precursor ion, which is the main objective of $MS^n$ [20]. However, given the sheer complexity of the metabolome – both with the amount of metabolites and their concentration range present in samples – the holistic untargeted analysis becomes a very challenging endeavour, with many difficulties that have to be overcome or minimized.

A popular way to address some of the issues caused by metabolome complexity is to couple the MS analysis with separation techniques such as HPLC (High Performance/Pressure Liquid Chromatography) and GC (Gas Chromatography) – LC-MS and GC-MS for example – to further improve analytical performance [4,9,21]. Their advantages and disadvantages have been well

described in [5,18,19]. This approach is particularly useful in untargeted metabolomics approaches. The objective of using these chromatographic separation techniques is to make the sample mixtures less complex by separating them based on a certain characteristic such as metabolite polarity, so every MS analysis will only detect a lower number of metabolites. This facilitates the detection of low-concentration metabolites that might not be detected due to the limits of the dynamic range and it helps distinguish compounds with very similar *m/z* that are not easily discriminated due to the resolution limits of the instruments used. A common disadvantage to all these hyphenated methods is the increase of time duration of the analysis by adding the chromatographic step. GC-MS has high reproducibility and low cost, but it can only analyse and detect volatile compounds (both non-volatile and thermolabile compounds are not detected), which hinders the objective of untargeted metabolomics experiments. Additionally, it may also require a lengthy sample preparation with sample chemical derivatization (to provide volatility to most of the metabolites so that they are able to be detected). This chemical derivatization may lead to the formation of by-products that were not present in the original sample, as well as the degradation of some metabolites, further influencing the results and the reproducibility of the derivatization [5,18,19]. On the other hand, LC separation followed by electrospray ionization (ESI) in LC-MS does not have these limitations [18]. The columns used in HPLC are usually reverse phase columns ($C_8$ or $C_{18}$ are the most common); this type of LC-MS is better suited to the analysis of non-polar or semi-polar compounds since polar and ionic compounds tend to elute with the solvent front (no separation in time is achieved). Thus, complementary analysis using separations like hydrophilic interaction liquid chromatography (HILIC) can help increase the coverage of the analysis [18,19]. In UPLC (Ultra Performance Liquid Chromatography) porous particles with $< 2$ μm diameter are used and this can help enhance the resolution as well as the sensitivity of the chromatographic separation in comparison to HPLC [5,18].

Fourier Transform Ion Cyclotron Resonance Mass Spectrometry (FT-ICR-MS) is an extreme resolution (over 1,000,000) mass spectrometry technique which has the highest *m/z* determination accuracy with an average error lower than 1 ppm [8,22]. This technique stores ions that travel in a circular trajectory in which the frequency of the rotation is characteristic of the *m/z* and the magnet of the instrument [23]. A disadvantage of FT-ICR-MS is the relatively slow time of acquisition of each transient, which complicates coupling with prior separation methods such as LC-MS [24]. The accumulation of transients also grants a very high dynamic range and sensitivity (limit of detection for compounds) for detection of compounds. The mass accuracy and resolution of this technique allows the identification of most metabolites based on their *m/z* alone without the need of a coupled separation method (simpler sample preparation), which enables unambiguous molecular formula assignments in low mass metabolites (up to $\approx$ 500 Da), [22]. Besides these alluring characteristics, since ions are stored, $MS^n$ experiments are easily done from FT-ICR-MS instruments that can apply an array of different fragmentation methods [24].

## 1.2 Challenges in Metabolomics Experiments

Despite the advanced techniques employed, metabolomics experiments still face a lot of serious challenges due to the extreme sensitivity of their data: a good part of them can be addressed during the experiment or corrected during data analysis but some remain intrinsic to the data and must be considered when analysing results.

At an individual level, every biological system is unique and will have a unique metabolome. Therefore, even two identical systems will have minor differences in their metabolome. This uninduced biological variation will lead to inherent variability in the data [7,25]. The metabolome is

extremely sensitive to experimental manipulation and environmental factors (slight changes in pH or growth medium, stress, temperature, among others), leading to considerable changes in metabolite concentration especially in secondary metabolites. Secondary metabolites are metabolites that are not directly essential for cell growth and survival but are important in the interactions of the organism with its environment for its continued survival, for example, acting as defence or resistance compounds or as signals [26]. Metabolome changes occur in a very short time span due to the small half-life of metabolites. Therefore, reducing this inter-individual variability between biological replicates is a goal that must be kept in mind when preparing the experimental protocol [7,22,27]. The wide range of metabolites in an extensive array of concentrations also hinders the detection of the least concentrated metabolites, which can be biologically significant (molecules such as signals) despite the high dynamic range of the instruments used. Furthermore, the high degree of difficulty in achieving unambiguous structural identification of the detected metabolites is a major bottleneck [28,29]. This is due to the variety of possible metabolites that complicate unambiguous formula assignments without extreme mass accuracy and resolution and to the $MS^n$ fragmentation not generating a consensus fragmentation pattern [30], coupled with the lack of reference spectra due to metabolomics being a still relatively new "omics" field [7,28,29].

A metabolomics dataset usually has a very high number of features (thousands) that represent the metabolites in comparison with the number of samples. A characteristic of these datasets that must be considered when applying statistical methods is that a lot of these features are highly correlated due to their relations, for being in the same metabolic pathways for example – the curse of dimensionality [31,32]. The variability present in each feature comes from the induced biological variation (that is, the intended variation to observe and analyse in the experiment) and the uninduced biological variation previously mentioned, which encompasses all the technical variation due to either the protocol or instrumental variations. This variation can lead to large intensity fluctuations not correlated with the biological response, which means that intensity data is highly variable. The different features in the dataset are also present in many different magnitudes. Many multivariate statistical methods will give more weight to higher magnitude features with larger absolute changes in concentration rather than low concentration metabolites. However, the biological importance of a metabolite does not depend on the concentration of metabolites. For example, signal molecules usually have very low concentrations and can be fundamental in characterizing two different phenotypes. Finally, metabolomics data is usually heteroscedastic (the variability/variance of its features is not constant), while many different statistical methods assume the data is homoscedastic [25].

Taking these issues into account, an objective of untargeted metabolomics experiments is the identification of some key features, characteristics and trends in the data that can help define and discriminate the studied systems. To achieve this goal, robust computational and statistical tools to treat and extract information from the data have been developed and applied (data analysis), [7]. However, many of the currently applied methods have been adapted to the metabolomics framework from previously established "omics", especially transcriptomics and proteomics [2,30] and, consequently, are not perfectly tailored to metabolomics data. In the next sections, a workflow of metabolomics data analysis will be presented, with special focus on the data pre-treatment that aims to eliminate the impact of the uninduced biological variation while maximizing the information from the induced biological variation.

## 1.3 Metabolomics Data Analysis

The raw spectral data goes through an extensive and time-consuming analysis. This analysis has the aforementioned objective of extracting information from the raw spectra and can be divided into pre-processing, pre-treatment and data analysis [25,31], which will be further explained in the next sections.

However, besides these steps, proper statistical planning prior to the execution of the experiment is of the utmost importance to take the most advantage of different statistical methods (examples mentioned in section 1.4), [31,33]. This planning must consider both the number of samples and replicates to use (depending on the aim of the work) and the procedure to obtain data relevant to the scientific question. It is important to randomise any step of the procedure that introduces bias in the results – for example, the order of sample analysis, who is handling the samples, where the samples are prepared or stored, etc. Another critical point is the knowledge of how the used analytical platforms work [18,33].

### 1.3.1 Data Pre-Processing and Feature Annotation

The main differences between analysis of MS and NMR data is in the pre-processing stage, since the initial steps of processing spectral data are unique to each method, due to the nature of the raw data obtained [34]. The product of the pre-processing stage is a 2D data matrix with features represented in one dimension and samples in the other. Therefore, these steps aim to improve the quality of the signal while reducing bias in the raw data, thus facilitating the retrieval of useful information [4,35,36]. Mass-spectrometry raw data processing includes spectra deconvolution, correction of the baseline and noise filtering, peak detection, or peak picking, peak alignment, and gap filling (if needed). In cases where MS is coupled with liquid or gas chromatography, retention time correction can also be employed [35,37]. Correction of the baseline is a noise filtering procedure used to remove low-intensity artefacts (born of instrumental or experimental noise) by estimating the baseline shape and subtracting it from the raw signal [4,36,38]. The peak detection step aims to identify and quantify all features (ions) in the spectra while trying to avoid false positives, using, for example, peak-based methods (detect ‘peak-like shapes’) or binning-based methods (split spectra into small *m/z* intervals) [4,36,39]. A common way of peak picking is using the signal-to-noise ratio (S/N) to filter the detected peaks. Spectral or peak alignment (before or after peak detection) is an essential step in multiple sample studies and intends to correct the slight shifts in *m/z* and retention time (if applicable) that exist between different samples [4,39]. Some common methods of spectral and peak alignment are well discussed by Alonso et al. [4]. These processing steps are normally performed by commercially or freely available software such as XCMS [40] and MZMine [41], simplifying user input.

NMR pre-processing includes chemical shift calibration, phasing and baseline correction, specific to NMR spectra followed by steps akin to peak detection, filtering and spectral alignment that are closer to the MS processing steps already mentioned. An in-depth review of the processing steps is featured in Emwas et al. [42].

Gap filling or missing value imputation is a bridge between the processing and pre-treatment step of the metabolomics data analysis workflow. Missing values arise in metabolomics datasets after peak or spectral alignments when a feature detected in a sample is not detected in another. The imputation consists of replacing those missing values by a certain value to facilitate the different kinds of data analysis performed downstream in the workflow (which do not adequately account for the presence of missing values) while maintaining the overall structure of the data. Missing values can be values missed completely at random (MCAR), missed at random (MAR) and missed not at random (MNAR),

[43,44]. For metabolomics data, MCAR missing values can be due to instrumental factors, such as stochastic fluctuations during data acquisition, while MAR missing values can be, for example, due to incorrect peak detection by the chosen algorithm [43]. In other words, these kinds of missing values are due to stochastic errors in the metabolomics workflow that lead to the lack of detection of features that are above the detection limit of the high-resolution method applied. MNAR missing values are, however, caused by some features of the data being under the detection limit (accounting for the baseline correction and noise filtering performed), [44] – low concentration or absent features in samples (can have a biological meaning). There are strategies to reduce the number of values that have to be imputed by filtering the number of features used in further analysis. These can be done by imposing a maximum percentage of missing values in any feature, filtering those that exceed this limit [43]. This threshold can be set on the overall dataset or it can be set on a group of technical replicates.

As far as the strategies to impute the missing values are concerned, there are a plethora of options available; some favour the imputation of MNAR and others MCAR/MAR [43]. As for methods that favour the imputation of MNAR, considering all missing values as such, some strategies replace all missing values by a constant small value that is usually either zero or half of the minimum intensity value on the dataset. Another strategy known as Quantile Regression Imputation of Left-Censored data (QRILC) [45] randomly imputes values from a small-value distribution (estimated by quantile regression), [43,44]. As for methods that favour the imputation of MAR/MCAR considering all missing values as such, these are usually replaced by the mean or median of all values in the corresponding feature or by using more complex methods like kNN (or k-Nearest Neighbours) imputation [46], Random Forest imputation [47], among many other methods that can be applied [43,44]. As always, the choice of the method has a considerable effect on the data matrix and on the results of further statistical analysis. Wei et al. [43] recommends the use of Random Forest imputation for MCAR/MAR and QRILC for MNAR, while Guida et al. [44] suggests that the type of missing value imputation to be used depends on the data analysis method that will be used subsequently, meaning that there is no single one-size-fits-all "best" method.

After peak alignment and obtaining a 2D dataset, feature annotation is an optional step that can be applied at any time in the workflow. Feature annotation is the annotation of *m/z* values with formulas or metabolite "names". The dataset can sometimes be filtered to only include features that were annotated. Annotation can be done by comparing *m/z* values with those of a database such as Chemspider ([48], http://www.chemspider.com/) or the Human Metabolome Database (HMDB, [49], https://hmdb.ca/), or by using algorithms that find suitable formulas that can be assigned to an *m/z* peak such as the SmartFormula algorithm of MetaboScape 4.0 (Brüker Daltonics). These algorithms tend to use the 7 golden rules proposed by Kind and Fiehn [50] that set some guidelines to restrict the possible formulas to assign to an *m/z* peak. These guidelines are in terms of setting the maximum absolute numbers of each element (most common are C, H, O, N, S, P), elements ratio to carbon ranges, presence of multiple heteroatoms, respect to Senior and Lewis chemical rules and whether the expected isotope pattern for the metabolite with a possible formula is observed [50]. Both these methods must take into account the fact that *m/z* peaks represent the protonated or de-protonated metabolites whether the analysis is performed in positive or negative mode and that they may form adducts with ions such as $Na^+$, $K^+$ or $Cl^-$. However, coverage of the metabolome of different organisms and biological systems is lacking even for the Human metabolome, which means that using databases will lead to incomplete metabolite assignments [7]. Nevertheless, using formula assignment algorithms can be used as a complementary approach to database annotation. The reliability of the formulas assigned to the *m/z* peaks, nonetheless, might not be the best, especially with the automated formula assignment algorithms. The assignment can be validated by $MS^2$ but, on the large scale of

metabolomics, it relies on the availability of reference data which can make unambiguous identification difficult [7,30]. Therefore, feature annotation is still a major bottleneck in metabolomics analysis.

### 1.3.2 Data Pre-Treatment

After all the pre-processing steps mentioned in the previous section, with optional feature annotation, a clean dataset is obtained and is ready to be treated. Data pre-treatment has the objective of highlighting relevant biological information within the dataset while reducing the effect of undesired variation [25] due to measurement or technical errors, slight changes in temperature, batch or operator variation, etc. At this stage, an extra filtering step can be applied to remove features with very low variance between the samples (since these features are non-informative) or, if possible, to remove features with low reproducibility between samples based on quality control samples [51]. Following this, pre-treatments can be divided into 3 different "categories" of treatments: normalizations, transformations, and centering and scaling. Each category contemplates multiple options of treatment and may also be applied in combination with other categories, exponentially increasing the number of options available [52]. Since the pre-treatments made to the data can considerably alter the results of the statistical analysis, a thoughtful deliberation of the advantages and disadvantages of each treatment should be made, taking into account both the goal of the metabolomics study and the statistical analysis that will be performed downstream [25,35].

In the usual metabolomics workflow, normalization is the first pre-treatment step. Normalization has the objective of removing between-sample variation by trying to eliminate the systematic bias that exists between them [35,36]. This is done by multiplying or dividing the intensity values of the samples by a certain normalization factor, which is specific to each sample, allowing quantitative comparison analysis between them. There are several methods to normalize data currently used in metabolomics:

- Normalization by a reference feature that is an internal or external standard present in every sample. In this case, the normalization factor is the peak area or intensity of said reference feature in each sample or that value multiplied by a constant. Since this feature has a known concentration across all samples, by equalizing the intensity of this feature on all samples, comparing the intensity of the features between samples becomes more reliable.
- Normalization by the total peak area sum of a sample. Each sample's normalization factor is the sum of intensities of all its features. This method assumes that the total metabolite concentration in each sample is identical. It is worth noting that this means that high concentration metabolites will contribute much more to the normalization factor, which means that if, for whatever reason, there is a considerable concentration change in these types of metabolites, it will affect the normalization factor, reducing its efficacy – a possible disadvantage [35].
- Quantile Normalization, which aims to make all samples have the same peak intensity distribution. It is different from the other methods as it does not use a conventional normalization factor. Instead, it creates a "reference spectrum" from the data and uses it to replace all the values in the dataset. This is done by first ranking all intensities in each sample. Then, by calculating the mean or median of all intensities with the same rank, the "reference spectrum" is obtained. Finally, all values are replaced by the intensity value in the "reference spectrum" with the same rank. After normalization, a dataset where each sample has the same set of intensity values but distributed between the features differently is obtained [35,53]. This

method achieves the goal of having all samples with the same distribution, although it is problematic in datasets with considerable amounts of missing values or missing values that were imputed with constant values, since it creates samples with a lot of identical values in the lowest (or highest) ranks.

- Probabilistic Quotient Normalization (PQN) is a method that assumes that the difference in intensity in most peaks is due to different dilutions. This method therefore starts by performing a normalization by the total peak area, scaling all samples to the same magnitude and then proceeds to calculate another normalization factor based on a reference spectrum. This reference spectrum can either be calculated as the mean or median intensity of each feature or it can be a separate sample from the study or from a database. All samples are divided by this reference spectrum and the median value of the quotients of all features of a sample from this operation is taken as the normalization factor (dilution factor) for each sample [54].

Transformations are a set of non-linear treatments whose main objective is to reduce heteroscedasticity and to make the data more symmetric (less skewed). A dataset is said to be heteroscedastic if the variance of its variables is not constant. This affects the reliability of the statistical analysis since most methods assume that the dataset is homoscedastic and that it has a symmetrical distribution [25,35,55]. Transformations can have a "pseudo scaling" effect since they reduce the intensity of the higher values more than that of the lower values, thus shortening their differences. However, it does not replace the proper scaling methods presented below [25]. The most common transformation methods are logarithmic or power transformations:

- Logarithmic Transformation is a straightforward transformation that usually applies either natural base or base 2 logarithms to the dataset. This transformation, besides the previously mentioned effects, also turns multiplicative relations into additive relations, which might be biologically relevant. Note: This transformation cannot be directly applied on a dataset with null values.
- Generalized Logarithmic Transformation (Glog) is a variation of the logarithmic transformation that introduces a transformation parameter λ, with the objective of stabilizing variable variance. This transformation is made by applying equation 1.1 to the dataset [55,56]. The standard logarithmic transformation stabilizes variance for most of its variables, except for low-intensity features, since as they get closer to zero their variance increases dramatically. The Glog transformation aims to correct this drawback with the λ factor. Ideally, λ = b/a, where *b* and *a* are a normal distribution (with mean 0) of the error/variance, dependent of (*b*) and independent of (*a*) the intensity in the dataset in a model where the variance of a variable x is given by equation 1.2 [55,56].

$$\tilde{x}_{ij} = \log_2\left(\frac{x_{ij} + \sqrt{(x_{ij}^2 + \lambda^2)}}{2}\right) \quad \text{(eq. 1.1)}$$

$$\text{Variance}(x_i) = b^2 \times x_i^2 + a^2 + \alpha \quad \text{(eq. 1.2)},$$

Where $\tilde{x}$ is the transformed intensity x of the metabolite i of sample j, λ is a transformation parameter and α is background noise.

- Power Transformation applies the square root to the values in the dataset ($\sqrt{x_{ij}}$). It is a simple transformation that, despite not transforming multiplicative relations into additive relations, it

tends to reduce heteroscedasticity, improve symmetry and can be used in datasets with null values [25].

Mean centering and scaling is the last category of metabolomics data pre-treatments and usually the last step in data pre-treatment. These procedures have the aim of "balancing" high and low intensity biologically important metabolites and, to that end, they are often coupled together. Mean centering is done by applying equation 1.3 to the dataset, which removes the average intensity of each metabolite in every sample, leading to a focus on the relevant biological changes of a metabolite between different samples. This results in datasets where each variable has a mean of zero [25]. However, metabolites with higher intensities will have greater absolute differences in concentration compared to low concentration metabolites (especially in datasets where transformations were not applied to reduce heteroscedasticity) [35]. Scaling complements mean centering since it specifically aims to transform these absolute differences into relative differences, such as fold changes. Therefore, scaling methods include both mean centering and scaling of the data, as can be seen in the equations that define the scaling methods (eqs. 1.4-1.8) that subtract the mean of the corresponding variable to each value (mean centering) and then divide it by a scaling factor, which varies between the scaling methods. These scaling factors are mostly based on a dispersion measure, such as standard deviation, with size measures (mean, median) being a possible alternative. Although these very useful methods help emphasize the importance of low concentration biologically important metabolites, they can also lead to an amplification of the error in these metabolites' intensity assuming that the errors are relatively large in small values (which, when scaled up, also scale up the errors) [25]. Some of the most common and discussed scaling methods are presented here [7,25,35,57]. In each equation, $\tilde{x}$ is the scaled intensity x of the metabolite i of sample j, s and $\bar{x}$ are the standard deviation and average intensity of metabolite i across all samples, respectively.

- Mean-centering:

$$\tilde{x}_{ij} = x_{ij} - \bar{x}_i \quad (\text{eq. } 1.3)$$

- Auto Scaling or unit variance scaling uses the standard deviation of each metabolite as a scaling factor – eq. 1.4. This directly achieves the proposed aim of centering and scaling data to give equal weight to all features for further statistical analysis. However, this is the method that best represents the aforementioned increase of error that scaling might lead to in uninformative features affected by noise, which variation may, then, be interpreted as important. So, in order to apply this method, a considerable analytical effort has to be made during peak filtering to remove noisy and uninformative features of the dataset to minimize its shortcomings [35].

$$\tilde{x}_{ij} = \frac{x_{ij} - \bar{x}_i}{s_i} \quad (\text{eq. } 1.4)$$

- Pareto Scaling is a slight modification to auto scaling that uses the square root of the standard deviation as the scaling factor – eq. 1.5. This combines elements of both mean centering and auto scaling and still increases the importance of low concentration metabolites while limiting the inflation of measurement errors and preserving the data structure. This way, the method limits the influence of the drawbacks of mean centering and auto scaling, but it still suffers from them to some degree. While sensitive to large fold-changes, it does not present any major drawback due to the compromise taken [7,35].

$$\tilde{x}_{ij} = \frac{x_{ij} - \bar{x}_i}{\sqrt{s_i}} \quad (\text{eq. } 1.5)$$

- Range Scaling uses the range between the maximum and the minimum value a variable has across all samples as a scaling factor (hence the name) – eq. 1.6. The idea behind the scaling factor is that it represents the biological range of each metabolite. A disadvantage of the method is that its scaling factor depends only on the maximum and minimum value of a metabolite, making it very sensitive to outliers on any extreme (the minimum tends to be an imputed value given to missing values) [25,58].

$$\tilde{x}_{ij} = \frac{x_{ij} - \bar{x}_i}{x_{i\,(\max)} - x_{i\,(\min)}} \quad (\text{eq. } 1.6)$$

- Vast Scaling or VAriable STability scaling is another extension of auto scaling that multiplies its result by the ratio between the mean and the standard deviation of metabolite i (prior to auto scaling) – eq. 1.7. The purpose of this extra factor is to diminish the inflation of measurement errors – the major drawback of autoscaling – and focus on more "stable" features. This way, the low-abundance metabolites (low average) with high relative error (high standard deviation) will have a low $\bar{x}_i/s_i$ ratio and will have less importance [59]. However, a problem that arises is that biologically relevant features that have considerable fold changes between samples will also be considered unstable (low $\bar{x}_i/s_i$ ratio) and may be overlooked. Thus, this method focuses on features that change subtly between samples – stable features.

$$\tilde{x}_{ij} = \frac{x_{ij} - \bar{x}_i}{s_i} \times \frac{\bar{x}_i}{s_i} \quad (\text{eq. } 1.7)$$

- Level Scaling uses the mean of a metabolite (or the median as an alternative) – a size measure – as a scaling factor, unlike all other prior methods that used a dispersion measure – eq. 1.8. It changes each value to the intensity percentage change from the mean of the metabolite, which helps see fold changes between samples [25].

$$\tilde{x}_{ij} = \frac{x_{ij} - \bar{x}_i}{\bar{x}_i} \quad (\text{eq. } 1.8)$$

Each of these scaling methods provides a different outlook to scaling the data and brings its own sets of advantages and disadvantages, giving weight to the claim that the choice of pre-treatment methods depends on the objective of the analysis and should be regarded as a challenging and key task in data analysis.

## 1.4 Metabolomics Data Analysis – Statistical Analysis

All these possible processes and treatments aim to enhance the results of the statistical analysis of the metabolomics data. This analysis is the next step of the metabolomics workflow and, like the previous ones, a plethora of different approaches can be taken. As already discussed in the challenges of metabolomics experiments (section 1.2), the "omics"-characteristic high dimensional data (number of features much higher than the number of samples) poses specific issues to the statistical analysis due to feature multicollinearity [31,60], which is not corrected by the previously described pre-treatments. Statistical analysis can either be univariate or multivariate analysis. These two types are not mutually exclusive and application of both strategies can help to maximize the extraction of meaningful

information [32]. This dissertation is mainly focused on multivariate statistical analysis so only a brief overview of univariate analysis will be given.

### 1.4.1 Univariate Analysis

Metabolomics data is clearly multivariate with a large number of features, although it can still be analysed in a univariate way. A univariate analysis is performed by doing successive tests one feature at a time [61]. These methods have the disadvantage of considering a multivariate dataset as multiple individual variables, which will therefore fail to consider the interactions between different metabolites expected from a dynamic system. They counterbalance this disadvantage by being relatively simple to use and easily interpretable [4,62]. Moreover, they can also be used to find informative features in the dataset as a feature filtering procedure prior to multivariate analysis in some cases [32]. Many different statistical tests have been developed that are specific to certain characteristics in the data, most importantly if the distribution is approximately normal (Shapiro-Wilk and Kolmogorov-Smirnov tests can be used to assess normality), if there is homogeneity of variances between groups or homoscedasticity (Bartlett and Levene's test can be used to assess it), if there are 2 or more groups to be tested or if the samples are paired/matched [32]. These are based on hypothesis tests that set a null hypothesis or $H_0$ that states that there are no differences between the tested groups. With the test that will be applied, a probability value (*p*-value) of type I errors, that is, false positives (when the hypothesis is rejected despite being true) is calculated. After setting a threshold of acceptable type I errors (the most common is 5%), if the *p*-value is below this threshold, the null hypothesis is rejected, meaning that there is a significant difference between the compared groups; if it is above the threshold, the null hypothesis cannot be rejected – there isn't a significant difference between the compared groups. A stricter/lower threshold for type I errors will lead to more type II errors – false negatives – and vice versa. Vinaixa et al. [32] gives examples of possible tests to apply based on the dataset studied: for datasets whose features follow an approximate normal distribution and are homoscedastic, parametric tests are applied to compare means such as the unpaired and paired t-tests for 2 unpaired and paired groups, respectively, one-way ANOVA with multiple comparison for more than 2 unmatched groups and repeated-measures ANOVA for more than 2 matched groups; if the distribution isn't approximately normal, non-parametric tests are applied to compare medians such as the Mann-Whitney U test, the Wilcoxon signed-rank test, the Kruskal Willis one-way analysis of variance and the Friedman tests for 2 unpaired groups, 2 paired groups, more than 2 unmatched groups and more than 2 matched groups, respectively. An extensive review on univariate analysis of metabolomics data is provided by Vinaixa et al. [32].

When the univariate tests are applied iteratively to all features in a multivariate dataset, the likelihood (and number) of false positives starts increasing and they become almost inevitable when thousands of features are tested – the problem of multiple testing. Therefore, multiple test correction procedures need to be applied to control the number of false positives while trying to prevent missing true positives. Common methods for this are the Bonferroni correction and the False Discovery Rate (FDR) [4,32,61,62]. The Bonferroni correction is a conservative approach to minimize type I errors (increasing type II errors) that changes the threshold to reject the null hypothesis by dividing the pre-defined threshold by the number of tests performed (number of features) – $threshold_{new} = threshold_{old}$ / number of tests. Minimizing the FDR is a less conservative approach that aims to minimize the ratio of false positives to true positives instead of just reducing the absolute number of false positives. It transforms the set thresholds (that indicate that *x*% of all tests will result in false positives) into a *q*-value that indicates that *x*% of all significant tests will be false positives [4,32,62]. The Benjamini-Hochberg procedure is the most common out of the FDR-based methods. This procedure orders every

*p*-value obtained from the statistical test for every feature from lowest to highest and considers features significant (rejecting null hypothesis) if the condition indicated in equation 1.9 verifies [63].

$$p - \text{value} \leq \frac{\text{rank of feature by lowest } p - \text{value}}{\text{number of features}} \times q - \text{value} \quad \text{(eq. 1.9)}$$

### 1.4.2 Multivariate Analysis

Multivariate analysis considers all metabolomic data features simultaneously to analyse the data. This global view means that they take into account all possible metabolite interactions (including metrics like covariance and correlations) and don't see them as independent, unlike in univariate analysis [4,62]. However, due to the fact that the thousands of features in metabolomics dataset represent metabolites that are correlated and connected with other metabolites through, for instance, metabolic pathways, and the fact that there is a limited number of samples, high multicollinearity in the data is unavoidable – the curse of dimensionality [32,57,64,65]. This is a difficult issue for most multivariate methods to efficiently deal with, especially methods that rely on building a model since these will be prone to be overfitted – the model will "learn" the training data used to build it too well relying on small feature's variations that are not significant and will perform poorly with additional data [32]. Besides this, the complex interactions between the different features muddled in the data between informative and non-informative features can be difficult to discern [65]. For all these reasons, robust and extensive validation of built models must be achieved (discussed in section 1.4.2.2). Multivariate analysis methods can either be unsupervised learning methods or supervised learning methods. Unsupervised methods analyse the data without information regarding the groups or classes the samples belong to and try to detect intrinsic patterns within the data [66,67]. Supervised methods analyse the data with *a priori* knowledge of the group memberships of the training data used in building predictive models suitable to classify new data [4,66].

#### 1.4.2.1 Unsupervised Learning Methods

As mentioned earlier, unsupervised learning methods are a type of multivariate analysis whose aim is to detect intrinsic patterns within the data to group or separate different samples without knowledge of their group membership or the number of groups (metadata), that is, results are purely data-driven [66]. Many different unsupervised methods are available, with the most commonly used being Principal Component Analysis (PCA). Other common methods are self-organizing maps and a plethora of different clustering analysis methods [4,64]. A brief overview of PCA and some clustering analysis methods, specifically agglomerative Hierarchical Clustering Analysis (HCA) and K-means Clustering Analysis, will be given.

Principal Component Analysis (PCA) is an approach first described by Hotelling [68] and which became one of the most widely used methods of multivariate analysis. It is a dimension reduction algorithm that focuses on reducing the dataset dimensions from its number of features (*pp*) to a small number of Principal Components (PCs hereafter) orthogonal to each other [64,69]. These principal components are directions in the *pp*-dimensional space (with *pp* being the number of features) designed to represent new coordinates of sample data and are calculated to preserve as much information as possible by maximizing the variance of the projections of the data point over these new coordinates. The PCs will then represent a new coordinate system, with a dimension lower (usually much lower) than *pp*. Therefore, the first PC is defined as the direction in the *pp*-dimensional space, which corresponds to the largest variance in the data (linear combination of the *pp* features of the

original dataset); the second PC is defined as the direction of the largest variance of the data orthogonal to the first PC; the third is orthogonal to the first and the second one and henceforth [38,64,70]. The new coordinates of each sample projected on each PC form the matrix of PC Scores [64]. The number of components to use is also a valid question. Common approaches are setting a threshold for the minimum accumulated explained variance by the different components (for example, it can be set at 80 or 90%) or observing when the variance explained by a new component compared to the variance explained by the prior component starts being less impactful (and therefore including it does not meaningfully improve the fitting of the PCA model to the data). Apart from this, common visualization of this descriptive method is usually done with the plot of the sample scores on the first 2 or 3 principal components; therefore, when the aim is to represent the data in these components only 2 or 3 are used, as seen in Fig. 1.1A [69,70]. The results of the PCA depend on the variance of the data. This means that they also depend on the units of the features in the dataset, hence the aforementioned importance of pre-treating the data, in this case specifically the importance of centering and scaling the data to standardize the features [70]. A more detailed review of PCA, explaining the intricacies of the method, as well as extensions of PCA, is presented by Jolliffe et al. [70]. Moreover, due to the popularity of the method, comprehensive textbooks on the subject can be found (such as [71]).

Clustering methods analyse the data based on similarity measures, using the distance between samples. Clustering methods can be divided into hierarchical clustering (agglomerative HCA is an example), partitioning clustering (such as K-means clustering), model-based clustering, grid-based clustering, density-based clustering and graph-based clustering [64,72]. A detailed review of the different clustering types and their applications in bioinformatics is provided by Andreopoulos et al. [72]. It is worth taking into account that most of these methods don't give an estimation of the reliability of their results and, therefore, it has to be estimated by other methods if needed [20].

In agglomerative hierarchical clustering analysis, all samples start in their own separate cluster and are progressively clustered together based on a similarity measure. The closest clusters are first clustered together and this process is done iteratively until one cluster with all samples is obtained. This similarity measure is based on a distance metric used – commonly the Euclidian distance between the samples but can also be Manhattan, Mahalanobis distances or even binary dissimilarity metrics such as Jaccard or Yule dissimilarities. The calculation of the distances between multi-sample clusters needs another criterion called Linkage. Some common Linkage methods are average linkage or UPGMA algorithm that considers the mean distance between all pairs of samples, one belonging to each of the clusters (may be computationally intensive), single linkage that considers the minimum distance between 2 samples (one belonging to each cluster – may merge clusters with only 2 samples close to each other, which is referred to as the chaining problem), complete linkage that considers the maximum distance between 2 samples (one belonging to each cluster – vulnerable to outliers), Ward's variance minimization linkage method where the "distance" to be minimized is the within-cluster variance [73] and centroid linkage where the distance between clusters is the distance between their centroids, among others [38,64,72]. The result of this method can be neatly represented as a dendrogram (tree) that facilitates both visualization and interpretation – Fig. 1.1B [4]. Different dendrograms can result from applying either different distance metrics or linkage methods. They can be compared by correlation metrics, with the two common metrics being the cophenetic correlation coefficient (Pearson correlation between all pairwise distances where two samples were clustered together between the two dendrograms [74]) and Baker's Gamma correlation coefficient (Kendall's correlation between the iterations of the algorithm where two samples were merged together in each dendrogram for all sample pairs [75]).

K-means clustering is a type of partitioning-based clustering method where *k* clusters are made with a centroid being representative of each cluster (*k* is defined by the user). Samples are attributed to the cluster defined by the closest centroid in the *pp*-dimensional space (*pp* being the number of features in the studied dataset) and the centroids are updated to minimize the distance to the samples that are in their cluster (which can be negatively affected by outliers). These two steps are repeated until the samples in each cluster do not change between iterations (or their shift is below a predefined threshold). Since the starting centroids are chosen randomly, this method does not always generate the same results, since the algorithm may stop at a local minimum instead of a global minimum, so repeating the algorithm might help find the best estimation of the global minimum. This type of methods might be applied over hierarchical clustering when a specific number of clusters is desired [64,72].

As with all techniques, it is important to assess the quality of the clustering. The Rand index is an external criterion that evaluates how similar the clustering made is to the natural group separation of the samples on the original dataset (defined *a priori* of the analysis). This index compares the *k* clusters made from K-means clustering analysis (each cluster is a group) to the original group memberships. Thus, each pair of samples is tested to see if their group relation by the original group memberships and by the clusters made from clustering are in "agreement". They are in "agreement" if the two samples that belong (or do not belong) to the same original group memberships were (or were not) clustered in the same cluster by the K-means clustering analysis. For example, considering the samples in Fig 1.1, the original group memberships and clustering performed are "in agreement" if 2 samples of the dGLO2 strain (same group) were clustered in the same cluster and are also "in agreement" if a sample from the dGLO2 strain and another from the dENO1 strain were clustered in different clusters. The Rand index is then the ratio of every pair of samples that is in agreement by total amount of pairs of samples [64,76].

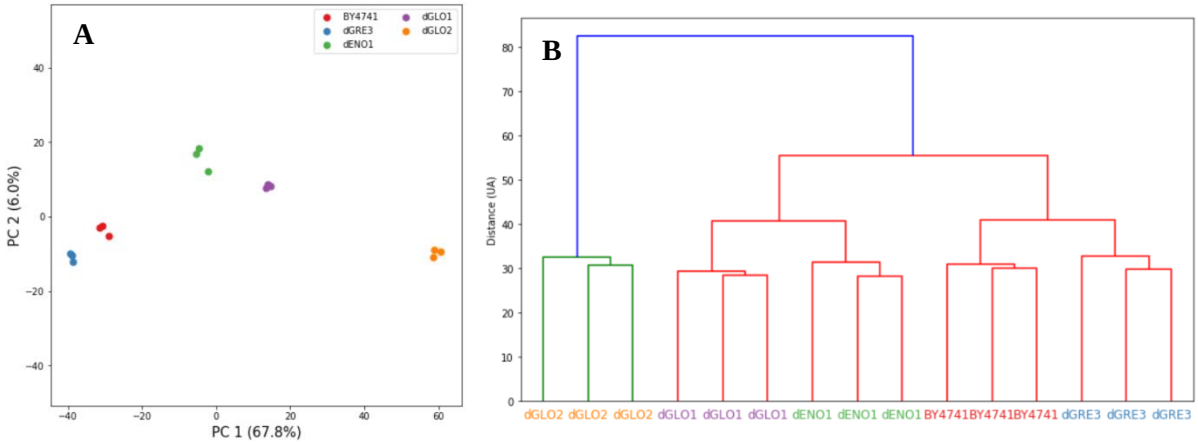



**Figure 1.1: Representation of a typical results figure from PCA (A) and Hierarchical Clustering (B).** A) Sample projection on the first 2 Principal Components with the explained variance of each Principal Component (PC) indicated in the respective axis. B) Dendrogram representation of Hierarchical Clustering. Both representations were made from an example dataset with 3 samples of 5 different groups that each represent a strain of yeast – the reference BY4741 strain – and 4 single gene deletions of this strain: deletion of the *GRE3* (dGRE3), *ENO1* (dENO1), *GLO1* (dGLO1) or *GLO2* (dGLO2) gene (Data from [77]).

### 1.4.2.2 Supervised Learning Methods

As mentioned earlier, supervised methods build models based on a set of data called the training set with *a priori* knowledge of their group membership (discrete groups) or of continuous response variables. The models can then be used to make classifiers (group memberships) or regressions

(continuous response variables), [4,66]. Regressions will try to predict the best fit of a test sample with a continuous response variable and the quality of its predictions can be assessed by calculating the root mean squared error in relation to the true response variables of the test data. Classifiers, on the other hand, will try to predict the group membership (discrete options) of samples of the test data based on the groups of the samples in the training data. These classifiers can then be assessed according to their predictive accuracy of group memberships, as well as their sensitivity and specificity among other possible performance metrics [66]. For this work, classifiers are the most relevant type of supervised methods.

As indicated before, validation of the models obtained by supervised learning methods is extremely important to avoid overfitting due to the high number of features, high feature collinearity and low sample number of the metabolomics datasets. Common ways to try and validate a model include (but are not limited to) external validation by holding out test samples, internal validation, bootstrapping, and permutation tests. Internal validation tends to be used when the number of samples is low and the loss of a few samples to test the model would have a significant impact on the model built, while external validation is better used when there is a greater amount of samples [78]. Internal validation is performed using methods such as *k*-fold cross-validation (CV) and leave-one-out cross-validation (LOOCV), [78] and it consists of splitting the dataset that will be used for the final model into its own training and testing set in different ways to observe the model's performance when predicting "unknown samples". *K*-fold CV splits the dataset into *k* different sample sets (5,7 or 10 are the most common), where one of them is iteratively the testing set and all the others are the training set until all sets have been the test data once. The mean of the performance metrics of the models built for each test set is therefore an estimate of the performance of the prediction model built with the full dataset. The splitting of data can be achieved in different ways (Fig. 1.2): standard *k*-fold CV groups can be constructed with venetian blinds (each sample is sequentially assigned a number from 1 to *k*, with *k* being the number of sets and each sample with the same number belongs to the same set) or contiguous blocks (each block of *n*/*k* samples forms a set, with *n* being the total number of samples) if the samples are already randomly ordered, or split into random subsets. Besides this, stratified *k*-fold CVs aim that all different sets have the same number of samples of each group. In LOOCV, each set is comprised by a single sample – in each iteration, a model is built with all the samples, except one that will be tested [4,52,64,78]. External validation works by leaving out a set of samples that are only used to test the model, while the rest of the samples are used to build the model. The two main parameters are the number of the samples excluded (around 30% of the total samples is the norm) and the procedure to select the samples to leave out, such as random sampling (closer to a "real" situation) or Kennard-Stone sampling (used to have a uniform distribution between training and test sets) [78].

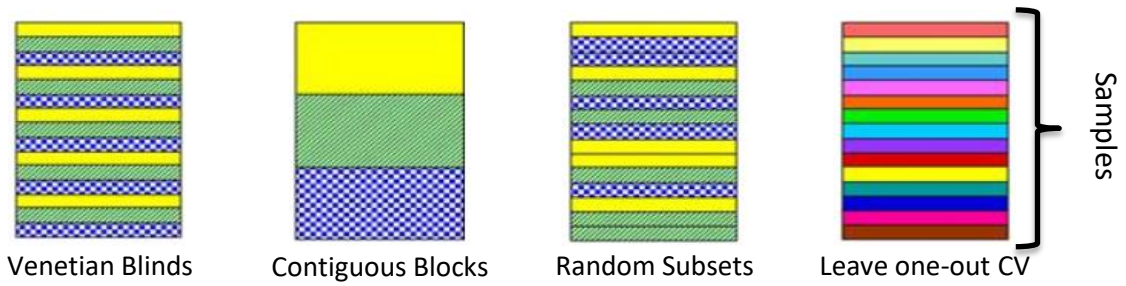



**Figure 1.2: Different strategies to split the dataset into *k* different groups of *m* samples each.** Each different pattern/colour represents a group. Adapted from https://wiki.eigenvector.com/index.php?title=Using_Cross-Validation.

The objective of a permutation test is to determine the probability of observing an equal or better performance of the predictive model built from a dataset compared to predictions based on pure chance, that is, the significance of the predictive accuracy of the model. This is done by randomly permuting the group labels of each sample, building the model (with the same parameters) with the permuted labels' dataset, and comparing the performance of its prediction model (using one of the prior validation techniques mentioned) – usually accuracy is the performance measure used with classifiers. The rationale is that if there is some intrinsic structure within the groups, the prediction model should have a greater accuracy compared to when the data is jumbled (the difference between the groups is absent). Thus, many label permutations are repeated and if our original prediction model has a better performance *x*% of the times, with *x* normally being equal to 95 or 99%, the prediction model's performance is significantly better than the models built with randomly labelled datasets, that is, the model is using information intrinsic to the different groups to classify them [4,64,79].

A brief overview of two different supervised learning methods, PLS-DA and Random Forest, will be given next.

**Partial Least Squares Discriminant Analysis (PLS-DA)**

Partial Least Squares (or Projection to Latent Structures) – Discriminant Analysis (PLS-DA) is a popular classification method in the field of metabolomics since it can analyse efficiently high-dimensional data with multicollinearity and does not assume any kind of distribution for the data. PLS-DA was created from the Partial Least Squares (PLS, also referred as Projection to Latent Structures) regression analysis by adding a decision rule for group membership to the regression results obtained. Like PCA, PLS is a dimension reduction algorithm. However, instead of maximizing variance of the projections of the samples on the principal components, PLS maximizes the covariance between the dataset (the samples) – X – and a vector or a matrix representing classes/group memberships – Y [31,64,78,80]. The PLS algorithm used is slightly different depending on whether there are 2 classes (PLS1-DA) or more than 2 (PLS2-DA) to predict. For 2 classes, the response variable will usually be a vector of 0s and 1s, where each number represents a categorical class. For more than 2 categorical classes, a "dummy matrix" is constructed with one-hot encoding, where each column represents a class with a sample having a 1 or a 0 in that column whether they belong to the class or not, respectively [78]. PLS components are projections called Latent Variables (LVs). Wold et al. goes into detail about the mathematical models that define the PLS regression and the PLS-DA classification analysis [81], which will not be explained here. As in PCA, we can compute Loadings (values of the contribution of each variable to each LV) and Scores (coordinates of each sample on the new components); however, since both the X variables and the Y response variables are considered when making a model, there are both X-loadings and X-scores, Y-loadings and Y-scores. Furthermore, another pair of matrices called X-weights and Y-weights is needed to give information about the combination of the variables to form the quantitative relation between X and Y [80,81]. According to Wold et al., the LVs in the model should be restricted to those which contribute to the predictive significance of the PLS or PLS-DA model using internal validation methods like cross-validation, such as the ones discussed earlier in this work [81]. For regression analysis, the performance of the model can be assessed by the value of (1 - Predictive Residual Sum of Squares (PRESS)/SS), where SS is the residual sum of squares of Y corrected for the mean – $Q^2$ – as the number of LVs increases [81].

The results of applying a PLS model on a test sample are a non-categorical prediction of the best estimate of the response variable for a given test sample. This will be called $y_{pred}$ here and it is a

number (almost always between 0 and 1) for 2 class systems and a vector with *n* numbers (*n* > 2) for more than 2 class systems where each number corresponds to a class where the closer to 1 it is, the more similar the sample is to the samples belonging to that class. From the $y_{pred,}$ a decision rule (DR) is then applied to decide the class or group membership of each test sample (transforming PLS into a PLS-DA). Besides, the X-scores could be used by employing a distance metric to calculate its distance to the different classes [78]; however, this is less common than the use of $y_{pred}$. The simplest and most common decision rule used is to simply see what the maximum value in the $y_{pred}$ vector is and assign the sample to the corresponding column/group (max DR rule). Nevertheless, other more complex DRs can be applied, such as transforming the $y_{pred}$ with a probability density function or setting a minimum threshold for a class to be assigned (leads to the possibility of unassigned samples, which can be an advantage when testing truly unknown samples). This threshold can be determined arbitrarily or by applying tools like probability density functions [78]. For 2 class models, the threshold or cut-off point is usually set at 0.5, where a higher value is assigned to the class represented as 1, and a lower value than 0.5 is assigned to the class represented as 0. This threshold can also be changed and optimized based on user preference. Besides, two threshold points might be set, creating a boundary and an interval of values between the thresholds where no assignment is made [78]. However, if most of the predictions made are close to the defined threshold, this could mean that the model has a low discriminatory power, putting its prediction capability in question [31]. Despite this, a study by Lee et al. shows that the "simple" max DR rule has a similar performance to other more complex rules, leading to higher model stability in exchange for slightly lower model accuracy [80].

With the PLS-DA models, there are many strategies that allow either variable selection to make a model with a lower number of variables (higher density of meaningful variables) or to select possible biologically important features that contributed more to the model to discriminate the different classes. These are filter methods that aim to identify important features based on a certain measure (which is the most relevant to this work); wrapped methods that aim to create robust models with a reduced amount of variables by iterating the following process – applying filter methods to obtain subsets of the data and refitting the model to find the subset that maximizes performance; finally, embedded methods where variable selection is made during the development of the model [82]. Filter measures can be applied to a direct set of parameters obtained by the PLS model, specifically the absolute values of X-weights (contribution of each feature to the covariance of X and Y in each LV) and regression coefficients (global measure of association between X and Y used for test sample prediction in each LV), [80,82]. Therefore, an ordered list of their absolute values can give us an indication of the most important features to build the PLS-DA model. Variable Importance/Influence in Projection (VIP) score is another method that estimates the importance of each feature to explain the variance in Y using equation 1.10, which takes into account the importance (variance explained) of each LV in the model in comparison to the total variance in Y and the contribution of each feature to each LV (represented as the X-weights) to select the features that are most important to the model [82–84]. Since the average of all VIP scores is 1, 1 is usually used as a minimum threshold to consider a feature as important or relevant. However, a careful consideration of a suitable threshold based on the results obtained is recommended.

$$\mathrm{VIP_i} = \sqrt{pp \times \frac{\sum_{a=1}^{nº\,LV}(q_a^2 \times t'_a \times t_a) \times (w_{ai} \,/\, \|w_a\|)^2}{\sum_{a=1}^{A} q_a^2 \times t'_a \times t_a}} \quad (\mathrm{eq.}\,1.10)$$

In equation 1.10, $q_a$ and $t_a$ are Y-loadings and X-scores of the $a^{th}$ LV, $w_{ai}$ is the x-weight of the $a^{th}$ LV for feature i and *pp* is the total number of features [82,84].

Many different variants of PLS-DA have been developed since the start of PLS-DA use in statistical analysis with orthogonal PLS-DA being the variant that has gained the most notoriety in recent years; this method separates the Y-correlated variation in X from the Y-uncorrelated variation in X (structure in X that is orthogonal to the Y response) – Orthogonal Signal Correction [7,52,64].

**Random Forest**

Random Forest is a classifier that consists of an ensemble of independent decision trees, each built with a subset of the training data. Each tree makes a decision about which group a tested sample belongs to. The global result for the random forest classifier is then based on the majority vote of all individual trees [64,85]. This allows for a better performance, less variance in the results and especially less overfitting compared to using single-decision trees built with all training samples [52,86]. A decision tree is an input-output model that makes binary decisions based on the value of a feature (each node on the tree represents one of these decisions) built so that it can split the samples in their respective groups (Fig. 1.3). With each decision made, the data subset is split into two different subsets. The logical rules (decisions) are chosen to decrease a certain "impurity measure" such as the Gini Index/Impurity. The Gini Index is a measure of the probability of a random sample being incorrectly labelled if randomly labelled by the distribution of the training samples' labels of the node where that sample is inserted. Each binary decision is then made so that misclassification of samples used to build the tree is decreased as much as possible until all subsets of samples only have samples of one group or until a maximum tree depth predefined by the user is reached. A test sample navigates the tree according to those binary decisions and it will be classified as belonging to the majority group of the terminal node that it reaches [52,64,86,87]. For random forest, each tree is independently built based on a random bootstrap sampling of the training data, with the forest being a set of de-correlated trees [52,86]. An important parameter of the model is, then, the number of trees to use. With more trees added to the forest, the generalization error of the prediction converges, which means that adding more trees won't lead to overfitting the model [85,86]. This means that an appropriate number of trees to use is one that is sufficient to approximate the converged generalized error (and, therefore, the convergence of the model accuracy) without being too computationally expensive. Random Forest can be applied to great effect with metabolomics data due to their ability to deal with collinear data while not assuming any distribution in the data and being resistant to outliers [64]. Furthermore, they are able to highlight features that were important to build the model. A way to perform this is by assessing how much a feature contributed to decreasing the impurity measure used across the different decision trees (more contributions means a higher importance of the feature in the model). When the impurity measure used is the Gini Index, this is called the Gini Importance or Mean Decrease Gini [87]. Other approaches to characterize feature importance include the Mean Decrease Accuracy, which assesses the feature's importance by assessing its impact on the performance of the model (by checking the performance when said feature has random permutated values), [87].

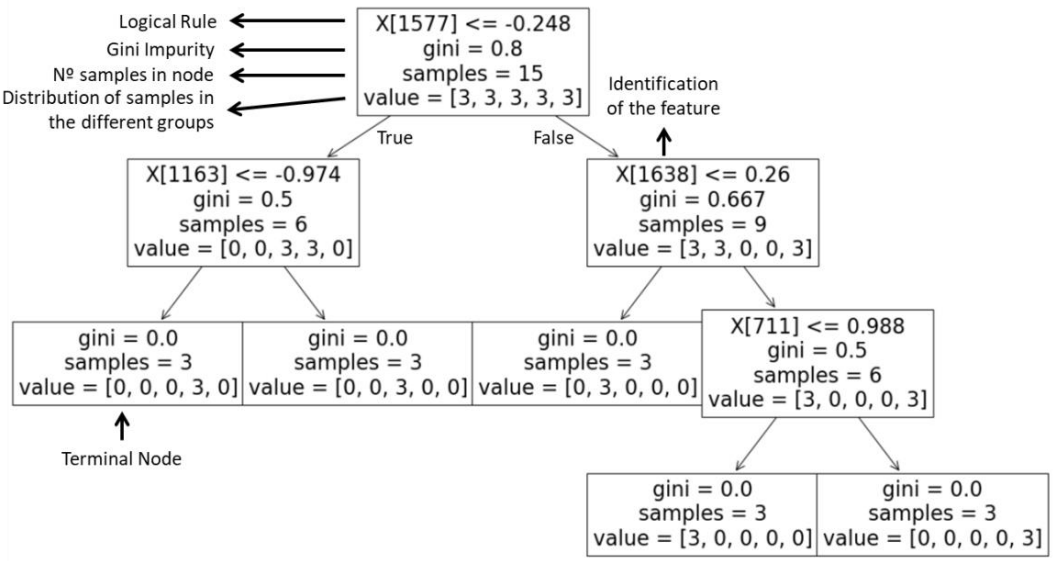



**Figure 1.3: Example of a small decision tree present in a Random Forest.** All the samples are a part of the training set used to build the example decision tree.

## 1.5 Analysis of the Chemical Diversity of a System's Metabolome

A system's metabolome has a huge chemical diversity, which at any given point is characteristic of the system under the conditions of its environment. Its chemical diversity comes from the thousands of metabolites that make its metabolome, from their complex interactions and spatial and temporal organization [88]. The compositional space these metabolites occupy even when only considering the 6 more common elements in biological compounds (C, H, O, N, S and P) is extremely vast, with millions of possible combinations in both number and structural organization of those six elements to make different compounds. As mentioned earlier, this makes the structural identification of metabolites a major bottleneck of the metabolomics analysis even for high resolution techniques, especially when considering that thousands of formulas can be fit in the same 1 Da interval, each with many different possible isomers. The actual metabolites present in a biological system undergo specific interactions consistent with the homeostasis of highly organized and complex systems. This organization is, in great part, achieved by the compartmentalization of molecules, which places different metabolites in specific subcellular localizations at specific times, where they can be used in a myriad of different functions, for example as energy storages, building blocks of macromolecules, signals, among others [6,88].

### 1.5.1 Representation of a System's Chemical Diversity

The analysis of the global chemical diversity of a system from high-resolution data helps to characterize it. Data from high-resolution methods can give us a global snapshot of the metabolome. As already discussed, this "snapshot" is incomplete and will be biased towards a subset of metabolites more easily identified by the method used. Despite this limitation, it still encompasses a substantial part of the chemical diversity of a metabolome, which can be further increased by using multiple different high-resolution methods [88]. The annotation of some spectral features found by these methods can greatly help the analysis of the chemical diversity of a sample. Fortunately, structural elucidation (unambiguous identification) of metabolites is not needed and that bottleneck can be overcome. The use of formula assignment algorithms to assign likely formulas to *m/z* peaks is a faster

process that may be used instead of structural elucidation, if needed. However, these can still be ambiguous when multiple suitable formulas have very similar masses (a problem that gets worse at higher masses). The complex chemical diversity of different samples is usually represented in simplified and visually clear ways that can be easily interpreted. Some of the more common representations of a system's chemical diversity are Van Krevelen and Kendrick Mass Defect (KMD) plots [88,89].

Van Krevelen plots represent the carbon, hydrogen and oxygen (CHO) chemical space by plotting the identified formulas on a plot of the ratio of the number of hydrogen to carbon atoms (H/C) over the ratio of the number of oxygen to carbon atoms (O/C) – Fig. 1.4A. The areas (combinations of both ratios) the different formulas occupy are sometimes used to estimate to which groups or categories they belong, such as aminoacids, carbohydrates, lipids, lignins, nucleic acids, aminoacids-like, or carbohydrates-like compounds, among others [90]. For example, this representation was done in the works of Gougeon et al. [91] and Roullier-Gall et al. [29] when analysing the chemical diversity in wines. However, the lack of concretely defined boundaries for each category and the considerable overlap among these different categories makes the classification often inefficient. This lack of accuracy associated with the difficulty in categorizing compounds solely based on their H/C and O/C ratios (not considering other relevant elements such as nitrogen) can lead to inconclusive results and a lack of robust conclusions [90].

KMD plots represent the Kendrick Mass Defect over the Kendrick nominal masses – Fig 1.4B. The Kendrick mass of a compound is given by standardizing the group $CH_2$ to exactly 14 (Kendrick mass) by eq. 1.11. The Kendrick Mass Defect (KMD) is the difference between the nominal Kendrick mass and the exact Kendrick mass [92]. This means that compounds with the same heteroatoms (class) and same amount of double bonds plus rings (type – changes by $H_2$ or exactly by 0.0134 KMD) will be represented in a horizontal line with their position on the x-axis being based on the number of $CH_2$ groups. Therefore, it allows the visualization of the different classes and types of compounds in the samples (due to their characteristic KMD) in a 2D space [92], although it can become difficult to distinguish groups when many compound classes are present.

$$\mathrm{m}_{\mathrm{Kendrick}} = \mathrm{m}_{\mathrm{IUPAC}} \times 14 \div 14.01565\ (\text{IUPAC mass of the group } \mathrm{CH_2}) \quad (\text{eq. } 1.11)$$

Furthermore, the different formulas being plotted in these types of graphs can also be colour-coded based on some pre-determined condition to further help the visualization of the chemical diversity. For example, in Adrian et al. [93], when analysing the chemical diversity in grapevines, formulas were coloured based on the chemical elements they possess, splitting them into these groups: CHO, CHON, CHOS, CHOP, CHONS, CHONP and CHONSP.

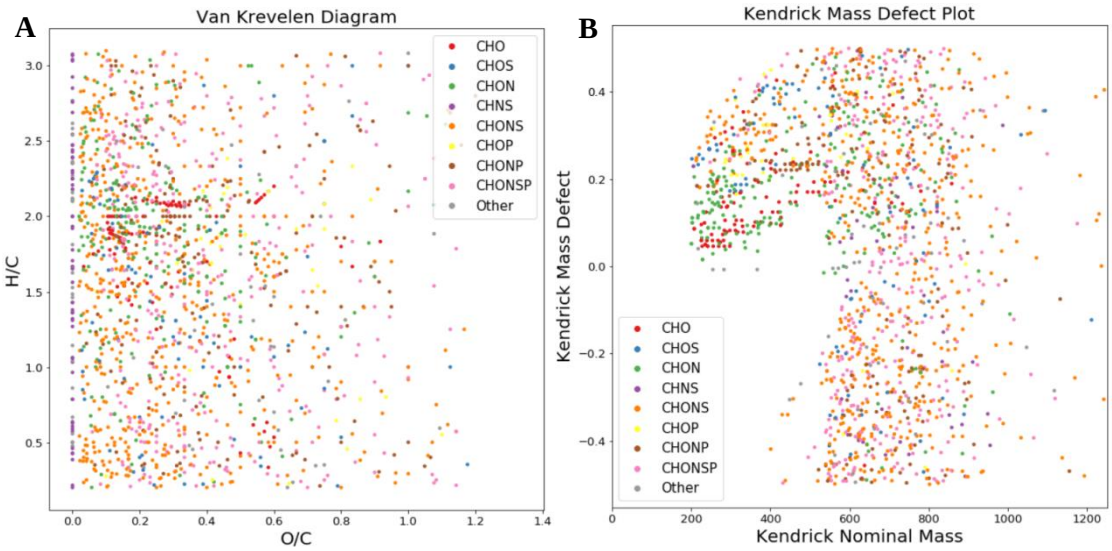



**Figure 1.4: Example of a Van Krevelen diagram (A) and a Kendrick Mass Defect plot (B) of metabolomics data.** Dots are coloured according to their elemental formula composition: (●) – CHO, (●) – CHOS, (●) – CHON, (●) – CHNS, (●) – CHONS, (●) – CHOP, (●) – CHONP, (●) – CHONSP and (●) – Other. Data from [77].

### 1.5.2 Mass-Difference Networks (MDiNs)

Recently, another representation method called Mass-Difference Networks (MDiNs), which allows a better discrimination of different compound classes when compared to Van Krevelen and KMD plots, has been used to complement the two previously mentioned types of plots. Mass-Difference Networks (MDiNs) use the list of masses obtained from high-resolution methods as nodes in a network. Edges are established between nodes that have a difference in masses very close to specific mass differences that represent certain chemical transformations – Fig. 1.5 [88,94]. This method was originally developed by Breitling et al. [95] and took advantage of the fact that many likely chemical formulas could be assigned from high-resolution data to build non-random and informative networks *ab initio*. These networks made from high-resolution data originate their own partial "metabolic networks" that also consider possible spontaneous non-enzymatic reactions and are not influenced or skewed by the known metabolic pathways that still are very incomplete in many biological systems [95,96]. The exact mass differences correspond to the changes in the elemental formula of metabolites due to common biochemical reactions such as: methylations (total change of $CH_2$ after substitution of –H hydrogen atom by a $–CH_3$ group), hydrogenations/dehydrogenations ($H_2$), hydroxylations (O), phosphorylations ($PO_3(H)$), among many other possibilities (such as aminoacids for bigger groups) that have characteristic masses, for example, $H_2$ = 2.01565 Da and $CH_2$ = 14.01565 Da [94,95]. These "Mass-Difference-based Building blocks" (MDBs) can be chosen from a set of common biochemical reactions or by observing the most common mass differences present in the dataset and to which chemical transformations they might correspond [95,97].

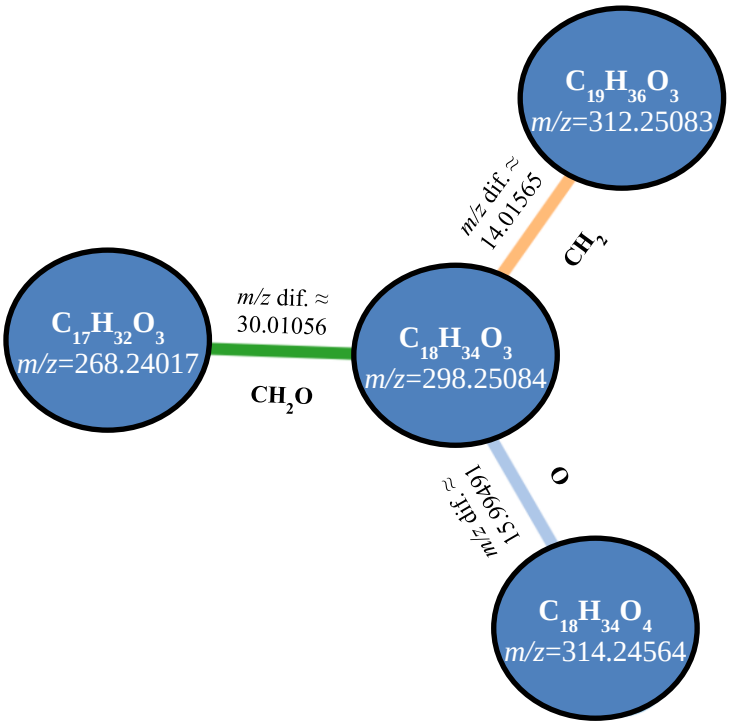



**Figure 1.5: Example of the concept of Mass-Difference Networks (MDiNs) in a 4 node example network.** *m/z* peaks are from a mass spectrum. dif. = difference.

These networks are also used as a method to expand the number of features assigned with formulas in a way similar to Kendrick Mass Defect analysis, but able to consider as many homologue series as the number of groups used to build the network [94,96]. This kind of formula assignment can be achieved if even one single elemental formula is known in a component of the generated network, since all edges correspond to specific additions or subtractions of MDBs. Therefore, by starting with a select list of formulas assigned to peaks with higher reliability, obtained for example through annotation with a database, you can assign formulas to many of the other *m/z* peaks through a chain of specific mass differences to the starting formulas (while keeping in mind certain characteristics of metabolites such as acceptable ratios between the different elements), [94] as an alternative approach to formula assignment algorithms. On one hand, this makes formula assignment biased to formulas "similar" to the starting set of formulas, instead of considering the complete metabolite chemical space; on the other hand, these formulas belong to known existing metabolites that have a high probability of being present in the sample and, therefore, it could be argued that formulas similar to these, especially when the difference can be explained by common biochemical reactions, also have a higher probability of being present in the same sample.

Visualization of the networks can be further enhanced by colouring the different edges based on the group or mass difference that defined that edge. This helps seeing trends of relations between functional groups in the network more easily [94].

## 1.6 Aim

This dissertation focuses on the exploration and development of new and reliable ways to analyse metabolomics datasets for the profiling and discrimination of samples into their respective design groups. In particular, two new methods were developed, as alternatives to traditional data pre-treatments, exploring new perspectives to look at metabolomics data:

1) Binary Similarity (or BinSim for short);
2) Sample Mass-Difference Networks (Sample MDiNs).

These two pre-treatment methods forgo the use of the highly variable intensity data and focus on the (less variable) occurrence of spectral features in the samples (as an effort to make metabolomics results more reproducible). Moreover, due to focusing on different aspects of the data, any of these methods could also be used as a complementary approach to the usual workflow to provide a more complete picture of the metabolomics data analysed.

1) Binary Similarity focuses simply on the presence and/or absence of features from a dataset – occurrence of spectral features – discarding its metabolites signal intensity data. Since most other mainstream available treatments focus on the different signal intensities, features with considerable amounts of missing values tend to be filtered. This method skips most of the peak filtering (since missing values are valuable information in this method), missing value imputation and choice of pre-treatment steps in the metabolomics analysis workflow for a faster and simpler analysis. Furthermore, it also ignores the oft-used intensity data, which is very prone to variance from sample to sample in metabolomics, thus it should give a new perspective on the data compared to other methods. So, with its speed, simplicity, less variability and new perspective, it can also be a way to complement the standard metabolomics data analysis workflow, even if not used as the primary analysis. Here, how well different statistical analysis methods (both unsupervised and supervised methods) can discriminate between different groups in datasets treated by BinSim compared to datasets treated with more established intensity-based pre-treatment methods were compared. Moreover, to observe if this approach was extracting relevant information from data features that are often not used despite being equally important, the important features to build classifier models with supervised methods from datasets treated with BinSim and with other intensity-based methods were compared.

2) Sample MDiNs is a method that focuses on building Mass-Difference Networks (MDiNs) for each sample in a dataset (sample networks), where each (neutral) mass in the sample represents a node and an edge is established if the difference between two masses (nodes) can be explained by a change in a metabolite derived from a simple biochemical reaction (enzymatic or non-enzymatic). Thus, the set of mass differences ("Mass-Difference-based Building blocks" – MDBs) used represent the mass derived from the overall change in the elemental formula of a metabolite after certain sets of reactions. This method will aim to build a characteristic network for each sample to represent their chemical diversity. The chemical diversity of a biological system is, in principle, characteristic of that system under certain conditions since it is tied to the metabolome. It follows that a network such as a MDiN, which aims to be a good descriptor of the chemical diversity could and should have specific properties that characterize a sample of that system. Therefore, here it is proposed that these sample MDiNs can be characteristic of their biological group and therefore used (as a data

analysis treatment) to represent metabolomics samples and discriminate them into their respective groups (be it species, strains, varieties, or any other kind) based on the networks characteristics and properties. As traditional methodologies, MDiNs aim to extract information and generalize it into something we can understand, however it focuses on the possible chemical transformations between the different spectral features rather than looking at each feature individually. Thus, the sample networks contained the information of the possible chemical transformations between its nodes, which is a very different perspective in comparison to other methods. This opens a plethora of options due to the versatility of network analysis methods that can be used to compare different networks. Through different of these network analyses that focus on diverse network characteristics (nodes, edges, topology of the network) on the sample networks, "secondary datasets" were built which allowed the comparison of the different samples. By a similar methodology to the previous point, statistical methods were used to try and discriminate the different samples into their respective groups based on those secondary datasets to test if any of them could be considered a viable way to treat data as far as the discrimination of samples is concerned.

These treatments, if viable, will contribute to a boost in the usual metabolomics data analysis workflow, by taking into account characteristics that are specific of metabolomics data (as opposed to most other methods, which were borrowed and adapted from the other "omics" sciences), looking at the data in considerably different ways and being able to quickly complement most dataset analysis.

# 2. Materials and Methods

Three metabolomics datasets (hereafter referenced as the Negative and Positive Grapevine Datasets (GD) and the Yeast Dataset (YD)) were systematically used for the development of the proposed methods and for comparison with more established intensity-based methods. These datasets were previously acquired at the Fourier-Transform Ion-Cyclotron-Resonance and Structural Mass Spectrometry Laboratory (FT-ICR-MS-Lisboa) group infrastructure, where this work was developed.

All processing, treatment and analysis of the different datasets mentioned from here on out were made using Python 3.7.4 programming language. Furthermore, this was developed by using several well-known Python packages. The main Python packages used were: numpy [98], pandas [99], scipy [100], scikit-learn [101], matplotlib [102], seaborn [103] and networkx [104]. Moreover, this work both contributed for and used extensively the package metabolinks available at https://pypi.org/project/metabolinks/.

The main scripts, *jupyter* notebooks and data used to produce this work are available in a git-hub repository at: https://github.com/aeferreira/similarity_share as well as some of the other analyses made to support this work that was not shown in this dissertation.

## 2.1 Datasets

### 2.1.1 Grapevine Datasets (Positive and Negative Ionization Modes)

The Grapevine Datasets were acquired by Marisa et al. [105] and are available in figshare data repository with the identifier m9.figshare.12357314 (https://doi.org/10.6084/m9.figshare.12357314), [106]. Grapevine Dataset samples were prepared as described in Marisa et al. [105]. Briefly, the leaf metabolome from eleven field grown *Vitis* genotypes (5 wild *Vitis* species and 6 *Vitis vinifera*) was analysed by Fourier Transform Ion Cyclotron Resonance mass spectrometry (FT-ICR-MS). For the analysis, extracted metabolite samples were diluted 1000-fold in methanol and human leucine enkephalin (Sigma Aldrich) was added for internal calibration of each mass spectrum ($[M+H]^+$ = 556.276575 Da or $[M-H]^-$ = 554.262022 Da). Formic acid (Sigma Aldrich, MS grade) was added at a final concentration of 0.1% (v/v) before the positive ion mode analysis. Samples were analysed by direct infusion on an Apex Qe 7-Tesla FT-ICR-MS (Brüker Daltonics). Spectra were acquired with an accumulation of 250 scans of 512Kb for each spectrum, at both positive ($ESI^+$) and negative ($ESI^-$) electrospray ionization modes, in the mass range of 100 to 1000 *m/z*. Data Analysis 5.0 (Brüker Daltonics, Bremen, Germany) was used to internally calibrate each mass spectrum using leucine enkephalin for single point calibration. Peaks lists (*m/z* and intensity) were retrieved, considering a minimum signal-to-noise ratio of 4.

Thus, the Grapevine dataset consists of FT-ICR-MS metabolomics data obtained in positive and negative modes of 3 biological replicates of 11 different grapevine genotypes. The information of the 11 different *Vitis* genotypes is presented in Table 2.1. Data from the positive and negative modes were treated independently.

**Table 2.1: Wild *Vitis* species, *V. vinifera* subsp. *Sylvestris* and *V. Vinifera* cultivars in the Grapevine Datasets.** Species, cultivar names, VIVC variety number, type of accession, origin (information adapted from https://www.vivc.de/) and abbreviations used to identify each species/cultivar are indicated. Table adapted from Marisa et al. [105].

| *Vitis* species | Subspecies (subsp.) or cutivar (cv.) | VIVC variety number | Abbreviation | Type of accession | Origin |
|---|---|---|---|---|---|
| *V. candicans Engelmann* | Vitis candicans engelmann | 13508 | CAN | Wild species | United States of America |
| *V. riparia Michaux* | Riparia Gloire de Montpellier | 4824 | RIP | Wild species | United States of America |
| *V. rotundifolia* | Muscadinia Rotundifolia Michaux cv. Rotundifolia | 13586 | ROT | Wild species | United States of America |
| *V. rupestris Scheele* | Rupestris du lot | 10389 | RU | Wild species | United States of America |
| *V. labrusca* | Isabella | 5560 | LAB | Wild species | United States of America |
| *V.vinifera* | Subsp. *sylvestris* | - | SYL | Wild plant | Portugal |
| | Subsp. *sativa* cv. Cabernet Sauvignon | 1929 | CS | Cultivated grapevine | France |
| | Subsp. *sativa* cv. Pinot Noir | 9279 | PN | Cultivated grapevine | France |
| | Subsp. *sativa* cv. Regent | 4572 | REG | Cultivated hybrid (crossing *V. vinifera* cv. Diana X cv. Chambourcin) | Germany |
| | Subsp. *sativa* cv. Riesling Weiss | 10077 | RL | Cultivated grapevine | Germnay |
| | Subsp. *sativa* cv. Trincadeira | 15685 | TRI | Cultivated grapevine | Portugal |

The total 33 samples were aligned together by a peak-based method using an in-house Python script made available in the metabolinks Python package (https://github.com/aeferreira/metabolinks) with 1 ppm *m/z* peak tolerance, generating a 2D-dataset with 5821 peaks in the negative mode and 30660 peaks in the positive mode.

### 2.1.2 Yeast Dataset

Yeast dataset was acquired by J. Luz [77] and is available at the git-hub repository: https://github.com/aeferreira/similarity_share ('5yeasts_notnorm.csv'). Briefly, the metabolome from five different yeast strains was analysed by FT-ICR-MS. Metabolites were extracted from cells collected at stationary phase of growth for each culture. For the analysis, extracted metabolite samples were diluted 100-fold in methanol:water (1:1) and human leucine enkephalin (Sigma Aldrich) was added for internal calibration of each mass spectrum ($[M+H]^+$ = 556.276575 Da). Formic acid (Sigma Aldrich, MS grade) was also added at a final concentration of 0.1% (v/v). Samples were analysed by direct infusion on SolariX XR 7-Tesla FT-ICR-MS, equipped with ParaCell (Brüker Daltonics). Spectra were acquired with an accumulation of 100 scans of 4Mb for each spectrum, at positive

electrospray ionization mode ($ESI^+$), in the mass range of 100 to 1200 *m/z*. Data Analysis 5.0 (Brüker Daltonics, Bremen, Germany) was used to internally calibrate each mass spectrum using leucine enkephalin for single point calibration. Peaks lists (*m/z* and intensity) were retrieved, considering a minimum signal-to-noise ratio of 4.

Thus, the Yeast dataset consists of FT-ICR-MS metabolomics data obtained in positive mode of 3 biological replicates of 5 different strains of *Saccharomyces cerevisiae*: the reference strain BY4741 (represented as BY) and 4 single-gene deletion mutants of this strain – ΔGLO1, ΔGLO2, ΔGRE3 and ΔENO1 represented respectively as dGLO1, dGLO2, dGRE3 and dENO1. These deleted genes are directly or indirectly related to methylglyoxal metabolism.

The raw data from the 15 samples were aligned using the MetaboScape 4.0 software (Brüker Daltonics, Germany) using the T-ReX (Time aligned Region complete eXtraction) algorithm with the following parameters: *m/z* delta = 1.10, Intensity Threshold = 0.00, Maximum Charge = +1. The peak lists were aligned in a bucket table, generating a 2D dataset with 21252 peaks.

Formulas were assigned to the *m/z* values using the MetaboScape 4.0 software (Brüker Daltonics), first with annotation from the HMDB [49] or YMDB [107] (Human and Yeast Metabolome Databases, respectively) metabolites list and, afterwards, with MetaboScape's *SmartFormula* algorithm (formula assignment algorithm) with the following parameters: *m/z* tolerance narrow 0.1 and wide 1.0 ppm and mSigma narrow 10 and wide 100. The elements considered for formula assignment were C, H, N, O, S, P with the 'Auto Upper Formula' option. The formulas assigned had to have at least one carbon and one hydrogen. The Senior and Lewis MetaboScape filter and the heuristic element count probability checks were applied. The allowed element ratios were the following: H/C – 0.2-3.1, O/C – 0.0-1.5, N/C – 0.0-1.3, S/C – 0.0-0.8, P/C – 0.0-0.3, P/O – 0.0-0.34. 17726 unique formulas were assigned to the 21252 peaks, 1652 of which were identified in multiple samples. The Yeast dataset filtered down to only peaks with assigned formulas will be referred as the Yeast Formula Dataset (YFD).

## 2.2 Binary Similarity – Data Pre-Treatment and Statistical Analysis

### 2.2.1 Data Pre-Treatment

Minor peak filtering was performed in the datasets by excluding peaks that only appeared in 1 sample since these features are uninformative, regardless of the pre-treatments made. Using this filtering, the negative Grapevine dataset was reduced from 5821 to 3629 peaks, the positive Grapevine dataset from 30660 to 7026 peaks and the positive Yeast dataset from 21252 to 1973 peaks.

After filtering, the same procedures described below were applied independently to the negative and positive GD (Grapevine Dataset), YD (Yeast Dataset) and YFD (Yeast Formulas Dataset): the Binary Similarity method proposed pre-treatment in this dissertation and with several combinations of more established, intensity-based, methods that were discussed in the introduction.

#### 2.2.1.1 Binary Similarity

The Binary Similarity treatment consisted of considering the occurrence of spectral features to construct a binary sample vector encoding feature presence as 1 and absence as 0, that is, changing all the intensity values (feature present in a sample) to 1 and change all missing values (feature not

present in a sample) to 0, obtaining a binary dataset (comprised of 0s and 1s). An example of this transformation is presented in Fig. 2.1.

| Samp./ Feat. | A-1 | A-2 | A-3 | B-1 | B-2 | B-3 |
|---|---|---|---|---|---|---|
| x | 163 | 124 | 189 | 200 | 176 | 223 |
| y | 115 | 108 | 101 | - | 98 | - |
| z | - | - | - | 165 | 126 | - |
| w | - | 98 | 103 | 879 | - | - |
| v | - | - | - | 176 | 215 | 245 |

| Samp./ Feat. | A-1 | A-2 | A-3 | B-1 | B-2 | B-3 |
|---|---|---|---|---|---|---|
| x | 1 | 1 | 1 | 1 | 1 | 1 |
| y | 1 | 1 | 1 | 0 | 1 | 0 |
| z | 0 | 0 | 0 | 1 | 1 | 0 |
| w | 0 | 1 | 1 | 1 | 0 | 0 |
| v | 0 | 0 | 0 | 1 | 1 | 1 |

**Figure 2.1: Example of the Binary Similarity (BinSim) treatment applied to an example dataset.** Feat. – Features; Samp. – Samples.

#### 2.2.1.2 Other Traditional Data Pre-Treatment Methods

Except for data transformed into a binary matrix, missing value imputation was performed by replacing missing values with half of the minimum intensity value present in the whole dataset. In this work, representative results of most established intensity-based methods will be presented. For that, one of each of the previously mentioned methods will be chosen: normalizations (normalization by a reference feature, in this case, leucine enkephalin – N), transformations (generalized logarithmic transformation – G) and centering/scaling (Pareto scaling – P). The methods were chosen due to their frequency in metabolomics data analysis. Since any of these methods can be used in combination with each other, results obtained with datasets treated in 3 different combinations (in the mentioned order) will be presented:

1. **P pre-treatment** – Pareto scaling only (eq. 1.5).
2. **NP pre-treatment** – Normalization by leucine enkephalin followed by Pareto scaling.
3. **NGP pre-treatment** – Normalization by leucine enkephalin, generalized logarithmic transformation (eq. 1.1 with λ equal to 1/10th of the minimum intensity value in the dataset as it is done in the commonly used software MetaboAnalyst 4.0 – see https://github.com/xia-lab/MetaboAnalystR) and Pareto scaling.

These pre-treatments were implemented in Python and are available in the metabolinks Python package (https://github.com/aeferreira/metabolinks).

After these pre-treatments, from each original dataset, 4 differently treated datasets were obtained which will be referred to as BinSim (Binary Similarity), P, NP and NGP depending on the treatment undertaken.

### 2.2.2 Statistical Unsupervised and Supervised Multivariate Analysis - BinSim

As already discussed, the objective of this part of the work is to compare the viability of using the simple pre-treatment method BinSim, which only focuses on the occurrence of spectral features in each sample, to the more established (intensity-based) pre-treatment methods used for discriminating different groups in the metabolomics data. Thus, with this objective in mind, the performance of different clustering and classification methods in discriminating the groups of the same datasets treated differently was compared. These methods were: unsupervised clustering analysis, more

specifically agglomerative hierarchical clustering and K-means clustering analysis, and supervised classification analysis, more specifically, Random Forest and PLS-DA.

#### 2.2.2.1 Statistical Unsupervised Analysis – Clustering

Agglomerative Hierarchical Clustering Analysis (HCA) with UPGMA (average) linkage method was performed on each dataset. The distance metric considered for the datasets treated with intensity-based pre-treatments (P, NP and NGP) was Euclidian, while for the BinSim treated datasets, binary dissimilarity metrics were chosen due to its binary nature. Nine different binary distance metrics (available in the scipy Python package [100]) were tested (Jaccard, Dice, Rogers-Tanimoto, Russell-Rao, Kulsinski, Yule, Sokal-Sneath and Sokal-Michener dissimilarities and Hamming distance [108]), from which 3 will be shown as they were considered representative of the results (to avoid repetition): Jaccard and Yule dissimilarities and Hamming distance.

- Jaccard dissimilarity: $d_{Jaccard}\,(S1, S2) = 1 - \frac{nº\,(S1 \cap S2)}{nº\,(S1 \cup S2)}$ $(\text{eq. } 2.1)$, where Si represents sample i and the intersection and union are based on the set of features each sample has.
- Yule dissimilarity: $d_{Yule}\,(S1, S2) = \frac{2 \times n_{10} \times n_{01}}{n_{11} \times n_{00} + n_{10} \times n_{01}}$ $(\text{eq. } 2.2)$, where $n_{ij}$ is the number of corresponding pairs of features in sample 1 and sample 2 equal to i and j respectively.
- Hamming distance: distance between 2 samples of the dataset is the proportion of disagreeing components (i.e. a feature is in one sample and not the other and vice-versa) of all the components in the dataset (including features missing in both samples).

The similarity between the dendrograms of the same dataset but with different pre-treatments performed was observed by using two correlation coefficients: the cophenetic correlation coefficient [74] and the Baker's gamma correlation coefficient [75]. These methods were adapted from the R package dendextend [109]. To test if the analysis led to a good discrimination of the different groups in each dataset, three metrics were developed and used to detect if the clustering of samples belonging to the same group is happening preferentially to the clustering of samples from different groups. The "correct clustering" percentage is the percentage of the groups (group based) whose samples all clustered together before any other sample clustered with a sample of said group – group "correctly clustered". Another metric used was called "Discrimination Distance" (DD). The DD of a dendrogram is the average DD of each group of the dataset. The DD of each group is 0 if it is not "correctly clustered" (defined in the same way as in the previous metric) or it is the distance between where all the samples of the group were correctly clustered and where another sample is clustered with that group, normalized by the maximum distance of any two samples in the HCA – see Fig. 2.2. Therefore, the DD will always be between 0 and 1. The final metric used was the "correct first cluster" percentage (sample-based). This is the percentage of samples whose first cluster was only with samples from its group (not all needed). As an example, in Fig. 2.2, the two right RL samples have a correct first cluster since the first time they cluster is with each other, while the remaining RL sample does not have a correct first cluster, since its first cluster is with both 3 CS samples and the 2 other RL samples. Thus, despite the RL group not being correctly well clustered, two of its samples have a correct first cluster. This means that the correct first cluster percentage will always be higher than the correct (group) clustering percentage. These metrics can give us information on how well the different groups are discriminated and it helps us better compare how different treatments in the same dataset affected the discrimination of different groups.

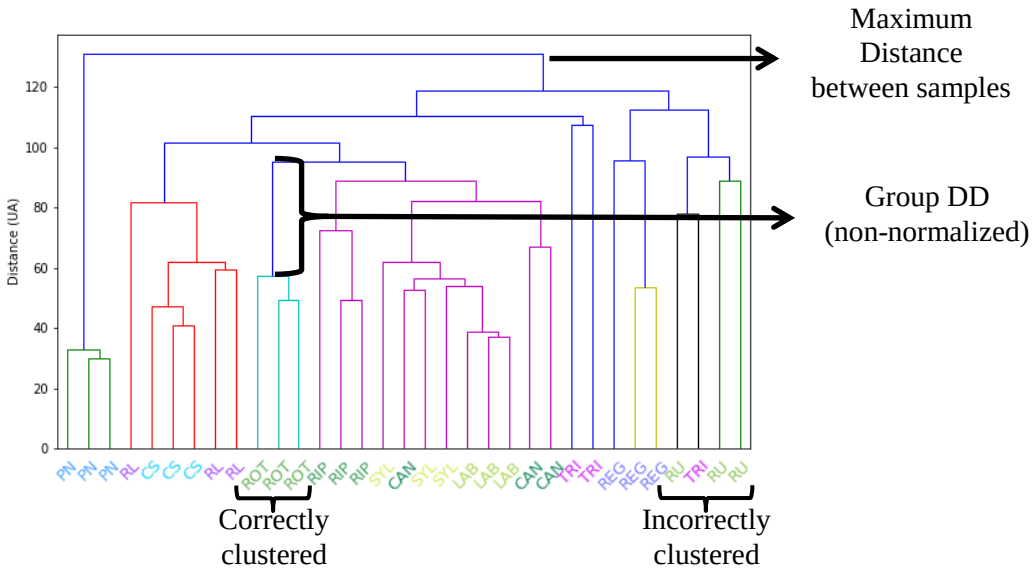



**Figure 2.2: Demonstration of "correctly" and "incorrectly" clustered groups and of the Discrimination Distance (DD) calculation for each group on an example dendrogram.**

K-means Clustering Analysis was performed on each differently treated dataset with cluster number equal to the total group number of the dataset (11 in the grapevine, 5 in the yeast data) and Euclidian distance metric (including for the BinSim treated datasets) using the scikit-learn Python module [101]. K-means clustering results can slightly vary due to the existence of local minima when trying to minimize the distance of the samples to the closest cluster centroid (K-means clustering optimization objective – minimize sum of squared distances of all samples to the closest cluster centroid) since the starting positions of the different cluster centroids is random [72]. Thus, the K-means clustering algorithm was iterated 150 times and the median results of all metrics used of the 15 cluster sets (10%) closest to the global minimum were taken. Three metrics were used to measure how well the groups were discriminated. Two metrics used had the same rationale as two of the ones used for the HCA (Discrimination Distance and correct clustering percentage – group-based metrics) but with a slight redefinition of the concept of "correct clustering". In this case, the distances are measured between cluster centroids and a clustering is correct if it contains all and only the samples of a single group. This is a stricter condition that the one imposed in the HCA, so a lower percentage of correct clustering is expected. The other metric used was the Rand Index adjusted for chance (sample-based metric).

#### 2.2.2.2 Statistical Supervised Analysis – Random Forest and PLS-DA

The classifiers chosen to use as a comparison test between the differently treated datasets were Random Forest and PLS-DA. These models were built using the scikit-learn Python module [101]. Due to the datasets used having a low number of replicates (samples in each group), validation of the model and the results was done by internal stratified 3-fold cross-validation (number of samples in each group is 3) [78]. The performance of the models was judged based on their prediction accuracy estimated by the average accuracy of the stratified 3-fold cross-validation. Since the combination of samples randomly selected to each fold can affect the results (even if slightly), especially with low sample size in each group, this process was iterated 200 times and the mean accuracy of all iterations

was taken as a global metric for the cross-validation evaluation. Moreover, permutation tests (with 1000 iterations each) were used to further assess if the models were significant.

The number of trees used for the Random Forest classifiers were tuned to 200, the number for which the predictive accuracy of the models did not increase further in any of the different datasets (Fig. 3.3 and Suppl. Fig. 6.4 – already stabilized with 100 trees in all datasets), while not being too computationally intensive. Other parameters were left as the default values used in the scikit-learn function. For each model built, the Gini Importance of each feature was calculated [87]. Then, an ordered list of the average importance of each feature across each iteration and each different combination of training and testing sets was compiled. The top 2% of features considered most important to build the models for each dataset were taken and their relevant characteristics were evaluated, namely, the number of samples and different groups those features appeared in.

Partial Least Squares (Projection to Latent Structures) – Discriminant Analysis (PLS-DA) classifiers were built for each differently treated dataset using the PLS2 – NIPALS algorithm implemented in the *PLSRegression* module from the scikit-learn Python package [101]. The default parameters in scikit-learn were used, except the scaling of the samples, not performed since data was already pre-treated. The number of components for the PLS-DA models were chosen to minimize the predictive residual sum of squares (maximize $Q^2$) computed from stratified 3-fold cross-validation choosing 11 components for the 4 Negative Grapevine Datasets (treated in different ways), 13 for the Positive GD ones, 4 for YD and the YFD ones – Fig. 3.4 and Suppl. Fig. 6.5. Group membership was encoded by the one-hot encoding method.

In group membership predictions, test samples were assigned to the group corresponding to the maximum value in $y_{pred}$ (vector with *n* numbers, each a measure of similarity to a group) output of the PLS – maximum DR. As it was done for the random forests, the top 2% of features considered most important to build the models were taken and the number of samples and different groups those features appeared in were counted. Here, the Variable Importance in Projection (VIP) was used to estimate the importance of each feature to build each model [82].

## 2.3 Sample Mass-Difference Networks – Data Pre-Treatment and Statistical Analysis

### 2.3.1 Mass-Difference Network Construction

An extra peak filtering step was applied to the Yeast Dataset (YD) to discard *m/z* peaks over 1000 and merging features that had the same formula assigned by the *SmartFormula* algorithm of MetaboScape 4.0 (Brüker Daltonics). After this step, the YD had 1893 *m/z* peaks. A Mass-Difference Network (MDiN) was built for the Yeast Dataset (YD), for the Negative and for the Positive Grapevine Datasets (GD) using the MetaNetter 2.0 plugin [110] of Cytoscape 3.8.1 [111] with an accepted error margin of 1 ppm and the transformations used described in Table 2.2. The nodes of the different networks were the neutral masses of the peaks for each of the network. These were annotated in the YD by the *Bucket Labels* given by the MetaboScape 4.0 software, in the Negative GD by adding the mass of a proton and in the Positive GD by subtracting the mass of a proton (≈ 1.0073 Da) from the *m/z* peaks. Edges were established between nodes with difference in masses very close to a specific set of mass differences (within 1 ppm deviation).

Each of the mass differences in the set of mass differences mentioned is called a “Mass-Difference-based Building block” (MDB). The MDB corresponds to the mass of a specific overall change in the

metabolite elemental formula due to a simple biochemical reaction (enzymatic or non-enzymatic). For example, a methylation corresponds to the substitution of a –H hydrogen atom in a metabolite by a –$CH_3$ methyl group, leading to an overall change of a $CH_2$ and a change in mass of 14.01565 Da. The choice of the set of MDBs is crucial in building the MDiNs, since the structure of the network directly depends on these. The objective was to choose a set of MDBs that represent changes caused by the most common and ubiquitous chemical reactions in biological systems, while still maintaining the metabolite formula charge neutrality. For example, to maintain neutrality, a phosphorylation would mean the overall addition of a $PO_3H$ group – addition of a $-PO_3^{2-}$ group + 2 $H^+$ (maintaining neutrality) to replace an H atom in a metabolite. To this end, the set of MDBs should encompass a considerable percentage of reactions that happen in a biological context with a relatively small number of groups – a total of 15 groups were picked. All the MDBs chosen represent changes in metabolites of no more than 5 atoms and less than 80 Da (small size). Each MDB should represent a set of chemically known reactions and a change in every main element in metabolites (C, H, O, N, S and P) is represented by at least one of the MDBs. To fulfil these conditions, representative MDBs were searched using BRENDA ([112], https://www.brenda-enzymes.org/). The list of MDBs chosen as best candidates were compared with works that used MDiNs such as Breitling et al. [95] and Tziotis et al. [94]. The MDBs that were considered to build the MDiNs were the following:

**Table 2.2: List of MDBs used to build the MDiNs.** Elemental Transformations (represented by the overall elemental change in the metabolite), their masses (Δ Mass, Da) which correspond to specific changes in the elemental composition of a metabolite and examples of types of reactions represented by each MDB.

| Elemental Transformations | Δ Mass (Da) | Reaction Types |
|---|---|---|
| **O (-NH)** | 0.984016 | Deamination |
| **$NH_3$ (-O)** | 1.031634 | Transamination |
| **$H_2$** | 2.015650 | Hydrogenations / Dehydrogenations |
| **$CH_2$** | 14.015650 | Methylations |
| **O** | 15.994915 | Oxygenations / Hydroxylations |
| **$H_2O$** | 18.010565 | Condensation / Dehydration / Cyclization |
| **NCH** | 27.010899 | Transfer of a formidoyl group |
| **CO** | 27.994915 | Formylation |
| **CHOH** | 29.002740 | Hydroxymethylation |
| **S** | 31.972071 | Transfer of a –SH group |
| **$C_2H_2O$** | 42.010565 | Acetylation |
| **CONH** | 43.005814 | Transfer of a carbamoyl group |
| **$CO_2$** | 43.989829 | Carboxylation / Decarboxylation |
| **$SO_3$** | 79.956815 | Sulphation |
| **$PO_3H$** | 79.966331 | Phosphorylation |

Analysis of the number of nodes, edges, size of the biggest component, diameter and radius of the networks and number of isolated nodes in the networks was made using the networkx Python module [104].

In this methodology, an edge is established between two masses, independently of the rest of the nodes (masses) or edges in the network. Thus, the edges established between a list of masses is the same whether that list of masses are the only nodes in the network or integrated in a bigger network. Thus, a subgraph of said list of masses in the bigger network built would be isomorphic to the network built with only the list of masses. Consequently, the MDiNs for each sample of each of the datasets were

built by inducing the subgraph of only the nodes that represent *m/z* peaks present in each sample – sample MDiNs or sample networks.

### 2.3.2 Mass-Difference Network Analysis and Secondary Dataset Construction

The analysis of the different sample MDiNs was done in 5 different ways: using 3 different measures of network centrality – degree, betweenness centrality and closeness centrality –, a metric based on the MDBs used to establish edges – "MDB influence" – and a metric to compare network topology – GCD-11 (Graphlet Correlation Distance using 11 graphlet orbits), [113]. The results from each of these analyses in each sample MDiN were used to compile a "secondary dataset" for each analysis method with different features according to the method. Thus, a set of 5 different secondary datasets was made for the YD, for the Negative and for the Positive GD.

For each of the three centrality measures used (degree, betweenness and closeness centrality), its value for each node in the different sample networks were compiled on a secondary dataset (one for each centrality measure), where each feature is representative of a node in the sample MDiNs, thus creating datasets with the same number of features to the original datasets.

Another secondary dataset was made by counting the number of times each MDB (mass difference / chemical transformation) was used to establish an edge between two masses in each sample MDiN. These counts were normalized by the total number of edges established in each sample network to represent the percentage of edges each MDB established in each sample network. This methodology is referred hereafter as "MDB influence" – the impact of each MDB in building the sample networks. The features of the dataset are the 15 MDBs used to build the sample networks (greatly reducing the number of features on the original datasets).

The last network analysis method used to build a secondary dataset was the Graphlet Correlation Distance including the 11 non-redundant orbits of up to 4-nodes graphlets (GCD-11) to analyse the topology of the network [113]. Fig 2.3 shows the orbits and graphlets of up to 4-nodes graphlets. Graphlets are small and non-isomorphic sub-graphs of a network and each graphlet can have multiple automorphic orbits if the nodes in the graphlet are not "symmetric" that is, are not in the same relative position [114]. In this method, a matrix is built where each row is the Graphlet Degree Vector of a node in the network constructed by counting the number of times that node is in each of the 11 orbits. The Spearman correlation between the columns (the orbits) of said matrix makes an 11 by 11 symmetric matrix called the Graphlet Correlation Matrix (GCM) that should be representative of the network topology. Usually, the Euclidian distance between the upper triangular matrices is used to compare two different networks (thus the Correlation Distance). Here, instead of this procedure to calculate a distance between two networks, the secondary dataset was built by compiling the upper triangular matrix of the GCM of each sample network as the sample information (a column). The features were, therefore, the 60 orbit *n* – orbit *m* Spearman correlations with *n* and *m* being 2 different of the 11 orbits [113,115].

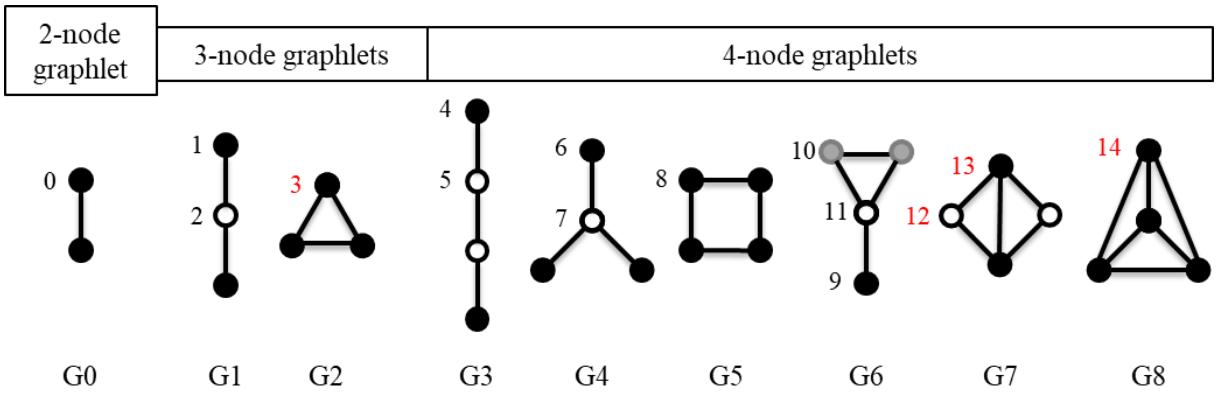



**Figure 2.3: Representation of all 9 unique graphlets up to 4 nodes (G0, G1, …, G8) and their 15 automorphism orbits (0, 1, …, 14).** The different orbits in each graph are represented by the different coloration of the nodes (nodes with the same coloration in a graphlet have equivalent orbits). Orbit numbers coloured black are the 11 non-redundant orbits used in the GCD-11 method, and orbit numbers coloured red are 4 redundant orbits of up to 4-node graphlets not used in the GCD-11 method. Figure based on [114].

As for nomenclature, each secondary dataset will be identified by the following system: Main Network – Analysis method. For example, the yeast dataset sample networks analysed by degree will be identified as: YD – Degree. The other 4 analysis methods would be mentioned as: YD – Betweenness, YD – Closeness, YD – MDB Influence and YD – GCD-11.

### 2.3.3 Statistical Unsupervised and Supervised Multivariate Analysis – MDiNs

The statistical multivariate analysis performed on the secondary datasets was similar to the described in the 'Statistical Unsupervised and Supervised Multivariate Analysis – BinSim' section (section 2.2.2). Briefly, the methods employed were the following:

Agglomerative Hierarchical Clustering Analysis (HCA) was performed on each dataset with UPGMA (average) linkage method and Euclidian distance metric. The quality of the discrimination of the different groups by the clustering was assessed with the Discrimination Distance, the correct clustering and correct first cluster percentages (metrics explained earlier). K-means Clustering Analysis was performed using the Euclidian distance metric with cluster number equal to the total group number in each case – 5 in the secondary datasets obtained from the YD network and 11 from the Negative and Positive GD. This was made with the scikit-learn Python module [101]. Each analysis was iterated 150 times and the median results of the 15 (10%) best set of clusters (evaluated by the minimization of the sum of squared distances to the cluster centers – objective function of the K-means clustering analysis algorithm) for 3 metrics were used: the Discrimination Distance, the correct clustering percentage, and the adjusted Rand Index.

Random Forest and PLS-DA models were built for each secondary dataset using the scikit-learn Python module [101] with validation of said models done by internal stratified 3-fold cross-validation. The performance of the models was assessed by their mean predictive accuracy of the test groups in each cross validation set over 200 iterations of random sampling of the data into 3 stratified folds. Random Forest models were made with 200 trees (after tuning – Suppl. Fig 6.10). The average importance of the 15 features from the MDB influence secondary datasets obtained from the YD, Negative GD and Positive GD to build the aforementioned Random Forest models were estimated by the Gini Importance (calculated using the scikit-learn Python library), [87,101]. Optimization of the number of components used to build the PLS-DA models was made by the minimization of the Predictive Residual Sum of Squares of the respective PLS Regressions (Suppl. Fig. 6.11). PLS-DA models were built (using the PLS2 – NIPALS algorithm implemented in the scikit-learn Python module [101]) with 5 components for the secondary datasets obtained from the YD network and for

the betweenness centrality, MDB influence and GCD-11 secondary datasets obtained from the Negative and the Positive GD and with 11 components for the degree and closeness centrality secondary datasets obtained from the Negative and the Positive GD. Biological group membership was encoded by the one-hot encoding method. The decision rule employed was the max DR rule – each sample was assigned to the group corresponding to the maximum value of the $y_{pred}$ output of the PLS-DA. Permutation tests for the Random Forest and PLS-DA models (with 1000 iterations each) were used to further assess the significance of the PLS-DA models (Suppl. Fig. 6.12).

# 3. Results and Discussion

This section encompasses two parts:

1) The first part will focus on comparing the performance of different unsupervised and supervised statistical analysis methods in discriminating samples into their respective groups in datasets treated with the Binary Similarity (BinSim) data pre-treatment, described in section 2.2.1.1 (Materials and Methods), and treated with traditional, intensity-based methods, described in section 2.2.1.2 (Materials and Methods).
2) The second part will focus on the development of a characteristic Mass-Difference Network (MDiN) for each sample considering a set of "Mass-Difference-based Building blocks" (MDBs) that represent sets of chemical reactions. The constructed sample networks will be analysed by different network analysis methods. Results from the analysis of different aspects of each network will be used to build a "secondary dataset" for each analysis method. Then, to test if the sample networks constructed can be characteristic of the group that the sample belongs to, the performance of unsupervised and supervised statistical analysis methods in discriminating samples into their respective groups based on the secondary datasets built will be evaluated. Finally, the performance of the discrimination achieved will be compared to the results obtained in 1).

## 3.1 Binary Similarity as a Data Pre-Treatment

The four datasets analysed were: the Negative and Positive Grapevine Dataset (Negative GD and Positive GD), the Yeast Dataset (YD) and the Yeast Dataset filtered for features with formulas assigned (YFD). Each of these datasets were treated in 4 different ways: the Binary Similarity (BinSim), the new treatment proposed in this dissertation, and 3 different combinations of 3 different treatments – normalization by the reference feature leucine enkephalin (N – a normalization treatment), generalized logarithmic transformation (G – a transformation treatment) and Pareto scaling (P – a centering/scaling treatment). These particular data treatments were chosen due to their commonality in metabolomics data analysis [25,35]. Thus, as an example, the datasets obtained from the Yeast Dataset, according to the pre-treatment applied, will be referred to as:

- **YD – P –** only treated with Pareto scaling.
- **YD – NP** – treated with normalization (by leucine enkephalin) followed by Pareto scaling.
- **YD – NGP** – treated with normalization (by leucine enkephalin) followed by generalized logarithmic transformation and Pareto scaling.
- **YD – BinSim** – treated with the Binary Similarity pre-treatment.

The Binary Similarity (BinSim) pre-treatment was envisioned as a reliable and simpler alternative to more established intensity-based pre-treatments. Furthermore, it was specifically created with metabolomics data analysis in mind. It focuses on the presence or absence of features from the different samples instead of the highly variable intensity-driven focus of the other pre-treatments. The BinSim pre-treatment consists of encoding all intensity values (feature present in a sample) as 1 and all missing values (feature not present in a sample) as 0, obtaining a binary dataset comprised of 0s and 1s.

To test the viability of the Binary Similarity method in highlighting relevant information from metabolomics datasets, the performance of multiple unsupervised and supervised statistical methods in discriminating the various groups present in each of the datasets treated with the BinSim pre-treatment

and the same datasets treated with the other methods mentioned was compared. Furthermore, the important features to build the classifier models built by supervised statistical methods for each differently treated dataset was analysed to observe if the BinSim treated data offers a new perspective on the same original data (discrimination achieved by looking at a different set of features). Since the Binary Similarity treatment was developed with the intention of also reducing the peak filtering steps, only minor peak filtering was performed in the datasets (only peaks that appeared in only one sample were excluded).

To avoid redundancy, only the results for the Negative GD and the YD are presented in this section. The complementing results of the Positive GD and YFD are presented in the Annexes.

### 3.1.1 Unsupervised Statistical Analysis – Hierarchical and K-means Clustering

Clustering techniques are unsupervised methods where the algorithms employed do not use the information of group membership of the samples [66]. These methods were used to see if there is an intrinsic pattern in the data brought out by the different treatments that allows the discrimination of samples belonging to different groups and how the success of the discrimination was affected when using the BinSim pre-treatment. Both Hierarchical Clustering and K-means Clustering were employed.

Hierarchical Clustering Analysis (HCA) was performed on each differently treated dataset with UPGMA (average) linkage method with the resulting dendrograms shown in Fig. 3.1 for the Negative GD and the YD – 4 in each case for each different treatment (for the BinSim pre-treatment, here are shown the results using only one binary distance metric – the Jaccard dissimilarity). For the Positive GD and YFD, the results are presented in the Suppl. Fig. 6.1.

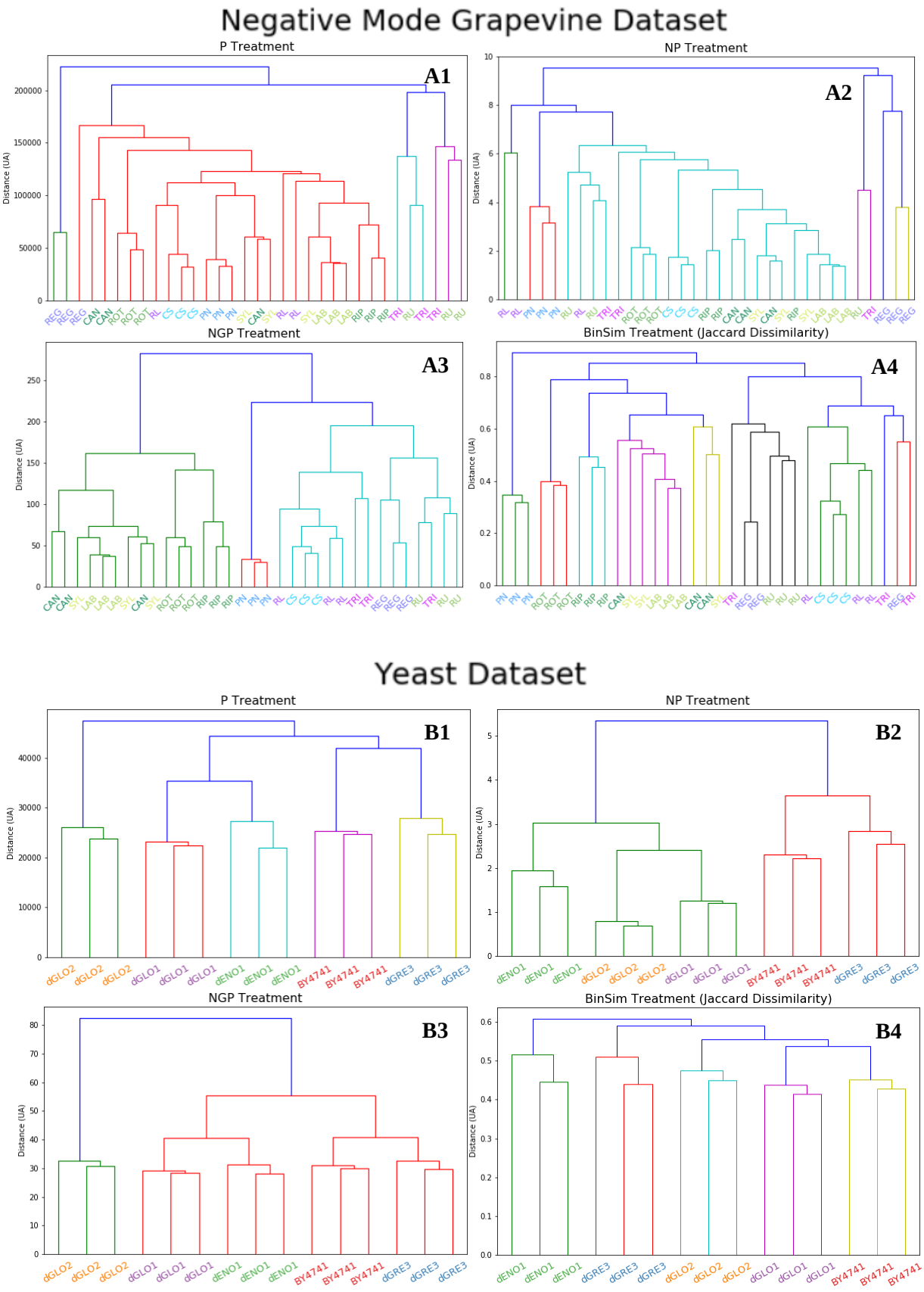



**Figure 3.1: Hierarchical Clustering Analysis (HCA) dendrograms of the Negative Grapevine Dataset (A) and Yeast Dataset (B).** The datasets were treated with the P (**1**), NP (**2**), NGP (**3**) pre-treatments using Euclidian distances or the BinSim pre-treatment (**4**) using Jaccard Dissimilarity distance metric. HCA was performed with UPGMA linkage method. Pre-treatments: P – Pareto scaling, N – Normalization by leucine enkephalin, G – Generalized logarithmic transformation, BinSim – Binary Similarity; *Vitis* genotypes abbreviations are indicated in Table 2.1.

For the Hierarchical Clustering analysis of the YD in Fig 3.1B, all 5 yeast strains have all their samples well discriminated from each other in all 4 dendrograms (4 treatments). All replicates are grouped with each other. This dataset represents the case of a dataset where the samples of each group are very distinct from each other as can be observed in Fig 3.1B to test if the BinSim pre-treatment keeps this information from the dataset. From the dendrograms made from the negative GD in Fig 3.1A, it can be seen that the different grapevine groups are less "well" discriminated. From the 11 different varieties, only around half of them have all their samples clustering together before clustering with any other sample – "correctly clustered". Thus, this dataset represents the case where the samples of each group are very similar to each other to test if the discrimination power is similar or even higher in datasets treated with BinSim pre-treatment in relation to the intensity-based methods when meaningful information is harder to extract. Overall, especially in the GD – NGP and GD – BinSim datasets (Fig 3.1A3-A4), the *Vitis vinifera* cultivars (in shades of blue and purple) tend to be closer together with each other than to the wild *Vitis* species (in different shades of green) except for the RU variety (*Vitis rupestris du lot*). Analysing each group, the PN cultivar samples tend to cluster well together and are distant from other varieties. ROT, RIP, LAB and CS varieties' samples also tend to be well clustered among the different treatments. On the other hand, the TRI, REG and RU varieties are less well defined, as their samples appear to be similar since they seem to mix up with cluster of the other varieties as easily as they cluster with samples of the same variety. All dendrograms obtained seem, at a glance, very similar independently of the treatment made and indeed they lead to the same conclusions about trends in the data as analysed here. This means that, at a first look, the data treated with BinSim retains as much information as the other treated datasets to discriminate between the different groups.

To have a more concrete and objective measure of this similarity, the cophenetic correlation coefficient [74] and the Baker's Gamma correlation coefficient [75] between all pairs of dendrograms were calculated and are presented, respectively, in Fig 3.2 and Suppl. Fig 6.2 for the Negative GD and the YD and in Suppl. Fig. 6.3 for the Positive GD and the YFD.

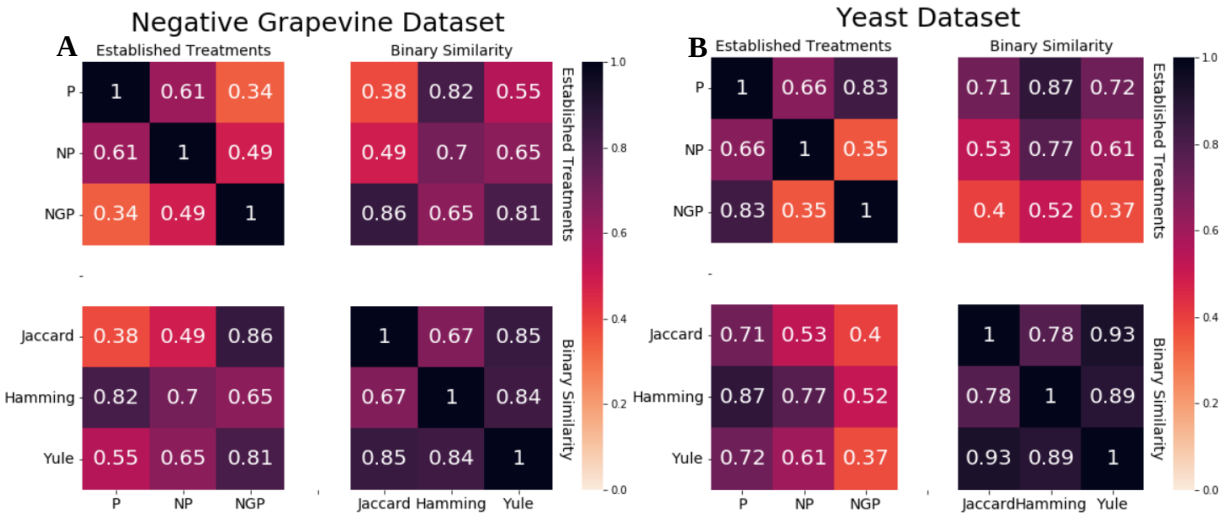



**Figure 3.2: Heatmaps of the Cophenetic Correlation Coefficient between the dendrograms of all differently treated dataset pairs of the Negative Grapevine Dataset (A) and of the Yeast Dataset (B).** For the datasets treated with the Binary Similarity pre-treatment, 3 representative binary distance metrics were used: Jaccard, Hamming and Yule dissimilarities/distances. For the others, Euclidian distance was used. Pre-treatments: P – Pareto scaling, N – Normalization by leucine enkephalin, G – Generalized logarithmic transformation.

The heatmaps in Fig. 3.2 show that all cophenetic correlation coefficients between the different dendrograms for each dataset (Negative GD and YD) are higher than 0, in fact, most of them show strong positive correlations. This was also observed in the Baker's Gamma correlation coefficient (Suppl. Fig 6.2) and in the correlations calculated for the Positive GD and YFD dendrograms (Suppl. Fig 6.3). Furthermore, there are not noticeable differences in correlations between different intensity-based pre-treatments and between these and the BinSim pre-treatment. In fact, for example, the Negative GD – NGP has a higher correlation with the Negative GD – BinSim (with the different binary distance metric) than to the GD – P or GD – NP treatment. As for the 3 different binary distance metrics used, Negative GD – BinSim and YD – BinSim with the Hamming distance seem to have slightly higher correlations with the P, NP and NGP treated datasets but the dendrograms made with all the three binary distance metrics are highly correlated (so any of the three distance metrics lead to very similar results) and are also positively correlated with the dendograms treated with the other treatments (computed using the Euclidian distance). This shows that all the treatments employed, including BinSim, are revealing the same trends in the data, leading to similar clustering.

Taking this into account, the next question is to test if the hierarchical clustering was joining samples of the same strain/variety/group preferentially, that is, if there was an intrinsic pattern in the data (after treatment) which led, without outside information, to the discrimination of samples from different groups. To this end, three metrics were used to analyse the discrimination of the samples in the clustering – the Discrimination Distance (DD), the correct clustering and correct first cluster percentages described in the section 2.2.2.1 (Materials and Methods). To reiterate, a "correct clustering" was here defined as all the samples in a group clustering together before clustering with a sample of any other group. The results obtained from these analyses are presented in Table 3.1 (Negative GD and YD) and in Supplementary Table 6.1 (Positive GD and YFD).

**Table 3.1: Discrimination Distance, correct clustering and correct first cluster percentages of the HCA of the Negative Grapevine and Yeast Datasets after different treatments.** Binary Similarity has 3 different results based on the distance metric used. Pre-Treatments: BinSim – Binary Similarity, P – Pareto Scaling, N – Normalization by a reference feature, G – Generalized Logarithmic Transformation.

| | Grapevine Dataset (ESI⁻) – 11 groups | | | | | | Yeast Dataset – 5 groups | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Dataset / Treatments / Metrics | P | NP | NGP | Jaccard (BinSim) | Hamming (BinSim) | Yule (BinSim) | P | NP | NGP | Jaccard (BinSim) | Hamming (BinSim) | Yule (BinSim) |
| Discrimination Distance | 0.10 | 0.12 | 0.14 | 0.12 | 0.12 | 0.15 | 0.31 | 0.22 | 0.22 | 0.14 | 0.19 | 0.35 |
| Correct Clustering (%) | 45 | 45 | 54 | 54 | 45 | 45 | 100 | 100 | 100 | 100 | 100 | 100 |
| Correct First Cluster (%) | 64 | 64 | 79 | 67 | 70 | 64 | 100 | 100 | 100 | 100 | 100 | 100 |

For the Yeast Dataset, as expected, all groups in all cases were perfectly discriminated with 100% correct clustering percentage. The Discrimination Distance changes slightly between datasets from YD – BinSim with the Yule dissimilarity metric and YD – P with higher DDs at 0.35 and 0.31 to the lower DD of the YD – BinSim with the Jaccard dissimilarity metric (0.14). The more informative results come from the GD where all datasets had correct (group) clustering percentages between 45 and 54 % (5 or 6 groups correctly clustered), DDs between 0.10 and 0.15 and correct first clusters between 64 and 79% (21 to 26 samples with a correct first cluster). Thus, very similar results were obtained between the different HCA with Negative GD – NGP results being slightly better followed by the Negative GD – BinSim results (using either Jaccard or Hamming distance metrics). Results for the

Positive GD and YFD in the Supplementary Table 6.1 are in line with this, with the BinSim treated datasets having similar results to the others. In fact, in the Positive GD, despite the overall poorer discrimination of the groups if compared to the Negative GD, the Positive GD – BinSim using the Yule or Jaccard dissimilarities achieves a considerable better discrimination than the other treatments – 45% of groups are correctly clustered compared to a maximum of 27% in other cases. Furthermore, the Positive GD – BinSim dendrograms using any of the three binary distance metrics had higher correct first cluster percentages (52, 61 and 70% – Suppl. Table 6.1) in comparison to the dendrograms made from the traditionally treated datasets (maximum of 48%) with more than half of the samples having a "correct" first cluster. Although these discrimination results are not ideal if the objective was to discriminate the different grapevine varieties based on this data, it means that even in a dataset with 'murky' information, BinSim performance in extracting information is comparable or slightly better to that of intensity-based methods.

It is worth noting that the correct (group) clustering percentage and DD are very sensitive to outliers with the "correct clustering" definition used, since just one stray sample from a group can lead to that entire group being labelled as not "well clustered". However, this is not a problem for the results obtained since each dataset used here only has 3 replicates. Hence, one sample being an outlier in the group corresponds to a hefty part of the group and should be considered. It would be remiss to not say that these methods are not suited to be directly applied to test the clustering efficiency for datasets with higher number of samples per group. In these cases, they should be adapted. For example, a change would be to consider a "correct clustering" as *x*% of samples of a group clustering together (80 or 90% for example) instead of all samples to account for possible outliers in bigger datasets. On the other hand, the correct first cluster percentage should be resistant to outliers as is.

To corroborate the results obtained, another clustering technique was used: K-means Clustering using the scikit-learn Python module [101]. This method clusters samples in a pre-determined number of clusters. The number of clusters chosen for each dataset was equal to their number of groups: Negative and Positive GD – 11 groups, YD and YFD – 5 groups. Correct clustering percentage and Discrimination Distance (group-based) metrics were adapted for this method (see section 2.2.2.1). Since, in this method, *k* clusters are made instead of samples being progressively clustered, the "correct clustering" definition was altered to a stricter "all and only the samples of a group are in one cluster." As such, lower correct clustering percentages are expected in relation to what was obtained with Hierarchical Clustering. Moreover, a 3$^{rd}$ metric was used, the Rand Index, a measure of the proportion of the pair of samples which are correctly clustered or correctly not clustered together (adjusted for the expected percentage of samples which would be in those situations randomly). The results are presented in Table 3.2 for the Negative GD and YD and in Supplementary Table 6.2 for the Positive GD and YFD. K-means Clustering analysis uses the projection of the samples in the *pp*-dimensional space with *pp* equal to the number of features, therefore, binary distance metrics cannot be used to cluster the samples. So, the distance metric used was Euclidian for all the different datasets, including those treated with BinSim.

**Table 3.2: Discrimination Distance, correct clustering percentage and adjusted Rand Index of the K-means Clustering analysis of the Negative Grapevine and Yeast Datasets after different treatments.** Pre-Treatments: P – Pareto Scaling, N – Normalization by a reference feature, G – Generalized Logarithmic Transformation.

| | Grapevine Dataset (ESI⁻) | | | | Yeast Dataset | | | |
|---|---|---|---|---|---|---|---|---|
| Dataset / Treatments / Metrics | P | NP | NGP | Binary Similarity | P | NP | NGP | Binary Similarity |
| Discrimination Distance | 0.09 | 0.13 | 0.17 | 0.16 | 0.73 | 0.39 | 0.37 | 0.86 |
| Correct Clustering (%) | 18 | 27 | 36 | 27 | 100 | 100 | 100 | 100 |
| Adjusted Rand Index | 0.52 | 0.48 | 0.59 | 0.53 | 1 | 1 | 1 | 1 |

As expected, the correct clustering percentage in the GD sharply decreased from the obtained with Hierarchical Clustering. However, once again, the GD – BinSim enables a similar discrimination to the other treatments, although GD – NGP has the higher correct clustering percentage (4 out of 11 groups) and a slightly higher Discrimination Distance (0.17) than GD – BinSim (0.16). The adjusted Rand Index results are all between 0.48 and 0.59 showing that the samples are being correctly placed in the clusters many more times than what would be expected at random, once again reaching the conclusion that there is some intrinsic information and characteristics in the dataset which allows the correct discrimination of the samples in their groups. GD – NGP also has the higher Rand index which indicates that the samples are being correctly clustered more times comparing to the other treatments, with GD – BinSim having again the 2$^{nd}$ highest value with 0.53. The analysis of the YD led again to the perfect separation of all samples in their respective clusters so all datasets have 100% correct clustering and Rand Index equal to 1. Thus, the information here lies in the Discrimination Distance where the YD – BinSim outperforms every other pre-treatment scoring 0.86, which means that each group is close to be equidistant to any other group in the *pp*-dimensional plane. The YD – P also has a very high DD at 0.73 while YD – NP and YD – NGP trail farther behind with DDs slightly below 0.4. In this case, the BinSim pre-treatment amplified further the already very meaningful differences between each group. On the other hand, in the Positive GD, GD – NP has a considerable better performance than all other treatments. However, it is only able to discriminate 23% (3 or 4 groups) of the groups with a 0.49 Rand index (Supplementary Table 6.2). The BinSim treated dataset here leads to worse results in the Positive GD (compared to the GD – NP) despite its better performance on the Positive GD with Hierarchical Clustering. Overall, K-means clustering analysis was not able to discriminate the different groups of the Positive GD regardless of the pre-treatment method, when compared to the Hierarchical Clustering Analysis.

From these clustering methods, we reach the conclusion that, in all datasets studied, data treated with the Binary Similarity revealed the same trends and information in the data leading to identical conclusions about it, whether it contained clearly distinct groups or more similar and less distinct groups. It was also observed that the dendrograms made were very similar and that all methods had similar results, with no single treatment having performed consistently better than others.

### 3.1.2 Supervised Statistical Analysis – Random Forests and PLS-DA Classifiers

Following the conclusion of the previous section and since the discrimination of the samples in their respective groups is a key part of this work, the natural progression was to compare the performance of different classifiers to discriminate the samples in their groups after pre-treatment with the different methods. Furthermore, the more important features to build the classifier models were computed to

assess if the BinSim pre-treatment gives relevance to parts of the data different from the other methods, as was hypothesized. For this comparison, two different classifiers were applied: Random Forest and PLS-DA using the scikit-learn Python package [101]. PLS-DA is a dimension reduction classifier that was chosen due to its popularity in the metabolomics data analysis workflow [31,78]. Random Forest is an algorithm based on the majority decision of an ensemble of decision trees [85] that make a series of binary decisions based on the values of one feature in each decision. This classifier was chosen due to the nature of the binary choices in its decision trees. The rationale was that the binary choices would be able to extract the information present in a binary dataset comprised of zeroes and ones, such as the ones obtained after the BinSim pre-treatment by choosing features where ones and zeroes being separated would lead to clear division of a single group or a set of groups. Therefore, a better performance of Random Forest classifiers with datasets treated with BinSim was expected, comparing to PLS-DA classifiers performance.

#### 3.1.2.1 Random Forest and PLS-DA Classifiers – Prediction Accuracy

Since the objective was the discrimination of the different groups in the data, the performance of the models was evaluated by their average predictive accuracy based on an internal stratified 3-fold cross-validation procedure. This internal stratified 3-fold cross-validation was chosen to mitigate the low number of samples in each group – only 3 [78]. Yet, this means that each group's training data is only comprised of 2 training samples for each model which weakens the reliability of the model. Knowing this, measures were taken to improve the fidelity of the results obtained. For Random Forest models, the number of trees was tuned to 200 since, as you can see in Fig. 3.3 and Suppl. Fig. 6.4, the average predictive accuracy of the datasets with all the different treatments had already stabilized and stopped increasing (fluctuating around a certain set of values) from 100 trees onwards, while also not being too computationally intensive. For PLS-DA models, the number of components were chosen based on the minimization of the predictive residual sum of squares (maximization of $Q^2$) estimated with stratified 3-fold cross-validation – Fig 3.4 (Negative GD and YD) and Suppl. Fig. 6.5 (Positive GD and YFD). The 4 different treatments on the same datasets had a similar performance ($Q^2$) with the increase in the number of components, so choosing the same number of components for each one was possible. The results obtained led to 11 components being chosen for the 4 Negative Grapevine Datasets, 13 for the positive GD and 4 for both the YD and the YFD.

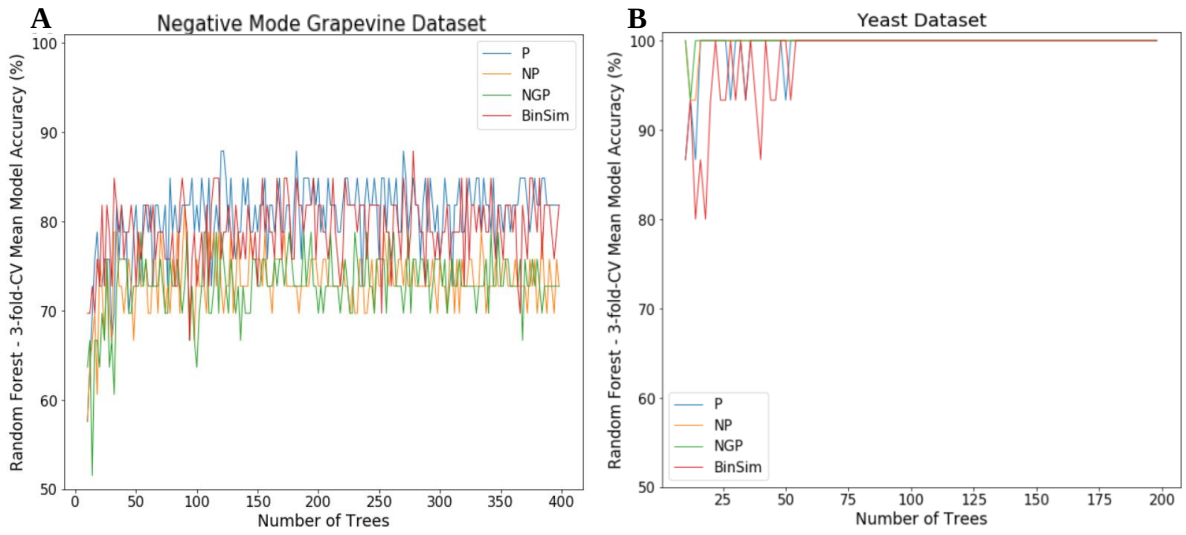



**Figure 3.3: Tuning of the number of trees used to build the Random Forest models.** Random Forest predictive accuracy as a function of the number of trees used in the forest for the Negative Grapevine (**A**) and Yeast (**B**) datasets with different pre-treatments. Accuracy was estimated by stratified 3-fold cross-validation. Pre-Treatments: BinSim – Binary Similarity, P – Pareto Scaling, N – Normalization by a reference feature, G – Generalized Logarithmic Transformation.

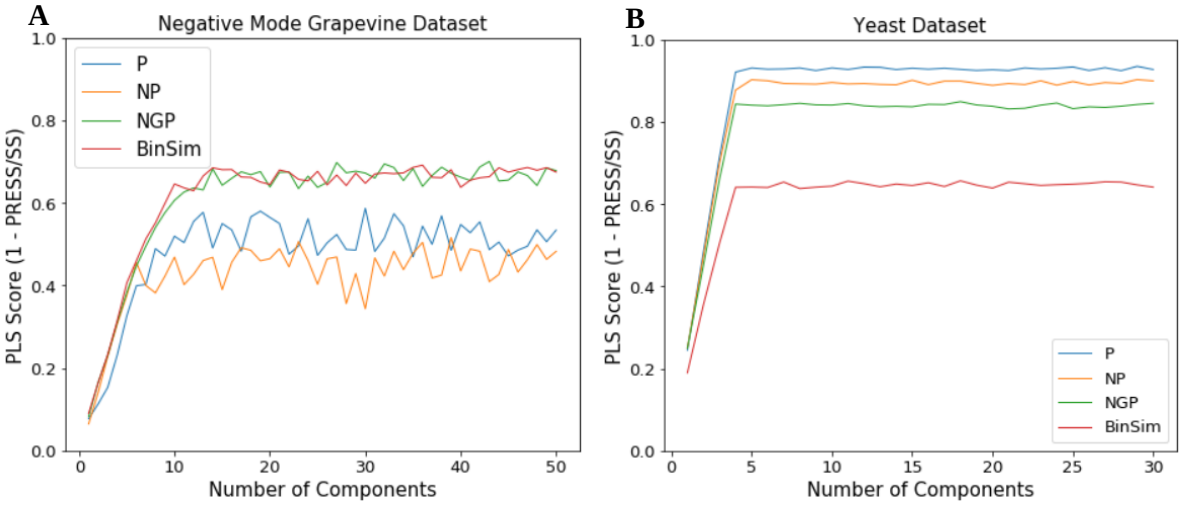



**Figure 3.4: Optimization of the number of components used to build the PLS-DA models.** 1- (Predictive Residual Sum of Squares (PRESS) / residual Sum of Squares (SS)) or $Q^2$ estimated by stratified 3-fold cross-validation of PLS regressions of the Negative Grapevine (**A**) and Yeast (**B**) datasets with different number of components. Pre-Treatments: BinSim – Binary Similarity, P – Pareto Scaling, N – Normalization by a reference feature, G – Generalized Logarithmic Transformation.

To further mitigate the issue of having only 2 training samples within each group, stratified 3-fold cross-validations were repeated 200 times (iterations) with random sampling of the folds. Figure 3.5 shows the distribution of the average prediction accuracy of the Random Forest and PLS-DA models for the Negative GD and YD estimated by stratified 3-fold cross validation for each of the 200 iterations on a violin plot with the average accuracy of the 200 iterations being presented below the graphs. Results for the Positive GD and YFD are shown in Suppl. Fig 6.6 (Supplementary Data). The significance of the accuracy of the models built was assessed by permutation tests that are shown in Suppl. Fig. 6.7, where the predictive accuracy of permuted labels models (1000 permutations),

estimated by stratified 3-fold cross-validation, were compared to the predictive accuracy of a random iteration of the corresponding non-permuted model. For all models built from different combinations of datasets and treatments, the predictive accuracy distribution of permuted labels models was considerably below the non-permuted model accuracy (*p*-value = 0.001), which means that the classifiers' accuracy resulted from significant information present in the data and not from random noise. This can also be concluded for the BinSim pre-treatment.

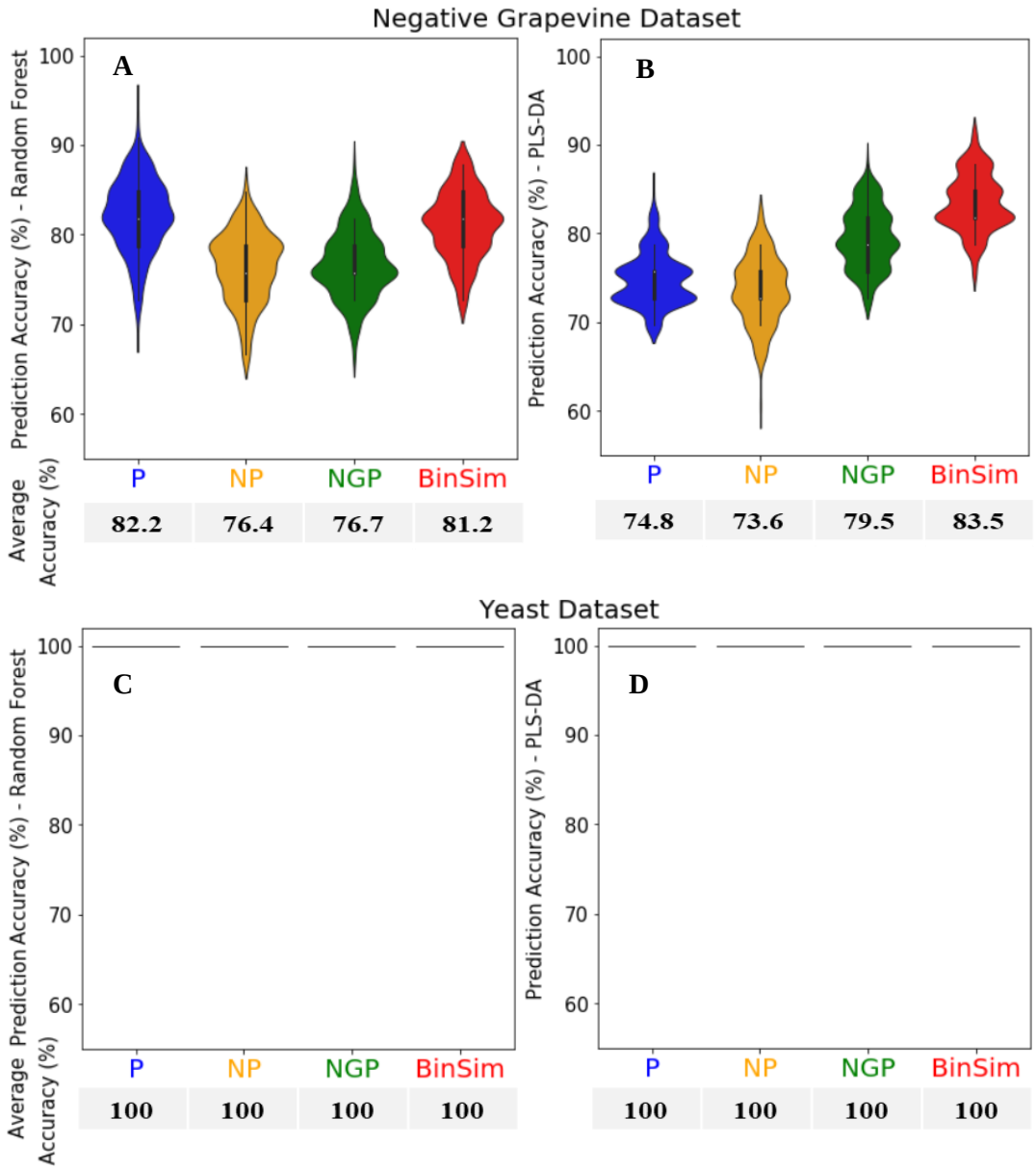



**Figure 3.5: Distribution of the prediction accuracy of Random Forest and PLS-DA models.** Violin plots of the distribution of the prediction accuracy of 200 iterations of Random Forest and PLS-DA models built based on the Negative Grapevine dataset (**A** and **B**, respectively) and on the Yeast dataset (**C** and **D**, respectively). Each iteration's accuracy is estimated by stratified 3-fold cross-validation. Each iteration randomly splits the dataset in three folds. Below the plots, the average prediction accuracy is presented. BinSim – Binary Similarity, P – Pareto Scaling, N – Normalization by a reference feature, G – Generalized Logarithmic Transformation.

The distributions of predictive accuracy for both Random Forest and PLS-DA models of the Negative GD are very similar, situated around the 80% mark, with a great overlap amongst the different treatments. This leads to the conclusion that the models built from each differently treated dataset have approximately the same discrimination power. The average accuracy of the different datasets in Random Forests varies between 76.4 and 82.2% and in PLS-DA between 73.6 and 83.5%. The GD – BinSim had the 2nd best accuracy in Random Forest models (Fig 3.5A) with 81.2% close to the 82.2% of the GD – P and higher than the GD – NP and GD – NGP which have accuracies around 77%. These results can also be observed in Fig. 3.3A (since the predictive accuracy was also used to tune the number of trees), where GD – BinSim and P had, almost always, a higher prediction accuracy than GD – NP and NGP independently of the number of trees used for the Random Forest, which gives more confidence to the results obtained. GD – BinSim also had the best accuracy in PLS-DA models (Fig 3.5B) with 83.5% higher than the 79.5% of the GD – NGP and much higher than the 74.8 and 73.6% of the GD – P and GD – NP respectively. As for the Positive GD (Suppl. Fig. 6.6A,B), similar results are observed. Once again, PLS-DA model prediction accuracy is around 80% for all datasets, with Positive GD – NGP and BinSim above 80% and GD – NGP having the best average accuracy (84%). Positive GD – P has a smaller average accuracy at 76.2% and a high spread of values in its distribution. On the other side, the Random Forest models have a much lower average accuracy (Suppl. Fig 6.6A). Positive GD – BinSim has the highest average accuracy at 71.1% closely followed by GD – P with 70.2%. However, these are much higher than the average accuracy of Positive GD – NP and GD – NGP that are below the 50% mark. This difference in accuracy between the GD – P and BinSim and the GD – NP and NGP can also be clearly observed in Suppl. Fig 6.4 in models built with different number of trees. Despite the much lower performance in this case, the BinSim treated dataset still performs as well or better than the other methods chosen for comparison.

Thus, despite the similar performances of the different models (especially on the Negative GD), the GD – BinSim models seem to perform slightly better than the other datasets. Even more, it seems more consistent since it is the best or close to the best dataset in all the 4 methods used (Random Forest, PLS-DA and also in Hierarchical Clustering and K-means Clustering analysis) in the 4 datasets studied. The information extracted from this dataset led to consistently good discrimination power (comparatively) between the different groups. Ideally, the prediction accuracy of the grapevine models should be closer to 100%, but to evaluate the viability of the BinSim pre-treatment, this works as a good proof of concept with difficult to discriminate data.

The good performance of BinSim treated datasets might also be a consequence of the big inherent variability of the intensity data of FT-ICR-MS [116] that reduces the efficiency of the other treatments. Since in every model built each group is represented by only 2 training samples, trying to discern intensity patterns that reliably discriminate the different groups is a very difficult endeavour and, therefore, more prone to errors since an incidental pattern in the 2 training samples might not be replicated by the corresponding test sample leading to misclassifications. A hypothesis is, then, that the BinSim pre-treatment robustness might be increased (in comparison to other treatments) on low to medium-sized datasets where the amount of training samples to build a classifier is limited and relying on intensity patterns may be even more prone to errors because of their variance. This could possibly be explored in future studies.

Surprisingly, and contrary to expectations, PLS-DA models based on the Negative GD – BinSim dataset have a better prediction accuracy than Random Forest models (83.5 against 81.2), as opposed to other treated datasets such as the Negative GD – P and GD – NP where Random Forest models had the higher performance. So, the binary decisions made in decision trees were not more suited to discriminate the groups present in a binary matrix in relation to other supervised classifiers. The same

was observed for the results of the Positive GD with PLS-DA models having an average accuracy of 81.0% to the 71.2% of the Random Forest models.

For the remaining datasets, both the YD (Fig 3.5C,D) and the YFD (Suppl. Fig 6.6C,D) in both Random Forests and PLS-DA models had a perfect predictive accuracy (100%) in the discrimination of the 5 groups on the models built. This leads to the conclusion that the BinSim pre-treatment did not discard substantial and essential information by ignoring intensity data to hamper a perfect discrimination. However, by analysing the key features chosen by the different algorithms, it can be tested if the discrimination happened based on a different set of features.

#### 3.1.2.2 Random Forests and PLS-DA Classifiers – Important Features

Now that it was established that the discrimination power of different statistical methods on BinSim treated datasets is similar to datasets treated with the other, intensity-based, methods, the next step is to assess if this is, as hypothesized, achieved by "looking at the information differently". That is, if the treatment made is actually giving more weight to information/features that are usually ignored and the different statistical methods are using that information for the discrimination of the different groups. To this end, the features more relevant to building the models were considered, followed by an evaluation of whether the sets of such features and their overall characteristics are very different between methods, giving a special relevance to the comparison with the BinSim pre-treatment. To determine which features are "important" for the development of the models, different "feature importance" metrics, specifically, the Gini Importance for Random Forest models [87] and the Variable Importance in Projection (VIP) for PLS-DA models [82] were used (estimated with stratified 3-fold cross-validation and averaged for the 200 iterations). The top 2% of features considered overall important were taken, that is, 73 features in Negative GD, 141 in the Positive GD, 39 in the YD and 33 in the YFD. When these methods are used for feature selection, usually a much greater number of features are selected since the interest is on features that influence the model in a noticeable way even by a small amount. For example, for the VIP metric, a usual cut-off point to keep a feature in feature selection is 1 [82]. For these datasets, this would mean hundreds and sometimes even more than a thousand features would be selected. However, here the aim is to compare the more essential features and it is in that interest that only such a small number of features were selected. The overlap of the sets of important features is represented in the Venn diagrams in Fig. 3.6A,B,E,F (Negative GD and YD) and Suppl. Fig. 6.8A,B,E,F (Positive GD and YFD), to see if there were more different and unique important features chosen in the GD – BinSim case. Since the BinSim pre-treatment focuses on the occurrence of spectral features, the number of different groups and, more importantly, the number of samples the features appeared in were considered as their main characteristics. Fig. 3.6C,D,G,H (Negative GD and YD) and Suppl. Fig. 6.8C,D,G,H (Positive GD and YFD) swarm plots show the distribution of the number of samples the important features appear in as well as the average number of samples and groups where they appear. Table 3.3 shows the average percentage of the percentage of "unique important features" (only important for the dataset/classifier combination with a specific treatment) for Random Forest, PLS-DA or both (that can be seen in the Venn diagrams of Fig. 3.6 and Suppl. Fig. 6.8) for each of the tested treatments when applied in the 4 main datasets (Negative and Positive GD, YD and YFD).

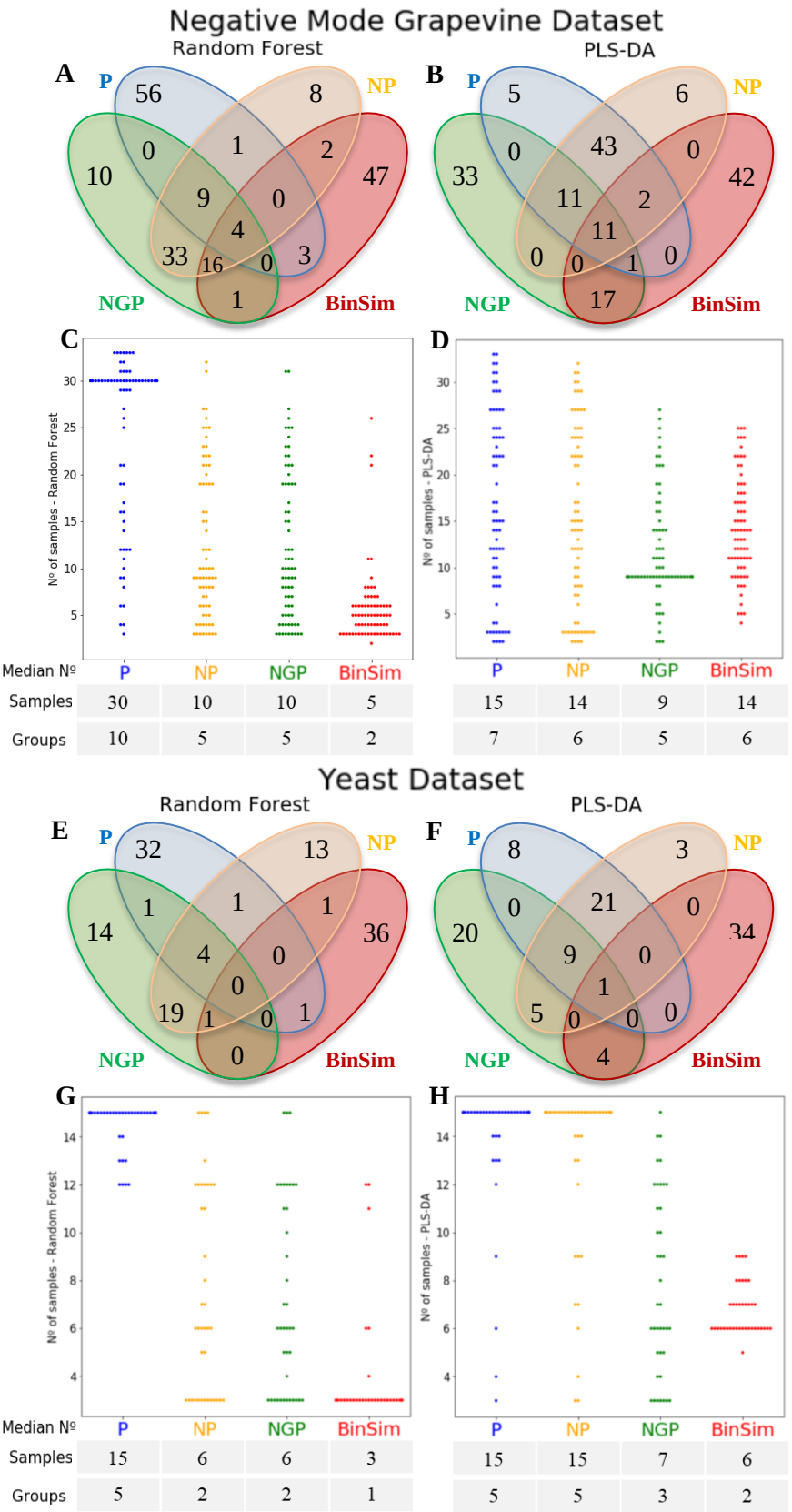



**Figure 3.6: Characteristics of the most important features used to build the Random Forest and the PLS-DA models.** Venn diagrams of the 2% most important features used to build the Random Forest (by the Gini Importance method) and the PLS-DA (by the VIP method) models built based on the Negative Grapevine Dataset (141 features, **A** and **B**, respectively) and Yeast Dataset (33 features, **E** and **F**, respectively). Distribution plots of the number of samples each important feature appears in their dataset on each differently treated Negative Grapevine Dataset (**C, D**) and Yeast Dataset (**G, H**) with the median number of samples and different groups they appear in below the plots. Pre-Treatments: BinSim – Binary Similarity, P – Pareto Scaling, N – Normalization by a reference feature, G – Generalized Logarithmic Transformation.

**Table 3.3: Percentage of unique features in each set of the 2% of most important features to build Random Forest or PLS-DA models.** The percentage is calculated by averaging the percentage of unique features for each of the 4 pre-treatments and for Random Forest or PLS-DA models. Pre-treatments: P – Pareto Scaling, N – Normalization by a reference feature, G – Generalized Logarithmic Transformation.

| | Percentage of unique important features in the different models (%) | | | |
|---|---|---|---|---|
| **Models** | **P** | **NP** | **NGP** | **Binary Similarity** |
| **Random Forest (RF)** | 76.2 | 22.9 | 23.1 | 75.6 |
| **PLS-DA** | 16.8 | 12.3 | 47.9 | 71.2 |
| **Combined (RF + PLS-DA)** | 46.5 | 17.6 | 35.5 | 73.4 |

For simplicity, the 2% of features chosen as important to build the PLS-DA or Random Forest models in the "X" treated dataset will be hereafter mentioned as the important features of the "X" (or "X" treated) dataset. Similar results for other PLS-DA model importance metrics (sum of the regression coefficients and the X-weights for each feature) were obtained so only the results for the VIP metric were shown. Accounting for both Fig. 3.6 and Suppl. Fig. 6.8, the Venn Diagrams show that BinSim treated datasets have the most unique important features in 6 of the total 8 (4 datasets with one of two models: Random Forest or PLS-DA) presented cases and have the 2$^{nd}$ most unique features after the P treated datasets in the 2 remaining cases (Random Forest models of the Negative and Positive GD), making it the treatment with the most average "unique important features" overall – 73.4% (Table 3.3). As for the distribution of the number of samples the features appear in, Random Forest and PLS-DA models chose features in BinSim treated datasets that consistently (in 7 of 8 cases) appeared in a lesser number of samples (and different groups) except for the PLS-DA models made from the Negative GD that, as we are going to see later, presented multiple oddities to all other cases. This seems to point towards a conclusion that, indeed, BinSim treated dataset emphasize the information present in a very different set of features with different characteristics, specifically, that have a bigger number of missing values, that is, features that are exclusive to one or very few groups in the dataset (approaching the concept of "biomarkers").

For the results in the 4 sets of datasets (Negative GD and YD presented in Fig. 3.6 and Positive GD and YFD presented in Suppl. Fig. 6.8), there are apparent patterns and similarities they all share with the exception of the PLS-DA models of the Negative GD (Fig. 3.6B,D) that breaks some of these trends. It can be seen that the distribution of the number of samples where the chosen features appear is very different in Random Forest and PLS-DA models, showing that the rationale to build classifier models for the two methods is quite different and demonstrating the importance of researchers knowing the concepts behind the statistical methods they use to take the most advantage of their data and the methods to answer the hypothesis proposed.

Starting with the intensity-based pre-treatments, both the important features of the P and NGP treated datasets have a similar distribution of the number of samples they appear in between the Random Forest and PLS-DA models (with the exception already mentioned). Most of the P treated datasets important features appear in almost every sample in the datasets (15 in the yeast datasets and 30 plus in the grapevine datasets) and, consequently, every group. In this facet, this is the complete opposite to the aim of the BinSim pre-treatment, as it is also apparent when observing the distribution of the number of samples important features appear between these two pre-treatments. The conclusion is that the information regarding the presence and absence of features is almost completely discarded and the

discrimination of the groups is made almost entirely by trying to discern different intensity patterns between the groups in features present in almost all samples.

On the other hand, the number of samples NGP datasets' important features appear in is very spread out without a particular bias for samples that appear in a high or low number of samples. Furthermore, there is a slightly higher occurrence of features that appear in multiples of 3 samples, especially on the yeast datasets (where there is a reason to believe there are more features that appear exclusively in some of the groups because of the excellent predictive accuracy of the models of these datasets treated with BinSim). This is important when recalling that in every dataset each group has 3 samples, so the higher occurrence can be explained by the fact that there may be some preference for features that only appear in some groups (if it was just a combination of appearing in 0, 1, 2 and 3 samples of each group, there wouldn't be a generalized increase of features that appear in exactly 3, 6 and 12 samples when compared with their direct "neighbours" in both yeast datasets). This is an indication that there is still some consideration for the presence and absence of the features, which tracks when realizing that, from the intensity-based methods, the NGP's logarithmic transformation decreases more the difference between higher values, such as different high-intensity values in comparison to the difference between smaller values, such as between imputed missing values and low intensity peaks, separating more the "missing values" (absence of the features) to the usual intensity values (present features), and giving them, therefore, a slightly greater importance (compared to other traditional methods).

The important features found when the models were built from datasets treated with NP have a very peculiar behaviour. The distribution of the number of samples where the features occur is almost identical to the one in NGP treated datasets for the Random Forest models (features well spread); while it is almost identical to the P treated datasets in PLS-DA models (mostly features that appear in almost all samples). The same can be seen in the Venn diagrams, where there is a great overlap of the important features of Random Forest models between NP and NGP treated dataset and of the important features of PLS-DA models between the NP and P treated datasets. This also helps explain the results presented in Table 3.3, where NP treated datasets have a low percentage of unique important features in each dataset (on average below 20%) compared to all other treatments due to this overlap to NGP datasets in Random Forests models and P datasets in PLS-DA models. A similar trend is found in the results of the P and NGP treated datasets where P datasets have a lot of unique important features in Random Forest models (even slightly higher on average that BinSim datasets) but a really low amount in PLS-DA models due to the overlap with the NP datasets' important features, while the opposite happens to the NGP datasets. This also explains why the average percentage of unique features in these two cases is between the low percentages of NP datasets and very high percentages in BinSim datasets. This finding also led to the identification of this exact pattern happening on the predictive accuracy of these models as seen in Fig. 3.5 and Suppl. Fig. 6.6. The predictive accuracy of models built based on the NP treated datasets is close to NGP treated datasets if the model is a Random Forest and to P treated datasets if the model is a PLS-DA.

Finally, regarding the new proposed pre-treatment, BinSim, its important features are quite different from the other pre-treatments, with an average of 73.4% of unique features, almost 30% higher than the next following average (Table 3.3). This is because, consistently, for both PLS-DA and Random Forest models, they exhibit a very small overlap with the sets of important features of datasets treated in other ways. For the remaining BinSim datasets' non-unique important features, these have a larger overlap, that is, also are chosen as important features, with the NGP treated datasets, making the latter the closest to BinSim datasets' important features. As explained earlier, from the intensity-based methods, NGP's logarithmic transformation decreases more the difference between higher intensity

values in comparison to the difference between smaller values (found between imputed missing values and low intensity peaks), giving some importance to the absence or presence of features that the BinSim treated datasets represent, hence explaining the slight NGP and BinSim similarity.

Furthermore, important features of the BinSim treated datasets appear in a much smaller number of samples and different groups than the important features in the other datasets (except for Negative GD PLS-DA models). However, the distribution of the feature occurrence by number of samples is very different in Random Forest and PLS-DA models. In Random Forest models, they mostly appeared in a very small number of samples ranging between 3 and 6, that is, features that were exclusive to one group or sometimes two groups. This was more apparent on the Yeast datasets where 33 of the 39 on the YD (Fig. 3.6G) and 27 of the 33 on the YFD (Suppl. Fig. 6.8G) important features appeared in only 3 samples (number of samples of each group). These features are chosen because they help to clearly identify and separate the samples belonging to one or sometimes two groups in one decision node in the binary decision trees made in the Random Forest models. For the Random Forest model of the Positive GD (Suppl. Fig. 6.8C), there is also a clear group of features that occur in a great number of samples from around 22 to 30 after a zone from 10 to around 21 samples where almost no important feature is present. This trend can be found in the other datasets and models, but is more noticeable here. This kind of feature represents the opposite of the concept of "biomarker" by being metabolites that are present in almost all groups with the exception of one or two.

On the other hand, in PLS-DA models, the important features tend to occur in approximately half of the samples. For example, in YD and YFD (Fig. 3.6H and Suppl. Fig. 6.8H), most features occur between 6 and 9 samples (with a higher number on 6 samples) of the 15 total samples, that is, in 2 or 3 of the 5 different yeast strains. This difference to Random Forests may be attributed to the fact that each component in PLS-DA is trying to maximize the group separation between all groups instead of prioritizing individual group separation. In the example of the yeast datasets, when features can only have 2 values (1 or 0), this happens when the contribution of a feature to a component separates half of the groups between each other, that is, 2 groups from the other 3 groups, therefore features that appear in 2 or 3 groups only (appear between 6 and 9 samples) are prioritized. This tendency is also maintained in the Negative and Positive GD (Fig. 3.6D and Suppl. Fig. 6.8D) where chosen features tend to appear in near half of the total samples in the dataset but with a much higher spread due to the higher number of groups and samples and the groups being less well defined.

Regarding the results for the important features of the PLS-DA models of the Negative Grapevine Dataset in Fig. 3.6B,D (often mentioned exception), the Venn diagram of the overlap of the important features is consistent with what was observed in the other cases; however the distribution of the occurrence of features by number of samples is quite different. Here, for the P treatment, the distribution of the occurrence of important features in the data is spread between appearing in low to high numbers of samples, instead of appearing almost exclusively in 30 plus samples like in all other results, while, for the NGP, the distribution is less spread out and is heavier at 9 which is also uncharacteristic of the NGP treated datasets. The important features of the Negative GD – NP follow their usual trend of being very close to P datasets in PLS-DA models. No discernible reason for these changes was found. The important features of the BinSim treated Negative GD are the only set that follows the trend displayed in other datasets with their distribution being very similar to the one observed in Positive GD (Suppl. Fig. 3.6D), however the change of the other distributions decreased the median of their distributions, making this the only set where the BinSim dataset's important features does not have the lowest value for the median.

One important note to take away from the analysis is that classifier models built after the BinSim pre-treatment use features that appear in a low number of samples, specifically features almost exclusive to only one group (acting as "biomarkers"). This is important since no artificial emphasis was put on this kind of features and the information stored in them is echoed in features that are present in almost every group except one or two. Therefore, this leads to the conclusion that there are more features that act like "biomarkers" than the opposite (only do not appear in one or two groups). These features have a lot of missing values. Features with many missing values are often discarded during peak filtering [43]. This indicates that these highly informative features that can greatly help sample discrimination are often filtered out of the dataset during peak filtering. When these features are kept, it is usually because feature filtering was done on an individual group basis (for example, if a feature is present in most of the samples of a single group, it is kept even if it is not found in any other samples). This kind of filtering would potentiate, in theory, the results of a method based on the occurrence of spectral features such as BinSim (highlights biomarker-like features), as was observed when this filtering was performed on the grapevine datasets leading to a high increase in the performance of different methods in BinSim treated datasets (results not shown). Nevertheless, this happens because samples of the same group were artificially made more similar to each other by this method. Thus, in my opinion, when using internal validation methods to specifically analyse the discrimination of samples into their respective groups, these types of filtering are not suitable since the "test samples" used to validate the models were made artificially closer to the training samples. An example of the use of these filters in discrimination analysis would be applying it only on the training samples when the model will be validated by an external metric so that "testing samples" are always treated as truly "unknown samples".

### 3.1.3 The Rationale and Benefits of Using Binary Similarity

The Binary Similarity (BinSim) pre-treatment was specifically created with metabolomics data analysis in mind as a simpler and reliable alternative to traditional pre-treatments. It focuses on the presence or absence of features from the different samples instead of the intensity-driven (in the case of mass spectrometry metabolomics data) perspective of the other pre-treatments. It considers only the occurrence of spectral features to construct a binary sample vector encoding feature presence as 1 and absence as 0, obtaining a binary dataset comprised of 0s and 1s. Therefore, this method requires lower amounts of peak filtering from the original data since the existence of missing values is also a source of information. Even more, since missing values are such a valuable source of information, the method benefits from lower amounts of peak filtering altogether. This method also allows skipping the choice of a missing value imputation method (since all missing values are changed to 0) and the subsequent choice of the pre-treatments to use. Hence, the application of this treatment is very simple and skips the ambiguity of choosing the 'best' methods to apply in the peak filtering, missing value imputation and pre-treatment (choices of which combinations and methods of scaling, normalizations and transformations to use) in the metabolomics workflow.

The presence or absence of features in a set of samples can be very helpful in the discrimination between samples from different sources (belonging to different biological groups) especially for metabolites that are exclusive to one group of samples (and act as biomarkers in the context) or to only a few of the studied groups; or the opposite with some key metabolites absent from just one or two groups. However, this kind of information tends to be overshadowed by the intensity data in the usual workflow due to 1) the extensive peak filtering usually performed, 2) the subsequent missing value imputation and 3) the nature of the traditional pre-treatments. The peak filtering tends to exclude features with higher amounts of missing values [43]. Depending on the method used for the filtering,

this can also include those features exclusive to one or a small number of groups, overlooking the importance of these features that can greatly help group discrimination. The missing value imputation on the remaining missing values is an almost mandatory step for further statistical analysis since many statistical methods do not work with the absence of values in the data [43]. When these values are treated as mistakes in the acquisition or processing of the data, that is, as MAR/MCAR values [43], their potential original importance is again eliminated (although this is the correct procedure to counteract errors in acquisition or processing if they are suspected to be such). When they are treated as MNAR values, that is, absent or in very low concentrations [44], they are usually replaced by small values and (only) partially retain their information and importance as low concentration/absent metabolites (decreases difference to present features). As such, low intensity features can be closer to (originally) missing values than to high intensity features and are considered more similar to them in many statistical methods, diminishing the importance of actually identifying a feature. Finally, traditional pre-treatments are all intensity transformations, from the mathematical point of view, so even when low amounts of peak filtering are performed (as it was done in this work), the focus falls mostly on the intensity patterns of the features, as was observed (Fig. 3.6).

Thus, the idea of the BinSim pre-treatment is to focus specifically on the occurrence of spectral features by discarding the intensity data. This idea is supported by intensity data being highly variable between different metabolomics experiments even in the same batch of analysis due to slight differences in the sample preparation, ionization efficiency of the samples, sample processing, etc., which reduces the reproducibility of metabolomics results. Lin et al. [116] shows the difficulty in having reproducible intensity and relative quantification results between two different laboratories that are analysing the same set of samples with the same protocol and different instruments, despite a good portion of the same metabolites being annotated in both analyses. So, it stands to reason that, by eliminating this high variance factor, it is possible to more consistently compare the different samples with higher reliability based on the (less variable) identified metabolites. In fact, the discrimination results obtained by different statistical methods in BinSim treated datasets were more consistent that with intensity-based pre-treatments with them being either the best or second-best results of the 4 pre-treatments compared in almost all cases between the different statistical analysis methods and between the different datasets used (perfect discrimination in the YD with all statistical methods; around half of the groups correctly clustered in unsupervised statistical methods and 80% predictive accuracy in supervised statistical classifiers in the Negative GD, for example). Furthermore, since BinSim also focuses on a different aspect of data, as observed by the very unique set of important features in building Random Forest and PLS-DA models from BinSim treated datasets (73.4% of the selected important features were unique on average), it gives a different perspective of them. It focuses on features that act as or close to “biomarkers” for specific biological groups in the dataset that are many times ignored or overshadowed due to factors explained above in this section. Thus, BinSim could also be used as a complementary approach to any of the other treatments mentioned. In fact, this observation is an indicator that both avenues of analysis should be made to give a more global and in-depth look at the data, instead of only looking at one aspect of it.

With this in mind, BinSim did perform consistently as well or slightly better than the other treatments used in this work in all different statistical methods used, and in all tested datasets for the profiling and discrimination of samples while using information different from the information used by other methods (as observed by the set of important features used to build Random Forest or PLS-DA models).

### 3.1.4 Chemical Formulas as Features in Analysis across Different Datasets

The results for the Yeast Dataset (YD – features were peaks *m/z*) and the Yeast Formula Dataset (YFD – features were the formulas assigned to peaks *m/z* when possible) were very similar with all different statistical analysis methods used. This was expected since most features in the YD had a formula assigned and were, therefore, also present in the YFD; hence the results from the YFD were only presented in the Annexes. Nevertheless, it was relevant to show that the perfect discrimination of the different yeast strains was still possible using only *m/z* peaks that had formulas assigned. Although more tests on different datasets are needed, these preliminary results show that this kind of feature engineering does not hinder (at least noticeably) information extraction by the different multivariate analysis methods. The importance of this lies in the much bigger versatility of using formulas as features in comparison to *m/z* peaks, specifically, to compare to other datasets.

As we mentioned earlier, intensity data of mass spectrometry experiments is highly variable [116] and depends heavily on the sampling, metabolite extraction, instrument used, instrument settings, ionization efficiency of the metabolite, processing of the data, etc. As such, even comparing between samples or technical replicates obtained in the same experiment batch can reveal these irregularities that are even higher when comparing samples that should be similar (belong to the same group such as culture extract from the same yeast strain in similar conditions) but were prepared at a different time by a different person, analysed at a different spectrometer and, maybe, with some slight differences in parameters. This was observed by Lin et al. [116] which noted the variability of intensity of the same metabolites and the same samples was present between different batches of the experiment performed by the same laboratory and was magnified further when repeated in different laboratories, showing the low reproducibility of relative quantitation in untargeted metabolomics experiments. On the other hand, they were able to get a good intersection of the same metabolites annotated. This low reproducibility of the intensity data between studies makes the building of classifiers and comparison across different datasets difficult. Thus, the fact that the BinSim pre-treatment focuses on occurrence of spectral features that has a lower variability than that of intensity data makes a possible cross dataset analysis more feasible.

However, the alignment of *m/z* peaks of different datasets that have slight shifts due to the small error in mass spectrometry analysis can be another problem for this analysis. By using formulas as features instead of *m/z* peaks, the alignment between the different datasets becomes much more straightforward (comparing the number of metabolites in common between the different datasets), especially if reliable formula assignments are possible. This means that there is significant potential in analysis across different datasets when using both the BinSim pre-treatment and formulas as features since both were shown in this work to keep the relevant information of the dataset to discriminate between the different groups. In my opinion, future studies could capitalize on these findings by building a classifier model from a dataset and observe if it can be used to reliably classify samples of another dataset with samples belonging to the same groups but obtained in slightly different conditions, for example in another mass spectrometer, to test the potential of building more general classifying models.

## 3.2 Mass-Difference Sample Networks as a Data Pre-Treatment

### 3.2.1 The Rationale of Using Mass-Difference Networks as a Data Pre-Treatment

The core idea behind this pre-treatment is related to BinSim. It relies on the concept that the set of metabolites that are identified in high-resolution methods is characteristic of the studied biological system and can be used for the discrimination of samples in a dataset, implicitly discarding highly variable signal intensity data. The information used from the original dataset is rather the set of features identified in each sample. Mass-Difference Networks focus on the possible chemical transformations between the identified metabolites based on their masses instead of just focusing on their presence or absence in the different samples. A cell's metabolome is very dynamic, with metabolites consistently changing and transforming into each other [6]. Usually, metabolism is represented by metabolic networks that trace every identified enzymatic chemical reaction of a certain biological system. MDiNs (Mass-Difference Networks) aim to construct a network representation of the metabolome using solely the information from metabolomics datasets [95] and not requiring unambiguous structural or elemental identifications to build the network. These networks are built in an *ab initio* fashion and aim to describe the chemical diversity of a system. To this end, they use the list of masses of identified metabolites as nodes in the network. Edges are established between masses with a difference close to one of specific mass differences (MDB – "Mass-Difference-based Building block"), [94,95]. Each mass difference chosen is the mass corresponding to a specific change in the elemental formula of a metabolite after a chemical reaction that is common in a biologic context; for example, a methylation corresponds to the incorporation of a $–CH_3$ methyl group by substitution with a –H hydrogen atom, leading to an overall change of $CH_2$, that is, a change in mass of 14.01565 Da. Therefore, the set of mass differences chosen is of critical importance in this process.

Mass-Difference Networks consider all possible reaction interactions between the list of masses (based on the set of MDBs used) and can take into account both enzymatic and non-enzymatic chemical reactions in the representation of the metabolome and its interactions. Apart from accepting non-enzymatic transformations, it is not restricted by the knowledge of the different metabolic pathways that make up the metabolic networks that may still be very incomplete in less studied biological systems [96]. Furthermore, it does not require the extensive metabolite identification that is required in the mapping of a dataset to traditional metabolic networks. In fact, since MDiNs are based on the differences between masses, it does not technically require any formula assignment to have been made to spectral features. Nonetheless, it is worth noting that there are some drawbacks to building the network without any prior metabolite formula identification such as precluding network construction from considering elemental ratio constraints and increasing the number of spurious connections that will be discussed further later on. Most of the times, only a small fraction of metabolites is confidently identified, which are, then, used as a benchmark to assign many more formulas using the MDiNs built by propagation of the chemical transformations that link the metabolites in a network component.

Thus, an MDiN built on the list of masses of a sample should be a representation of the chemical diversity of its metabolome akin to the metabolic networks, while being more easily applied and built, and not depending on prior knowledge of the metabolic pathways in the studied biological systems. This methodology is more complex than that of BinSim but should generate networks with plentiful information that can be analysed by network-oriented methods from analysing the local characteristics of each individual node to the global characteristics of the networks. Consequently, comparing the characteristics of these networks could be an efficient way to discriminate and classify the samples into their respective groups. Furthermore, using multiple analysis methods to compare sample MDiNs could give complemental information beyond the quality of discrimination achieved.

### 3.2.2 Mass-Difference Network Construction and Limitations

MDiNs were built for 3 datasets: the Negative and Positive Grapevine Datasets (Negative and Positive GD) and the Yeast Dataset (YD) as described in section 2.3.1 (Materials and Methods). The only information given to build the networks was the list of masses of the *m/z* peaks for each dataset after conversion to represent neutral states of the metabolites. A network for each of the 3 datasets was built with a representation of their largest components in Fig. 3.7B and Fig. 3.8 and an overall look at the full network built from the YD and a close-up view of its most populated area in Fig. 3.7A,C, respectively. Table 3.4 shows the main characteristics of each network. The Yeast Formula Dataset (formulas as features) was not used since this method uses a list of masses (from the *m/z* peaks after adjustment for neutrality). The sample networks (or sample MDiNs) for each sample in each of these datasets were constructed by inducing a subgraph with only the nodes that correspond to the masses (*m/z* peaks) present in each sample. As such, each sample MDiN represents the chemical diversity of that sample.

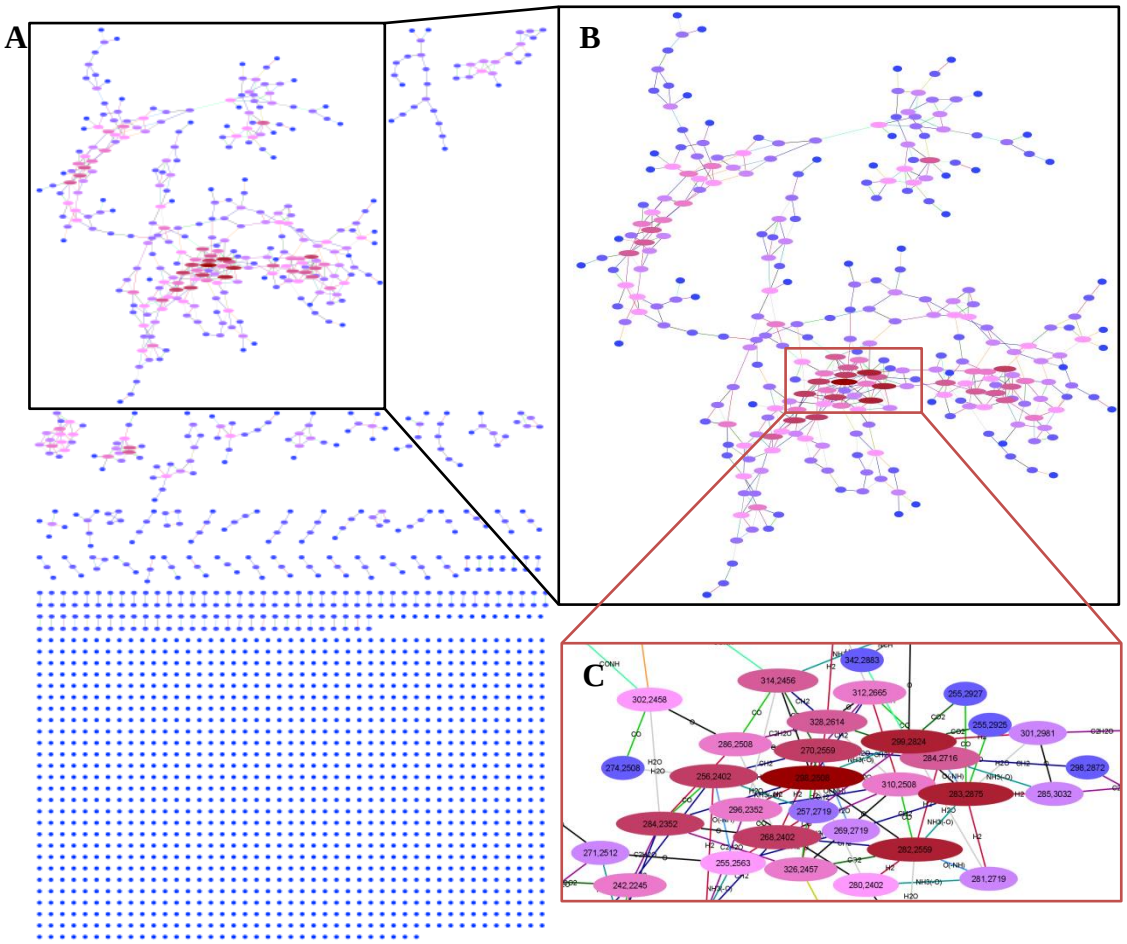



**Figure 3.7: Mass-Difference Network built from the complete Yeast Dataset.** A) Overview of the Yeast dataset network constructed. B) Close-up of the biggest Yeast dataset network component. C) Detailed view of the highest populated area in the network. Node size reflects the node degree and node colour changes from blue (●) to dark red (●) with higher degree. Edge colour represents the MDB (representing a set of chemical reactions) used to establish said edge: (—) – O(-NH), (—) – $NH_3$(-O), (—) – $H_2$, (—) – $CH_2$, (—) – O, (—) – $H_2O$, (—) – NCH, (—) – CO, (—) – CHOH, (—) – S, (—) – $C_2H_2O$, (—) – CONH, (—) – $CO_2$, (—) – $SO_3$, (—) – $PO_3H$. Network representations were made with the Cytoscape 3.8.1 [111].

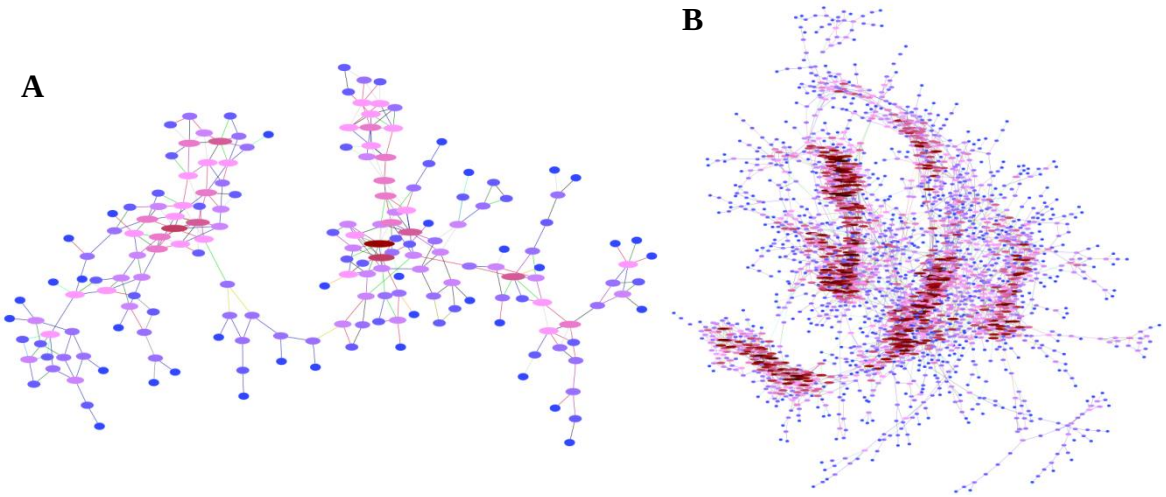



**Figure 3.8: Mass-Difference Network built from the Negative (A) and Positive (B) Grapevine Dataset.** Node size reflects the node degree and node colour changes from blue (●) to dark red (●) with higher degree. Edge colour represents the MDB (representing a set of chemical reactions) used to establish said edge: (—) – O(-NH), (—) – $NH_3$(-O), (—) – $H_2$, (—) – $CH_2$, (—) – O, (—) – $H_2O$, (—) – NCH, (—) – CO, (—) – CHOH, (—) – S, (—) – $C_2H_2O$, (—) – CONH, (—) – $CO_2$, (—) – $SO_3$, (—) – $PO_3H$. Network representations were made with the Cytoscape 3.8.1 [111].

**Table 3.4: Characteristics of the Mass-Difference Networks of the Yeast Dataset, the Negative Grapevine Dataset and the Positive Grapevine Dataset.**

| Network Characteristics | Yeast Dataset (YD) Network | Negative Grapevine Dataset (GD) Network | Positive Grapevine Dataset (GD) Network |
|---|---|---|---|
| **Number of Nodes** | 1893 | 3629 | 7026 |
| **Number of Edges** | 810 | 1005 | 6597 |
| **Biggest Component Size** | 275 | 183 | 2482 |
| **Diameter** | 31 | 27 | 49 |
| **Radius** | 16 | 14 | 25 |
| **Number of Nodes without Edges** | 1205 | 2452 | 3110 |

The network in Fig. 3.7 as well as Table 3.4 show that from approximately 1/3 to half of the nodes are connected to, at least, one other node. This means that the majority of nodes do not establish any connections and are, therefore, as uninformative to these methods as completely missing features. Despite this, the remaining nodes establish many different connections with each other, most of them in the largest components of each network. The Negative GD Network is comparatively less inter-connected than the YD Network while the Positive GD Network is a lot more inter-connected, probably due to the sheer number of nodes in a relatively small mass window, obtaining a component with almost 2500 nodes much larger than the largest components of the other networks. The 3 networks all have very high diameter and radius, with the radius being almost perfectly half of the diameter. This means that the networks are highly spread with many long stretches of node chains (Fig. 3.7 and 3.8). Near the center, there are zones of the network with greater interconnectivity between the nodes visible in the redder areas of the networks in Fig. 3.7C and Fig. 3.8, which act as the main hubs of the network – higher degree of the nodes in the area. In the Positive GD network (Fig. 3.8B) there are 3 to 4 different zones of high interconnectivity. The network topology with low number of nodes with high degree (that make up the hubs of the network) and a high number of nodes

with low degree (distribution approximates a power law) is characteristic of many different biological systems, including metabolic networks [117].

Suppl. Table 6.3 shows that the top 3 main transformations (MDBs) used to establish edges in the 3 networks were $CH_2$ (methylations), $H_2$ (Hydrogenations) and O (Oxygenations and Hydroxylations). Since they represent some of the most common reactions and differences between metabolites in biological systems, this is a signal that the difference between the masses is not random and is skewed towards differences congruent with these types of biological reactions – networks are being built as expected.

The sample networks (not shown) induced from the full networks share their main characteristics and topology on a smaller scale, because they have fewer nodes. This means that the sample networks cannot be easily discriminated by their topology without a closer and more in-depth analysis.

As mentioned before, there are some disadvantages in only using mass lists due to the lack of extra restrictions for establishing edges. False positives can happen between two masses (metabolites), generating an edge between them, whose MDB does not correspond to their actual differences in elemental compositions. This is more likely to happen with higher masses, where the combination of possible formulas within a 1 ppm error margin increases exponentially [50]. Those spurious connections can thus happen between 2 very different metabolites, for example, between X and Y, if X has a very close theoretical mass to a metabolite that could be transformed from or into Y. This could be observed in some instances in the YD network, where a conflict between formulas existed when propagating formula assignment from metabolites with more reliable formula assignments – annotated with the HMDB [49] or YMDB [107]. Moreover, elemental ratio constraints (for example, number of oxygen to number of carbons ratio) can only be applied when formula assignment is taken into account, while the network is being built from the formulas assigned to a few selected nodes. Without considering formula assignments (only using mass lists), this cannot be done. Consequently, there may be additions in succession of different groups due to chemical transformations that can lead to formulas that would theoretically occupy a chemical space not usually occupied by metabolites. In case the masses are very similar to each other, in some instances, two masses can be linked by the same chemical transformation (mass difference) in the same “direction” (both add or both subtract the same group due to a chemical transformation) to the same node. This can be seen in the upper right corner of Fig. 3.7C where nodes 255.2927 and 255.2925 (blue) are both linked to node 299.2924 (red) by a $CO_2$ edge. This would mean that those two peaks are representing metabolites with the same elemental formula, which results from a problem of either the peak selection and alignment in the pre-processing stages or with the mass error tolerance being too high in the MDiN building stage. Finally, all the connections established are only hypothetical based on the mass difference of the metabolites.

Despite these problems, the overall structure of the networks should be robust to these issues so it should not considerably affect our results. Moreover, the objective is to test this approach as a generalized treatment of metabolomics datasets for sample discrimination, many of which do not have the needed assigned formulas to mitigate these issues. So, it is in the interest of this work to test if this treatment is viable with the least possible available information (and not for the common use of MDiNs which is metabolite formula assignment).

### 3.2.3 Mass-Difference Network Analysis

Since the application of MDiNs as a metabolomics data analysis pre-treatment (for sample discrimination) is novel, several different analyses methods were applied to test if meaningful

discriminatory information can be gathered only from the structure of MDiNs. Thus, for the different sample networks obtained from the 3 main networks, their individual node centrality and global network (node-independent) characteristics (considering the number of times each MDB was used to establish edges and their network topology) were computed. As an aside, since the global network characteristics are independent of the identity of their nodes, they can be used to compare sample MDiNs whose peaks were not previously aligned and compiled in the same dataset. The analysis of the results for each sample network was grouped together, for each method, to make "secondary datasets" with features depending on the method used. Statistical methods were subsequently used to analyse if the information gathered in the different secondary datasets is discriminatory in the sense of separating groups or predicting the correct assignment of samples into those groups. The statistical methods used were the same as in sections 3.1.1 and 3.1.2: Hierarchical Clustering and K-means Clustering analysis (unsupervised), Random Forests and PLS-DA classifiers (supervised).

For individual analysis of the nodes, three centrality measures were used: degree, betweenness centrality and closeness centrality. The values of each node in the network in each measure were compiled on a secondary dataset for each of the centrality measures. That is, the features of the secondary datasets are still the mass lists of the original datasets. However, the information of each feature (mass) is the relation to other masses as described by the different centrality measures and not information contained in the feature itself. As an example, if the degree of a feature is 0, then that feature either is not present or does not establish any connections, if the degree is 1, then the feature is present and establishes exactly one connection and so forth with degree 2, 3, etc. Thus, the key information associated with a feature is its possible relation with other features.

To observe if the frequency of mass differences between the mass lists of different samples could indicate that certain sets of chemical reactions were being over or under stimulated in some biological groups and could be meaningful information for group discrimination, the percentage of edges associated with each MDB (mass difference) in sample networks was compiled to make a secondary dataset. This method will be referred to as MDB influence hereafter since it represents the impact of each MDB in building the sample networks. The rationale is that if, for example, oxidizing compounds or enzymes are more present or expressed in a biological system in relation to another, it is expected the presence of more metabolites whose difference corresponds to an oxidation reaction (O or $H_2$) and, therefore, more metabolites/masses would have a difference corresponding to those chemical transformations. In this case, the number of features is condensed to the number of groups used – 15 (where each feature is an MDB) – from originally thousands of features. The important features to build the classifier models can then indicate which chemical transformations have different prominences in the biological systems studied, which is a concept that is further explored in section 3.2.6.

To analyse the general network topology of each sample network, a method called GCD-11 was applied. The GCD-11 method was chosen to analyse the topology of the network since prior studies such as Tantardini et al. [115] and Yaveroğlu et al. [113] show it performs very well in the comparison and classification of networks when compared to many other methods. GCD-11 is the Graphlet Correlation Distance, using 11 non-redundant orbits for up to 4-node graphlets [113]. Graphlets are small and non-isomorphic subgraphs of a network and each graphlet can have multiple automorphic orbits if the nodes in the graphlet are not in the same relative position [114]. As explained in greater detail in the Materials and Methods (section 2.3.2), the features of this method are the 60 orbit *n* – orbit *m* (*n*, *m* are 2 of the 11 different orbits) Spearman correlations of the counts of orbits *n* and *m* of all nodes in a network. These 60 features represent the network topology. In this case, the features do

not have a clear and interpretable biological significance, which prevents an in-depth analysis of the features considered important in classifiers built from these secondary datasets.

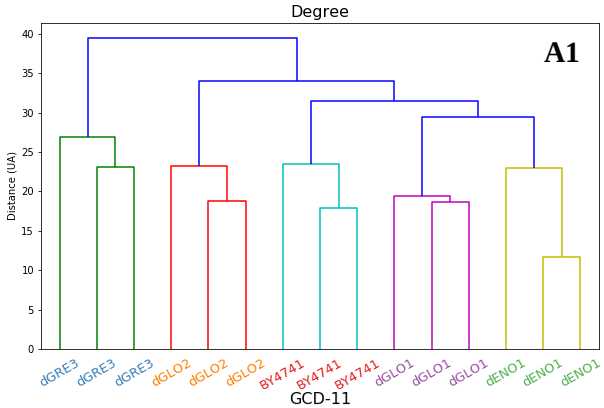
Each secondary dataset is identified by the following general notation: Main Network – Analysis method. For example, the yeast dataset sample networks analysed by degree are identified as: **YD – Degree**. The other 4 analysis methods are referred to as: **YD – Betweenness**, **YD – Closeness**, **YD – MDB Influence** and **YD – GCD-11**. When addressing all the three secondary datasets built from a certain network analysis method, they are referred to as: analysis method secondary datasets; for example, degree secondary datasets.

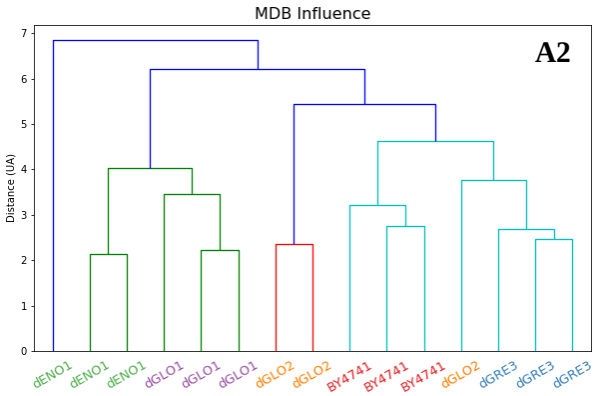
### 3.2.4 Unsupervised Statistical Analysis – Hierarchical and K-means Clustering

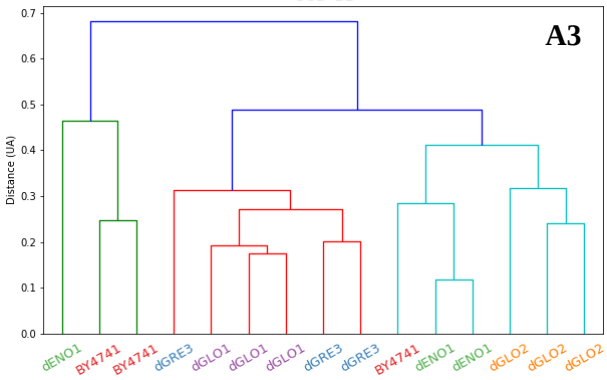
Hierarchical Clustering Analysis (HCA) and K-means Clustering analysis were performed to observe if the samples of the same biological groups in the different secondary datasets clustered preferentially with each other rather than with samples of other groups due to an intrinsic pattern in the data.

HCA was performed with the UPGMA linkage method and the Euclidian distance metric. The resulting dendrograms corresponding to degree, MDB influence and GCD-11 sample network analysis are presented in Fig. 3.9, while the remaining dendrograms (after betweenness and closeness centrality analysis) are presented in Suppl. Fig. 6.9. Furthermore, an evaluation summary using and Discrimination Distance metrics, the correct clustering and first correct cluster percentages of the HCA performed on the secondary datasets obtained is also presented in Fig. 3.9A4, B4, C4.

## Yeast Dataset



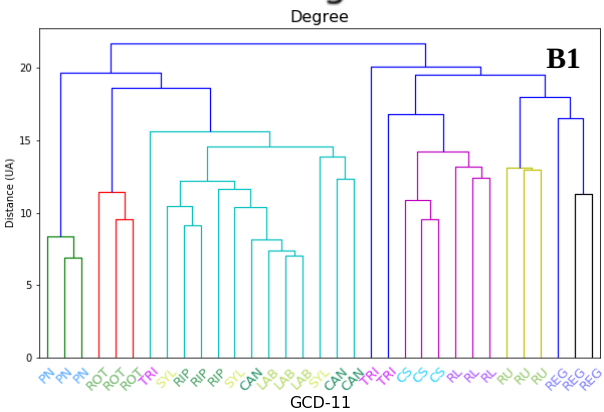



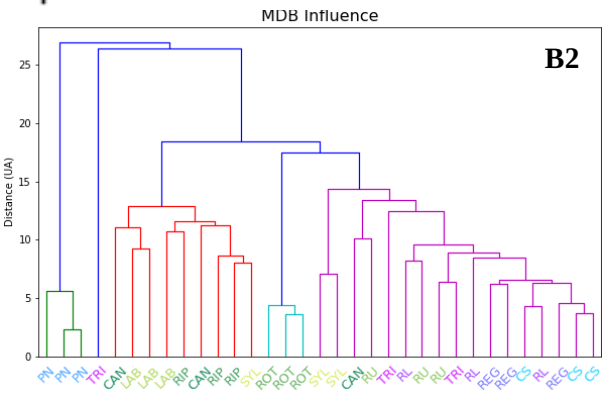



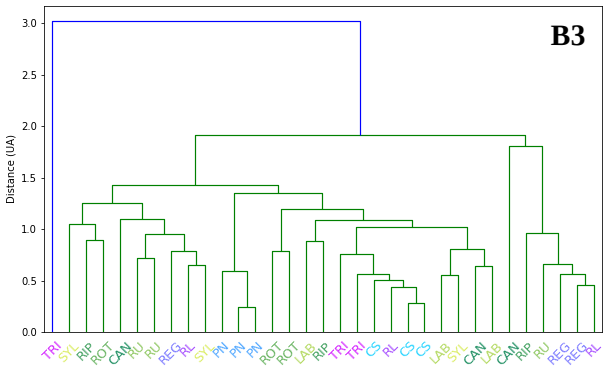

**A4**

| Yeast Dataset Sample Networks | Degree | Betweenness Centrality | Closeness Centrality | MDB Influence | GCD-11 |
|---|---|---|---|---|---|
| Discrimination Distance | 0.24 | 0.00 | 0.19 | 0.09 | 0.05 |
| Correct Clustering (%) | 100 | 0 | 80 | 60 | 40 |
| Correct First Cluster (%) | 100 | 53 | 93 | 87 | 80 |

## Negative Mode Grapevine Dataset



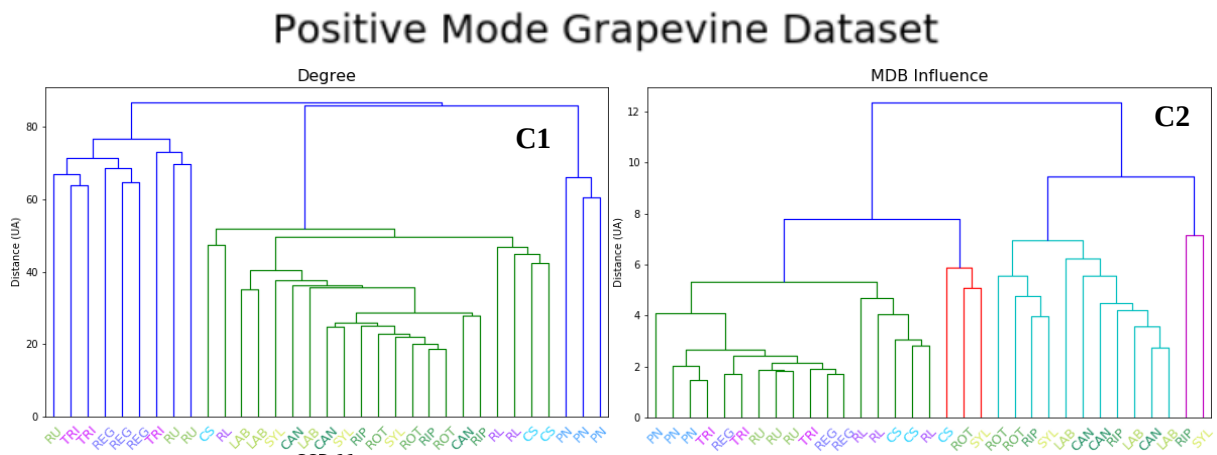




**B4**

| Negative Grapevine Dataset Networks | Degree | Betweenness Centrality | Closeness Centrality | MDB Influence | GCD-11 |
|---|---|---|---|---|---|
| Discrimination Distance | 0.13 | 0.02 | 0.01 | 0.12 | 0.02 |
| Correct Group Clustering (%) | 64 | 9.1 | 55 | 18 | 9.1 |
| Correct First Cluster (%) | 76 | 33 | 55 | 42 | 27 |

Figure continues in the next page →




C4

| Positive Grapevine Dataset Networks | Degree | Betweenness Centrality | Closeness Centrality | MDB Influence | GCD-11 |
|---|---|---|---|---|---|
| Discrimination Distance | 0.02 | 0.00 | 0.01 | 0.002 | 0.00 |
| Correct Group Clustering (%) | 18 | 0 | 9.1 | 9.1 | 0 |
| Correct First Cluster (%) | 42 | 18 | 33 | 15 | 30 |

**Figure 3.9: Hierarchical Clustering Analysis (HCA) of the different secondary datasets obtained from sample MDiNs.** Dendrograms of the HCA of the Yeast (**A**), Negative Grapevine (**B**) and Positive Grapevine (**C**) sample networks after degree (**1**), MDB influence (**2**) or GCD-11 (**3**) network analysis of each one. HCA was performed with UPGMA linkage method and Euclidian distance metric. (**4**) Summary of the discrimination observed (based on 3 different metrics) after HCA of the datasets obtained after the different network analysis methods used on the sample networks. *Vitis* genotypes abbreviations are indicated in Table 2.1.

K-means Clustering analysis was performed on the 5 secondary datasets built from sample networks for each of the 3 main datasets studied using the Euclidian distance metric with the scikit-learn Python module [101]. The number of clusters chosen for each dataset was equal to the number of groups: YD – 5 groups, Negative and Positive GD – 11 groups. The results of an analysis of how well the samples were discriminated by K-means clustering analysis using the discrimination distance, correct clustering percentage and the adjusted Rand Index metrics are presented in Table 3.5.

**Table 3.5: Discrimination Distance, correct clustering percentages and adjusted Rand Index of the K-means Clustering analysis performed on the secondary datasets obtained from network analysis of each sample network for the Yeast, Negative and Positive Grapevine Datasets.**

| | Yeast Dataset Network | | | | | Negative Grapevine Dataset ($ESI^-$) Network | | | | | Positive Grapevine Dataset ($ESI^+$) Network | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Metrics | Degree | Betweenness Centrality | Closeness Centrality | MDB Influence | GCD-11 | Degree | Betweenness Centrality | Closeness Centrality | MDB Influence | GCD-11 | Degree | Betweenness Centrality | Closeness Centrality | MDB Influence | GCD-11 |
| Discrimination Distance | 0.71 | 0.00 | 0.41 | 0.10 | 0.08 | 0.21 | 0.00 | 0.11 | 0.08 | 0.02 | 0.00 | 0.01 | 0.00 | 0.00 | 0.00 |
| Correct Clustering (%) | 100 | 0 | 60 | 20 | 20 | 45 | 0 | 27 | 18 | 9.1 | 0 | 9.1 | 0 | 0 | 0 |
| Adjusted Rand Index | 1 | 0.31 | 0.81 | 0.56 | 0.51 | 0.48 | 0.23 | 0.43 | 0.25 | 0.24 | 0.23 | 0.14 | 0.23 | 0.26 | 0.21 |

Analysing both Fig. 3.9 and Table 3.5, it can be seen that, in all three main datasets, using the degree centrality measure to characterise each sample network leads to the most correct discrimination of samples in both clustering techniques applied. Observing the YD results, YD – Degree was the only secondary dataset that allowed a perfect discrimination of the samples in both methods (Fig. 3.9A1 and Table 3.5). YD – Closeness (another centrality measure) allows the 2$^{nd}$ best discrimination, while YD – Betweenness (the last centrality measure) leads to the worst discrimination with 0% of groups correctly clustering in both methods, only 53% of samples with correct first clusters (HCA) and an adjusted Rand Index (K-means clustering) of 0.31 (much lower than those of other analysis methods – Table 3.5). YD – MDB Influence results are below those of YD – Closeness with 60% correct group clustering in HCA (87% of samples with correct first clusters) and 20% in K-means clustering. YD – GCD-11 results (topology of the sample networks) are slightly worse than YD – MDB Influence with 40% correct group clustering and 80% of samples with correct first clusters in HCA and 20% correct clustering in K-means clustering (same as YD – MDB Influence) with a lower Discrimination Distance ($0.08 < 0.10$) and adjusted Rand Index ($0.51 < 0.56$) that show that the well discriminated group was more separated from other samples (Discrimination Distance) and that the samples of groups not completely well clustered were more partially well clustered in the YD – MDB Influence case (Rand Index).

This trend of the quality of the discrimination of samples achieved from the secondary datasets was also observed for both clustering metrics in the Negative GD results, despite the overall quality being lower due to the groups being less distinct in this dataset. Despite this, HCA on the GD – Degree led to a very good result: 7 of the 11 groups (64%) were correctly clustered/discriminated with 76 % of samples (25 out of 33) with correct first clusters (Fig. 3.9B1) and K-means clustering led to 5 groups being correctly discriminated (45%) with a Rand Index of 0.48.

The performance of the discrimination of the samples of the Positive GD was poor with both clustering methods like it was with the intensity-based and BinSim pre-treatments (section 3.1.1). In the best discrimination, obtained with HCA of GD – Degree, only 2 groups were correctly clustered and 42% of samples had correct first clusters – albeit a very low quality of separation is apparent in the dendrogram in Fig. 3.9C1. K-means Clustering analysis was not able to correctly cluster a single group in 4 of the 5 secondary positive grapevine datasets and they all had a low Rand Index (between 0.21 and 0.26). GD – Betweenness was the exception, with one group being correctly clustered. Nonetheless, the low Rand Index when compared to the others (0.14) indicates that this was probably happenstance rather than better discrimination due to information from the GD – Betweenness.

Based on these results, from the different sample MDiN characteristics, the analysis and comparison of the networks based on the centrality of their nodes, especially their degree (which keeps a high number of features), was the most successful approach, far better than focusing on the global characteristics of the network that greatly reduced the number of total features (topology and impact of each MDB in establishing the MDiNs – GCD-11 and MDB influence, respectively). This indicates that there is a significant information content in the network that is not being translated to the few features of those methods. From these latter two, the MDB influence analysis allows the clustering methods to discriminate the different samples slightly better than the GCD-11 topology analysis, even though the information is concentrated in only 15 features. A more in-depth discussion is presented later when combined with the results from supervised analysis.

### 3.2.5 Supervised Statistical Analysis – Random Forests and PLS-DA

As performed in section 3.1, after clustering analysis, the discrimination of the different groups in the data was evaluated by using supervised statistical methods that build classifiers whose purpose is to classify and discriminate samples. The classifiers chosen were Random Forests and PLS-DA. Random Forests and PLS-DA models were built from each of the secondary datasets for the 3 studied datasets using the scikit-learn Python module [101]. After tuning, Random Forest models were built with 200 trees (Suppl. Fig. 6.10). The number of components of PLS-DA models was chosen to minimize the predictive residual sum of squares estimated with stratified 3-fold cross-validation. For the secondary datasets based on the Yeast sample networks, models were built with 5 components (Suppl. Fig. 6.11A). For the secondary datasets obtained from network analysis of the degree or closeness centrality of each node (high feature number) on the Negative or the Positive GD sample networks, PLS-DA models were built with 11 components (Suppl. Fig. 6.11B). For the secondary datasets obtained from the MDB influence or GCD-11 network analysis (low feature number) on the Negative or Positive GD sample networks, PLS-DA models were built with 5 components since $Q^2$ started to drop with higher component numbers (Suppl. Fig. 6.11B). Models with 5 components were also made after betweenness centrality analysis of the Negative and Positive GD sample networks, despite the high number of features of its secondary datasets (Suppl. Fig. 6.11B). The quality of the discrimination was evaluated according to the distribution of the predictive accuracy of 200 iterations of the classifiers, estimated through randomly sampled stratified 3-fold cross-validation (Fig. 3.10). Permutation tests in Suppl. Fig. 6.12 all show that the predictive accuracy of the distribution of permuted labels Random Forest and PLS-DA models was considerably below that of the reference non-permuted models built from the secondary datasets ($p$-value < 0.02), which means that the classifiers' accuracy resulted from significant information present in the data and not from random noise.

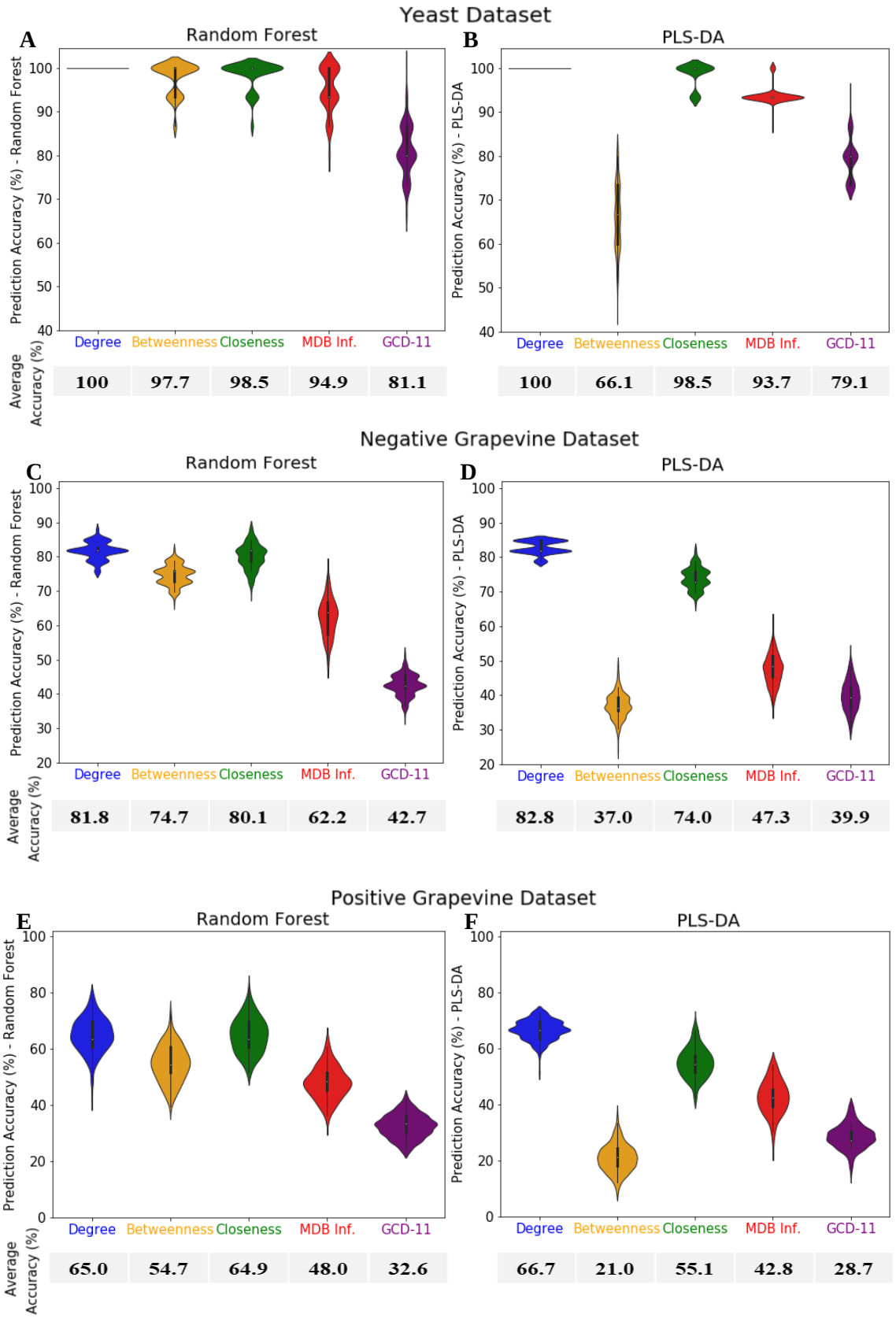



**Figure 3.10: Distribution of the prediction accuracy of Random Forest and PLS-DA models built from the different secondary datasets.** Violin plots of the distribution of the prediction accuracy of 200 iterations of Random Forests and PLS-DA models built from the secondary datasets obtained from the sample networks of the Yeast Dataset (**A** and **B**, respectively), Negative Grapevine Dataset (**C** and **D**, respectively) and Positive Grapevine Dataset (**E** and **F**, respectively). Each iteration's predictive accuracy is estimated by randomly sampled stratified 3-fold cross-validation. Below the plots, the average prediction accuracy is presented. MDB Inf. – MDB Influence.

Once again, models built on secondary datasets obtained from the degree analysis of the sample networks outperform all others in the 3 main datasets studied, closely followed by closeness. As for the general performance on the 3 main datasets, the classification analysis of the different secondary datasets from the YD has a greater predictive accuracy, with a perfect discrimination of the sample being reached in some cases (Fig 3.10A,B). Maintaining the same trend from the clustering analysis, analysis of the secondary dataset from the Negative GD yields better results than the poor performance of the classifiers built from all secondary datasets of the Positive GD, with the highest average predictive accuracy reached being a low 66.7%, around 16% lower than the maximum achieved in the Negative GD – 82.8% (Fig 3.10C-F). Despite these differences in absolute accuracies, the trends observed between the five network analyses methods used was identical for the datasets.

Regarding the three different centrality measures used to analyse the networks, the degree secondary datasets outperformed the others in all statistical methods used. As previously seen, this was already the trend observed in the clustering methods. With these classifiers, Random Forest and PLS-DA classifiers have a perfect discrimination with the YD – Degree, 81.8 and 82.8% average accuracy, respectively, with the Negative GD – Degree and 65.0 and 66.7% average accuracy, respectively, with the Positive GD – Degree (Fig 3.10). Both classifier methods achieve similar accuracies based on these datasets. Classifiers of the closeness centrality secondary datasets always have a similar but slightly inferior performance to the degree secondary datasets. Regarding the betweenness centrality secondary datasets, Random Forest classifiers had an average predictive accuracy of 30 to 40% higher than PLS-DA models (97.7 to 66.1% for the YD, 74.7 to 37% for the Negative GD and 54.7% to a very low 21.0% for the Positive GD) – Fig 3.10. This difference meant that PLS-DA classifiers built from the betweenness centrality secondary datasets had the worst performance of the 5 metrics following the results from clustering analysis while Random Forest classifiers had the 3$^{rd}$ best performance (below degree and closeness secondary datasets). However, the conclusion to be extracted is that analysis based on the degree of each node is the most suitable one to be performed on the sample networks among the centrality measures based on node characteristics which create secondary datasets with an equal number of features to that of masses.

Regarding the methods that analyse the networks as a whole, classifiers built from the MDB influence secondary datasets always outperform, in terms of predictive accuracy, those from GCD-11 secondary datasets, despite both methods performing poorer than those built from the degree secondary datasets (Fig 3.10). These results are in line with the results of the clustering techniques. Moreover, the performance of Random Forest classifiers of these datasets slightly outperforms that of PLS-DA classifiers. This might be due to the low feature number in MDB influence (15) and GCD-11 (60) secondary datasets. As a dimension reduction algorithm, one of the main appeals of PLS-DA comes from its proficiency in reducing greatly the number of features of metabolomics datasets to just a few components. Since MDB influence and GCD-11 secondary datasets already have a low number of features, this appeal of dimension reduction algorithms is reduced.

The low performance of the different statistical methods in analysing GCD-11 secondary datasets was partially expected. Despite the proven suitability of this method to classify different networks according to their topology [113,115], the topology of the different sample networks was very similar and, in turn, similar to their mother networks shown in Fig 3.7 and 3.8. Therefore, the difference in topology would have to be identified in the finer structure of these networks. The low predictive accuracy of classifiers of YD – GCD-11 (around 80%), Negative GD – GCD-11 (around 40%) and

Positive GD – GCD-11 (around 30%) shows that GCD-11 was not sensitive enough to allow a quality discrimination based on the finer structure of the topology of the network (Fig 3.10).

The MDB Influence analysis (specific to MDiNs) stands, then, as the 2nd most promising method from the 5 studied. Despite the worse performance in comparison with the closeness secondary datasets, since closeness centrality focuses on the same aspect of the network as the degree analysis and underperforms comparatively, it is less useful than the MDB influence analysis. As an aside, although Random Forest models built from Negative GD – MDB Influence show a 62.2% (Fig. 3.10C) average prediction accuracy, this is probably an underestimation (variance due to the random 3-fold split of the dataset) if we consider Suppl. Fig. 6.10B (number of trees tuning), where the prediction accuracy seems closer to 70%. This performance could be important due to the significance of the features of the MDB influence secondary datasets – MDBs chosen to build the MDiNs. Each feature represents the percentage of times the respective MDB was used to construct an edge in each sample, that is, it should be representative of the number of different compounds that are being created/destroyed by the set of chemical reactions each MDB represents. In theory, this can inform if a certain biological system has, for example, the phosphorylation or the oxidation of compounds over or under stimulated in comparison to another system. This discrimination between different groups becomes harder with the presence of more groups due to the limited number of features. Although the 94/95% predictive accuracy of the classifiers for YD – MDB Influence (Fig. 3.10A,B) and the 60 to 70% predictive accuracy in discriminating between 33 samples of 11 different groups of the Negative GD with only 15 features (Fig. 3.10C) is impressive and clearly shows that there is meaningful information grasped in this way, it is undeniable that it underperforms in relation to the degree secondary datasets and the analysis of datasets treated in the ways discussed in section 3.1. However, this information can still give us a perspective on some metabolic differences between the biological groups as exemplified in the next section (3.2.6).

### 3.2.6 Potential of MDB Influence Secondary Dataset Features

From the different network analysis methods employed, MDiN analysis based on the degree of the nodes allowed for the best discrimination results. Nonetheless, as discussed, the information associated with their features is the possible relations (by biochemical transformations) with other features represented by the number of connections each feature (node) has in the MDiN. Consequently, classifier models would give more importance to features whose pattern of possible chemical transformations is different from biological group to group. Since it is the changes on the presence of the neighbouring nodes and the pattern of edges around it in the MDiN that determines the importance of the features (nodes) to build the different classifier models from the degree secondary datasets, these important features can't be conclusively identified as key metabolites in the discrimination of the groups. On the other hand, despite the worse discrimination achieved in classifier models built from the MDB influence dataset, the most important features to build these models can give an indication of the chemical transformations that have different prominence in the biological systems, which can be a new way of comparing the samples [118]. In turn, these can indicate metabolic differences in those reaction types in the biological systems under study. This is a promising use of the analysis based on MDB influence and can help characterize differences in the metabolomes or orient future research paths. The MDB influence differences are clearer when only comparing 2 different groups but can also be seen between multiple groups as shown in the example below.

As an example, consider the Random Forest models built from the MDB influence secondary datasets obtained from the sample networks. Random Forest models were chosen since they discriminate the

biological systems of each dataset better than the PLS-DA models built from the same secondary datasets (higher average predictive accuracies). Table 3.6 shows the importance of each of the 15 MDBs (used to build the sample MDiNs) in building the Random Forest classifiers, estimated by the Gini Importance [87]. The results show that the "$PO_3H$" MDB that represents mostly the phosphorylation of metabolites is the most important MDB to separate the groups of YD – MDB Influence, while oxygenations (oxidations) and hydroxylations represented by the "O" MDB are the most important for both the Negative and Positive GD – MDB Influence (and the "$PO_3H$" is closer to the least important feature – 13$^{th}$ and 15$^{th}$ place, respectively). Looking at the YD results, since it is known that the groups are distinct between each other to the point that many different treatments and statistical methods discriminated them perfectly, it can be seen in Table 3.7 that there is, in fact, a distinction of the "$PO_3H$" edges established in each group with a high number of edges (phosphorylations) in the strains BY4741 and ΔGRE3, an average amount in ΔGLO2 and a very low amount in the strains ΔENO1 and ΔGLO1. So, for a study with the goal of characterizing differences in the metabolism of these strains, although other treatments can show a perfect discrimination of the groups and indicate certain key metabolites for this discrimination, this could point to this global metabolic change on metabolite phosphorylation or steer a study on the possible causes that lead to different phosphorylation of metabolites (for example, what caused or can justify the lack of metabolite phosphorylation in the ΔENO1 and ΔGLO1 strains) as a more global look at the metabolome and its differences.

**Table 3.6: Gini Importance of the features from the MDB influence secondary datasets obtained from the sample networks to build the respective Random Forest models**. Gini Importance is calculated by the scikit-learn Python module also used to build the Random Forest models [101]. The Gini Importance of all features adds up to 1. MDB – Mass-Difference-based Building block.

| Yeast Dataset | | | Negative Grapevine Dataset | | | Positive Grapevine Dataset | | |
|---|---|---|---|---|---|---|---|---|
| Place | MDB | Gini Importance | Place | MDB | Gini Importance | Place | MDB | Gini Importance |
| 1 | PO3H | 0.086223 | 1 | O | 0.087312 | 1 | O | 0.086307 |
| 2 | CO | 0.084046 | 2 | H2 | 0.086574 | 2 | CONH | 0.081555 |
| 3 | CH2 | 0.081853 | 3 | NH3(-O) | 0.084466 | 3 | NH3(-O) | 0.076146 |
| 4 | S | 0.080721 | 4 | CO2 | 0.082026 | 4 | CH2 | 0.074113 |
| 5 | O | 0.078283 | 5 | SO3 | 0.078643 | 5 | H2O | 0.071581 |
| 6 | CO2 | 0.076783 | 6 | CH2 | 0.078133 | 6 | CO | 0.069252 |
| 7 | H2 | 0.073881 | 7 | CO | 0.075951 | 7 | NCH | 0.068762 |
| 8 | NH3(-O) | 0.073731 | 8 | C2H2O | 0.072885 | 8 | CO2 | 0.067989 |
| 9 | CONH | 0.071418 | 9 | CONH | 0.068147 | 9 | O(-NH) | 0.067651 |
| 10 | NCH | 0.067919 | 10 | NCH | 0.064084 | 10 | H2 | 0.065988 |
| 11 | C2H2O | 0.057347 | 11 | H2O | 0.059136 | 11 | S | 0.063798 |
| 12 | SO3 | 0.055605 | 12 | O(-NH) | 0.056158 | 12 | SO3 | 0.054317 |
| 13 | H2O | 0.051184 | 13 | PO3H | 0.044968 | 13 | CHOH | 0.053717 |
| 14 | O(-NH) | 0.041219 | 14 | CHOH | 0.040145 | 14 | C2H2O | 0.051256 |
| 15 | CHOH | 0.019786 | 15 | S | 0.021372 | 15 | PO3H | 0.047567 |

**Table 3.7: $PO_3H$ feature of the MDB influence secondary dataset built from the Yeast sample networks before and after normalization.** Normalized values represent the percentage of edges established in the sample network due to the respective MDB. MDB – Mass-Difference-based Building block.

| MDB | BY4741 | | | dGRE3 | | | dENO1 | | | dGLO1 | | | dGLO2 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $PO_3H$ (not normalized) | 19 | 20 | 23 | 17 | 17 | 16 | 5 | 7 | 6 | 8 | 10 | 10 | 12 | 13 | 11 |
| $PO_3H$ (normalized) | 10.5 | 7.9 | 11.3 | 9.2 | 8.0 | 6.8 | 4.9 | 4.8 | 3.8 | 3.1 | 4.8 | 5.3 | 5.9 | 7.0 | 5.8 |

### 3.2.7 Comparison of Sample MDiNs to Other Pre-Treatments

A summary of the results presented throughout this dissertation about the performance of different statistical methods after pre-treatments based on signal intensities and the new pre-treatments proposed in this work is presented in Table 3.8. The results from the analysis on the YD show that degree analysis after the construction of sample MDiNs allowed a perfect assignment of the 15 samples into their respective groups with high Discrimination Distances on clustering analysis. This good performance is au par with the intensity-based pre-treatments and the BinSim pre-treatment. The results from the analysis on the Negative GD show that degree analysis of the sample MDiNs also led to very good discrimination results even when compared to intensity-based pre-treatments or BinSim, exhibiting higher correct clustering fractions on both HCA (7 of the 11 groups were well clustered) and K-means clustering (5 of the 11 groups). This trend was also maintained in the Random Forest and PLS-DA models with more than 80% predictive accuracy in both, once again higher than the intensity-based treatments in most cases. This similar performance to the intensity-based pre-treatments, although with worse absolute values, was also mostly observed in the results from the discrimination analysis on the Positive GD with similar results obtained with the clustering techniques. Both Random Forest and PLS-DA models built from the Positive GD – Degree had a sub-par performance with a maximum of only around 66% predictive accuracy. For the Random Forest models, this wasn't very different to what was observed with intensity-based and BinSim pre-treatments (5% below the two best pre-treatments and 15% higher than the others). However, with the PLS-DA models, this was far below (around 15%) the performance on the Positive Grapevine Dataset treated with the pre-treatments presented in section 3.1 of this work (around 80% predictive accuracy). From the different metrics used to evaluate sample discrimination in the 4 statistical methods presented and on the 3 benchmark datasets, this, then, stands as the exception where the discrimination obtained with the sample MDiNs (using the degree analysis) was not better or similar to intensity-based treatments. As for the analysis on the datasets generated by the MDB influence analysis of the sample networks, the discrimination achieved by the different statistical methods was poorer than the discrimination achieved in datasets treated in other ways. Thus, despite the potential highly informative and biological significance of the features of the secondary dataset built by this type of analysis, when the objective is just the discrimination of the different samples, this method should only be picked when other alternatives mentioned in this work can't be applied.

Like the BinSim pre-treatment, the analysis and discrimination of metabolomics dataset by building sample MDiNs appears viable compared to the more established intensity-based pre-treatments, leading to a good discrimination of samples. However, it should be pointed out that, from a computational point of view, this is a more complex pre-treatment than BinSim, whose simplicity was one of its main advantages. The complexity of MDiN-based methods stems from the need to build

secondary datasets. For these, it is necessary to choose a set of MDBs that represent the set of chemical reactions relevant to the analysed system, then build the different MDiNs and finally use a network analysis method (after the assessment in this work, it is recommended to use the node degree analysis to achieve the better discrimination). The poor results obtained from the Positive GD PLS-DA models may indicate that the sample MDiNs are a less robust pre-treatment (in comparison to BinSim) and, therefore, requires a more careful consideration of when to apply it. However, these problems are contraposed by the great versatility that the sample network analysis provides. While the BinSim pre-treatment is tailored to the discrimination of different groups while highlighting very specific key metabolites, the sample MDiNs can also achieve a very good discrimination of the samples while retaining other information that can be extracted from these networks for further analyses, granting a great versatility to this method. For example, MDB influence analysis is still possible, giving further insights, as discussed in the previous section; analysis on the main hubs of the partial metabolic networks built can be used to 1) characterize the sample and biological group and 2) further compare with the other biological groups. Therefore, it allows a great amount of other biologically relevant information to be extracted from this fresh perspective that MDiNs grant. Moreover, MDiN network analysis is still viable on sample MDiNs that come from different datasets whose peaks were not previously aligned and do not have formulas assigned. Analysing datasets with both drawbacks is usually impossible by other methods or individual node network analysis methodologies, but can still be done by falling back on comparing the global characteristics of the sample networks such as the MDB influence analysis that, although less efficient, may still provide meaningful insight and discrimination of the samples. These examples show the great benefit of the versatility of sample MDiN based analysis.

Finally, since this methodology, like the BinSim pre-treatment, forgoes the highly variable intensity data for the less variable encoding based on the presence/absence of features, its potential in analysis across different datasets analysis is similar to the great potential of BinSim with all the points discussed in section 3.1.4 also applying here. Moreover, using formulas assigned to compare different unaligned datasets as suggested in section 3.1.4, would allow MDiN construction with these formulas in mind, diminishing the number of spurious connections between masses and establishing maximum ratios between the different chemical elements which would further improve the quality of the MDiNs. Furthermore, since the information in MDiNs comes from the relations between features, these might be more robust to missed identifications of some metabolites, alleviating the impact of data variability on the analysis between different metabolomics datasets, potentiating even more the advantages of these types of analysis. Finally, as a last resort, if alignment of the different datasets is not possible, MDiNs still tolerates a comparison based on the global network characteristics, specifically, MDB influence analysis that may also give meaningful insights as discussed above. Therefore, as suggested for the BinSim pre-treatment, future studies could test the plausibility of classification of samples across different datasets based on sample MDiNs since, in theory, sample MDiNs could also outperform the currently available methodologies for analysis across different datasets.

**Table 3.8: Summary of the results of the performance of the different statistical methods in discriminating samples into their respective group.** Results summary of the Hierarchical clustering analysis (HCA), K-means clustering analysis, Random Forest and PLS-DA classifiers of the Yeast Dataset, Negative and Positive Grapevine Datasets after being treated in one of the following ways: one of three combinations of intensity-based pre-treatments, Binary Similarity pre-treatment, degree or MDB influence analysis of sample MDiNs. BinSim – Binary Similarity, P – Pareto Scaling, N – Normalization by a reference feature, G – Generalized Logarithmic Transformation, MDiN – Mass-Difference Network.

| Statistical Analysis Results | Intensity-Based Pre-Treatments | | | BinSim | MDiN Degree | MDiN MDB Influence |
|---|---|---|---|---|---|---|
| | P | NP | NGP | | | |
| **Yeast Dataset** | | | | | | |
| **HCA** | | | | | | |
| Discrimination Distance | 0.31 | 0.22 | 0.22 | 0.14 | 0.24 | 0.09 |
| Correct Clustering (%) | 100 | 100 | 100 | 100 | 100 | 60 |
| Correct First Cluster (%) | 100 | 100 | 100 | 100 | 100 | 87 |
| **K-means Clustering** | | | | | | |
| Discrimination Distance | 0.73 | 0.39 | 0.37 | 0.86 | 0.71 | 0.10 |
| Correct Clustering (%) | 100 | 100 | 100 | 100 | 100 | 20 |
| Adjusted Rand Index | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 0.56 |
| **Random Forest** | | | | | | |
| Prediction Accuracy (%) | 100 | 100 | 100 | 100 | 100 | 94.9 |
| **PLS-DA** | | | | | | |
| Prediction Accuracy (%) | 100 | 100 | 100 | 100 | 100 | 93.7 |
| **Negative Grapevine Dataset** | | | | | | |
| **HCA** | | | | | | |
| Discrimination Distance | 0.10 | 0.12 | 0.14 | 0.12 | 0.13 | 0.12 |
| Correct Clustering (%) | 45 | 45 | 54 | 54 | 64 | 18 |
| Correct First Cluster (%) | 64 | 64 | 79 | 67 | 76 | 42 |
| **K-means Clustering** | | | | | | |
| Discrimination Distance | 0.09 | 0.13 | 0.17 | 0.16 | 0.21 | 0.08 |
| Correct Clustering (%) | 18 | 27 | 36 | 27 | 45 | 18 |
| Adjusted Rand Index | 0.52 | 0.48 | 0.59 | 0.53 | 0.48 | 0.25 |
| **Random Forest** | | | | | | |
| Prediction Accuracy (%) | 82.2 | 76.4 | 76.7 | 81.2 | 81.8 | 62.2 |
| **PLS-DA** | | | | | | |
| Prediction Accuracy (%) | 74.8 | 73.6 | 79.5 | 83.5 | 82.8 | 47.3 |
| **Positive Grapevine Dataset** | | | | | | |
| **HCA** | | | | | | |
| Discrimination Distance | 0.03 | 0.02 | 0.03 | 0.04 | 0.02 | 0.002 |
| Correct Clustering (%) | 27 | 27 | 18 | 45 | 18 | 9.1 |
| Correct First Cluster (%) | 33 | 27 | 48 | 52 | 42 | 15 |
| **K-means Clustering** | | | | | | |
| Discrimination Distance | 0.00 | 0.04 | 0.00 | 0.06 | 0.00 | 0.00 |
| Correct Clustering (%) | 0 | 23 | 0 | 9.1 | 0 | 0 |
| Adjusted Rand Index | 0.22 | 0.49 | 0.31 | 0.23 | 0.23 | 0.26 |
| **Random Forest** | | | | | | |
| Prediction Accuracy (%) | 70.2 | 45.8 | 47.0 | 71.1 | 65.0 | 48.0 |
| **PLS-DA** | | | | | | |
| Prediction Accuracy (%) | 76.2 | 79.2 | 84.6 | 80.6 | 66.7 | 42.8 |

# 4. Conclusion

The main aim of this study was to develop and test the viability of two different data pre-treatments, which were called Binary Similarity (BinSim) and sample MDiNs, specifically tailored to the characteristics of high-resolution and high-accuracy metabolomics data. These new methods highlight relevant but usually overshadowed information to give a new perspective on the data complementing what is usually obtained from the standard metabolomics pre-treatments and workflows, which are signal intensity driven. Their viability was assessed by comparing the performance of different supervised and unsupervised statistical analysis in discriminating groups defined in FT-ICR-MS datasets after treatment with one of the methods proposed in this dissertation or with some of the most established and common pre-treatments used in the metabolomics workflow.

Both the BinSim and the sample MDiNs treatments forgo the intensity data from the Mass Spectrometry datasets for the information of which feature is present or absent from which sample. The idea was to augment the consistency of the results by discarding the highly variable intensity data for the less variable identification of metabolites.

The Binary Similarity pre-treatment consisted of changing all intensity values to 1 if features are present in a sample and all missing values to 0. This greatly simplified the metabolomics workflow steps before the statistical analysis since it skips the choice of different missing value imputation methods and combinations and parametrization of other steps, requiring a lower amount of peak filtering. The four datasets treated by BinSim allowed all the different statistical methods employed, both supervised (Hierarchical Clustering and K-means Clustering analysis) and unsupervised analysis (Random Forest and PLS-DA classifier models), to discriminate the different groups within the datasets with an accuracy as high as that achieved using the intensity-based pre-treatments. Moreover, results were more consistent with BinSim than with any of the other 3 treatments used as a reference, allowing often the best or second-best discrimination by the different statistical methods. Finally, the features estimated as more important to build the different Random Forest and PLS-DA classifier models for the BinSim were very different and unique when compared to the other treatments and had very different characteristics, being mainly features that appeared in only one (biomarker-like) or only a few of the biological groups studied in the dataset, that is, features that appear in a low number of samples.

Sample Mass-Difference Networks (MDiNs) consists of translating the list of masses (*m/z* peaks in MS datasets) into characteristic networks that represent the chemical diversity of each sample's metabolome by linking masses with differences that indicate they may be transformed into each other by a simple biochemical reaction (MDB), originating a metabolic-like network. Thus, the information on display is the possible transformations (through chemical reactions) of the different metabolites identified with a set of MDBs used to build the networks. The MDiNs associated with each sample allow a versatility in the way the data can be manipulated, when compared to other traditional treatments or BinSim. For the discrimination of samples, an individual node-centric network analysis, specifically, node degree network analysis allowed the different statistical methods to have a performance of group discrimination and prediction on par or slightly better than the original datasets treated with intensity-based pre-treatments on 2 out of 3 benchmark datasets. With the last benchmark dataset, the sample discrimination using sample MDiNs was also on par in almost all statistical methods used, except for the poorer performance of PLS-DA models. This may indicate that this treatment is less robust than BinSim, for example. This point and the complexity of building and analysing the sample MDiNs stood as the main disadvantages of this approach. They are contraposed

by the versatility of this type of analysis. For example, it was shown how MDB influence analysis of the MDiNs may have the capacity to identify metabolic differences by evaluating the prominence of MDBs (chemical reactions) between biological groups.

The promising results obtained for the two pre-treatments proposed in this dissertation associated with their theoretical mitigation of the low reproducibility of metabolomics data, by discarding intensity, show a great potential for the use of both methods in different contexts in the future. Thus, future studies could use these results to go further beyond and test the viability of sample discrimination in cross metabolomics dataset (same biological groups but obtained either in a different instrument, different protocol, different condition, etc.) analysis using these pre-treatments proposed. Finally, there are also possible avenues to further explore the versatility and the potential of the sample MDiNs in metabolomics data analysis, for example, going more in-depth on the possible biological significance of the MDB influence analysis.

# 5. References


1. Fiehn O. Metabolomics – the link between genotypes and phenotypes. *Plant Mol Biol*. 2002;48(1):155-171. doi:10.1023/A:1013713905833

2. Bartel J, Krumsiek J, Theis FJ. Statistical methods for the analysis of high-throughput metabolomics data. *Comput Struct Biotechnol J*. 2013;4:e201301009-e201301009. doi:10.5936/csbj.201301009

3. Nicholson JK, Holmes E, Lindon JC. Chapter 1 - Metabonomics and Metabolomics Techniques and Their Applications in Mammalian Systems. In: Lindon JC, Nicholson JK, Holmes EBT-TH of M and M, eds. Elsevier Science B.V.; 2007:1-33. doi:10.1016/B978-044452841-4/50002-3

4. Alonso A, Marsal S, Julià A. Analytical methods in untargeted metabolomics: State of the art in 2015. *Front Bioeng Biotechnol*. 2015;3(MAR):1-20. doi:10.3389/fbioe.2015.00023

5. Zhang A, Sun H, Wang P, Han Y, Wang X. Modern analytical techniques in metabolomics analysis. *Analyst*. 2012;137(2):293-300. doi:10.1039/c1an15605e

6. Roessner U, Hansen MAE. The Chemical Challenge of the Metabolome. *Metabolome Anal*. Published online February 2, 2007:15-38. doi:10.1002/9780470105511.ch2

7. Worley B, Powers R. Multivariate Analysis in Metabolomics. *Curr Metabolomics*. 2013;1(1):92-107. doi:10.2174/2213235X11301010092

8. Han J, Danell RM, Patel JR, et al. Towards high-throughput metabolomics using ultrahigh-field Fourier transform ion cyclotron resonance mass spectrometry. *Metabolomics*. 2008;4(2):128-140. doi:10.1007/s11306-008-0104-8

9. Ramautar R, Berger R, van der Greef J, Hankemeier T. Human metabolomics: strategies to understand biology. *Curr Opin Chem Biol*. 2013;17(5):841-846. doi:10.1016/j.cbpa.2013.06.015

10. Grissa D, Pétéra M, Brandolini M, Napoli A, Comte B, Pujos-Guillot E. Feature Selection Methods for Early Predictive Biomarker Discovery Using Untargeted Metabolomic Data. *Front Mol Biosci*. 2016;3:30. doi:10.3389/fmolb.2016.00030

11. Mamas M, Dunn W, Neyses L, Goodacre R. The role of metabolites and metabolomics in clinically applicable biomarkers of disease. *Arch Toxicol*. 2011;85:5-17. doi:10.1007/s00204-010-0609-6

12. Wolfender J-L, Rudaz S, Choi YH, Kim HK. Plant metabolomics: from holistic data to relevant biomarkers. *Curr Med Chem*. 2013;20(8):1056-1090. doi:10.2174/0929867311320080009

13. Tang J. Microbial metabolomics. *Curr Genomics*. 2011;12(6):391-403. doi:10.2174/138920211797248619

14. Cuperlovic-Culf M, Culf AS. Applied metabolomics in drug discovery. *Expert Opin Drug Discov*. 2016;11(8):759-770. doi:10.1080/17460441.2016.1195365

15. Putri SP, Nakayama Y, Matsuda F, et al. Current metabolomics: Practical applications. *J Biosci Bioeng*. 2013;115(6):579-589. doi:10.1016/j.jbiosc.2012.12.007

16. Roberts LD, Souza AL, Gerszten RE, Clish CB. Targeted metabolomics. *Curr Protoc Mol Biol*. 2012;Chapter 30:Unit30.2-30.2.24. doi:10.1002/0471142727.mb3002s98

17. Ferreira AEN, Sousa Silva M, Cordeiro C. Metabolic Network Inference from Time Series. In: Wolkenhauer O, ed. *Systems Medicine: Integrative, Qualitative and Computational Approaches*. vol. 3. Oxford: Elsevier; 2021:127–133. doi:10.1016/B978-0-12-801238-3.11347-9

18. Dunn WB, Ellis DI. Metabolomics: Current analytical platforms and methodologies. *TrAC Trends Anal Chem*. 2005;24(4):285-294. doi:10.1016/j.trac.2004.11.021

19. Theodoridis GA, Gika HG, Want EJ, Wilson ID. Liquid chromatography–mass spectrometry based global metabolite profiling: A review. *Anal Chim Acta*. 2012;711:7-16. doi:10.1016/j.aca.2011.09.042

20. Brown M, Dunn WB, Ellis DI, et al. A metabolome pipeline: from concept to data to knowledge. *Metabolomics*. 2005;1(1):39-51. doi:10.1007/s11306-005-1106-4

21. Kuehnbaum NL, Britz-McKibbin P. New Advances in Separation Science for Metabolomics: Resolving Chemical Diversity in a Post-Genomic Era. *Chem Rev*. 2013;113(4):2437-2468. doi:10.1021/cr300484s

22. Brown SC, Kruppa G, Dasseux J-L. Metabolomics applications of FT-ICR mass spectrometry. *Mass Spectrom Rev*. 2005;24(2):223-231. doi:10.1002/mas.20011

23. Scigelova M, Hornshaw M, Giannakopulos A, Makarov A. Fourier transform mass spectrometry. *Mol Cell Proteomics*. 2011;10(7):M111.009431-1-M111.009431-19. doi:10.1074/mcp.M111.009431

24. Ghaste M, Mistrik R, Shulaev V. Applications of fourier transform ion cyclotron resonance (FT-ICR) and orbitrap based high resolution mass spectrometry in metabolomics and lipidomics. *Int J Mol Sci*. 2016;17(6). doi:10.3390/ijms17060816

25. van den Berg RA, Hoefsloot HCJ, Westerhuis JA, Smilde AK, van der Werf MJ. Centering, scaling, and transformations: improving the biological information content of metabolomics data. *BMC Genomics*. 2006;7:142. doi:10.1186/1471-2164-7-142

26. Verpoorte R. Secondary Metabolism BT - Metabolic Engineering of Plant Secondary Metabolism. In: Verpoorte R, Alfermann AW, eds. Springer Netherlands; 2000:1-29. doi:10.1007/978-94-015-9423-3_1

27. Johnson CH, Gonzalez FJ. Challenges and opportunities of metabolomics. *J Cell Physiol*. 2012;227(8):2975-2981. doi:10.1002/jcp.24002

28. Matsuda F. Technical Challenges in Mass Spectrometry-Based Metabolomics. *Mass Spectrom (Tokyo, Japan)*. 2016;5(2):S0052. doi:10.5702/massspectrometry.S0052

29. Roullier-Gall C, Witting M, Gougeon RD, Schmitt-Kopplin P. High precision mass measurements for wine metabolomics. *Front Chem*. 2014;2(NOV):1-9. doi:10.3389/fchem.2014.00102

30. Schrimpe-Rutledge AC, Codreanu SG, Sherrod SD, McLean JA. Untargeted Metabolomics Strategies-Challenges and Emerging Directions. *J Am Soc Mass Spectrom*. 2016;27(12):1897-1905. doi:10.1007/s13361-016-1469-y

31. Gromski PS, Muhamadali H, Ellis DI, et al. A tutorial review: Metabolomics and partial least squares-discriminant analysis – a marriage of convenience or a shotgun wedding. *Anal Chim Acta*. 2015;879:10-23. doi:10.1016/j.aca.2015.02.012

32. Vinaixa M, Samino S, Saez I, Duran J, Guinovart JJ, Yanes O. A guideline to univariate statistical analysis for LC/MS-based untargeted metabolomics-derived data. *Metabolites*.

2012;2(4):775-795. doi:10.3390/metabo2040775

33. Liland K. Multivariate methods in metabolomics – from pre-processing to dimension reduction and statistical analysis. *TRAC-TRENDS Anal Chem*. 2011;30:827-841. doi:10.1016/j.trac.2011.02.007

34. Riekeberg E, Powers R. New frontiers in metabolomics: from measurement to insight. *F1000Research*. 2017;6:1148. doi:10.12688/f1000research.11495.1

35. Karaman I. Preprocessing and Pretreatment of Metabolomics Data for Statistical Analysis. *Adv Exp Med Biol*. 2017;965:145-161. doi:10.1007/978-3-319-47656-8_6

36. Katajamaa M, Oresic M. Data processing for mass spectrometry-based metabolomics. *J Chromatogr A*. 2007;1158:318-328. doi:10.1016/j.chroma.2007.04.021

37. Cambiaghi A, Ferrario M, Masseroli M. Analysis of metabolomic data: Tools, current strategies and future challenges for omics data integration. *Brief Bioinform*. 2017;18(3):498-510. doi:10.1093/bib/bbw031

38. Hansen MAE. Data Analysis. *Metabolome Anal*. Published online February 2, 2007:146-187. doi:10.1002/9780470105511.ch5

39. Sugimoto M, Kawakami M, Robert M, Soga T, Tomita M. Bioinformatics Tools for Mass Spectroscopy-Based Metabolomic Data Processing and Analysis. *Curr Bioinform*. 2012;7(1):96-108. doi:10.2174/157489312799304431

40. Tautenhahn R, Patti GJ, Rinehart D, Siuzdak G. XCMS Online: a web-based platform to process untargeted metabolomic data. *Anal Chem*. 2012;84(11):5035-5039. doi:10.1021/ac300698c

41. Pluskal T, Castillo S, Villar-Briones A, Oresic M. MZmine 2: modular framework for processing, visualizing, and analyzing mass spectrometry-based molecular profile data. *BMC Bioinformatics*. 2010;11:395. doi:10.1186/1471-2105-11-395

42. Emwas AH, Saccenti E, Gao X, et al. Recommended strategies for spectral processing and post-processing of 1D 1 H-NMR data of biofluids with a particular focus on urine. *Metabolomics*. 2018;14(3):1-23. doi:10.1007/s11306-018-1321-4

43. Wei R, Wang J, Su M, et al. Missing Value Imputation Approach for Mass Spectrometry-based Metabolomics Data. *Sci Rep*. 2018;8(1):1-10. doi:10.1038/s41598-017-19120-0

44. Di Guida R, Engel J, Allwood JW, et al. Non-targeted UHPLC-MS metabolomic data processing methods: a comparative investigation of normalisation, missing value imputation, transformation and scaling. *Metabolomics*. 2016;12(5):93. doi:10.1007/s11306-016-1030-9

45. Lazar C. imputeLCMD: A collection of methods for left-censored missing data imputation Version 2.0. R package. Published 2015. https://rdrr.io/cran/imputeLCMD/

46. Troyanskaya O, Cantor M, Sherlock G, et al. Missing value estimation methods for DNA microarrays. *Bioinformatics*. 2001;17(6):520-525. doi:10.1093/bioinformatics/17.6.520

47. Stekhoven DJ, Bühlmann P. Missforest-Non-parametric missing value imputation for mixed-type data. *Bioinformatics*. 2012;28(1):112-118. doi:10.1093/bioinformatics/btr597

48. Pence HE, Williams A. Chemspider: An online chemical information resource. *J Chem Educ*. 2010;87(11):1123-1124. doi:10.1021/ed100697w

49. Wishart DS, Feunang YD, Marcu A, et al. HMDB 4.0: The human metabolome database for

2018. *Nucleic Acids Res*. 2018;46(D1):D608-D617. doi:10.1093/nar/gkx1089

50. Kind T, Fiehn O. Seven Golden Rules for heuristic filtering of molecular formulas obtained by accurate mass spectrometry. *BMC Bioinformatics*. 2007;8:1-20. doi:10.1186/1471-2105-8-105

51. Schiffman C, Petrick L, Perttula K, et al. Filtering procedures for untargeted LC-MS metabolomics data. *BMC Bioinformatics*. 2019;20(1):334. doi:10.1186/s12859-019-2871-9

52. Xi B, Gu H, Baniasadi H, Raftery D. Statistical analysis and modeling of mass spectrometry-based metabolomics data. *Methods Mol Biol*. 2014;1198:333-353. doi:10.1007/978-1-4939-1258-2_22

53. Bolstad BM, Irizarry RA, Åstrand M, Speed TP. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*. 2003;19(2):185-193. doi:10.1093/bioinformatics/19.2.185

54. Dieterle F, Ross A, Schlotterbeck G, Senn H. Probabilistic Quotient Normalization as Robust Method to Account for Dilution of Complex Biological Mixtures. Application in 1H NMR Metabonomics. *Anal Chem*. 2006;78(13):4281-4290. doi:10.1021/ac051632c

55. Rocke D, Durbin-Johnson B. Approximate variance-stabilizing transformations for gene-expression microarray data. Bioinformatics 19, 966-972. *Bioinformatics*. 2003;19:966-972. doi:10.1093/bioinformatics/btg107

56. Durbin BP, Hardin JS, Hawkins DM, Rocke DM. A variance-stabilizing transformation for gene-expression microarray data. *Bioinformatics*. 2002;18(SUPPL. 1):105-110. doi:10.1093/bioinformatics/18.suppl_1.S105

57. Gromski PS, Xu Y, Hollywood KA, Turner ML, Goodacre R. The influence of scaling metabolomics data on model classification accuracy. *Metabolomics*. 2015;11(3):684-695. doi:10.1007/s11306-014-0738-7

58. Smilde AK, Van Der Werf MJ, Bijlsma S, Van Der Werff-Van Der Vat BJC, Jellema RH. Fusion of mass spectrometry-based metabolomics data. *Anal Chem*. 2005;77(20):6729-6736. doi:10.1021/ac051080y

59. Keun H, Ebbels T, Antti H, et al. Improved Analysis of Multivariate Data by Variable Stability Scaling: Application to NMR-Based Metabolic Profiling. *Anal Chim Acta*. 2003;490:265-276. doi:10.1016/S0003-2670(03)00094-1

60. Palermo G, Piraino P, Zucht H-D. Performance of PLS regression coefficients in selecting variables for each response of a multivariate PLS for omics-type data. *Adv Appl Bioinform Chem*. 2009;2:57-70. doi:10.2147/aabc.s3619

61. Antonelli J, Claggett BL, Henglin M, et al. Statistical workflow for feature selection in human metabolomics data. *Metabolites*. 2019;9(7):1-15. doi:10.3390/metabo9070143

62. Saccenti E, Hoefsloot H, Smilde A, Westerhuis J, Hendriks M. Reflections on univariate and multivariate analysis of metabolomics data. *Metabolomics*. 2013;10. doi:10.1007/s11306-013-0598-6

63. Benjamini Y, Hochberg Y. Controlling The False Discovery Rate - A Practical And Powerful Approach To Multiple Testing. *J R Stat Soc, Ser B*. 1995;57:289-300. doi:10.2307/2346101

64. Smolinska A, Hauschild A-C, Fijten R, Dallinga J, Baumbach J, Van Schooten F. Current breathomics - A review on data pre-processing techniques and machine learning in metabolomics breath analysis. *J Breath Res*. 2014;8:27105. doi:10.1088/1752-7155/8/2/027105

65. Hendriks MMWB, Eeuwijk FA va., Jellema RH, et al. Data-processing strategies for metabolomics studies. *TrAC - Trends Anal Chem*. 2011;30(10):1685-1698. doi:10.1016/j.trac.2011.04.019

66. Cuperlovic-Culf M. Machine Learning Methods for Analysis of Metabolic Data and Metabolic Pathway Modeling. *Metabolites*. 2018;8(1). doi:10.3390/metabo8010004

67. Sathya R, Abraham A. Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification. *Int J Adv Res Artif Intell*. 2013;2. doi:10.14569/IJARAI.2013.020206

68. Hotelling H. Analysis of a complex of statistical variables into principal components. *J Educ Psychol*. 1933;24(6):417-441. doi:10.1037/h0071325

69. Jolliffe IT. Principal Component Analysis. BT - International Encyclopedia of Statistical Science. Published online 2011:1094-1096. doi:10.1007/978-3-642-04898-2_455

70. Jolliffe IT, Cadima J. Principal component analysis: a review and recent developments. *Philos Trans A Math Phys Eng Sci*. 2016;374(2065):20150202. doi:10.1098/rsta.2015.0202

71. Jolliffe IT. *Principal Component Analysis*. 2nd Ed. Springer, New York, NY; 2002. doi:https://doi.org/10.1007/b98835

72. Andreopoulos B, An A, Wang X, Schroeder M. A roadmap of clustering algorithms: finding a match for a biomedical application. *Brief Bioinform*. 2009;10(3):297-314. doi:10.1093/bib/bbn058

73. Ward JH. Hierarchical Grouping to Optimize an Objective Function. *J Am Stat Assoc*. 1963;58(301):236-244. doi:10.1080/01621459.1963.10500845

74. Sokal R, Rohlf F. The comparison of dendrograms by objective methods. Taxon 11: 33-40. *Taxon*. 1962;11:33-40. doi:10.2307/1217208

75. Baker FB. Stability of Two Hierarchical Grouping Techniques Case 1: Sensitivity to Data Errors. *J Am Stat Assoc*. 1974;69(346):440-445. doi:10.2307/2285675

76. Hubert L, Arabie P. Comparing partitions. *J Classif*. 1985;2(1):193-218. doi:10.1007/BF01908075

77. Luz J. Metabolomic effects of single gene deletions in Saccharomyces cerevisiae. Master Thesis in Biochemistry. Published online 2020.

78. Lee LC, Liong C-Y, Jemain AA. Partial least squares-discriminant analysis (PLS-DA) for classification of high-dimensional (HD) data: A review of contemporary practice strategies and knowledge gaps. *Analyst*. 2018;143(15):3526-3539. doi:10.1039/c8an00599k

79. Westerhuis JA, Hoefsloot HCJ, Smit S, et al. Assessment of PLSDA cross validation. *Metabolomics*. 2008;4(1):81-89. doi:10.1007/s11306-007-0099-6

80. Lee LC, Liong C-Y, Jemain AA. Statistical comparison of decision rules in PLS2-DA prediction model for classification of blue gel pen inks according to pen brand and pen model. *Chemom Intell Lab Syst*. 2019;184(November 2018):94-101. doi:10.1016/j.chemolab.2018.11.014

81. Wold S, Trygg J. The PLS method -- partial least squares projections to latent structures -- and its applications in industrial RDP ( research , development , and production ). *PLS Ind RPD Prague*. 2004;1(June):1-44. doi:10.1109/JMEMS.2011.2159097

82. Mehmood T, Liland KH, Snipen L, Sæbø S. A review of variable selection methods in Partial

Least Squares Regression. *Chemom Intell Lab Syst*. 2012;118:62-69. doi:10.1016/j.chemolab.2012.07.010

83. Farrés M, Platikanov S, Tsakovski S, Tauler R. Comparison of the variable importance in projection (VIP) and of the selectivity ratio (SR) methods for variable selection and interpretation. *J Chemom*. 2015;29(10):528-536. doi:10.1002/cem.2736

84. Galindo-Prieto B, Eriksson L, Trygg J. Variable influence on projection (VIP) for orthogonal projections to latent structures (OPLS). *J Chemom*. 2014;28(8):623-632. doi:10.1002/cem.2627

85. Breiman L. Random Forests. *Mach Learn*. 2001;45(1):5-32. doi:10.1023/A:1010933404324

86. Probst P, Boulesteix A-L. To Tune or Not to Tune the Number of Trees in Random Forest. *J Mach Learn Res*. 2017;18(1):6673–6690.

87. Louppe G, Wehenkel L, Sutera A, Geurts P. *Understanding Variable Importances in Forests of Randomized Trees*. Vol 26.; 2013.

88. Schmitt-Kopplin P, Hemmler D, Moritz F, et al. Systems chemical analytics: introduction to the challenges of chemical complexity analysis. *Faraday Discuss*. Published online 2019. doi:10.1039/c9fd00078j

89. Hertkorn N, Ruecker C, Meringer M, et al. High-precision frequency measurements: indispensable tools at the core of the molecular-level analysis of complex systems. *Anal Bioanal Chem*. 2007;389(5):1311-1327. doi:10.1007/s00216-007-1577-4

90. Rivas-Ubach A, Liu Y, Bianchi TS, Tolić N, Jansson C, Paša-Tolić L. Moving beyond the van Krevelen Diagram: A New Stoichiometric Approach for Compound Classification in Organisms. *Anal Chem*. 2018;90(10):6152-6160. doi:10.1021/acs.analchem.8b00529

91. Gougeon RD, Lucio M, Frommberger M, et al. The chemodiversity of wines can reveal a metabologeography expression of cooperage oak wood. *Proc Natl Acad Sci U S A*. 2009;106(23):9174-9179. doi:10.1073/pnas.0901100106

92. Hughey CA, Hendrickson CL, Rodgers RP, Marshall AG, Qian K. Kendrick Mass Defect Spectrum: A Compact Visual Analysis for Ultrahigh-Resolution Broadband Mass Spectra. *Anal Chem*. 2001;73(19):4676-4681. doi:10.1021/ac010560w

93. Adrian M, Lucio M, Roullier-Gall C, et al. Metabolic Fingerprint of PS3-Induced Resistance of Grapevine Leaves against Plasmopara Viticola Revealed Differences in Elicitor-Triggered Defenses. *Front Plant Sci*. 2017;8(February):1-14. doi:10.3389/fpls.2017.00101

94. Tziotis D, Hertkorn N, Schmitt-Kopplin P. Kendrick-analogous network visualisation of ion cyclotron resonance Fourier transform mass spectra: improved options for the assignment of elemental compositions and the classification of organic molecular complexity. *Eur J Mass Spectrom (Chichester, Eng)*. 2011;17(4):415-421. doi:10.1255/ejms.1135

95. Breitling R, Ritchie S, Goodenowe D, Stewart ML, Barrett MP. Ab initio prediction of metabolic networks using Fourier transform mass spectrometry data. *Metabolomics*. 2006;2(3):155-164. doi:10.1007/s11306-006-0029-z

96. Moritz F, Kaling M, Schnitzler JP, Schmitt-Kopplin P. Characterization of poplar metabotypes via mass difference enrichment analysis. *Plant Cell Environ*. 2017;40(7):1057-1073. doi:10.1111/pce.12878

97. Kunenkov E V, Kononikhin AS, Perminova I V, et al. Total Mass Difference Statistics Algorithm: A New Approach to Identification of High-Mass Building Blocks in Electrospray Ionization Fourier Transform Ion Cyclotron Mass Spectrometry Data of Natural Organic

Matter. *Anal Chem*. 2009;81(24):10106-10115. doi:10.1021/ac901476u

98. Harris CR, Millman KJ, van der Walt SJ, et al. Array programming with NumPy. *Nature*. 2020;585(7825):357-362. doi:10.1038/s41586-020-2649-2

99. McKinney W. Data Structures for Statistical Computing in Python. In: van der Walt S, Millman J, eds. *Proceedings of the 9th Python in Science Conference*. ; 2010:56-61. doi:10.25080/Majora-92bf1922-00a

100. Virtanen P, Gommers R, Oliphant TE, et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat Methods*. 2020;17(3):261-272. doi:10.1038/s41592-019-0686-2

101. Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: Machine Learning in Python. *J Mach Learn Res*. 2011;12(85):2825-2830. http://jmlr.org/papers/v12/pedregosa11a.html

102. Hunter JD. Matplotlib: A 2D Graphics Environment. *Comput Sci Eng*. 2007;9(3):90-95. doi:10.1109/MCSE.2007.55

103. Waskom M, Botvinnik O, Gelbart M, et al. mwaskom/seaborn: v0.11.0 (Sepetmber 2020). Published online 2020. doi:10.5281/zenodo.4019146

104. Hagberg AA, Schult DA, Swart PJ. Exploring Network Structure, Dynamics, and Function using NetworkX. In: Varoquaux G, Vaught T, Millman J, eds. *Proceedings of the 7th Python in Science Conference*. ; 2008:11-15.

105. Maia M, Ferreira AEN, Nascimento R, et al. Integrating metabolomics and targeted gene expression to uncover potential biomarkers of fungal / oomycetes - associated disease susceptibility in grapevine. *Sci Rep*. Published online 2020:1-15. doi:10.1038/s41598-020-72781-2

106. Maia M, Figueiredo A, Silva MS, Ferreira A. Grapevine untargeted metabolomics to uncover potential biomarkers of fungal/oomycetes-associated diseases. Published online 2020. doi:10.6084/m9.figshare.12357314.v1

107. Ramirez-Gaona M, Marcu A, Pon A, et al. YMDB 2.0: a significantly expanded version of the yeast metabolome database. *Nucleic Acids Res*. 2017;45(D1):D440-D445. doi:10.1093/nar/gkw1058

108. Choi S-S, Cha S-H, Tappert C. A Survey of Binary Similarity and Distance Measures. *J Syst Cybern Inf*. 2010;8:43-48.

109. Galili T. dendextend: an R package for visualizing, adjusting and comparing trees of hierarchical clustering. *Bioinformatics*. 2015;31(22):3718-3720. doi:10.1093/bioinformatics/btv428

110. Burgess KE V, Borutzki Y, Rankin N, Daly R, Jourdan F. MetaNetter 2: A Cytoscape plugin for ab initio network analysis and metabolite feature classification. *J Chromatogr B Analyt Technol Biomed Life Sci*. 2017;1071:68-74. doi:10.1016/j.jchromb.2017.08.015

111. Shannon P, Markiel A, Ozier O, et al. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res*. 2003;13(11):2498-2504. doi:10.1101/gr.1239303

112. Jeske L, Placzek S, Schomburg I, Chang A, Schomburg D. BRENDA in 2019: a European ELIXIR core data resource. *Nucleic Acids Res*. 2019;47(D1):D542-D549. doi:10.1093/nar/gky1048

113. Yaveroğlu ÖN, Malod-Dognin N, Davis D, et al. Revealing the Hidden Language of Complex

Networks. *Sci Rep*. 2014;4(1):4547. doi:10.1038/srep04547

114. Milenković T, Pržulj N. Uncovering biological network function via graphlet degree signatures. *Cancer Inform*. 2008;6:257-273. doi:10.4137/cin.s680

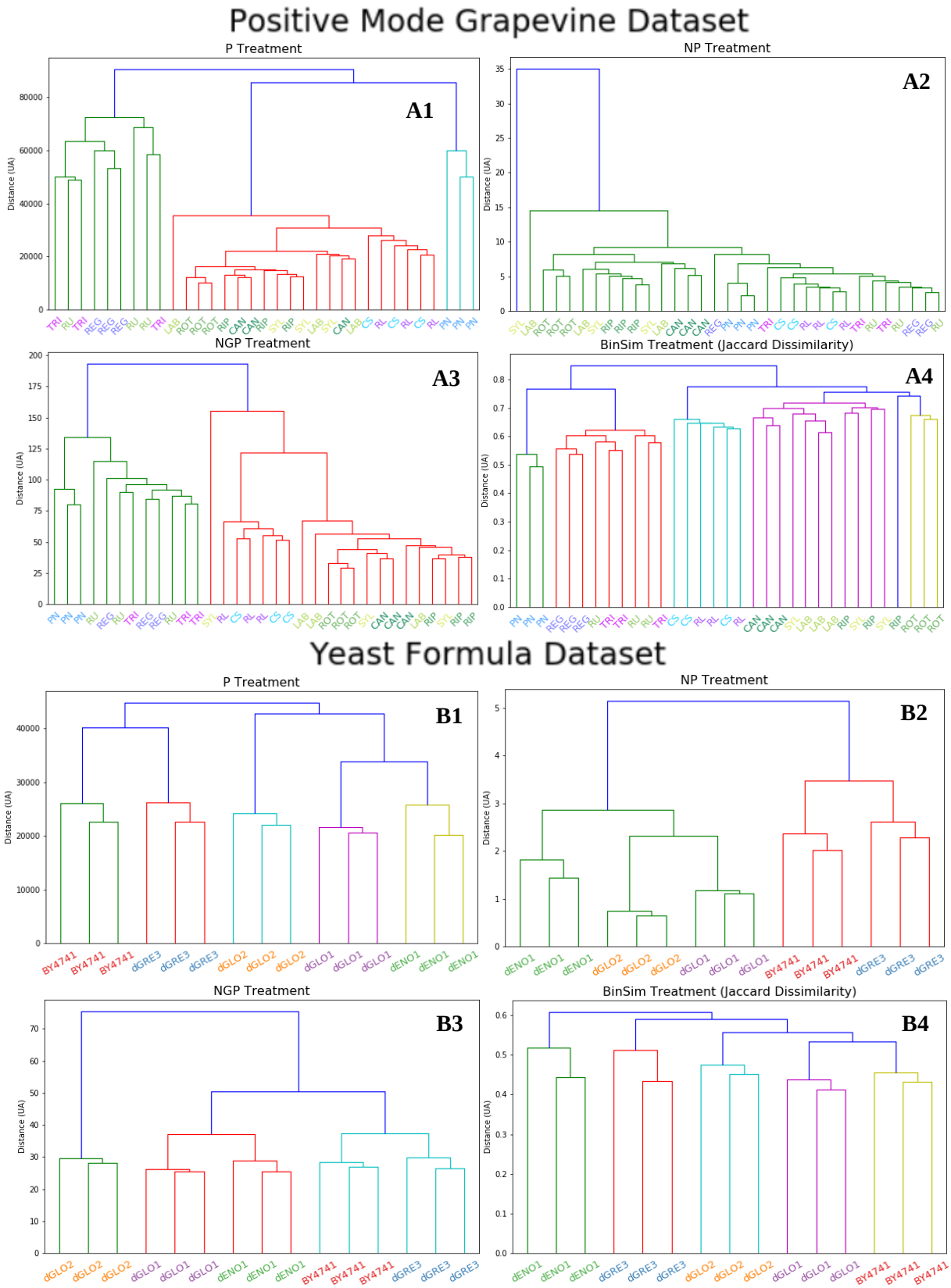
115. Tantardini M, Ieva F, Tajoli L, Piccardi C. Comparing methods for comparing networks. *Sci Rep*. 2019;9(1):1-19. doi:10.1038/s41598-019-53708-y

116. Lin Y, Caldwell GW, Li Y, Lang W, Masucci J. Inter-laboratory reproducibility of an untargeted metabolomics GC–MS assay for analysis of human plasma. *Sci Rep*. 2020;10(1):1-11. doi:10.1038/s41598-020-67939-x

117. Barabási AL, Oltvai ZN. Network biology: Understanding the cell's functional organization. *Nat Rev Genet*. 2004;5(2):101-113. doi:10.1038/nrg1272

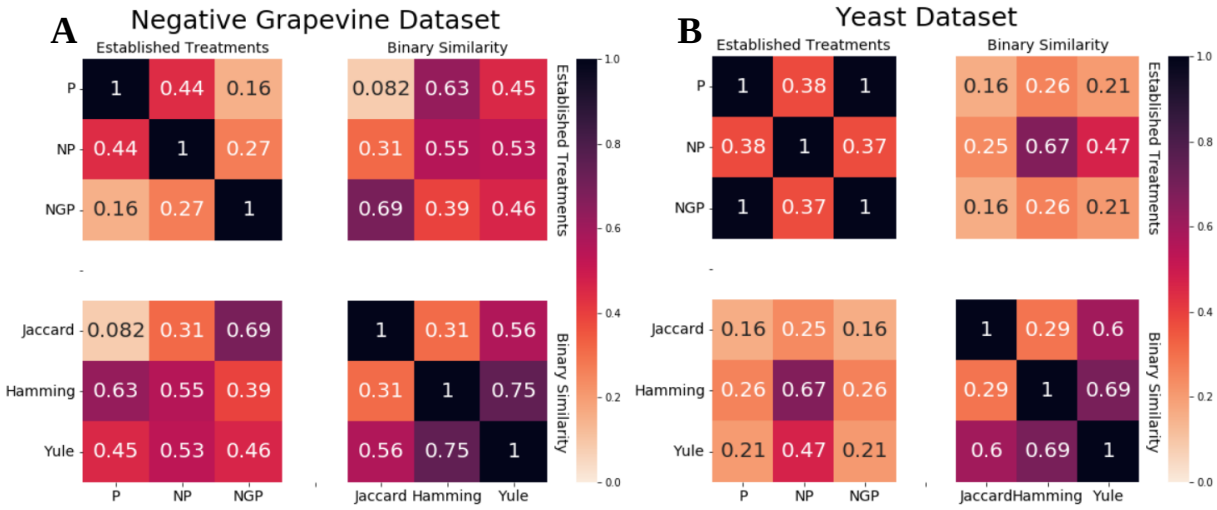
118. Longnecker K, Kujawinski E. Using network analysis to discern compositional patterns in ultrahigh resolution mass spectrometry data of dissolved organic matter: Network analysis and mass spectrometry. *Rapid Commun Mass Spectrom*. 2016;30. doi:10.1002/rcm.7719

# 6. Annexes

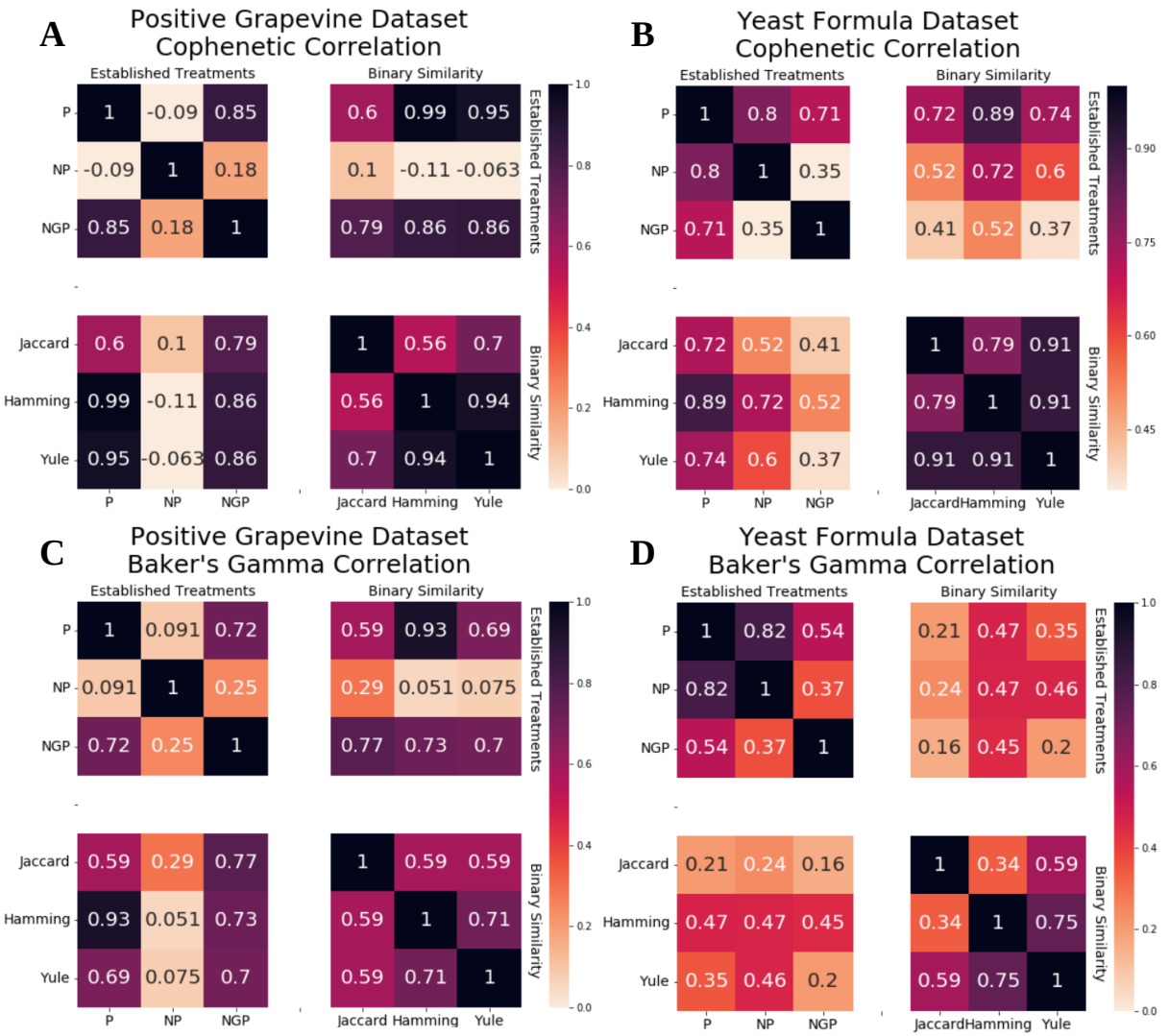



**Suppl. Figure 6.1: Hierarchical Clustering Analysis (HCA) dendrograms of the Positive Grapevine Dataset (A) and Yeast Formula Dataset (B).** The datasets were treated with the P (**1**), NP (**2**), NGP (**3**) pre-treatments using Euclidian distances or BinSim pre-treatment (**4**) using the Jaccard Dissimilarity distance metric. HCA was performed with UPGMA

linkage method. Pre-treatments: P – Pareto scaling, N – Normalization by leucine enkephalin, G – Generalized logarithmic transformation; BinSim – Binary Similarity; *Vitis* genotypes abbreviations are indicated in Table 2.1.




**Suppl. Figure 6.2: Heatmaps of the Baker's Gamma Correlation between the dendrograms of all differently treated dataset pairs of the Negative Grapevine Dataset (A) and of the Yeast Dataset (B).** For the datasets treated with the BinSim pre-treatment, 3 representative binary distance metrics were used: Jaccard, Hamming and Yule dissimilarities/distances). For the others, Euclidian distance was used. Pre-treatments: P – Pareto scaling, N – Normalization by leucine enkephalin, G – Generalized logarithmic transformation.



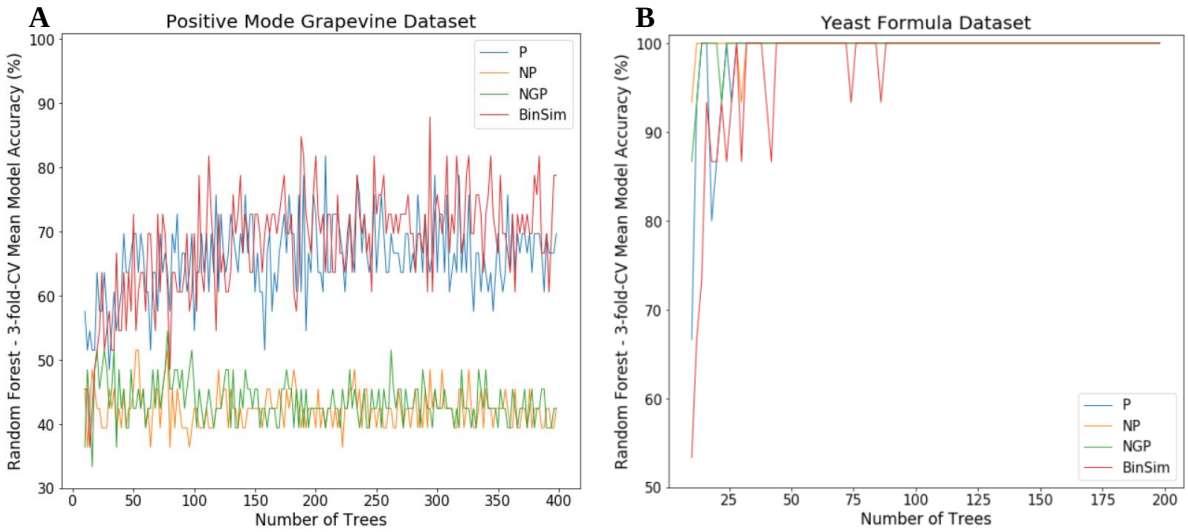

**Suppl. Figure 6.3: Heatmaps of the Cophenetic Correlation (A,B) and the Baker's Gamma Correlation (C,D) between the dendrograms of all differently treated Positive Grapevine Dataset (A) or Yeast Formula Dataset, respectively.** For the datasets treated with the Binary Similarity pre-treatment, 3 representative binary distance metrics were used: Jaccard, Hamming and Yule dissimilarities/distances). Pre-treatments: P – Pareto scaling, N – Normalization by leucine enkephalin, G – Generalized logarithmic transformation.

**Suppl. Table 6.1: Discrimination Distance, correct clustering and correct first cluster percentages of the HCA of the Positive Grapevine and Yeast Formula Datasets after different treatments.** Binary Similarity has 3 different results based on the distance metric used. Pre-Treatments: BinSim – Binary Similarity, P – Pareto Scaling, N – Normalization by a reference feature, G – Generalized Logarithmic Transformation.

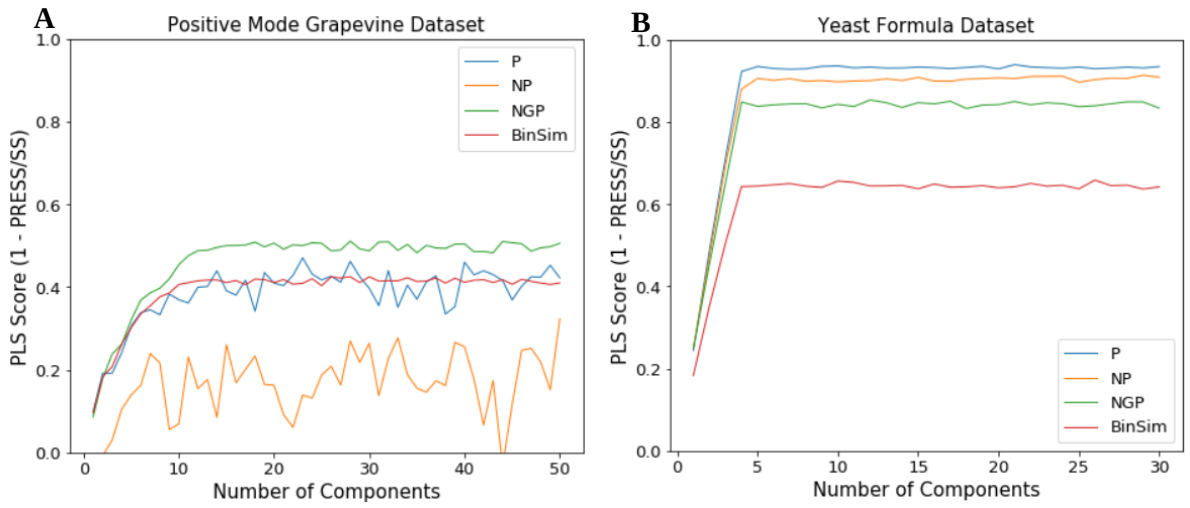
| | Grapevine Dataset ($ESI^+$) – 11 groups | | | | | | Yeast Formula Dataset – 5 groups | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Metrics \ Dataset Treatments | P | NP | NGP | Jaccard (BinSim) | Hamming (BinSim) | Yule (BinSim) | P | NP | NGP | Jaccard (BinSim) | Hamming (BinSim) | Yule (BinSim) |
| Discrimination Distance | 0.034 | 0.015 | 0.025 | 0.043 | 0.03 | 0.082 | 0.30 | 0.22 | 0.22 | 0.14 | 0.19 | 0.34 |
| Correct Clustering (%) | 27 | 27 | 18 | 45 | 18 | 45 | 100 | 100 | 100 | 100 | 100 | 100 |
| Correct First Cluster (%) | 33 | 27 | 48 | 52 | 61 | 70 | 100 | 100 | 100 | 100 | 100 | 100 |

**Suppl. Table 6.2: Discrimination Distance, correct clustering percentage and adjusted Rand Index of the K-means Clustering analysis of the Positive Grapevine and Yeast Formula datasets after different treatments.** Pre-Treatments: P – Pareto Scaling, N – Normalization by a reference feature, G – Generalized Logarithmic Transformation.

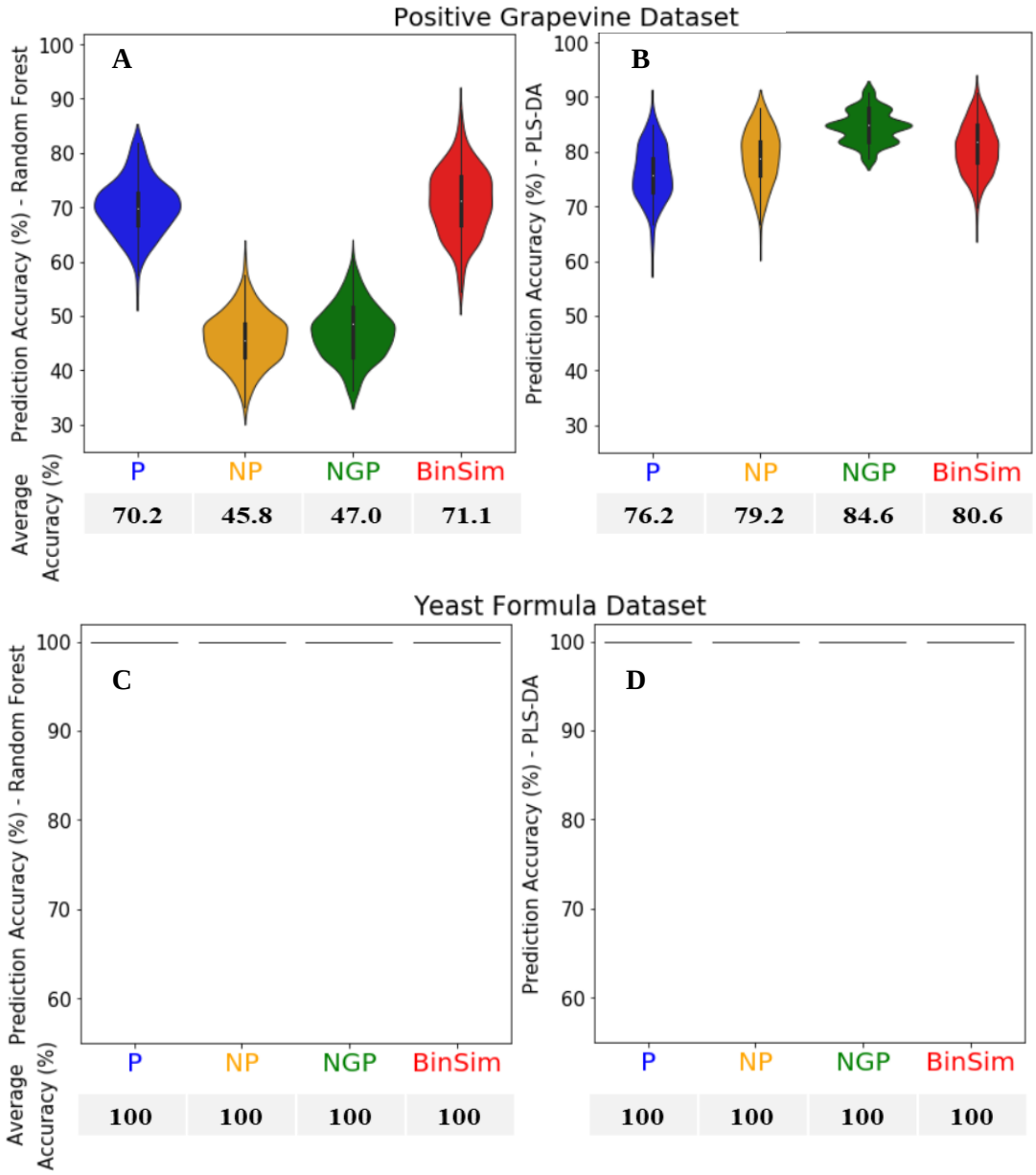
| | Grapevine Dataset (ESI$^+$) | | | | Yeast Formula Dataset | | | |
|---|---|---|---|---|---|---|---|---|
| Dataset / Treatments / Metrics | P | NP | NGP | Binary Similarity | P | NP | NGP | Binary Similarity |
| Discrimination Distance | 0 | 0.035 | 0 | 0.066 | 0.72 | 0.39 | 0.36 | 0.85 |
| Correct Clustering (%) | 0 | 23 | 0 | 9.1 | 100 | 100 | 100 | 100 |
| Adjusted Rand Index | 0.22 | 0.49 | 0.31 | 0.23 | 1 | 1 | 1 | 1 |




**Suppl. Figure 6.4: Tuning of the number of trees used to build the Random Forest models.** Random Forest predictive accuracy as a function of the number of trees used in the forest for the Positive Grapevine (**A**) and Yeast Formula (**B**) datasets with different pre-treatments. Accuracy was estimated by stratified 3-fold cross-validation. Pre-Treatments: BinSim – Binary Similarity, P – Pareto Scaling, N – Normalization by a reference feature, G – Generalized Logarithmic Transformation.




**Suppl. Figure 6.5: Optimization of the number of components used to build the PLS-DA models.** 1- (Predictive Residual Sum of Squares (PRESS) / residual Sum of Squares (SS)) or $Q^2$ estimated by stratified 3-fold cross-validation of PLS regressions of the Positive Grapevine (**A**) and the Yeast Formula (**B**) datasets with different number of components. Pre-Treatments: BinSim – Binary Similarity, P – Pareto Scaling, N – Normalization by a reference feature, G – Generalized Logarithmic Transformation.




**Suppl. Figure 6.6: Distribution of the prediction accuracy of Random Forest and PLS-DA models.** Violin plots of the distribution of the prediction accuracy of 200 iterations of Random Forest and PLS-DA models built based on the Positive Grapevine dataset (**A** and **B**, respectively) and on the Yeast Formula dataset (**C** and **D**, respectively). Each iteration's accuracy is estimated by stratified 3-fold cross-validation. Each iteration randomly splits the dataset in three folds. Below the plots, the average prediction accuracy is presented. Pre-Treatments: BinSim – Binary Similarity, P – Pareto Scaling, N – Normalization by a reference feature, G – Generalized Logarithmic Transformation.

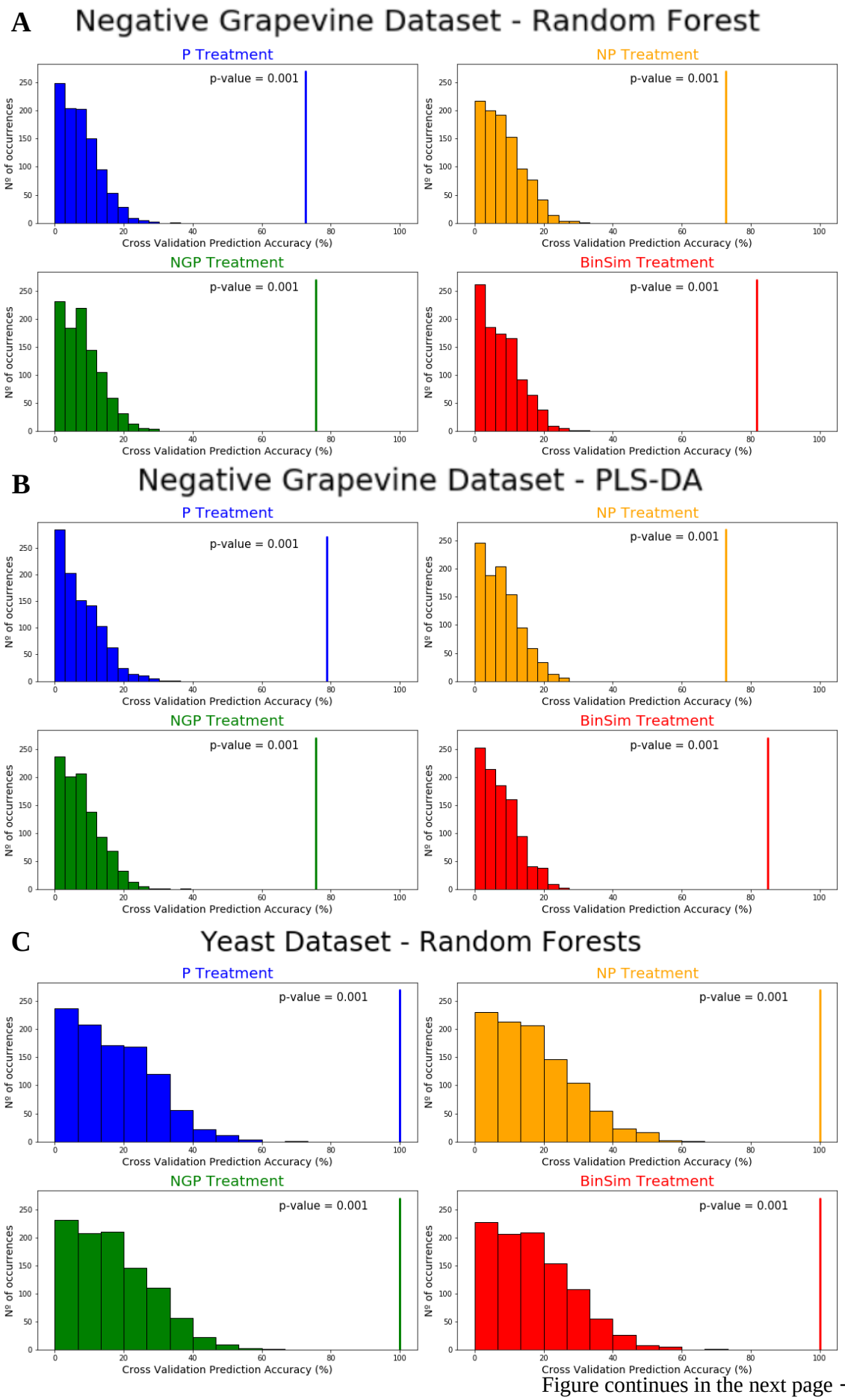



Figure continues in the next page →

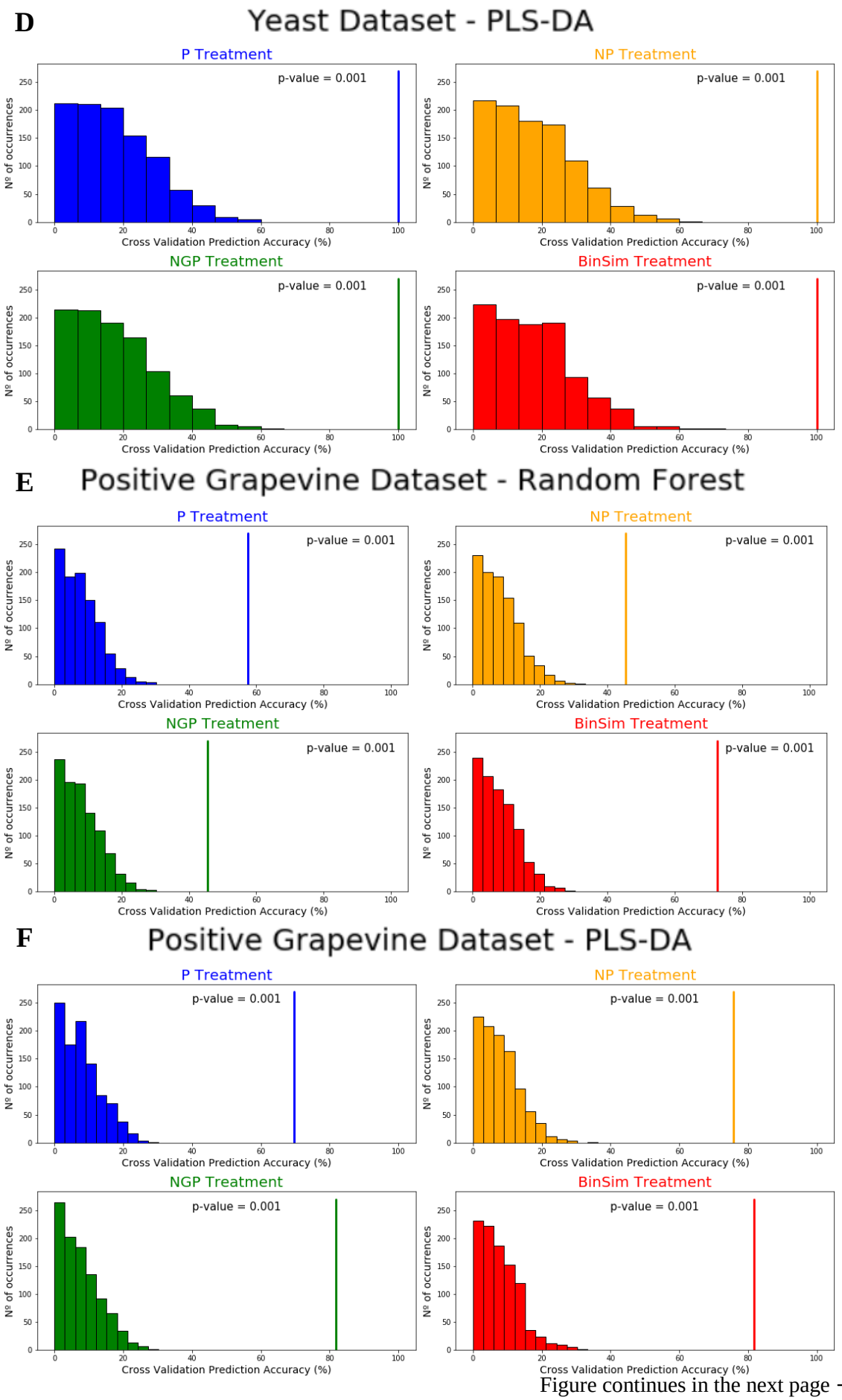
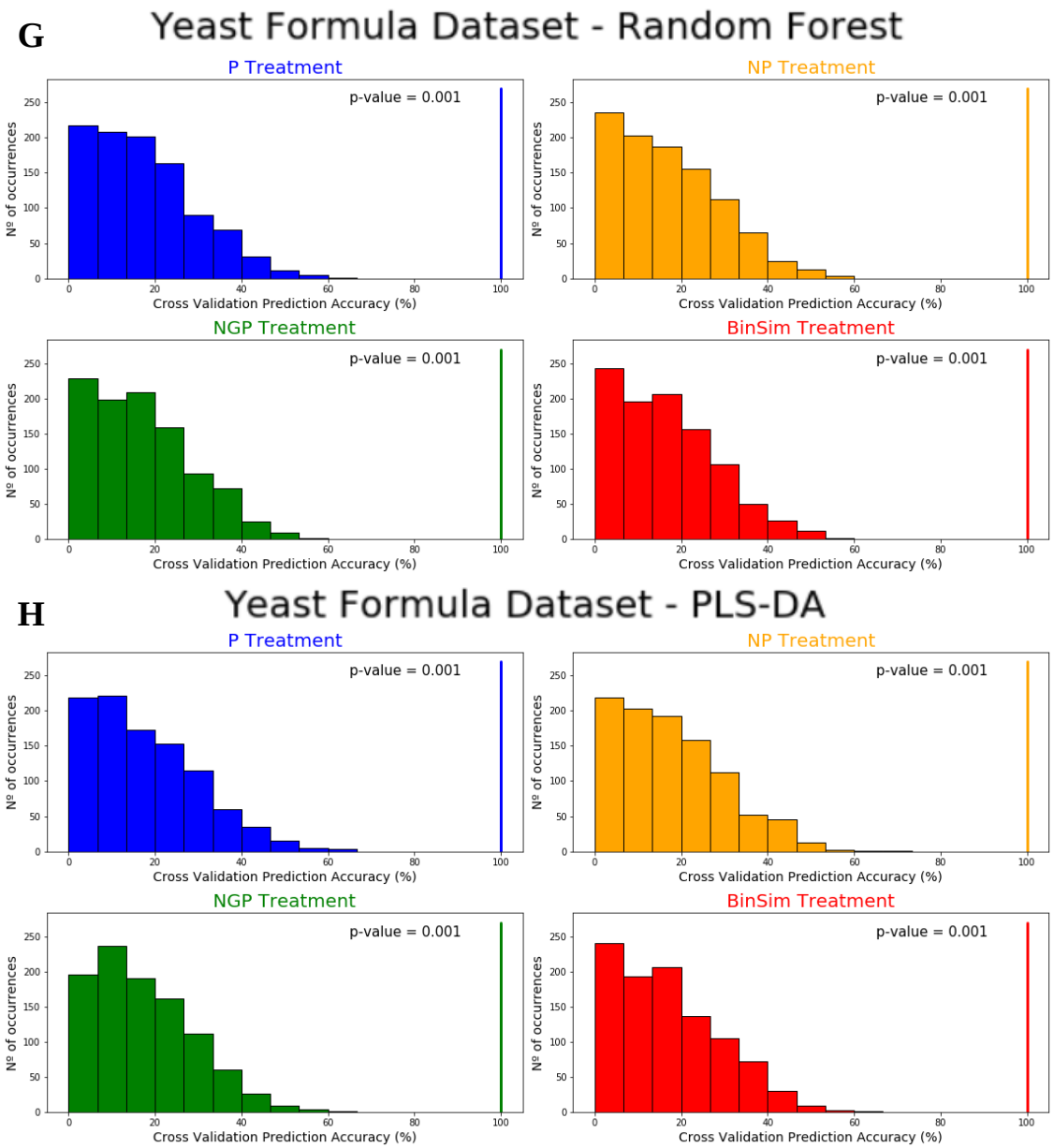
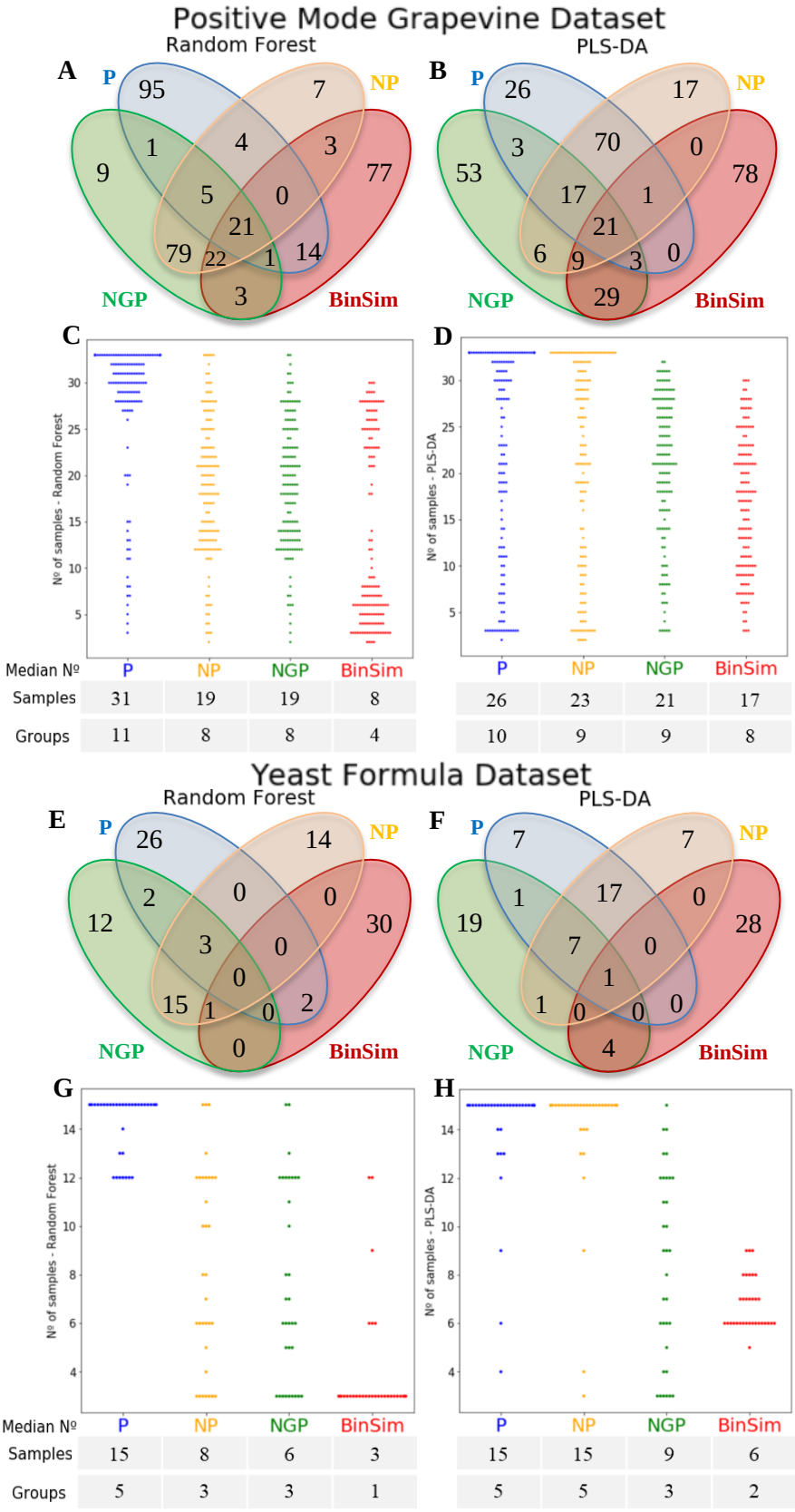



Figure continues in the next page →




**Suppl. Figure 6.7: Permutation test of the Random Forest and PLS-DA models built with each different set of datasets.** Significance diagnostic showing the distribution of predictive accuracy of the Random Forest and PLS-DA models built from each differently treated Negative Grapevine (**A** and **B**, respectively), Yeast (**C** and **D**, respectively), Positive Grapevine (**E** and **F**, respectively) and Yeast Formula (**G** and **H**, respectively) datasets in permutation tests and the *p*-values of the test for accuracy. 1000 permutations were randomly sampled. Vertical lines show the accuracy of model with non-permuted labels. Accuracy was estimated by stratified 3-fold cross-validation. *P*-value is: (nº permutations with higher prediction accuracies than the non-permutated dataset + 1)/(nº of permutations + 1). Pre-Treatments: BinSim – Binary Similarity, P – Pareto Scaling, N – Normalization by a reference feature, G – Generalized Logarithmic Transformation.

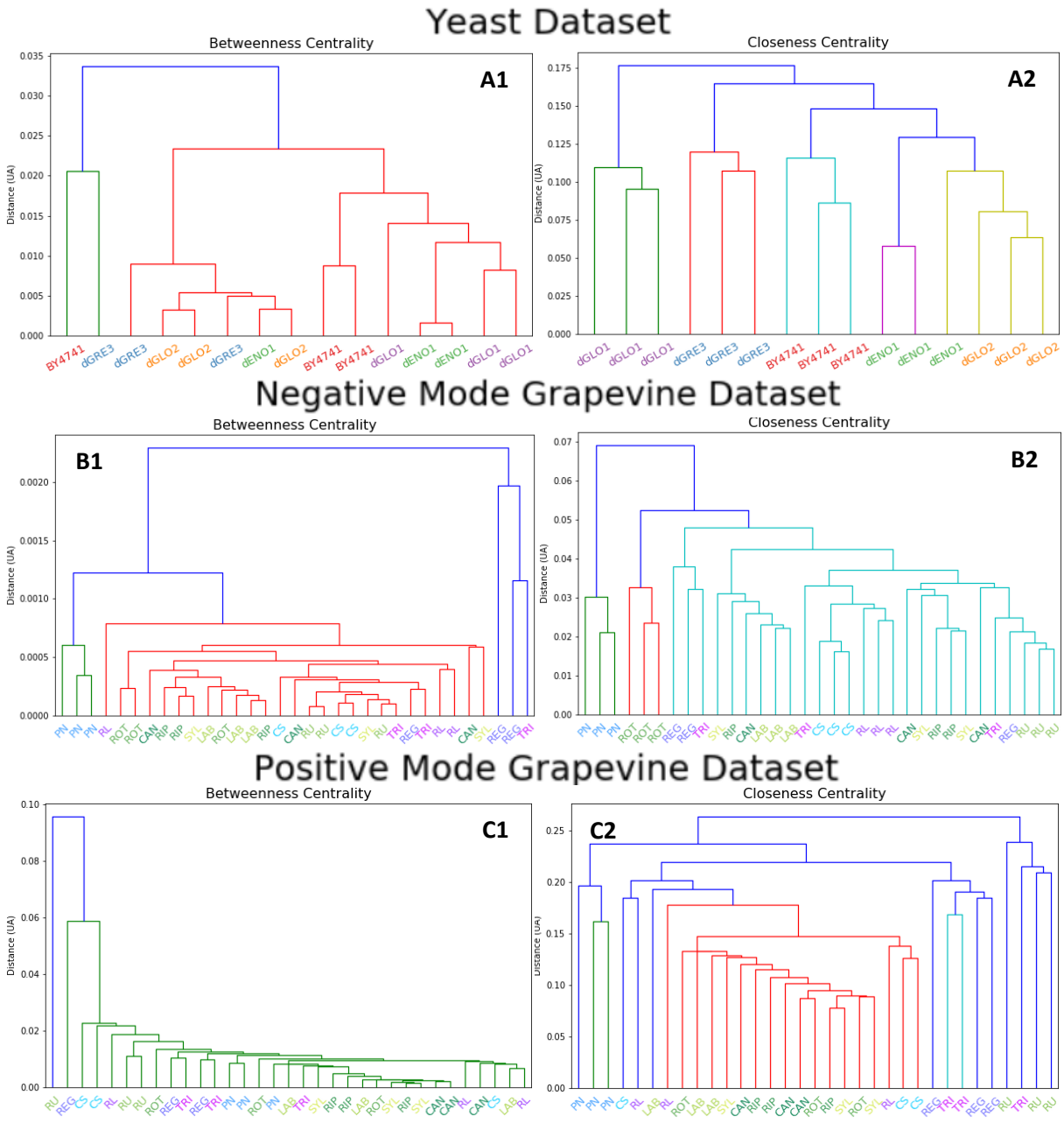



**Suppl. Figure 6.8: Characteristics of the most important features used to build the Random Forest and the PLS-DA models.** Venn diagrams of the 2% most important features used to build the Random Forest (by the Gini Importance method) and the PLS-DA (by the VIP method) models from each differently treated Positive Grapevine Dataset (141 features, **A** and **B**, respectively) and Yeast Formula Dataset (33 features, **E** and **F**, respectively). Distribution plots of the number of samples each important feature appears in their dataset on each differently treated Positive Grapevine Dataset (**C, D**) and Yeast Formula Dataset (**G, H**) with the median number of samples and different groups they appear in below the plots. Pre-

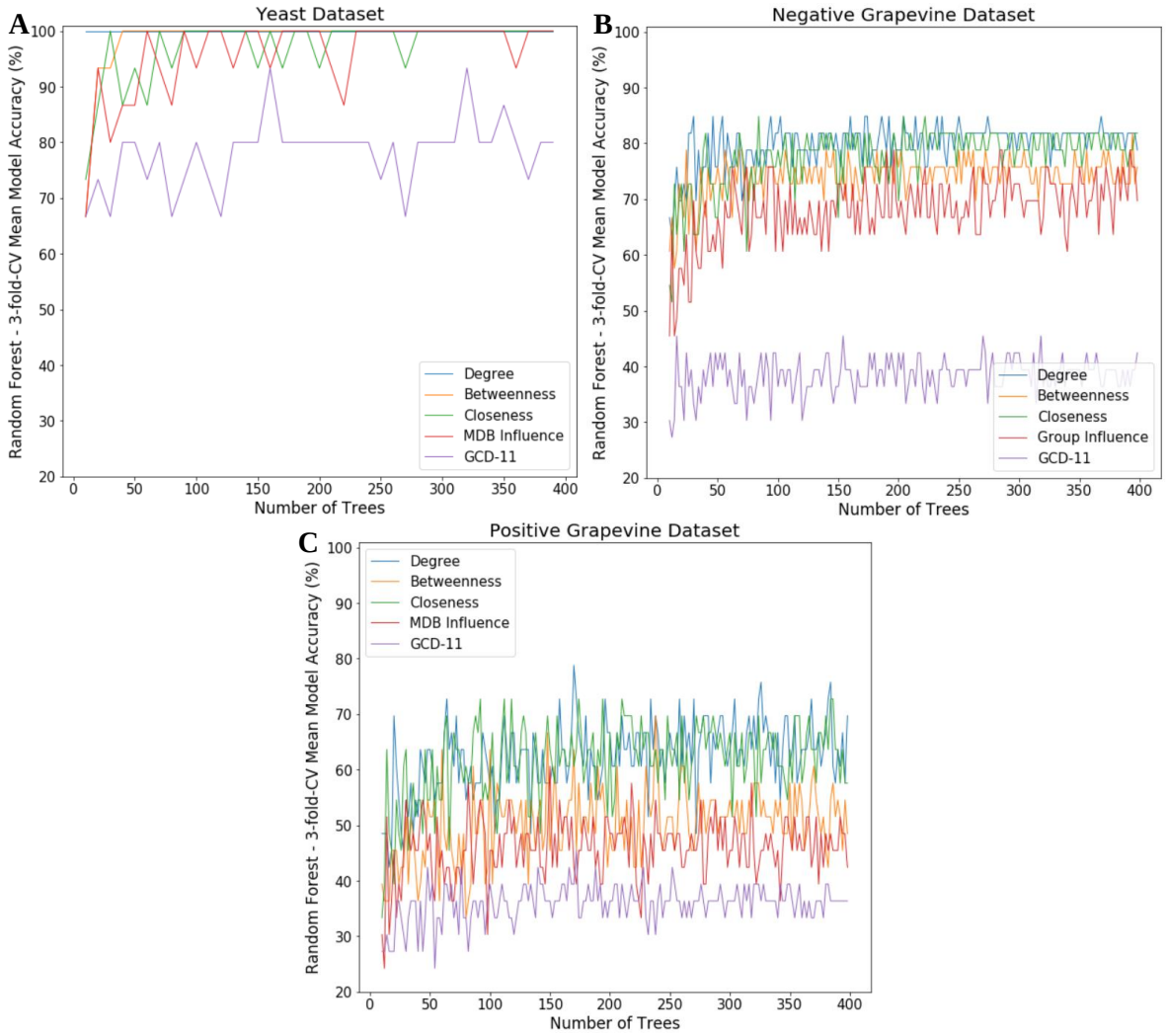
Treatments: BinSim – Binary Similarity, P – Pareto Scaling, N – Normalization by a reference feature, G – Generalized Logarithmic Transformation.

**Suppl. Table 6.3: Impact of each MDB in building the 3 full networks.** Counts of the number of edges that each mass difference corresponding to a specific elemental transformation established in the complete YD network, Negative GD network and Positive GD network.

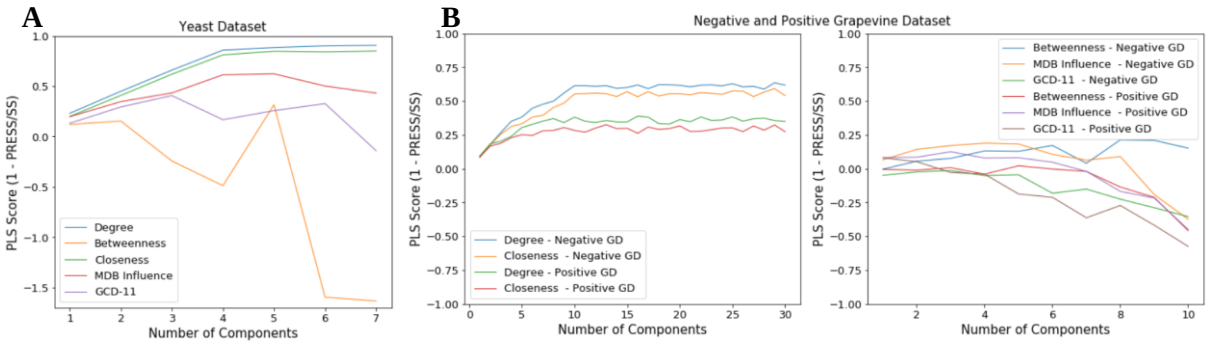
| Elemental Transformation (MDB) | Yeast Dataset (YD) | Negative Grapevine Dataset (GD) | Positive Grapevine Dataset (GD) |
|---|---|---|---|
| **O (-NH)** | 38 | 50 | 291 |
| **$NH_3$ (-O)** | 32 | 120 | 214 |
| **$H_2$** | 101 | 138 | 763 |
| **$CH_2$** | 152 | 173 | 1229 |
| **O** | 100 | 135 | 821 |
| **$H_2O$** | 96 | 58 | 735 |
| **NCH** | 27 | 34 | 289 |
| **CO** | 78 | 111 | 612 |
| **CHOH** | 6 | 36 | 98 |
| **S** | 13 | 8 | 118 |
| **$C_2H_2O$** | 62 | 39 | 544 |
| **CONH** | 19 | 24 | 261 |
| **$CO_2$** | 39 | 61 | 386 |
| **$SO_3$** | 11 | 9 | 121 |
| **$PO_3H$** | 36 | 9 | 115 |




**Suppl. Figure 6.9: Hierarchical Clustering Analysis (HCA) of the different secondary datasets obtained from sample MDiNs.** Dendrograms of the HCA of the Yeast (**A**), Negative Grapevine (**B**) and Positive Grapevine (**C**) sample networks after betweenness (**1**) or closeness centrality (**2**) network analysis of each one. HCA was performed with UPGMA linkage method and Euclidian distance metric. *Vitis* genotypes abbreviations are indicated in Table 2.1.




**Suppl. Figure 6.10: Tuning of the number of trees used to build the Random Forest models from the secondary datasets built from sample networks.** Random Forest Model predictive accuracy as a function of the number of trees used in the forest of the secondary datasets built from the Yeast Dataset (A), Negative Grapevine Dataset (B) and Positive Grapevine Datasets (C) sample networks. Accuracy was estimated by stratified 3-fold cross-validation.




**Suppl. Figure 6.11: Optimization of the number of components used to build PLS-DA models from the different secondary datasets.** 1- (Predictive Residual Sum of Squares (PRESS) / residual Sum of Squares (SS)) or $Q^2$ estimated by stratified 3-fold cross-validation of PLS regressions with different number of components of the secondary datasets obtained from the sample networks of the Yeast Dataset (**A**), Negative and Positive Grapevine Datasets (**B**). GD – Grapevine Dataset.

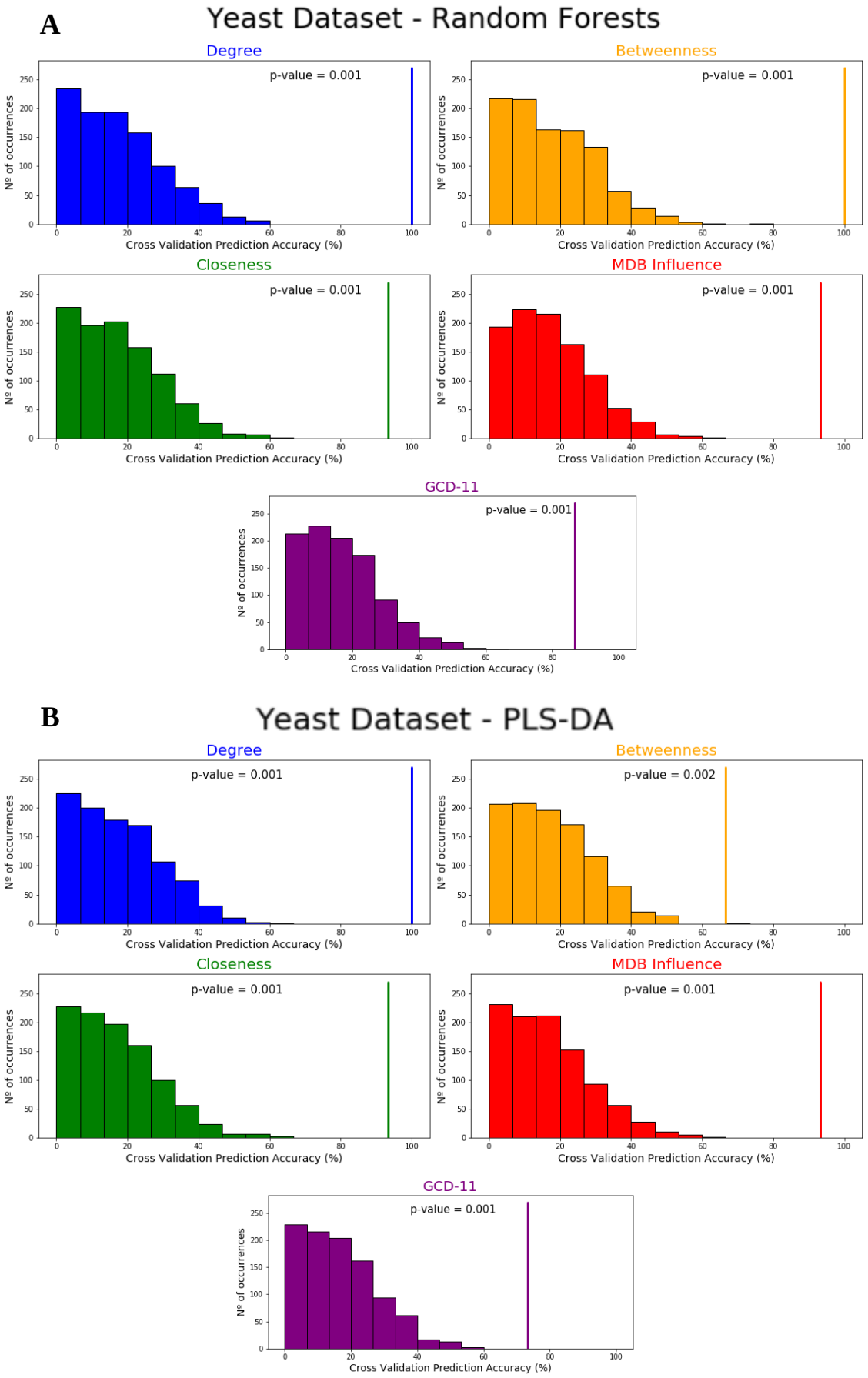



Figure continues in the next page →

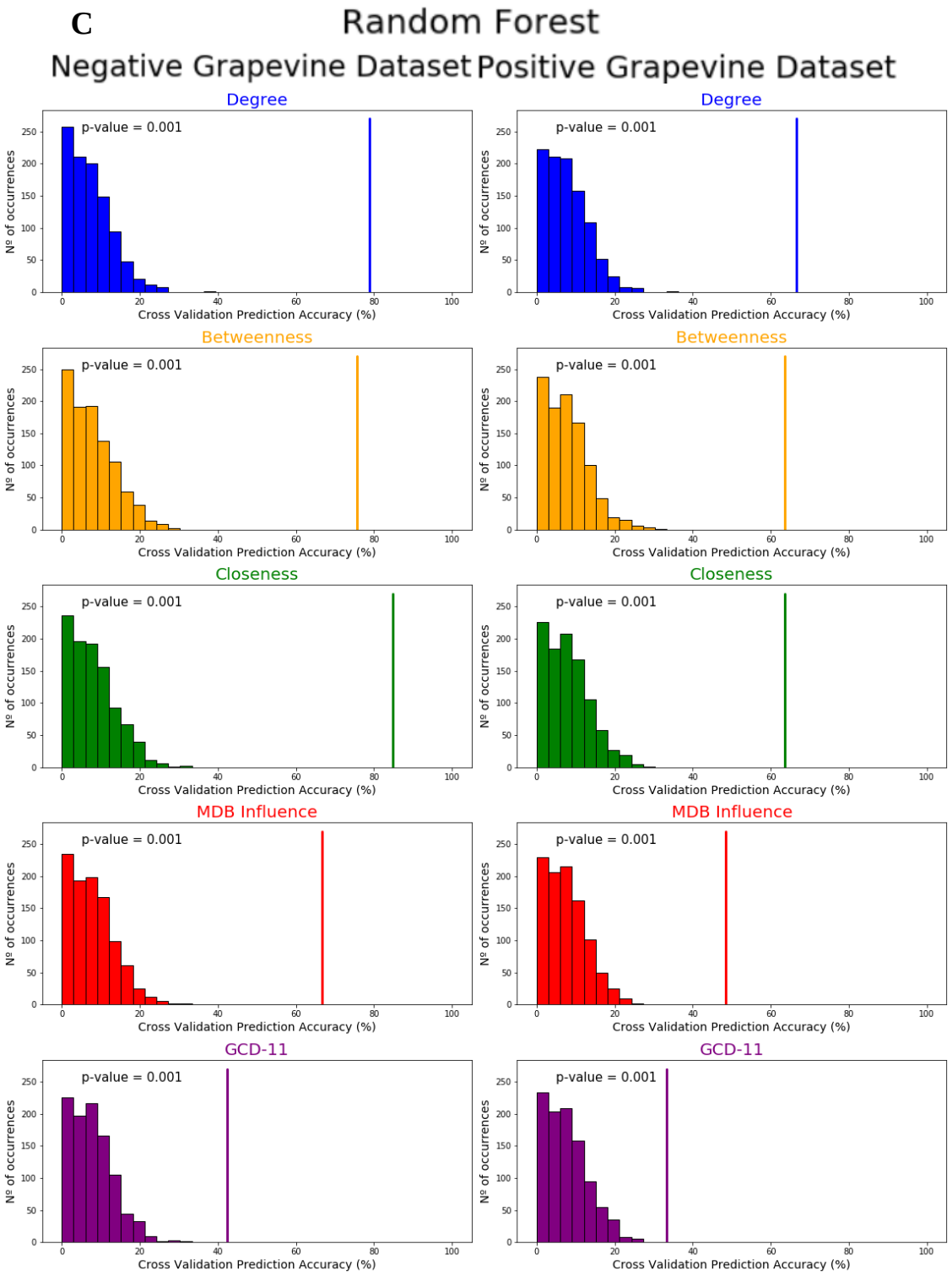
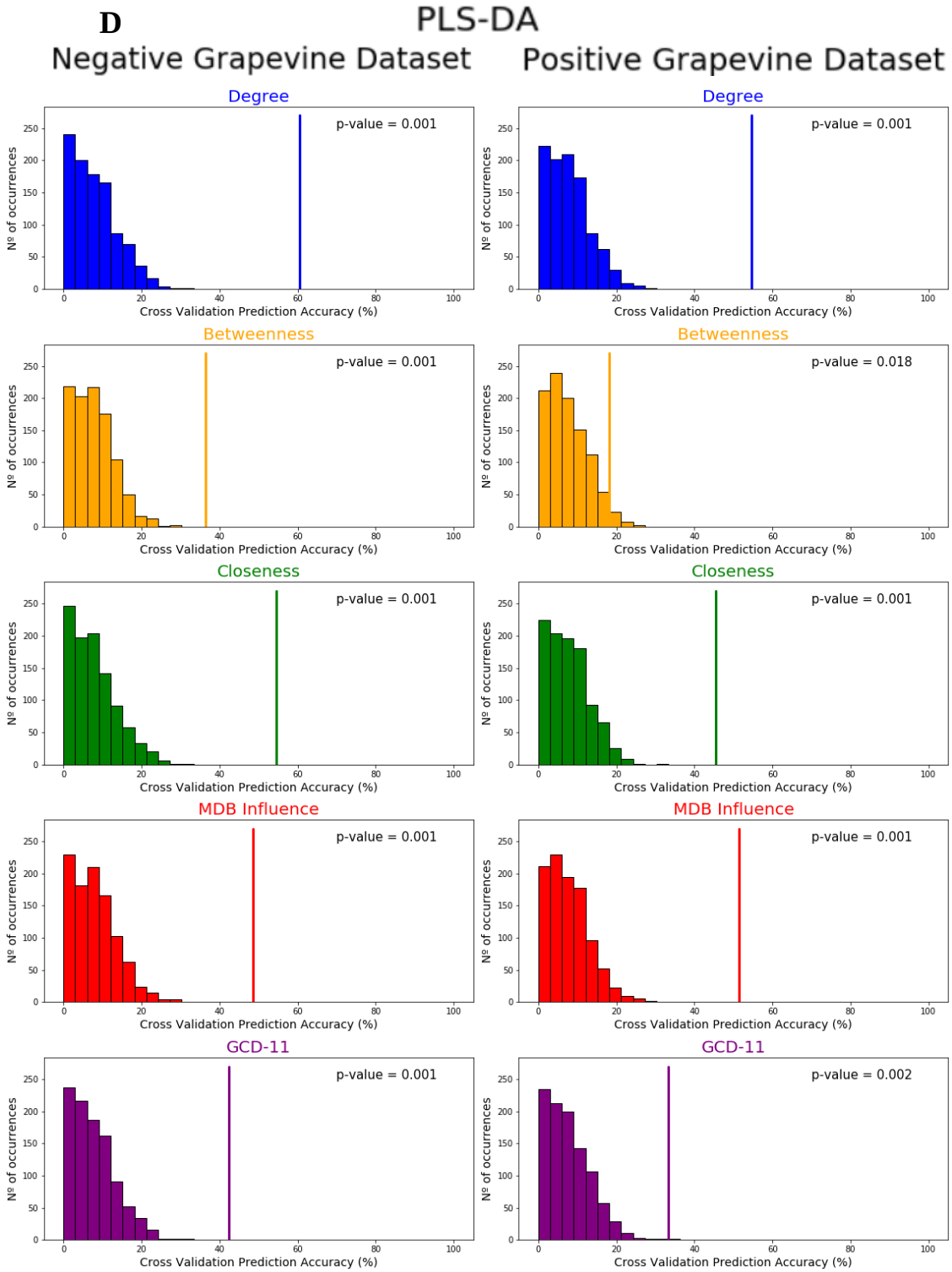



Figure continues in the next page →




**Suppl. Figure 6.12: Permutation test of the Random Forest and PLS-DA models built based on each set of secondary datasets.** Significance diagnostic showing the distribution of predictive accuracy in permutation tests and the *p*-values of the test for accuracy of the Random Forest and PLS-DA models built based on the secondary datasets generated from the sample networks of the Yeast (**A** and **B**, respectively), Negative and Positive Grapevine (**C** – Random Forests – and **D** – PLS-DA) datasets. 1000 permutations were randomly sampled. Vertical lines show the accuracy of model with non-permuted labels. Accuracy was estimated by stratified 3-fold cross-validation. *P*-value is: (nº permutations with higher prediction accuracies than the non-permutated dataset + 1)/(nº of permutations + 1).