Development of Metabolomics Approaches to Decipher Chemical Interactions in Microbial Communities

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät

der Eberhard Karls Universität Tübingen

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

(Dr. rer. nat.)

vorgelegt von

M.Sc Abzer Kelminal Pakkir Mohamed Shah

aus Palayamkottai, Indien

Tübingen

2025

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	30.01.2026
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatter/-in:	Prof. Dr. Daniel Petras
2. Berichterstatter/-in:	Prof. Dr. Nadine Ziemert

Declaration of Authorship / Eidesstattliche Erklärung

I hereby declare that I have written this dissertation independently and that the work presented here is the result of my own research. All sources of information, data, ideas, and quotations taken from the work of others have been appropriately acknowledged and cited. I affirm that no part of this thesis has been submitted in any previous application for a degree.
In preparing this thesis, I used digital writing assistants (e.g., ChatGPT 5) solely for language refinement, clarity, and editing. These tools were not used to generate scientific ideas, results, or interpretations. All scientific concepts, analyses, and conclusions in this dissertation originate from my own work and from collaborations explicitly described in the Author Contributions section. I have endeavored to provide accurate references and follow good scientific practice to the best of my knowledge.

Hiermit erkläre ich, dass ich diese Dissertation selbstständig verfasst habe und dass die hierin präsentierten Arbeiten das Ergebnis meiner eigenen Forschung sind. Alle Informationen, Daten, Ideen und Zitate, die aus den Arbeiten anderer übernommen wurden, sind ordnungsgemäß angegeben und zitiert. Ich versichere, dass kein Teil dieser Dissertation bereits in einem früherenden Antrag auf einen akademischen Grad eingereicht wurde.
Bei der Erstellung dieser Dissertation habe ich digitale Schreibassistenzsysteme (z. B. ChatGPT 5) ausschließlich zur sprachlichen Überarbeitung, zur Verbesserung der Klarheit und zur stilistischen Bearbeitung verwendet. Diese Werkzeuge wurden nicht zur Generierung wissenschaftlicher Ideen, Ergebnisse oder Interpretationen eingesetzt. Alle wissenschaftlichen Konzepte, Analysen und Schlussfolgerungen in dieser Dissertation stammen aus meiner eigenen Arbeit oder aus den im Abschnitt „Author Contributions“ ausdrücklich beschriebenen Kooperationen. Ich habe durchgehend korrekte Referenzen angegeben und die Grundsätze guter wissenschaftlicher Praxis nach bestem Wissen und Gewissen eingehalten.

Place, Date / Ort, Datum: Tübingen, 08.12.2025

Name: Abzer Kelminal Pakkir Mohamed Shah

Acknowledgement

I would first like to express my deepest gratitude to my supervisor, Prof. Dr. Daniel Petras, for giving me the opportunity to pursue this PhD and for his continuous support, guidance, and trust throughout these years. On both a professional and a personal level, I am deeply thankful for his mentorship. He created an environment where I felt supported, encouraged, and able to grow. I am grateful not only for his scientific guidance, but also for the warmth and kindness he extended to me during these four years. I am equally grateful to my thesis committee members, Prof. Dr. Hannes Link and Prof. Dr. Nadine Ziemert, for their valuable feedback and encouragement. I would also like to thank Dr. Lisa Maier for generously agreeing to join my defense as an additional oral examiner.

I would also like to thank the University of Tübingen for providing a supportive academic environment throughout my PhD, and for the opportunities that shaped my scientific journey over the past four years. My sincere thanks extend to all my lab members as well in Tübingen and in Riverside, whose support, discussions, and kindness made this journey both productive and enjoyable. I am grateful for the supportive environment at the UCR lab and for the warm welcome I received there. I extend my heartfelt thanks to Dr. Mingxun Wang for hosting the web tools associated with my three dissertation chapters and for providing insightful feedback and guidance along the way. His help has been instrumental in shaping the tools and ensuring their accessibility to the broader scientific community.

I am especially grateful to the collaborators of the Virtual Multi-Omics Lab, whose contributions, ideas, and dedication were essential to the success of my first project. I would also like to acknowledge our collaborators from the Universitätsklinikum Tübingen, Prof. Dr. Lisa Maier and Dr. Anne Griesshammer, for their scientific insight and teamwork. My gratitude extends to colleagues who played major roles across my dissertation chapters: Dr. Paolo Stincone, Dr. Jarmo Kalinski, Dr. Axel Walter, Karoline Steuer-Lodd, and Rithi Krishnakumar. In addition to these contributors, many others supported different aspects of this work, especially my lab members, who tested the web apps, shared valuable feedback, and helped refine the tools.

Throughout my PhD, I was fortunate to be part of several collaborations, many thanks to Daniel for opening those doors, each allowing me to learn from exciting science and contribute in meaningful ways. I am also grateful for the opportunities to present my work at workshops and conferences; the discussions and interactions at these events have inspired me and shaped my scientific thinking. There are far more people than I can list here whose support influenced various phases of this journey. Though I cannot name everyone individually, I remain truly grateful for every conversation, project, and partnership that expanded my skills, encouraged new ideas, and helped me grow into the person I am today.

I am sincerely grateful to all the members of the Functional Metabolomics Lab in Tübingen, with whom I have grown both personally and professionally over the past four years. While I value everyone in the lab, I would like to extend a special thanks to Paolo, who has been with me from my very first day until the completion of this thesis. More than a colleague, he has been a true

friend, someone who welcomed me into Tübingen, encouraged me to go out, and helped me build the friendships that shaped my early years here. I am deeply appreciative of his kindness and unwavering support throughout this journey. I am also grateful for his constant check-ins and for taking the time to read and provide feedback on my thesis draft. I would also like to offer a special thanks to Giovanni, Karo, Tilman, Shane, and Simon, with whom I shared many fun moments outside the lab, memories that made my time in Tübingen even more meaningful. I extend my gratitude to Sibgha and Lana, their friendship, empathy, and thoughtful conversations have been a source of comfort and grounding. Thanks to cute little Karlito, watching him grow over the years has brought me unexpected joy and reminded me of the simple beauty and meaning in everyday life.

My heartfelt thanks go to my friends in Tübingen, in Jena, and in India for being my emotional anchors throughout these four years. My friends from Jena are an extended family to me, filling my life with comfort, laughter, and a sense of belonging. A special thanks to Vasiya, Mano, and Priya from India, and to Priya from Jena, whose presence has been a constant source of strength. I am equally thankful for the friends I made in Tübingen through friends-of-friends circles, especially Aishwarya, Lobus, and many others whose names I may unintentionally miss but whose kindness and companionship I will always cherish. Every encounter, every conversation, and every moment shared has shaped me in meaningful ways, and I remain sincerely grateful for the role each of you played in my journey. I would like to give special thanks to Sharmi (Dr. Sharmila Sekar), for guiding me through the thesis documentation and for her constant reassurance.

Most importantly, I am profoundly grateful to my family. My mum, dad and my aunt have been my greatest sources of strength. I am especially thankful to my dad for always having the right words to calm me when I felt overwhelmed. I also thank my brothers for their love, patience, and encouragement throughout this long journey. Special thanks to my cute little nephew, Umar, for brightening my days. To all my friends and family, I want to quote Sheldon from *The Big Bang Theory*: *“I apologize if I haven’t been the friend you deserve, but I want you to know that, in my way, I love you all.”*

Above all, my faith and personal spirituality guided me through moments of doubt and exhaustion, reminding me that ease follows hardship and that perseverance is always rewarded.

“فَإِنَّ مَعَ الْعُسْرِ يُسْرًا”

In a nutshell, this was a journey filled with its own ups and downs, full of learning, both personally and professionally. I am grateful for everyone who was part of it. This thesis is as much theirs as it is mine. As much as this marks the end of my PhD journey, it is also the beginning of a new chapter, one I look forward to with hope and excitement. And if you are someone like me who reads acknowledgements in theses and books for inspiration: “If I can do it, you can do it as well”

உள்ளுவ தெல்லாம் உயர்வுள்ளல் மற்றது
தள்ளினுந் தள்ளாமை நீர்த்து.

Table of Contents

List of Tables

Chapter 2

Chapter 3

Abbreviations

ASV – Amplicon Sequence Variant
CFU – Colony-Forming Units
ChemProp – Chemical Proportionality
ChemProp1 – Two-timepoint log-ratio–based ChemProp score
ChemProp2 – Multi-timepoint correlation-based ChemProp score
CLR – Centered Log-Ratio Transformation
CorrOmics – Cross-omics correlation framework developed in this thesis
EDA – Exploratory Data Analysis
FBMN – Feature-Based Molecular Networking
FBMN-STATS – Statistical workflow integrating FBMN with metabolomics data analysis
FDR – False Discovery Rate
FASST – Fast Alignment and Spectrum Search Tool
GC – Gas Chromatography
GNPS– Global Natural Products Social Molecular Networking platform
HCA – Hierarchical Cluster Analysis
IIN – Ion Identity Networking
LC – Liquid Chromatography
MALDI – Matrix-Assisted Laser Desorption/Ionization
m/z – Mass-to-Charge Ratio
MI – Mutual Information
MS/MS – Tandem Mass Spectrometry
NMR – Nuclear Magnetic Resonance
OTU – Operational Taxonomic Unit
PCA – Principal Component Analysis
PCoA – Principal Coordinates Analysis
QIIME – Quantitative Insights Into Microbial Ecology
QC – Quality Control
RF – Random Forest
RT – Retention Time
SynCom – Synthetic Community
TIC – Total Ion Current
USI – Universal Spectrum Identifier

Summary

Microbial communities shape nearly every environment on earth, from the human gut to soil and marine ecosystems, through dense networks of chemical interactions. These interactions are mediated by metabolites that microbes produce, modify, exchange, or degrade, influencing processes such as drug metabolism, nutrient cycling, and host physiology. Untargeted LC-MS/MS metabolomics provides a direct window into this chemical layer, yet interpreting the high-dimensional, sparse, and largely unannotated data remains a major challenge. This thesis develops computational approaches that transform raw metabolomics data into mechanistic and ecological insight, focusing on three complementary goals: statistical exploration, biotransformation inference, and cross-omics integration.

Chapter 1 presents **FBMN-STATS**, a statistical workflow that integrates Feature-Based Molecular Networking (FBMN) with robust exploratory and differential analyses. Available as R, Python, and QIIME 2 workflows and as an interactive web application, FBMN-STATS guides users through preprocessing, multivariate and univariate analyses. It provides a reproducible and accessible framework for analyzing LC-MS/MS datasets within the GNPS ecosystem.

Chapter 2 introduces **ChemProp2**, a method to infer directional chemical relationships from time-resolved metabolomics data. While FBMN groups structurally related features, it lacks directional information. ChemProp2 addresses this by quantifying precursor-product patterns through anti-correlated temporal trajectories and cascade scoring, revealing multi-step transformation pathways. Applied to a gut synthetic community treated with 50 clinical drugs, ChemProp2 uncovered sequential degradation products and linked several transformations to microbial dynamics.

Chapter 3 extends beyond metabolite-metabolite relationships and links chemical patterns to microbial ecology. Because metabolite trajectories are strongly influenced by microbial abundance dynamics, we developed **CorrOmics** to perform scalable, correlation-based integration of LC-MS/MS features with microbial profiles. The tool incorporates a target-decoy FDR strategy to reduce false positives and supports hierarchical binning to stabilize taxa-level interpretation. Benchmarking with a 13-strain synthetic community demonstrated the strengths and limitations of correlation-based cross-omics analysis, emphasizing the role of experimental design and normalization.

Together, FBMN-STATS, ChemProp2, and CorrOmics form a cohesive workflow that advances metabolomics from statistical exploration to mechanistic inference and ecological integration. All tools are openly accessible via GNPS2, with source code provided through the Functional Metabolomics Lab GitHub

Zusammenfassung

Mikrobielle Gemeinschaften prägen nahezu jede Umgebung auf der Erde, vom menschlichen Darm über den Boden bis hin zu marinen Ökosystemen, durch dichte Netzwerke chemischer Wechselwirkungen. Diese Wechselwirkungen werden durch Metaboliten vermittelt, die Mikroben produzieren, modifizieren, austauschen oder abbauen und die Prozesse wie den Arzneimittelstoffwechsel, den Nährstoffkreislauf und die Physiologie des Wirts beeinflussen. Die nicht zielgerichtete LC-MS/MS-Metabolomik bietet einen direkten Einblick in diese chemische Ebene, doch die Interpretation der hochdimensionalen, spärlichen und weitgehend unannotierten Daten bleibt eine grosse Herausforderung. Diese Arbeit entwickelt computergestützte Ansätze, die rohe Metabolomikdaten in mechanistische und ökologische Erkenntnisse umwandeln, wobei der Schwerpunkt auf drei sich ergänzenden Zielen liegt: statistische Untersuchung, Rückschlüsse auf die Biotransformation und omicsübergreifende Integration.

Kapitel 1 stellt **FBMN-STATS** vor, einen statistischen Workflow, der Feature-Based Molecular Networking (FBMN) mit robusten explorativen und differentiellen Analysen integriert. FBMN-STATS ist als R-, Python- und QIIME 2-Workflow sowie als interaktive Webanwendung verfügbar und führt Benutzer durch die Vorverarbeitung sowie multivariate und univariate Analysen. Es bietet einen reproduzierbaren und zugänglichen Rahmen für die Analyse von LC-MS/MS-Datensätzen innerhalb des GNPS-Ökosystems.

Kapitel 2 stellt **ChemProp2** vor, eine Methode zur Ableitung gerichteter chemischer Beziehungen aus zeitaufgelösten Metabolomikdaten. Während FBMN strukturell verwandte Merkmale gruppiert, fehlen ihm gerichtete Informationen. ChemProp2 behebt dieses Problem, indem es Vorläufer-Produkt-Muster durch antikorrelierte zeitliche Verläufe und Kaskadenbewertung quantifiziert und so mehrstufige Umwandlungswege aufzeigt. Angewandt auf eine synthetische mikrobielle Gemeinschaft, die mit 50 klinischen Medikamenten behandelt wurde, deckte ChemProp2 sequenzielle Abbauprodukte auf und verband mehrere metabolische Umwandlungen mit der mikrobiellen Dynamik.

Kapitel 3 geht über die Beziehungen zwischen Metaboliten hinaus und verbindet chemische Muster mit der mikrobiellen Ökologie. Da die Trajektorien von Metaboliten stark von der Dynamik der mikrobiellen Abundanz beeinflusst werden, wurde **CorrOmics** entwickelt, um eine skalierbare, korrelationsbasierte Integration von LC-MS/MS-Merkmalen mit mikrobiellen Profilen durchzuführen. Das Tool beinhaltet eine Target-Decoy-FDR-Strategie zur Reduzierung von Fehlalarmen und unterstützt hierarchisches Binning zur Stabilisierung der Interpretation auf Taxa-Ebene. Ein Benchmarking mit einer synthetischen Gemeinschaft aus 13 Stämmen zeigte die Stärken und Grenzen der korrelationsbasierten Cross-Omics-Analyse auf und unterstrich die Bedeutung von Versuchsdesign und Normalisierung.

Zusammen bilden FBMN-STATS, ChemProp2 und CorrOmics einen zusammenhängenden Computer gestützten Arbeitsablauf, der die Metabolomik von der statistischen Untersuchung zur

mechanistischen Schlussfolgerung und ökologischen Integration weiterentwickelt. Alle Tools sind über GNPS2 frei zugänglich, der Quellcode wird über das Functional Metabolomics Lab GitHub bereitgestellt.

Chapter 1

General Introduction

1.1 Microbial Communities as Drivers of Life and Chemistry

Microorganisms rarely exist in isolation. Instead, they assemble into highly interactive intra- and inter-species communities that inhabit nearly every environment on Earth[1]. Through processes such as metabolite exchange, cross-feeding, signaling, and co-metabolism, these communities continually modify their surroundings and fundamentally enable ecosystem function[2]. As such, microbial communities and the chemical dynamics among them, drive major terrestrial and marine biogeochemical cycles, including carbon, nitrogen, and sulfur cycling, that underpin global ecosystem productivity and climate regulation[3,4].

In human host-associated systems, microbial communities play equally essential roles, modulating immune development, digesting complex nutrients, and influencing metabolic and inflammatory disease risk[5–8]. When this balance is disrupted, a state known as dysbiosis, microbial composition and function can shift in harmful ways, increasing susceptibility to infection, inflammation, and disease[9]. Dysbiosis can arise from many intrinsic and extrinsic factors, including antibiotics, other medications, pathogenic infections, diet changes, stress, and underlying health conditions. These stressors may cause temporary, long-term, or even permanent microbiome shifts such as changes in community structure, reduced diversity, or the loss of functionally important microbes. Such changes can alter microbial function and human host-microbe interactions, contributing to broader impacts on physiology and disease risk[10].

The metabolites exchanged within these communities often possess potent bioactivities, including antibiotic, anticancer, and immunosuppressant properties, underscoring their relevance for natural product discovery[3,4]. Microbial communities also display emergent properties that individual species cannot achieve alone, such as degrading complex substrates (e.g., cellulose, plastics) or resisting pathogen invasion[11–13].

A central question in the study of microbial systems is how diverse species coexist despite differences in their competitive ability. Positive species interactions, where one species provides nutrients, detoxifies inhibitory compounds, or physically supports another, has emerged as a key stabilizing mechanism[14]. Understanding these principles is essential for interpreting natural ecosystems and for engineering synthetic consortia in medicine, agriculture, and environmental applications[1,13].

1.2 Chemical Communication and Secondary Metabolism

Microbes constantly sense and reshape their chemical environment. Quorum sensing (QS) exemplifies this, where autoinducers accumulate with cell density and initiate coordinated gene expression across the population[15]. Initially studied as a molecular mechanism, QS is now recognized as an ecological and evolutionary system involving cooperation, conflict, signaling, and chemical manipulation[16].

Beyond QS, microbes produce diverse secondary metabolites, including pigments, alkaloids, antibiotics, terpenoids, and toxins, that mediate competition, defense, and cooperation[17]. These metabolites are tightly regulated, often strain-specific, and enriched under nutrient limitation or stationary-phase conditions, when microbial growth slows due to resource depletion and stress-response pathways become activated[18–20]. These chemical exchanges play fundamental roles in structuring community dynamics and shaping ecological outcomes[21]. Therefore, to understand microbial communities, we must understand their chemistry.

1.3 Omics approaches to study microbial communities

Microbial communities shape ecosystem structure, nutrient turnover, biomass productivity, and the chemical environment they inhabit. Their interactions such as competitive, cooperative, or commensal, drive processes such as nutrient cycling, secondary metabolite exchange, and community succession[22,23]. Modern biology seeks to understand these processes using multi-omics approaches, where each layer reveals a different aspect of microbial function: **genomics** identifies genetic potential[24], **transcriptomics** displays gene expression, which reflects regulatory activity and shows how cells respond to stimuli[25], and **proteomics** quantifies translated/functional proteins and their modified proteoforms[26]. Yet these layers primarily describe **what microbes could do**, not what they are doing. Metabolomics fills this gap by directly observing what microbes are doing, what they synthesize, transform, and cross-feed. In other words, **metabolomics** captures the exchanged chemical currency.

Metabolomics is arguably the most dynamic omics layer, responding rapidly to environmental, temporal, and physiological changes[27]. Its close link to the phenotype makes it powerful yet challenging, because metabolite levels are highly context dependent. Since metabolite levels reflect the current state of biochemical reactions, they can change within seconds in response to nutrient availability, stress, microbial interactions, or disease. This makes the metabolome far more sensitive to environmental and temporal variation than other omics layers, such as proteomics or transcriptomics, which respond more slowly due to regulatory and biosynthetic steps. For example, early metabolic shifts during drug-induced toxicity can appear well before any detectable changes in gene or protein expression. Thus, metabolomic measurements capture not only which pathways are present but how active they are at that specific moment. Metabolomics is primarily studied using mass spectrometry (MS) and nuclear magnetic resonance (NMR)-based platforms, with liquid chromatography-tandem mass spectrometry (LC-MS/MS), gas chromatography-mass spectrometry (GC-MS), matrix-assisted laser desorption/ionization mass spectrometry (MALDI-MS), and high-resolution NMR being the most widely applied[28,29]. Over the

past decades, improvements in sensitivity, mass accuracy, chromatographic separation, and imaging technologies have greatly expanded the number of detectable metabolites and enabled more detailed biochemical profiling. However, the interpretation of these complex and high-dimensional datasets continues to be a major challenge, especially in untargeted workflows where metabolite identification and biological context are often difficult to resolve[30–33].

Untargeted LC-MS/MS is particularly valuable for studying microbial communities because it detects thousands of molecular features and enables tracking of metabolic exchange, nutrient transformation, and chemical interactions within and between populations, even when many molecules remain structurally unknown[34–36]. However, such untargeted datasets are high-dimensional and sparse[31,37]. Thus, computational approaches are indispensable for transforming raw LC-MS/MS signals into meaningful biological insight.

1.4 Interpreting Complex LC-MS/MS Data Through Computational Approaches

1.4.1 Need for Reproducible and Scalable Computational Workflows

Modern metabolomics increasingly depends on computational infrastructure. Early analyses relied on manual spreadsheets, which were labor-intensive and irreproducible. Reproducible environments such as R and Python notebooks have transformed biological data analysis[38–40].

Untargeted LC-MS/MS generates thousands of features defined by m/z, retention time, and intensity, requiring tools for feature detection, alignment, and annotation[41] to extract meaningful molecular insights from the data. Feature-Based Molecular Networking (FBMN) on GNPS partially addresses this challenge by grouping structurally related metabolites through spectral similarity, enabling annotation propagation within molecular families and try to give information about distribution of these molecules among variable groups[42]. However, statistical interpretation of FBMN results often remains a bottleneck, particularly when moving from network structure to biological insight.

To bridge this gap, **Chapter 2 introduces FBMN-STATS**, a reproducible and transparent framework that integrates FBMN outputs with downstream statistical analysis. The workflow supports preprocessing and exploratory data analysis (EDA), including univariate, multivariate, and visualization-driven approaches that help users identify trends, clusters, outliers, and biologically relevant patterns before formal hypothesis testing. These analyses are implemented through notebooks and an interactive web app, allowing users to move seamlessly from feature networking to quantitative interpretation.

1.4.2 Statistical Interpretation of Untargeted Metabolomics: From Features to Patterns

EDA refers to the process of getting to know the dataset, visualizing it, checking its structure, detecting patterns, and identifying outliers before applying formal statistics. In untargeted metabolomics, this first requires careful preprocessing, including blank removal, normalization, and imputation, to ensure that observed patterns reflect biology rather than technical noise. Once the data are cleaned, multivariate methods such as Principal Component Analysis (PCA) help reveal global sample-level structure, clustering, and separation trends. In parallel, univariate tests enable hypothesis-driven comparisons to identify individual features that differ significantly between conditions. Best-practice guidelines also encourage evaluating effect sizes in addition to p-values, as significance alone is not always biologically meaningful[43].

Tools like MetaboAnalyst[44] provide powerful statistical capabilities, but they do not integrate seamlessly with FBMN outputs or downstream annotation tools like SIRIUS[45]. FBMN-STATS addresses this gap with a unified, reproducible environment that connects structural context with statistical patterns.

1.4.3 From Patterns to Mechanisms: Inferring Biotransformations with ChemProp2

Statistical analyses reveal which metabolites change across conditions, but not why they change. A major challenge in metabolomics is understanding how molecules are transformed or connected through biochemical processes[27,46]. In microbial systems, many abundance shifts arise from biotransformations, chemical conversions mediated by microbial enzymes, shaping processes such as drug metabolism, nutrient cycling, environment chemistry, and xenobiotic turnover[47–49].

These reactions have broad biological and ecological consequences:

- beneficial conversions, such as transforming dietary components into bioactive metabolites[50]
- harmful conversions, such as generating toxic intermediates[51]
- clinically relevant effects, including microbial inactivation or modification of therapeutic compounds[52,53]

They are also central to questions such as how gut bacteria metabolize orally administered drugs[48,54,55], how pesticides reshape soil microbial chemistry[56], and how marine ecosystems respond chemically to stressors like algal blooms or nutrient shifts[57,58].

While many methods such as PCA, PLS-DA, and differential abundance tests[59,60] can detect compositional differences between samples, and network-based methods like FBMN[42] can group structurally related features, but none provides a means to track directionality between metabolites. Predictive tools, including machine learning and deep-learning models for drug discovery and metabolite prediction[61–63] simulate biochemical outcomes but do not extract mechanistic relationships directly from experimental time-series data.

To bridge this gap, **ChemProp2** (Chapter 3) quantifies directional relationships between connected features across time. By identifying precursor-product pairs with opposing temporal trends, ChemProp2 generates mechanistic hypotheses for microbially or environmentally driven transformations. Its cascade-scoring framework further reveals multi-step pathways, shifting analysis beyond pattern recognition toward biochemical interpretation. Rather than replacing predictive models, ChemProp2 complements them as an intuitive, reproducible, and visually interpretable tool for hypothesis generation grounded in empirical data.

1.4.4 Linking Chemistry to Microbial Ecology: The Need for Multi-Omics Integration

While ChemProp2 focuses on metabolite-metabolite relationships, **biological interpretation often requires linking chemistry to community structure**. Microbial activity, ecological interactions, and environmental gradients frequently shape metabolite trajectories[64,65]. Thus, understanding the biological drivers of these chemical changes requires integrating metabolomics with complementary omics layers[66,67]. Microbiome profiling provides complementary information on taxonomic structure and ecological dynamics[67]. Correlation-based integration offers a transparent way to explore metabolite-microbe relationships.

Positive correlations may arise:

- when a microbe and metabolite show coordinated responses to the same experimental condition, reflecting a shared environmental or experimental driver rather than a direct interaction[68,69].
- They can also arise from community-level ecological succession, where both features follow similar multi-stage trajectories across the experiment instead of responding to a single factor[70,71].

- They may reflect microbial production or release of a metabolite, causing their abundance profiles to co-vary through a direct biochemical relationship[72,73].

In contrast, negative correlations may reflect consumption, detoxification, antagonism, or opposing ecological niches[65,74]. However, correlation is not causation and can arise from shared external drivers or compositional effects.

Model-driven multi-omics tools can be powerful[75,76], but their complexity often makes results hard to interpret at the feature level. A simpler and more intuitive framework is therefore valuable compared to the existing complex models. Chapter 4 introduces CorrOmics, a scalable correlation-based framework for exploring metabolite-microbe associations. By combining simple correlations with a target-decoy FDR strategy, CorrOmics provides an interpretable first layer of multi-omics integration. It bridges the gap between chemical transformation mapping (Chapter 3) and ecological interpretation of microbial communities.

1.5 Software Development and Code Availability

All three computational tools: FBMN-STATS, ChemProp2, and Corromics, are implemented as interactive Streamlit web applications[77]. Streamlit enables user-friendly, reproducible dashboards that allow complex analyses without programming expertise. The full source code for all tools is openly available on the Functional Metabolomics Lab GitHub page (https://github.com/Functional-Metabolomics-Lab/). The web applications are hosted on GNPS2:

- Chapter 2: FBMN-STATS: https://fbmn-statsguide.gnps2.org/ (v2.0.2)
- Chapter 3: ChemProp2: https://chemprop.gnps2.org/ (v1.0.1)
- Chapter 4: Corromics: https://corromics.gnps2.org/ (v1.0.0)

The thesis is structured into three research chapters that together form a cohesive workflow:

1. statistical exploration of LC-MS/MS data,
2. inference of directional chemical transformations,
3. cross-omics integration of metabolite-microbe relationships.

Chapter 2

Statistical Analysis of Feature-based Molecular Networking Results from Non-Targeted Metabolomics Data

Note: This chapter has been published as: Pakkir Shah, et al., "Statistical analysis of feature-based molecular networking results from non-targeted metabolomics data," **Nature Protocols**, 2024.

Abzer K. Pakkir Shah[1,2], Axel Walter[1,2,3], Filip Ottosson[4], Francesco Russo[4], Marcelo Navarro-Diaz[2], Judith Boldt[1,5,6], Jarmo-Charles J. Kalinski[1,7], Eftychia Eva Kontou[1,8], James Elofson[9], Alexandros Polyzois[1,10], Carolina González-Marín[1,11], Shane Farrell[12,13], Marie R. Aggerbeck[1,14], Thapanee Pruksatrakul[1,15], Nathan Chan[16], Yunshu Wang[16], Magdalena Pöchhacker[1,17], Corinna Brungs[18], Beatriz Cámara[19], Andrés Mauricio Caraballo-Rodríguez[20], Andres Cumsille[19], Fernanda de Oliveira[21,20], Kai Dührkop[22], Yasin El Abiead[20], Christian Geibel[2], Lana G. Graves[23,24], Martin Hansen[14], Steffen Heuckeroth[25], Simon Knoblauch[2], Anastasiia Kostenko[9], Mirte C. M. Kuijpers[26], Kevin Mildau[1,27,28], Stilianos Papadopoulos Lambidis[2], Paulo Wender Portal Gomes[20], Tilman Schramm[2,29], Karoline Steuer-Lodd[2,29], Paolo Stincone[2], Sibgha Tayyab[2], Giovanni Andrea Vitale[2], Berenike C. Wagner[2], Shipei Xing[20], Marquis T. Yazzie[9], Simone Zuffa[20,30], Martinus de Kruijff[31], Christine Beemelmanns[31,32], Hannes Link[2], Christoph Mayer[2], Justin J.J. van der Hooft[1,28,33], Tito Damiani[18], Tomáš Pluskal[18], Pieter Dorrestein[20], Jan Stanstrup[34], Robin Schmid[1,18], Mingxun Wang[1,16], Allegra Aron[1,9], Madeleine Ernst[4,#], Daniel Petras[1,2,29,#]

1. Virtual Multi-Omics Laboratory, The Internet
2. University of Tuebingen, Interfaculty Institute of Microbiology and Infection Medicine, Tübingen 72076, Germany
3. Applied Bioinformatics, Department of Computer Science, University of Tübingen, Tübingen, Germany
4. Section for Clinical Mass Spectrometry, Danish Center for Neonatal Screening, Department of Congenital Disorders, Statens Serum Institut, Artillerivej 5, DK-2300 Copenhagen S, Denmark
5. Leibniz Institute DSMZ - German Collection of Microorganisms and Cell Cultures, Braunschweig, Germany
6. German Center for Infection Research (DZIF), Partner Site Braunschweig-Hannover, Braunschweig, Germany
7. Department of Biochemistry and Microbiology, Rhodes University, 6140, Makhanda, South Africa

8. The Novo Nordisk Foundation for Biosustainability, Technical University of Denmark, Kemitorvet 220, 2800 Kongens Lyngby, Denmark
9. Department of Chemistry and Biochemistry, University of Denver, Colorado 80210, United States
10. Boyce Thompson Institute and Department of Chemistry and Chemical Biology, Cornell University, Ithaca, New York 14853, United States
11. Universidad EAFIT, Carrera 49, Cl. 7 Sur #50, Medellín, Antioquia, Colombia
12. Bigelow Laboratory for Ocean Sciences, East Boothbay, ME 04544, United States
13. School of Marine Sciences, Darling Marine Center, University of Maine, Walpole, ME 04573, United States
14. Department of Environmental Science, Aarhus University, Frederiksborgvej 399, 4000 Roskilde, Denmark
15. National Center for Genetic Engineering and Biotechnology (BIOTEC), National Science and Technology Development Agency (NSTDA), Thailand Science Park, Pathum Thani, 12120, Thailand
16. University of California Riverside, Department of Computer Science, 900 University Ave, Riverside CA, United States
17. Department of Food Chemistry and Toxicology, University of Vienna, Vienna, Austria
18. Institute of Organic Chemistry and Biochemistry of the Czech Academy of Sciences, Prague, Czech Republic
19. Laboratorio de Microbiología Molecular y Biotecnología Ambiental, Centro de Biotecnología DAL, Universidad Técnica Federico Santa María, Valparaíso, Chile
20. Skaggs School of Pharmacy and Pharmaceutical Sciences, University of California San Diego, 9500 Gilman Dr., San Diego, CA, 92093, United States
21. Department of Biotechnology, Engineering School of Lorena, University of São Paulo, Lorena, São Paulo 12602-810, Brazil
22. University of Jena, Department of Bioinformatics, Jena, Germany
23. University of Tuebingen, Department of Environmental Systems Analysis, Tübingen 72076, Germany
24. Leibniz Institute of Freshwater Ecology and Inland Fisheries (IGB), Berlin, Germany
25. Institute of Inorganic and Analytical Chemistry, University of Münster, Münster, Germany
26. Department of Ecology, Behavior and Evolution, University of California San Diego, 9500 Gilman Dr., San Diego, CA, 92093, United States
27. Department of Analytical Chemistry, University of Vienna, Vienna, Austria
28. Bioinformatics Group, Wageningen University & Research, 6708 PB Wageningen, the Netherlands
29. University of California Riverside, Department of Biochemistry, Riverside, CA, United States
30. Collaborative Mass Spectrometry Innovation Center, Skaggs School of Pharmacy and Pharmaceutical Sciences, University of California San Diego, 9500 Gilman Dr., San Diego, CA, 92093, United States
31. Helmholtz Institute for Pharmaceutical Research Saarland (HIPS), Helmholtz Centre for Infection Research (HZI), Campus E8, 66123 Saarbrücken, Germany
32. Saarland University, Saarbrücken 66123, Germany

33. Department of Biochemistry, University of Johannesburg, Johannesburg 2006, South Africa
34. Department of Nutrition, Exercise and Sports, University of Copenhagen, 1958 Frederiksberg C, Denmark

Correspondence should be addressed to Madeleine Ernst or Daniel Petras

2.1 Abstract

Feature-Based Molecular Networking (FBMN) is a popular analysis approach for LC-MS/MS-based non-targeted metabolomics data. While processing LC-MS/MS data through FBMN is fairly streamlined, downstream data handling and statistical interrogation are often a key bottleneck. Especially users new to statistical analysis struggle to effectively handle and analyze complex data matrices. In this protocol, we provide a comprehensive guide for the statistical analysis of FBMN results, focusing on the downstream analysis of the FBMN output table. We explain the data structure and principles of data clean-up and normalization, as well as uni- and multivariate statistical analysis of FBMN results. We provide explanations and code in two scripting languages (R and Python) as well as the QIIME2 framework for all protocol steps, from data clean-up to statistical analysis. All code is shared in the form of Jupyter Notebooks (https://github.com/Functional-Metabolomics-Lab/FBMN-STATS). Additionally, the protocol is accompanied by a web application with a graphical user interface (https://fbmn-statsguide.gnps2.org/), to lower the barrier of entry for new users and for educational purposes. Finally, we also show users how to integrate their statistical results into the molecular network using the Cytoscape visualization tool. Throughout the protocol, we have used a previously published environmental metabolomics dataset for demonstration purposes. Together, the protocol, code, and web application provide a complete guide and toolbox for FBMN data integration, clean-up, and advanced statistical analysis, enabling new users to uncover molecular insights from their non-targeted metabolomics data. Our protocol is tailored for the seamless analysis of FBMN results from Global Natural Products Social Molecular Networking (GNPS and GNPS2) and can be easily adapted to other MS feature detection, annotation, and networking tools.

2.2 Introduction

The field of metabolomics aims to characterize and quantify the detectable spectrum of small molecules in order to catalog and understand the metabolic dynamics within biological systems[46,78]. Phenotypes or environmental factors that distinguish samples within a given set are

often reflected in the chemical profiles of such small molecules across samples. Therefore, the characterization of chemical distinctions and gradients between samples provides crucial information for describing and understanding molecular mechanisms[79,80]. Metabolomics studies usually employ targeted or non-targeted approaches[46]. Targeted metabolomics is typically hypothesis-driven and aims to quantify known metabolites, often using internal standards and experimental methodology optimized for the study. Non-targeted metabolomics, on the other hand, aims to detect a maximum number of metabolites in order to comprehensively characterize the chemical profiles within a given sample set.

To uncover molecular insights from non-targeted liquid chromatography tandem mass spectrometry (LC-MS/MS) data, several software tools are available that assist with mining and annotating metabolites, including feature detection and annotation tools[41]. FBMN is a popular analysis approach that integrates feature-detection tools with molecular networking for metabolite annotation and annotation propagation[42] in the GNPS cloud ecosystem[81]. FBMN is routinely applied in many biological disciplines, including clinical studies[82,83], plant[84–86] and environmental science[32,87–89], as well as microbiome investigations[69,90,91] , and the functional analysis of natural products[92–94]. While platforms such as GNPS have improved the way that we identify and characterize metabolites, the statistical analysis of non-targeted metabolomics data remains a challenge for many researchers. Resources like MetaboAnalyst[44,95] provide powerful solutions for the statistical analysis of metabolomics data, but lack shareability and transparency as well as straightforward means of integration with the complex multi-layer information from FBMN results and other downstream annotation tools (e.g., SIRIUS). Most tools and analysis approaches that can be used to achieve this are typically custom scripts or different software tools that are scattered across different platforms. This makes it especially difficult for users new to the field to effectively manage and analyze their data. Moreover, while there are several tools available for individual clean-up and analysis steps (see alternative approaches section), there is a lack of comprehensive, user-friendly guidance that covers the entire spectrum of data preparation and statistical analysis of FBMN results.

In this protocol, we provide a detailed guide that starts with FBMN results, promoting an end-to-end pipeline from feature detection, subsequent data clean-up, and statistical analysis to spectrum annotation. This step-by-step guide is complemented with ready-made code for the popular statistical scripting platforms R and Python, the QIIME2 (Quantitative Insights Into Microbial Ecology 2) toolkit (https://github.com/Functional-Metabolomics-Lab/FBMN-STATS), as well as a web application with a graphical user interface (GUI) designed to simplify the process

(https://fbmn-statsguide.gnps2.org/) as well as a downloadable GUI application (https://www.functional-metabolomics.com/resources) for new users and educational purposes. The protocol can be seamlessly integrated with FBMN results from GNPS (https://gnps.ucsd.edu) and GNPS2 (https://gnps2.org).

2.2.1 FBMN from LC-MS/MS Data

Liquid chromatography-mass spectrometry (LC-MS) is one of the most prominent metabolomics techniques, with applications in numerous research fields[31,96–98]. Specifically, LC coupled with tandem mass spectrometry (LC-MS/MS) has been widely used because it provides a broad coverage of chemical space allowing for the simultaneous semi-quantitative detection and qualitative annotation of many metabolites, including both organic and inorganic compounds, over a wide dynamic range[30,32,99–101].

In addition to providing the molecular mass, retention time, and isotopic pattern of a metabolite, MS/MS provides structural information about the detected species. This is achieved through the fragmentation of precursor ions into product ions and the measurement of their mass-to-charge ratios (*m/z*) and abundances. This can be conducted using Data-Dependent Acquisition (DDA) or Data-Independent Acquisition (DIA) methods. DDA involves selecting ions observed in MS1 survey scans for further fragmentation in subsequent MS/MS scans (See **Figure 1**). It operates by selecting the “top N” peaks in each duty cycle, where “N” is a user-defined number. These peaks are chosen based on their intensity in a narrow isolation window and other user-defined criteria through an automated process[88]. Conversely, the DIA method fragments all ions within a predefined larger and sequential isolation window during each scan, directing the fragmented ions to a high-resolution mass spectrometry analyzer. Consequently, the MS/MS spectra show high complexity that demands the use of advanced processing and annotation tools, that have only recently become available to the scientific community. To date, most of the non-targeted metabolomics studies are performed in DDA mode due to its generation of simpler data and more straightforward analysis procedures compared to DIA[88].

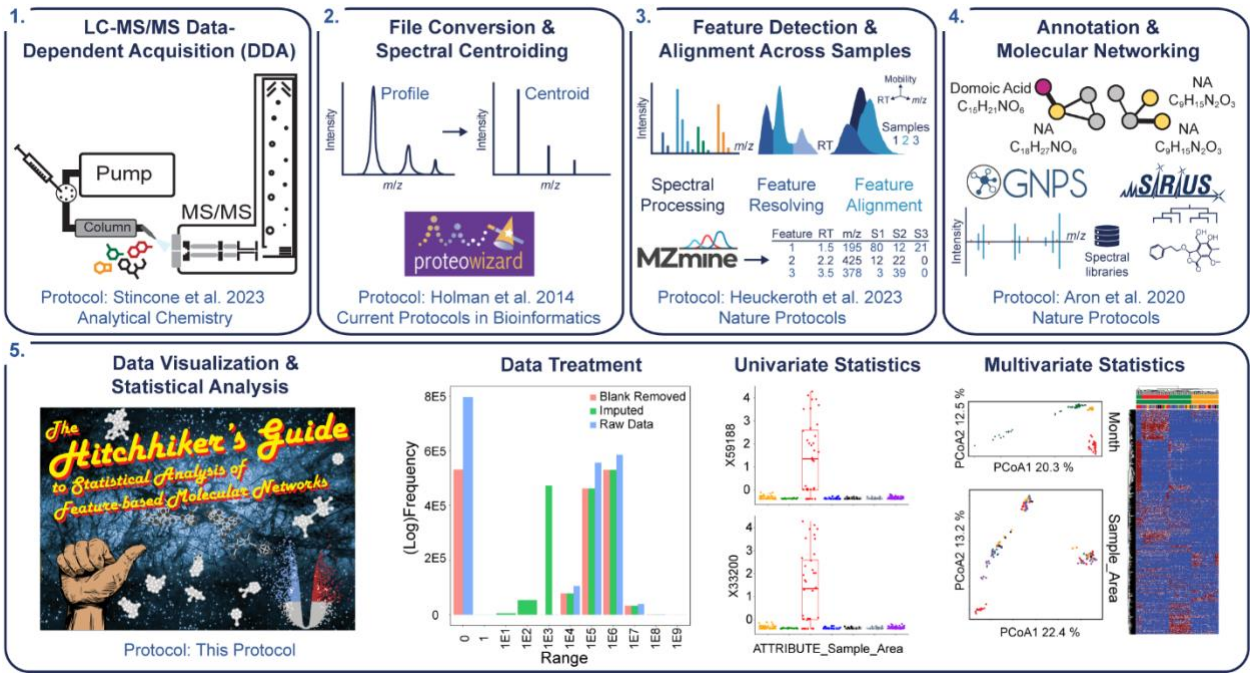


Figure 1: Flowchart of LC-MS/MS-based metabolomics experiment. 1. Data-dependent acquisition of MS/MS spectra. 2. Centroiding and file conversion. 3. Feature detection. 4. Feature annotation, network propagation, and clustering. 5. Data clean-up, statistical analysis, and visualization (blank removal, imputation, normalization, and scaling, followed by data visualization and statistical analysis.)

The resulting MS/MS spectra of product ions from either method can be used in several ways to indicate a candidate structure: 1) via spectral matching against spectral libraries of experimental reference spectra or *in silico* generated spectra[102,103]; 2) via machine learning-based structural predictions using experimental MS/MS-generated molecular fingerprints against structural databases[104,105]; 3) and via *de novo* structure prediction using molecular structure fingerprint prediction combined with neural networks[106].

Non-targeted LC-MS/MS metabolomics is a powerful and versatile research approach that enables high-throughput analysis and simultaneous detection of many small molecules, making it an excellent method for gaining insights into biological systems (For more information on Experimental Design and LC-MS/MS Data Acquisition, refer to **Box 1**). However, mining the vast amount of data created by non-targeted metabolomics experiments remains a challenging task despite a range of available resources that guide in the qualitative and quantitative aspects on non-targeted metabolomics. Qualitative data exploration has been democratized by platforms such as GNPS[36], by providing MS reference libraries, data analysis workflows, and compute resources for the community. Molecular networking (MN) is GNPS' core concept and is based on the comparison of all MS/MS spectra within a dataset by modification-aware similarity metrics,

which network features by their similar fragmentation patterns that are often reflective of structural similarity. FBMN[42] and Ion Identity Molecular Networking (IIMN)[107] add feature detection, improving the (semi) quantitative quality within MN results. FBMN builds upon the classical MN by harnessing both MS1 information, such as isotope patterns and retention time, and ion mobility separation when used. FBMN can distinguish isomers with similar MS/MS spectra, which might remain obscured in classical MN, through chromatographic or ion mobility separation.

IIMN enhances MS/MS-based spectral networks by adding connectivity based on the MS1 feature shape correlation. It efficiently tackles the issue of unconnected ion adducts in Molecular Networking by connecting ions from the same molecules into groups called ion identity networks (IIN). This helps remove redundancy in MS-based metabolomics.

Box 1 - Experimental Design for LC-MS/MS Data Acquisition

To obtain high-quality and representative LC-MS/MS data, proper planning of the sampling and mass spectrometry analysis is essential. While this protocol article focuses on data analysis, the integrity and reliability of the results are significantly influenced by meticulous sample handling to avoid introducing bias. No single platform is capable of encompassing the entire metabolome. Therefore, the selection of an analytical approach should align with the specific research objectives and the sample characteristics[162]. We strongly emphasize the importance of rigorous sample preparation for achieving reliable and robust outcomes from the FBMN. For researchers new to these types of experiments, we advise seeking guidance from a statistician and analytical chemists to guarantee optimal experimental design, instrument performance (e.g., system suitability tests), and analysis pipeline. Before proceeding with further processing and statistical analysis, raw LC-MS/MS data should be manually inspected by the user[163–165]. Below, we provide a short checklist for guidance:

- **Experimental design and power calculation** are crucial when determining the suitable number of samples and replicates. In non-targeted metabolomics experiments, it is often challenging to predict certain values, like the feature **coefficient of variation** and the **expected effect size**, which are crucial for estimating the required sample size and the power of the study. To navigate this, reviewing previous studies with similar biological systems and research questions can provide an approximate estimation of these values. As a general rule of thumb, when the effect size is smaller, one might need more samples or replicates.
- **Replicates (technical and biological)** to measure instrument and biological variance.

- **System and Process Blanks** to identify and correct for contamination that may be introduced during the sample collection, preparation, or measurement process. Some common blank samples include[166] solvent and extraction blanks. A **solvent blank** consists of only the solvent used to dissolve the sample and is used to identify the contaminants present in the solvent. Adding this blank periodically in an analytical run reduces carryover. Blanks should be added into the same well plates or vials to cover similar contaminations. An **extraction blank** is prepared by adding a known volume of solvent to a blank matrix such as water and extracting it the same way as a sample. This extracted blank is measured along with the real sample to find the contaminants introduced during the extraction process.
- **Control samples,** e.g., negative and positive controls. Depending on the experimental design, control samples are essential and should be included in the same number as the treatment samples.
- **Quality Control (QC) samples** are needed to measure instrument performance. These can be in the form of pooled QC (for example, a combination of aliquots from each sample) or standard mixtures (such as a combination of reference standard chemical compounds or isotopically labeled compounds that can also serve as internal standards). These mixtures should span a broad chemical spectrum and cover a wide retention time range.
- **Randomization of sample injection order**. It is suggested to randomize the injection order throughout the samples. However, we recommend injecting blanks at the start of the queue to prevent carry-over, which could lead to the removal of actual features from the samples during the blank removal step. Depending on the experimental design, it might also be useful to select certain sample types with the injection order, e.g., KO (knockout) vs. WT (wild type) mutant strains or low vs. high biomass samples, to avoid carryover between them.
- **Managing Batch effects**: Batch effects are systematic differences in sample measurements when samples are run as multiple batches or groups. In most cases, when the sample sizes exceed the measurement assay, it is often necessary to measure the samples in multiple batches. This might lead to varying mass spectra among the samples within a batch and among different batches[167]. Several factors could contribute to these batch effects such as variability in instrument conditions, RT shifts, and gradual contamination of LC columns when measuring multiple samples over a long period. These are often unavoidable issues, hence it is common to treat these effects post-sample

measurement[168]. To correct these unwanted variations, we first need to identify their presence, remove or adjust the variations for further statistical analysis, and assess the performance of our method[169]. See the section on Batch correction for more details.

- **Internal Standards (IS)** can be added to every sample to track instrument performance, and if desired, quantify predefined metabolites. If no internal standards are available, "housekeeping features" such as ubiquitous contaminants or metabolites can be used to control for mass and retention time drift.

2.2.2 FBMN Workflow

This protocol addresses the downstream analysis of FBMN results from non-targeted metabolomics data. This section offers a structured approach leading up to FBMN and prior to our statistical workflow. As highlighted in **Figure 1**, the non-targeted LC-MS/MS analysis workflows can be split into five main stages:

1. Data acquisition;
2. File conversion with centroiding of the raw data;
3. Feature detection and an optional ion identity networking;
4. Feature annotation, which encompasses spectral library matching, *in silico* spectrum annotation, annotation propagation, and spectral clustering methods like spectral networking;
5. Data refinement, visualization and statistical analysis.

Stages 1 to 4 result in a comprehensive feature table, capturing all detected features (*m/z*, RT, and MS/MS spectra) alongside quantitative information (e.g., peak area) for each sample, including the feature annotations. The focus and main work of this article is predominantly on the fifth stage, involving the refinement of the feature-sample matrix through cleanup steps such as blank removal, imputation, normalization, scaling, and ultimately, data visualization and statistical analysis. Accordingly, the emphasis is on guiding users through stage 5, but it is crucial to be aware of how the preceding processing can influence results. We encourage users to review the protocols outlined in **Figure 1** for a comprehensive understanding of the processing stages taken to produce the feature table used in our protocol. We will also briefly discuss each stage, as this information serves as a foundational starting point for the remainder of the protocol.

1. Data acquisition

Data acquisition very much depends on the experiment. Some guidelines on good experimental design can be found in **Box 1**.

2. File Conversion

Raw data acquisition in MS instruments involves first generating spectra in profile mode, also called continuous mode typically resulting in a Gaussian shape. In high-resolution instruments, each chemical entity is represented by signals of *m/z* values within a 5-20 ppm window, depending on the instrument's resolution. Researchers with access to vendor-specific software are encouraged to examine the raw data directly from these platforms to assess data quality. To simplify data for downstream analysis, a process called centroiding or "peak-picking", is employed: This process converts each Gaussian peak in the m/z dimension into a singular peak, thereby reducing data complexity. Centroiding can be performed on-the-fly during data acquisition, where the profile information is lost, or post-acquisition through file conversion tools like the Proteowizard's msconvert[108] or the Thermo dedicated ThermoRawFileParser[109]. For novices, msconvert offers an accessible starting point, supporting binary files from various vendors with an intuitive GUI. When selecting the "vendor" option for centroiding in msconvert, the algorithm provided by the vendor of the instrument is applied[110]. Nonetheless, vendor-specific tools, such as ThermoRawFileParser, may offer more precise algorithms than msconvert's default, making them preferable for accuracy (refer to **Figure 1.2**) as they can include the removal of flagged, e.g., noise or internal calibrant, signals[109]. Note that the conversion of vendor-specific formats into universally accessible, open formats like 'mzML' is often needed for compatibility with most feature detection software and to make data accessible without vendor-specific tools.

3. Feature Detection

The process of converting raw data from the preceding stage into a table of putative metabolic features, along with their relative abundances per sample, involves a pre-processing workflow that uses a series of algorithms (See **Figure 1.3**). The resulting table as shown in **Table 1** is referred to as a '**feature quantification table**'. Open-source tools such as the R package XCMS[111] (often used with the package CAMERA[112] for feature grouping), MZmine 3[113], MS-DIAL[114], or OpenMS[115], in addition to vendor-specific tools can be utilized for this purpose. For advanced optimization and fine-tuning, multiple tools such as NeatMS[116], MetaClean[117], and mzRAPP[118] exist to assess feature quality.

For the present protocol, we chose MZmine 3 as a feature detection package due to the following reasons. First, it provides an interactive and user-friendly GUI that can assist researchers without programming skills. Second, harmonized data exchange formats enable its tight integration with downstream annotation tools such as GNPS and SIRIUS. Third, MZmine offers a tool (called *Processing wizard*) for the automatic generation of data-processing workflows that can be used

by new users as a starting point for parameter optimization[119]. For detailed descriptions and recommendations for processing parameter optimization, we direct the reader to the recently published MZmine 3 protocol[119] and the software's online documentation (https://mzmine.github.io/mzmine_documentation/index.html, https://mzmine.github.io/). Here, we want to emphasize that suboptimal parameters can dramatically impact the quality of the resulting feature quantification table and impair all downstream steps. For instance, a feature table with more than 20,000 features might significantly prolong or even cause the FBMN process to fail. Therefore, during the feature detection phase, distinguishing features from noise is crucial, such as ensuring the features fall above the noise threshold. This necessitates particular attention, especially from new users at this stage.

Although the present protocol uses MZmine 3 as feature detection software, users of alternative platforms such as MS-DIAL, XCMS, and OpenMS can also integrate their data using their FBMN task ID. FBMN accommodates input tables from these platforms, standardizing them for the workflow. Utilizing an FBMN ID ensures the feature table automatically conforms to our protocol's requirements. For those inputting data manually, we advise referencing our example dataset to understand necessary adjustments, thereby allowing for tailored adaptation of their data to meet protocol requirements.

Table 1. **Example of a Feature Quantification Table as generated by MZmine.**

The table illustrates how a typical quantification output from MZmine appears. Adducts and charges are assigned by running specific modules (de-isotoper and ion identity networking modules) in MZmine[107,113]

		m/z	RT (min)	Adduct	Charge	**Sample 1**	**Sample 2**	**…**	**Sample N**
Feature 1		97.1082	4.6	$[M+H]^+$	+1	2.08e07	9.47e06	…	3.27e08
Feature 2		518.3032	2.0	$[M+H]^{+2}$	+2	1.88e07	5.56e05	…	2.11e06
⋮		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Feature K		83.1017	1.6	$[M+Na]^+$	+1	4.77e04	8.13e03	…	5.17e09

4. Feature Annotation, Spectral Networking and Annotation Propagation

The Metabolomics Standard Initiative (MSI) outlines four levels of metabolite annotation and structural identification through mass spectrometry to guide researchers in differentiating the level of identification rigor for the reported metabolites[120]. Level 1 annotations require fully characterized compounds such as authentic standards, analyzed under identical conditions as the experimental samples. To ensure accurate annotation confidence at level 1 for a feature, it is necessary to match two or more independent orthogonal data dimensions to those of an analytical standard. These dimensions typically include precursor *m/z* (MS1), RT, and MS/MS fragmentation patterns and significant differences in any of the available data dimensions preclude level 1 annotation. Generally, a feature is only described as "identified" when a level 1 annotation was achieved while for those with lower confidence, the term "annotation" is preferred.

In the context of tandem mass spectrometry, level 2 confidence is assigned when data are confidently matched to reference MS/MS spectra (precursor and fragment *m/z* values)[120], as facilitated through e.g., FBMN on GNPS. The assignment of level 2 annotations usually requires manual curation of spectral matches, since spectral matching scores are not fool-proof and especially scores between spectra of different compounds from classes with high isomerism and structural analogism, such as steroids and other terpenoids, can produce highly similar MS/MS spectra. Moreover, spectral libraries contain entries with insufficient fragmentation to allow level 2 annotation, such as matching based on low-intensity fragments and the residual precursor ion. Matching to such entries can be avoided by tuning parametrization of the FBMN workflow, such as the minimum number of matched fragments and noise level filters, but more stringent requirements may lead to losses of acceptable matches.

Level 3 annotations are given to features for which chemical classes can be inferred through molecular network connectivity with other annotated features, analog spectral matching to reference MS/MS spectra (rational *m/z* deviation, high similarity score), or *in silico* prediction tools such as SIRIUS[45,105], CSI:FingerID[104] or CANOPUS[106,121]. The results of these approaches can be contradictory for the same feature and manual curation is required to maintain good confidence. Finally, level 4 refers to unknown features that can be consistently detected (e.g., defined *m/z* value, RT, and MS/MS spectrum), but not annotated to any of the higher levels.

Feature annotation is essential in mass spectrometry-based metabolomics studies, especially to understand the biological significance of the detected features. Feature annotation entails several approaches, including database searches, spectral matching, and *in silico* annotation strategies. *In silico* annotation tools, such as SIRIUS, MS2Query, Network Annotation Propagation (NAP),

Dereplicator, and Dereplicator+, predict metabolite identities based on spectral similarities[45] and can only lead to MSI levels 2, 3 or 4.

Another popular method that combines feature annotation through spectral similarity scoring with visualization is molecular networking (MN), as shown in **Figure 1.4**. MN elucidates the structural relationships between metabolites based on similarities in their fragmentation patterns, highlighting potential biological pathways and processes. The utility of MN spans across various fields, such as natural products[122], agriculture[123] , and clinical microbiology[124]. Using the MS-Cluster algorithm on the GNPS (https://gnps.ucsd.edu), and GNPS2 (https://gnps2.org) web platforms, MN generates a molecular network by calculating spectral similarities between each MS/MS spectra pair and provides structural annotations of varying reliability[81] by scoring spectral similarity against public spectral database entries The .mzML or .mzXML spectra files can be analyzed directly on GNPS using the classical MN workflow[36] which is based solely on the comparison of MS2 spectra in a dataset without any inflow of chromatographic information from the MS1 dimension.

For more precise quantitative insights, FBMN has emerged as a significant advancement by incorporating MS1 peak intensities, retention times, isotope patterns, and when applicable, ion mobility separation. Consequently, FBMN distinguishes between isomers with near-identical fragmentation spectra, but different retention times[42] and allows feature peak areas to feed into the data exploration. FBMN on the GNPS web platform conveniently accepts the output of feature detection and alignment tools such as Mzmine[113,125] (see 'Feature Detection' above), MS-DIAL[114], and XCMS[111] that assemble feature lists and associate feature fragmentation spectra from raw MS/MS data.

5. Data Refinement, Visualization and Statistical Analysis

The feature quantification table (see **section 'Feature Detection'** and **Table 1**), contains a list of features, such as *m/z* and RT values, as well as their relative abundances per sample. This table represents the basic dataframe for statistical analyses, which can help reveal distribution patterns between sample types and determine which features are responsible for distinguishing between them. The challenge lies in prioritizing the important features in a large dataset, considering chemical and biological relevance, as well as statistical significance. In part, chemical and biological relevance can be understood through the FBMN workflow and our protocol offers a toolset to augment an FBMN analysis with information on the statistical significance of feature distribution and sample compositions.

To refine the feature quantification table, the protocol involves cleanup steps such as blank removal, imputation, normalization, and scaling. After data refinement, we apply multivariate and univariate statistical analyses for a deeper exploration of samples. The multivariate techniques showcased in our workflow include Principal Coordinates Analysis (PCoA), Hierarchical Clustering Analysis (HCA), heatmaps, and Random Forest. The univariate techniques involve parametric tests such as Analysis of Variance (ANOVA) and t-tests, as well as non-parametric counterparts like the Kruskal-Wallis test.

PCoA

Initially, dimensionality reduction techniques like Principal Component Analysis (PCA) and PCoA are employed for data exploration and visualization. These unsupervised techniques generate 2- or 3-dimensional plots grouping similar samples, despite starting with a high number of dimensions (up to thousands of features). The key insight is that only a few top dimensions are needed to visualize the most critical data aspects, effectively reducing complexity. This can give valuable impressions on the presence of sample clusters and how they relate to metadata categories[126].

PCoA is a popular ordination technique used alongside PCA to visualize sample similarities by calculating distance matrices between samples. PCoA groups samples based on their dissimilarity or distances whereas PCA focuses on their correlation or covariance[127]. The process begins by computing a dissimilarity matrix to capture the sample differences. This matrix is then transformed using multidimensional scaling (MDS) to produce a new set of points called Principal Coordinates (PCos) in a lower-dimensional space. The distance between samples in these coordinates reflects the original sample differences[128]. It is important to mention that MDS can be categorized into metric MDS (as in PCoA) and non-metric MDS[127]. In this protocol, we focus solely on metric MDS.

Coupled with Permutational Multivariate ANOVA (PERMANOVA), these techniques enable a comprehensive exploration of sample similarity by calculating correlations or distance matrices. Although not demonstrated in our workflow, t-Distributed Stochastic Neighbor Embedding (t-SNE) is another widely used technique for dimension reduction, recognized for its effectiveness in managing complex datasets[129]. For advanced feature extraction in high-dimensional data, tools such as Knowledge Discovery by Accuracy Maximization (KODAMA) can be considered. It functions as an unsupervised learning algorithm to perform feature extraction from noisy and high-dimensional data[130].

PERMANOVA

In multivariate analysis like PCA, it is crucial to measure confidence in observed relationships or separation between objects. This is often achieved via statistical significance tests, which provide a p-value as a measure of the confidence level. For ordination techniques that do not assume a specific data distribution, parametric statistical testing is not applicable[131]. In such cases, resampling methods such as bootstrap, jackknife[132], and permutation tests[133] are used to assess the statistical confidence of the results. These methods generate multiple samples or permutations from original data to estimate variability and assess the significance of observed relationships[139].

Alternatively, non-parametric methods such as PERMANOVA can be used[134]. PERMANOVA allows for multivariate ANOVA and tests for differences between object classes. It enables any dissimilarity metric and calculates a test statistic by comparing the dissimilarities between objects within and between classes. Here, the p-values are determined through permutation[131].

HCA and Heatmaps

Another unsupervised approach commonly used in metabolomics is the use of hierarchical clustering to group samples with similar relative abundance profiles of features. Unlike PCA, which focuses on capturing the maximum variance between samples, clustering aims to group samples with “similar” profiles. The results are often visualized as dendrograms[128].

The results of such analyses are often visualized as dendrograms in combination with a heatmap. This combination is ideal for hypothesis generation by providing an initial data overview. The heatmap displays the feature distribution across samples in a two-dimensional plot through cells that are colored according to the relative abundances of features in samples, with feature and/or sample rows and columns grouped according to their similarity profiles. A dendrogram is drawn beside the heatmap to further illustrate the hierarchical relationship between the samples and features.

Compound class ontologies such as ClassyFire[135] or NPC[136] categorize compounds based on shared structural features or biosynthetic origins and serve as high-level annotations of the data. CANOPUS[121] predicts these compound classes from tandem mass spectrometry data without searching in structure databases. Analyzing the distribution and variety of compound classes, along with their up- and down-regulation using these data visualization methods, can yield biological insights that may not be attainable when solely considering *m/z* and retention time values.

However, caution is advised with unsupervised methods; it's essential to critically assess the clusters and dimensions produced. For instance, heatmaps can help validate clusters by revealing inconsistencies or alignments among samples and features. While users typically select the number of clusters or dimensions for analysis, heuristic methods exist to suggest optimal numbers, though they are not definitive[137].

In addition to that, traditional cluster heatmaps also have limitations. Their data representation in two-dimensional format can be restrictive when processing complex multidimensional data. Furthermore, their static nature does not allow for data to be sorted along different axes, filtered, or focused on specific elements, making the representation of a vast number of elements quite challenging. Regardless of these limitations, heatmaps are preferred in biological and biomedical data representation because their visual format simplifies data interpretation and comparison. To overcome these limitations, more advanced versions such as XCMS interactive heatmaps are available that offer a more versatile and dynamic data visualization experience[138]. For further details on applying these unsupervised and supervised methods, including their advantages and limitations, refer to the respective sections under 'Procedure'.

Supervised Classification: RF

We use Random Forest (RF) as a key supervised classification technique in this protocol. While previously discussed unsupervised methods allows for the discovery of groups or trends in the data without prior assumptions about predetermined labels or categories, supervised analysis uses labeled data to guide the analysis toward specific objectives such as biomarker discovery, classification, and prediction. In supervised analysis, the algorithm is trained on labeled data to predict the response variable (or dependent variable) based on the predictor variables (or independent variables)[139].

Supervised learning is categorized into classification and regression problems based on the type of response variable: classification for categorical or discrete variables (e.g., cancer vs non-cancer samples), and regression for continuous variables. Popular supervised models in metabolomics include logistic regression, partial least square discriminant analysis (PLS-DA), support vector machines (SVM), k-nearest neighbor (KNN), and RF. Here, we focused on RF due to its advantages such as the low risk of overfitting, ease of implementation, interpretability, and minimal hyperparameter tuning requirements[140].

However, to maintain simplicity, we are not discussing the hyperparameter tuning in the main protocol. For advanced users interested in exploring beyond RF, we offer instructions on using the gradient boosting method via XGBoost which includes tuning various hyperparameters.

XGBoost employs gradient boosting to sequentially correct errors in a series of decision trees, offering a different approach to model enhancement. These additional instructions are provided in a separate Jupyter Notebook (link) in our GitHub Repository. Additionally, we would like to point to PLS-DA, another supervised multivariate technique that is frequently used in metabolomics studies simply due to the availability of the model in several software packages and ease of use with default settings. It handles collinear and noisy data well and offers comprehensive results such as classification prediction accuracy, scores, and loadings plots. Yet, its prediction accuracy may lag behind methods like RF, especially with datasets handling fewer features. Therefore, PLS-DA might not be suitable for those who want to significantly reduce the feature numbers and then use the model on them[141]. While we do not dismiss the utility of PLS-DA, we suggest considering alternative models as well. For a comprehensive comparison of different machine learning-based classification tools, we recommend the study of Mendez *et al*.[142] in which they evaluate eight machine learning algorithms across ten clinical metabolomics datasets for binary classification[142].

Univariate Statistics

While multivariate analyses offer a comprehensive overview of the data, univariate statistical analyses allow us to focus on specific attributes. Primarily, univariate analysis in metabolomics helps identify individual metabolites that significantly differ between experimental groups, potentially serving as biomarkers for certain conditions or indicators of specific biological processes. It can also reveal impacts on specific metabolic pathways if related metabolites change significantly. However, it is worth noting that univariate analysis does not account for metabolite correlations and interactions, hence, it is best used in conjunction with multivariate analysis for a holistic data interpretation.

For example, our test dataset consists of numerous features collected at seven diverse sample sites. Here, univariate analyses can assess feature differences across these sites. In the case of two site comparisons, the t-test can be used to examine significant feature differences (p-value < 0.05). For a comparison involving more than two sample groups, we can use ANOVA. In the event of significant differences, we represent these findings through a bar graph that captures the distribution of a ‘significant’ feature across sample conditions. Post-hoc tests are also introduced as supplementary tools to identify which groups’ average values significantly differ.

When conducting multiple univariate tests simultaneously, as is common in metabolomics, there is an increased risk of false positives. To manage this, the False Discovery Rate (FDR) gauges

the expected false positives among significant results. While the classical Bonferroni correction addresses false positives, it could increase the false negative rate. The following are some advanced methods that focus on maximizing true discoveries without escalating the false positive error rate[143]:

- **Benjamini-Hochberg (BH):** Commonly used in metabolomics for being less conservative than Bonferroni. It ranks p-values and adjusts them, targeting the expected false positives among all positives, rather than across all tests. It calculates FDR as Expected (False Positive/ (False Positive + True Positive)).
- **Benjamini-Yekutieli (BY):** An iteration of BH that is suitable when tests have dependencies.
- **Storey's q-value:** This approach estimates the proportion of true null hypotheses (i.e., no effect) among all hypotheses and then computes a q-value for each test, which is the FDR analog to the p-value[144].

In metabolomics, it is crucial to apply FDR correction methods to univariate results to ensure that the identified significant metabolites are not just statistical artifacts but potentially biologically relevant. However, it is important to note that differences observed at this stage are statistical rather than definitive biological distinctions. Follow-up experiments are necessary to confirm these as genuine biological differences. In all our univariate tests, we apply the BH metric to our p-values, aiming to robustly identify statistical differences in the data.

Testing for normality is often one of the first steps in univariate analysis and is crucial in deciding whether to use parametric or non-parametric tests. Parametric tests like t-test or ANOVA assume data follows a normal distribution, characterized by a symmetric bell-shaped curve with two key parameters: mean and standard deviation. Thus, before applying any statistical test, it is common to evaluate for normality with tests such as the Shapiro-Wilk test or the Kolmogorov-Smirnov test. Notably, Shapiro-Wilk is more suitable for small sample sizes ($N < 50$)[145]. Here, "normal" applies to the entire population, and not just the sample data. In these normality tests, the null hypothesis states that the data follows a normal distribution. The resulting 'p-value' from such tests is compared to a pre-specified alpha level (e.g., 0.05). Here, $p < 0.05$ rejects the null hypothesis, suggesting that the data does not follow a normal distribution. However, $p \geq 0.05$ does not provide any evidence against the null hypothesis given the available data, so there is no evidence that the data is not normally distributed.

Normality becomes less critical with large samples due to the Central Limit Theorem. In such cases, one can consider doing parametric tests instead. When the data does not follow a normal distribution, one can follow non-parametric tests, such as the Mann-Whitney U test or the Kruskal-Wallis test[145].

2.2.3 Aim of the Protocol

The goal of this protocol is to provide an integrated pathway for downstream data clean-up and statistical analysis of FBMN results derived from non-targeted LC-MS/MS data (see **Figure 1.5**) that is straightforward, transparent, and flexible. We aim to complement the structural insights gained from FBMN with downstream statistical findings, thereby enhancing the understanding of the molecular network. Integrating FBMN results with statistical analyses poses several challenges, often necessitating users to reformat, upload, and process the feature table with different external tools, to ultimately manually combine the outcomes. Our approach addresses this gap by offering a detailed guide and comprehensive solution to directly process and analyze the data after FBMN in one pipeline, as shown in **Figure 2**.

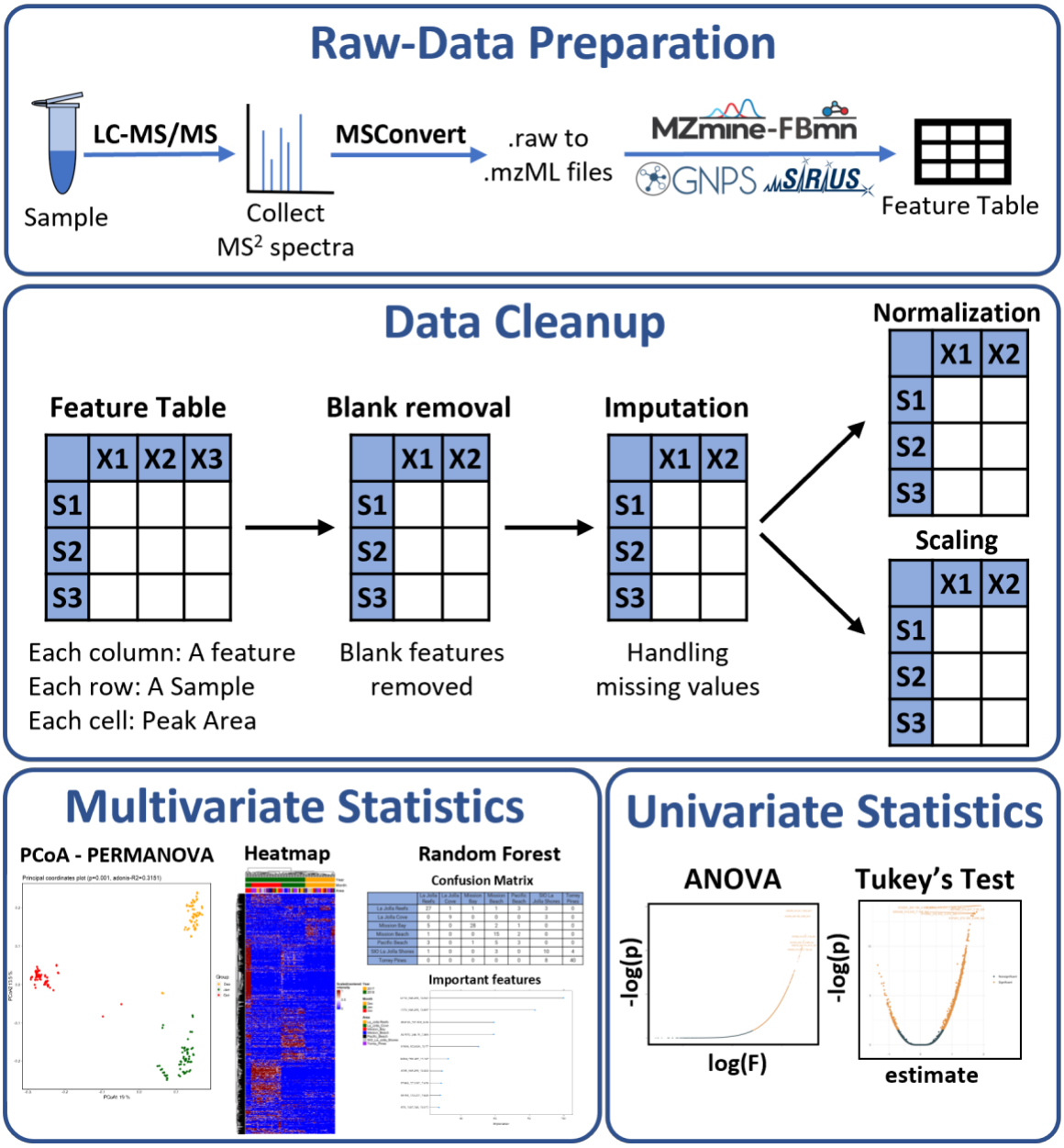


Figure 2: Overview of the Data Analysis Pipeline: Integrating four core segments, the flowchart starts with sample collection and LC-MS/MS data acquisition, transitions to raw data conversion into mzML format, and results in generating a feature quantification table under "Raw Data Preparation". Note, "Raw Data Preparation" is not part of the current protocol. More information on this can be found in the first three blocks of **Figure 1**. The current protocol commences with the "Data Cleanup" phase, which requires a feature quantification table and an associated metadata table produced in the "Raw Data Preparation" section. The feature table contains the relative intensity values for all detected features across the measured samples. The supplementary metadata table contains information on the different categories of the measured samples. The "Data Cleanup" phase emphasizes refining the feature on the feature table through blank removal, imputation, and normalization strategies like Total-Ion-Current (TIC), Probabilistic

Quotient Normalization (PQN), and scaling. Subsequently, the “Multivariate Statistics” segment showcases techniques such as Principal Coordinate Analysis (PCoA) plots, and heatmaps for effective data portrayal. In addition, the users are introduced to robust techniques including Random Forest classification. In the “Univariate Statistics” segment, tests such as ANOVA and Kruskal-Wallis test are discussed.

This protocol provides a transparent, user-friendly, and versatile approach. Transparency and reproducibility are ensured through shareable notebooks and compiled results. To promote usability, the pipeline is provided in popular scripting languages, R and Python, and is presented through Jupyter Notebooks and Google Colab notebooks for cloud-based applications. The pipeline is highly adaptable as well, allowing users to modify and apply the code according to their specific needs. This flexibility is demonstrated by its recent application in a publication on environmental metabolomics, which utilized codes from our Jupyter Notebooks[146]. The protocol is also beginner-friendly, helping new users to easily learn and use Jupyter Notebooks for statistical analyses, thus lowering barriers for those not familiar with R or Python. Finally, tool versatility is provided through customizable workflow modules that can be used as part of the pipeline or in combination with other feature detection, annotation, and data analysis tools, with or without the use of FBMN.

In addition to the R and Python Notebook, this pipeline is also provided within the widely used QIIME2 framework. QIIME2 is a widely recognized next-generation microbiome bioinformatics platform, designed to enable sophisticated analyses in microbiome science[147]. It offers a plugin-based architecture supporting a wide range of functionalities, from sequence analysis to machine learning. QIIME 2 enables comprehensive data integration across various types, such as metabolomics and metagenomics, supported by robust visualization tools for in-depth data exploration[147]. Acknowledging its widespread use, our protocol provides a streamlined workflow within a Jupyter notebook, specifically designed to facilitate analysis for users already familiar with QIIME 2. More information on this can be found in Supplementary Information (SI).

Additionally, we have developed a web application with a graphical user interface (GUI), which can be accessed at https://fbmn-statsguide.gnps2.org/ or downloaded as standalone GUI application from https://www.functional-metabolomics.com/resources. The main manuscript focuses on the concepts and step-to-step guide for the R workflow, while the SI contains step-to-step guides for the Python, QIIME2, and Web app workflows. Though most steps are consistent across the workflows, any differences are addressed and complemented with alternative solutions

in the SI. This protocol is made for both newcomers and experienced researchers in the metabolomics field:

- **For Beginners:** It introduces essential tools, resources, and workflows. The guidelines and code provided make it easier to understand common data processing and analysis steps, facilitating navigation through the complexities of the field. The provided tools utilize common programming languages (R, Python), the QIIME2 platform, and a GUI, allowing users with diverse computational backgrounds to perform data analyses.
- **For Experts:** It accelerates data analysis, ensuring faster interpretation of FBMN results and enables straightforward sharing of statistical data analysis workflows and results.

As inputs, the protocol requires a feature table and its corresponding metadata table. For specifications on the layout and requirements of these tables, please refer to **Figure 2** and 'Required Files' in the 'Materials' section. The feature quantification table, a product of feature detection software, contains a matrix of the relative intensities measured for each feature within each sample. The metadata table provides essential information about the samples and the conditions under which they were measured. Throughout its execution, users receive:

- Intermediate tables after each data cleanup step, aiding in comparison with the original feature table. Tabular outputs for clustering results from HCA, a list of statistically significant features as determined by ANOVA or Kruskal-Wallis tests, and RF outputs indicating feature importance. Significant features refer to those that differ notably in at least one group when comparing multiple groups. Such features can be further investigated to determine if they really cause the differences we observed between groups or samples.
- Visual outputs, such as PCoA score plots, heatmaps, volcano plots for significance tests, and box plots, showcasing group differences for significant features.

This protocol also helps with mapping the results of some of the statistical approaches (e.g., clustering, significant features) back to the FBMN network view. This is facilitated by importing these output feature tables, with feature IDs and the relevant information, into Cytoscape in order to examine the molecular network (See section 3.5 'Integrating Statistical Results into a Molecular Network'). Moreover, as all our resources are publicly available on GitHub, users can actively raise issues or provide suggestions on GitHub.

2.2.4 Limitations and Challenges

Our protocol for FBMN is aimed at offering advanced statistical analysis solutions for a broad range of users. We thus offer notebooks and code in different scripting languages (R, Python, and QIIME2) and platforms (Jupyter and Colab) as well as a web application to suit the specific needs and preferences within the metabolomic community. Readers are encouraged to refer to **Figure 3 and Figure 4** for a quick overview of the available options and guidance on selecting the most suitable one based on their familiarity with the tools and programming languages, and the size of your datasets.

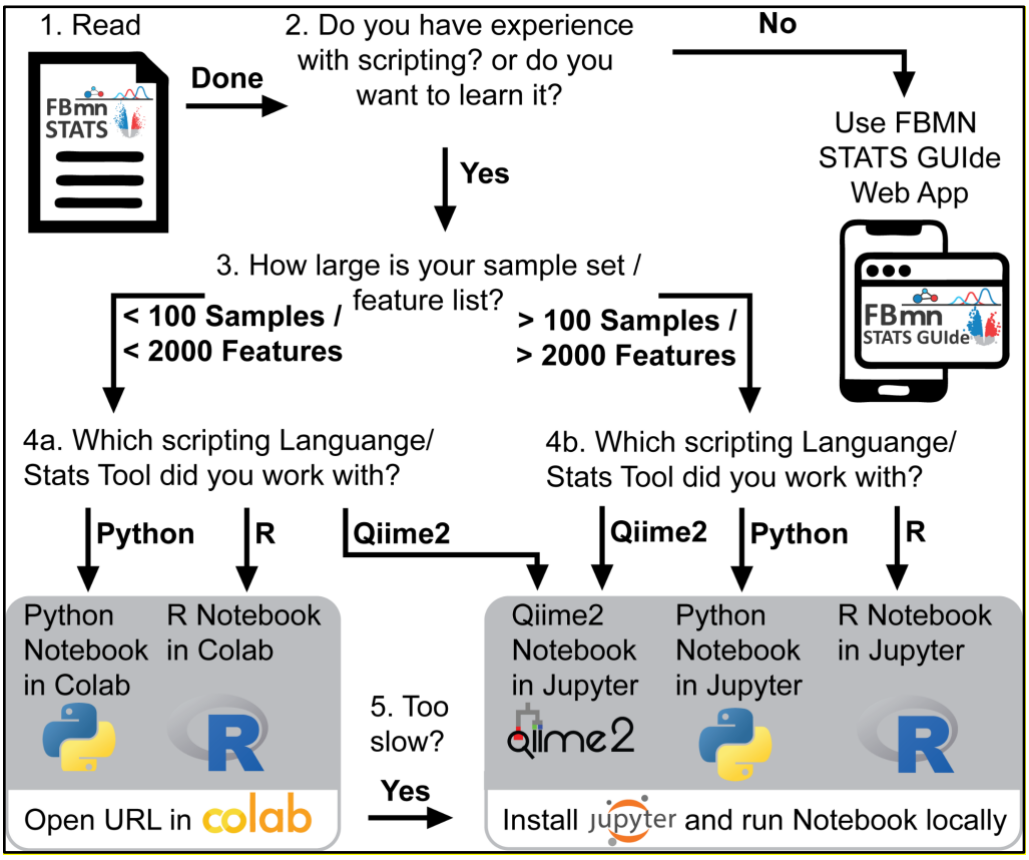


Figure 3: Decision tree to guide choosing which notebook/app to use. The number of samples and features mentioned here is for conceptual guidance only and should not be considered as fixed thresholds.

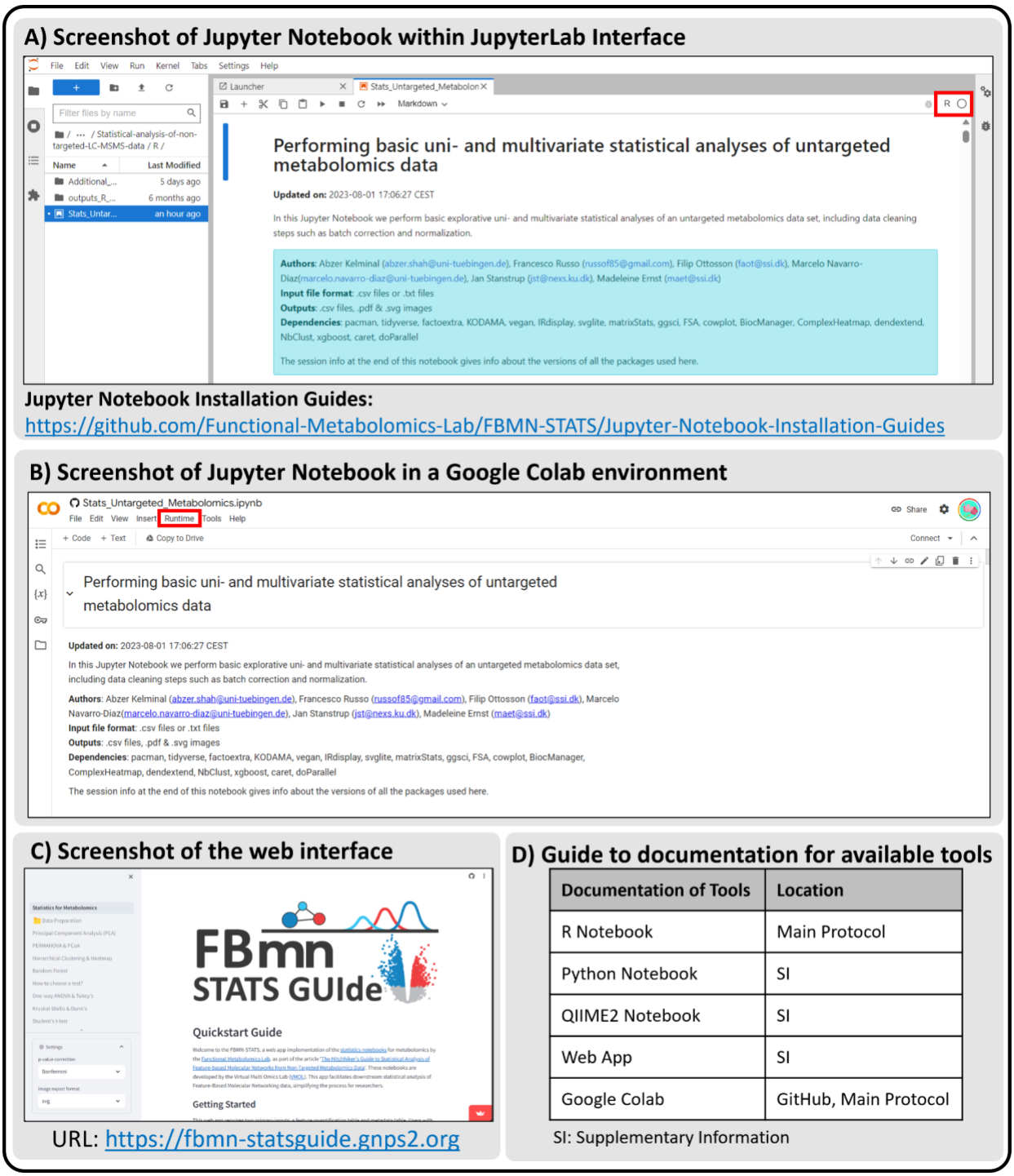


Figure 4: Interface Previews and Documentation Guide. A) Screenshot of JupyterLab Interface with a Jupyter Notebook, running locally, highlighting the kernel selection area (R, Python) in red. Guide for Jupyter Notebook installation is available on our GitHub repository. B) Screenshot of Google Colab Interface with a Jupyter Notebook. Here, the 'Runtime' tab for kernel management is emphasized in red. Colab facilitates cloud-based notebook execution without local installations. C) Screenshot of the web application interface of FBMN-STATS along with the URL entry point. D) Documentation Locations Table,

providing a centralized reference for where to find guidance for each tool, which is dispersed across the main protocol, Supplementary Information (SI), and README files in the FBMN-STATS GitHub repository.

This broad range of choices, while useful, comes with its own set of challenges. For instance, in the R Google Colab notebook, package installation can be time-consuming. Also, the inclusion of `readline` commands, although beneficial for customization, can appear cryptic to beginners. Despite these challenges, we primarily focus on R in the main protocol based on R's extensive range of statistical packages and its widespread use in statistical analysis.

On the other hand, installing packages in the Python Google Colab notebook is relatively faster. While Python excels in machine learning applications, it has comparatively fewer statistical packages than R. There is one vital point to note regarding the 'scikit-bio' package's incompatibility with Windows. Thus, Windows OS users are advised to either use the Google Colab version or consider the Windows Subsystem for Linux (WSL) for local operations. Furthermore, while Google Colab stands as a user-friendly platform, it is not devoid of limitations. One of the main concerns is that runtime automatically disconnects if the user leaves the Colab session inactive for 90 minutes or after 12 continuous hours of usage. This leads to the loss of the data and files they were working on from the Cloud session. Additionally, users must be aware of the 77 GB disk space limitation and ensure timely downloading of their results.

Both the R and Python notebooks comprise over 80 steps, with a significant portion dedicated to data organization. While these notebooks function smoothly with smaller datasets when run on the Cloud, their performance can lag with larger datasets (e.g., those with over 100 samples and more than 2,000 features), especially given the constraints of Google Colab. Despite this, the protocol still functions for large datasets, albeit with increased complexity and computational demands.

In such scenarios, local execution is advisable. For local execution, we have provided guidance on using the Anaconda Navigator, a user-friendly GUI platform, to set up Jupyter notebooks. However, MacOS users might encounter installation challenges. As an alternative, MacOS users can opt for the 'pip install' method or follow the additional installation guide provided in the GitHub repository. The Streamlit web application for the protocol, although user-friendly, has its own set of challenges. Notably, there's a data restriction of 200 MB, and larger datasets might inadvertently slow down the app or even lead to server crashes. Additionally, it offers less flexibility and fewer graphical data representation options compared to comprehensive tools like MetaboAnalyst[44,95]. Although it serves as an introductory tool to the concepts, the notebooks

facilitate more tailored analyses for specific research questions. For complex and detailed investigations, scripting is indispensable. Thus, the GUI is optimally employed as an educational tool, with more advanced resources recommended for in-depth research applications.

Lastly, the QIIME2 notebook is broadly used and applicable for both the microbiome and metabolomics communities. Our additional Jupyter notebook lets users import data directly from a GNPS job link. However, this notebook cannot be accessed in the cloud. Users need to either install QIIME2 and GNPS packages on their computer or use Docker. This might be difficult for some, but it is a good option for those familiar with QIIME2[147]. In all cases, users should always consider the size of their data, their computer's power, and their own skill level while using the protocol.

Although we highlight the importance of experimental design and data acquisition for reducing biological and analytical biases (see **Box 1**), we acknowledge certain limitations in our dataset and workflow, such as the absence of guidance on power calculations. We also acknowledge that our protocol does not include uncertainty propagation, a process important for detailed statistical analysis that tracks how initial uncertainties, such as measurement errors, influence the final results. This omission is mainly due to the immense computational resources needed for such estimation in high-dimensional metabolomics data. Despite discussing the limitations of univariate and multivariate analysis methods, our protocol prioritizes data exploration and hypothesis generation, where immediate precision in uncertainty estimation is not critical[148].

2.2.5 Alternative Open-Source Data Analysis Workflows and Protocols

There have been many efforts in the community to provide and teach statistics solutions for non-targeted metabolomics data analysis, and multiple scripts, web apps, and software tools are available for data clean-up, statistical analysis, and visualization. While we believe that such a streamlined solution for FBMN results, as described in our protocol, has not yet been provided, we would like to point out the many other tools, workflows, and applications that are available.

Table 2: **Overview of alternative statistics tools and scripting solutions for statistical analysis of non-targeted metabolomics data. All tools listed here are open source and freely available.**

Tool Name	Tool Type	Raw Data Processing	Blank Removal	Matrix Transformations	Uni-Variate Statistics	Multi-variate Statistics	Export for Downstream	Customizable	Estimated runtime for respective sample data	URL	Reference
GUI											
MetaboAnalyst	Web App (GUI)	Y	Y	Y	Y	Y	Y	N	Hours (bottleneck: raw data processing)	www.metaboanalyst.ca/	44,95
Workflows											
Galaxy-M	Workflow	Y	Y	Y	Y	Y	N	N	Hours (bottleneck: raw data processing)	github.com/Viant-Metabolomics/Galaxy-M	111,149
Workflow4Metabolomics	Workflow	Y	Y	Y	Y	Y	N	N	Hours (bottleneck: raw data processing)	github.com/workflow4metabolomics	150
UmetaFlow	Workflow	Y	Y	Y	Y	Y	N	N	Days (bottleneck: formula and structural predictions)	github.com/biosustain/snakemake_UmetaFlow	151

Chemometrics Tutorials	Workflow / Tutorial	N	N	Y	Y	Y	N	Y	Hours (bottleneck: Power Analysis)	github.com/Gscorreia89/chemometrics-tutorials	
QIIME2 metabolomics plugin	Language	N	N	N	Y	Y	N	N	NA	library.QIIME2.org/plugins/q2-metabolomics/10/	147
R Libraries											
mixOmics	Package	N	N	Y	Y	Y	Y	Y	Minutes (bottleneck: parameter tuning)	mixomics.org/	75
MetaboanalystR	Package	Y	Y	Y	Y	Y	Y	Y	Minutes - hours (bottleneck: raw data processing)	www.metaboanalyst.ca/docs/RTutorial.xhtml	152,153
omu	Package	N	N	Y	Y	Y	Y	Y	NA	cran.r-project.org/web/packages/omu/vignettes/Omu_vignette.html	154
metabolomicsR	Package	N	N	Y	Y	Y	Y	Y	NA	github.com/XikunH	155

										an/metabolomicsR	
MAIT	Package	Y	N	Y	Y	Y	Y	Y	Hours (bottleneck: raw data processing)	bioconductor.org/packages/release/bioc/html/MAIT.html	[156]
ropls	Package	N	N	Y	Y	Y	Y	Y	NA	bioconductor.org/packages/release/bioc/html/ropls.html	[157]
MSStats	Package	N	Y	Y	Y	Y	Y	Y	Minutes (bottleneck: data cleaning)	github.com/Vitek-Lab/MSstats	[158]
Python Libraries											
TidyMS	Package	Y	Y	Y	N	N	Y	Y	Minutes (bottleneck: data curation pipeline)	github.com/griquelme/tidyms	[159]

We summarized those that, in our opinion, are the most commonly used open-source software tools in **Table 2.** This table provides an overview of their functions, purpose, tool type, and when applicable, references to related protocols and guidelines. Additionally, the table includes a column named 'Estimated runtime for respective sample data,' offering insights into the processing time required by each tool for the sample data they provided or an estimation for our sample data. We present these times in terms of minutes, hours, and days, also noting potential

bottlenecks. This information aims to give users a rough benchmark while acknowledging the diversity in sample data and computational resources across different tools. Further details can be explored through the references listed in the 'Reference' column. We also indicated where in the data processing pipeline these tools have application by indicating yes ("Y") or no ("N") in columns related to raw data processing (generation of feature quantification table, see **section 'Feature Detection'**), data clean-up steps (involving quality control, missing value imputation, normalization, scaling, and transformation), and multivariate and univariate analyses.

We do note that this table is by no means exhaustive. All of these options are workflow-dependent and vary based on factors such as the structure of the acquired feature quantification table and the chosen data analysis techniques[160], and typically require specific file and table formats.

Acknowledging the extensive array of existing platforms, the unique value of our protocol lies in its versatility and accordance with FAIR research principles[161]. Processed feature tables generated after each step, such as post-blank removal, normalization, and scaling, can be integrated with other analysis software like MetaboAnalyst, or amended with newly created scripts. Additionally, the Jupyter notebooks provide easily shareable and reproducible data analysis records, thereby promoting maximum transparency. Our protocol significantly reduces the learning curve associated with Jupyter Notebooks, making advanced statistical analysis accessible to researchers with minimal R or Python experience. Finally, by demonstrating the integration of statistical analysis results into molecular network visualization in Cytoscape, we offer researchers a method to enrich molecular network analyses with diverse informational layers, thereby enhancing the interpretability and impact of their findings from FBMN workflows.

2.2.6 Expertise Needed to Implement the Protocol

We aimed to make this protocol accessible to a broad range of researchers, from absolute beginners to experts. As we provide different options for executing the code (Web application, Colab, and Jupyter notebooks), the protocol should be useful for users both new to metabolomics data analysis, who want to perform a fixed set of processing and statistical analysis, as well as users that require customizable options and need to analyze large datasets. To guide readers through the different options and help to choose which option is most suitable, we generated a decision tree displayed in **Figure 3**. Furthermore, for a preview of the user interfaces across different platforms, **Figure 4: A-C** provides screenshots of a Jupyter Notebook, Google Colab, and the web application. Recognizing that the documentation for each tool is distributed across various sources, we also include **Figure 4D**, which outlines the locations where users can find

comprehensive documentation for all the tools mentioned. At a minimum, we recommend having some general background in statistics and a basic understanding of LC-MS/MS data structure, as well as knowledge about the system and the experimental design of the dataset which should be analyzed. Furthermore, we strongly recommend consulting with a statistician to ensure accurate interpretation and application of the results.

Overview of the Procedure

This protocol primarily focuses on the R workflow, given its broad adoption in metabolomics data analysis and the extensive libraries it offers for this purpose. However, recognizing the diversity of our audience and the growing popularity of other platforms, we've also developed workflows in Python and QIIME2 as well as a web application (details in the SI document). In the following sections, we provide step-by-step instructions of the R Jupyter notebook. Code blocks are included to illustrate the main algorithms and functions.

The Procedure is structured into five sections: Preliminary Setup for the Notebook, Data Cleanup, Multivariate Analysis, Univariate Analysis, and Integrating Statistical Results into a Molecular Network.

'Preliminary Setup for the Notebook' involves initial steps 1-12, preparing the dataset for analysis.

'Data Cleanup' covers steps 13-30, focusing on blank removal, imputation, normalization, and scaling. If blank removal was done in MZmine, skip steps 21-23 and proceed directly to Step 24. The final step, Step 30, allows selection of a processed dataset, stored as `cleaned_data`, for further analysis.

'Multivariate Analysis' consists of the following sections:

- It starts with steps 31-40 for PCoA and PERMANOVA, followed by Hierarchical Clustering (steps 41-45) and Heatmaps (steps 46-50).
- Step 38, which involves setting colors, is a prerequisite for executing steps 47, 56, 64, 68, 71, 76, and 80, as all these steps require the color settings established in Step 38.
- Clustering in Step 42 utilizes the distance matrix from Step 36. Also, Step 45 requires `cleaned_data`. Hence it is recommended to perform clustering sequentially after PCoA and PERMANOVA.
- Heatmaps (steps 46-50) use `cleaned_data` but are independent of previous multivariate tests.

- Random Forest Classification (Steps 51-56) also relies on `cleaned_data` from Step 30 and is not dependent on the previous multivariate steps 32-50.

'Univariate Analysis' (Steps 57-80)

- Other than Step 51, this section is not connected to the multivariate section.
- The metadata from Step 51 is also essential for the 'Test for Normality' (steps 57-59), which informs whether to proceed with ANOVA and Tukey's Post Hoc (steps 60-68) or Kruskal-Wallis and Dunn's Post Hoc (steps 72-80). Note that Dunn's Post Hoc utilizes variables from Kruskal-Wallis, and Tukey's from ANOVA.
- The T-Test (steps 69-71) is an independent parametric analysis for two-group comparisons and can be performed after normality testing (steps 57-59).
- While Univariate Analysis is not directly tied to Multivariate steps, it's advisable to conduct at least one multivariate analysis, like PCoA, to discern global patterns before continuing univariate analysis.

The last step in the notebook, 'Exporting Results' is detailed for Google Colab users, instructing on how to zip and download the session's result files. This step is unnecessary for Jupyter Notebook users as files are stored locally. Hence, it is only included in the Notebooks and not in the protocol.

Integrating Statistical Results into a Molecular Network uses RF output from Step 56 as an example to integrate the results into the FBMN Molecular Network in Cytoscape.

We recommend initially executing the notebook using the provided example dataset. Once familiar, proceed with your own data. This approach ensures a smooth transition from learning to applying the workflow.

2.3 Materials

2.3.1 Software Used

In our protocol, a variety of software options are offered to accommodate different user preferences and system capabilities. These include optional tools such as Google Colab for cloud-based operations, local installation of Jupyter Notebook, Streamlit web application, and docker installation of QIIME2.

System Configuration

The protocol's development and testing were conducted on a robust system, featuring an 11th Gen Intel(R) Core(TM) i7-11700 processor with 8 cores (16 CPUs) and a clock speed of 2.5 GHz, designed for efficient multitasking and rapid data processing. Complementing this, the system is equipped with 32 GB of RAM, enabling smooth handling of large datasets, and a 1 TB SSD, ensuring swift data access and processing. Operating under Windows 11 Pro, the system is compatible with all necessary software for the protocol. Additionally, it includes an Intel(R) UHD Graphics 750 and an NVIDIA GeForce RTX 3070 GPU to meet a diverse range of graphical processing needs. The system is also connected to the University's Ethernet network, with a network speed of approximately 900 Mbps for both download and upload essential for efficient data transfer and cloud-based operations. The processing times outlined in the 'Procedure' section of this protocol are based on tests conducted with our example dataset on the Colab platform. Though the protocol is designed for compatibility with modern laptops or PCs, optimal performance may vary based on individual system specifications.

Resources

For the feature quantification table, MZmine 3 (version 3.3.0) was used, installed on the computer with the above-mentioned system configurations, and executed using a batch file, which is available along with all example input files in the Functional Metabolomics Lab GitHub Repository (https://github.com/Functional-Metabolomics-Lab/FBMN-STATS/tree/main/data). This batch file outlines each step taken in MZmine 3.

To cater to various user needs, the pipeline is accessible through several mediums. Google Colab offers a no-installation-required approach, ideal for those who prefer cloud-based solutions. The Streamlit web application (https://fbmn-statsguide.gnps2.org/) provides an intuitive GUI, suitable for users who favor visual interfaces. Here, users can directly upload all input files by simply entering the task ID from their FBMN job on GNPS. For larger datasets, or those seeking more control, a local installation of Jupyter Notebook is recommended. To assist users in choosing the most suitable method, a decision tree is provided in **Figure 3**.

Recommendation for Beginners and Supplementary Information

As a default for beginners, we recommend using the Colab Notebook with R code to follow this protocol. In addition to the R code, Python and QIIME2 versions are also available in our GitHub Repository (https://github.com/Functional-Metabolomics-Lab/FBMN-STATS/tree/main), with SI detailing files beyond the R code.

2.3.2 Required Files

Feature quantification table (.csv):

- Refer to **Table 1** for an example. This table, typically generated from LC-MS/MS metabolomics studies, includes all mass spectral features (integrated peak areas) and their relative intensities across samples. While we used MZmine 3 to obtain the feature quantification table (in .csv format), users from other platforms, such as MS-DIAL, XCMS, and OpenMS can integrate their tables by leveraging their FBMN task ID for seamless incorporation into the workflow. Manual data input requires referencing our dataset for necessary adjustments. See 'Resources' above for downloading the example feature quantification table (.csv) used throughout this protocol.

Metadata (.txt):

- Refer to **Table 3** for an example. The .txt file can be created in a word editor or spreadsheet programs such as Excel or Google Sheets. The metadata table needs to be created by the user, providing additional context for the measured samples, such as sample type, species, tissue type, and collection time point. See 'Resources' above for downloading the example metadata (.txt) file used throughout this protocol.
- For the datasets to be fully considered for public meta-analysis, we suggest using a standardized metadata format with controlled vocabulary. For guidance, users can refer to the ReDU metadata template (https://mwang87.github.io/ReDU-MS2-Documentation/HowtoContribute/). Make sure the metadata format in this protocol is compatible with GNPS workflows (https://ccms-ucsd.github.io/GNPSDocumentation/metadata/).
- The first column in the metadata, labeled **'filename'**, should match the exact filenames as reported in the feature quantification table. Following this, one should include additional columns to the metadata that begin with **'ATTRIBUTE_'** (e.g., ATTRIBUTE_groups, ATTRIBUTE_timepoint).
- Ensure metadata does not contain any empty cells for each of the attribute columns. Each cell must have a defined value. Fill all empty cells in the metadata table with 'NA'.
- Make sure to include a fully populated column titled '**ATTRIBUTE_Sample_Type**' to define various sample types (eg: Blanks, Samples, Control, QC).
- In our example metadata, columns like ATTRIBUTE_Replicate, ATTRIBUTE_Sample_Type, ATTRIBUTE_Batch, ATTRIBUTE_Month, and

ATTRIBUTE_Year all contain group-based information. This type of grouping will assist in selecting different categories for statistical analysis throughout this guide. Use consistent naming for the groups within ATTRIBUTE columns to avoid confusion and for ease of interpretation (e.g., use 'Control' consistently instead of variations like 'control, control sample').

- You can also include columns with continuous numerical data, such as ATTRIBUTE_Injection_order or ATTRIBUTE_timepoint. To ensure statistical power, it is essential to use biological replicates (we suggest at least three) for each sample type within the experimental design.

Table 3: **Sample metadata layout.**

The first column, 'filename', lists the filenames along with their specific extensions (preferably 'mzML' or the older 'mzXML'), exactly matching the column names in the feature quantification table. Two example "ATTRIBUTE_" columns are also included: "ATTRIBUTE_groups", which showcases sample categorical data (i.e., different sample types such as Control, Sample, and Blanks), and "ATTRIBUTE_timepoint", which is an example for numerical data.

filename	**ATTRIBUTE_groups**	**ATTRIBUTE_timepoint_hours**
control_rep1.mzML	Control	1
⋮	⋮	⋮
Sample_type1_rep1.mzML	Sample_type1	4
Sample_type1_rep2.mzML	Sample_type1	4
⋮	⋮	⋮
blank.mzML	Blank	NA

2.3.3 Additional Input Files

Besides the feature quantification table and metadata, the R and Python notebooks can also integrate molecular annotation files (either in .txt or .tsv format). These include SIRIUS, CANOPUS, and GNPS annotations, which enrich our understanding of each feature during analysis. While the inclusion of SIRIUS and CANOPUS files is optional, they can provide valuable insights.

GNPS annotations can be obtained from the Feature-Based Molecular Networking (FBMN) analysis. The process requires MS/MS fragmentation patterns in the ".mgf" format, a feature quantification table, both obtained with e.g., MZmine 3 (see **section 'Feature Detection'**), and a metadata file. The .mgf file carries spectral information about specific MS/MS scans designated for each feature and feature IDs match with feature IDs in the feature quantification table. All of these files need to be uploaded to the GNPS platform.

The metabolite annotation requires a user-defined mass tolerance. Subsequently, MS/MS patterns are matched against the GNPS database using a modified cosine similarity[77], resulting in a molecular network that allows for the identification of compound names for all library hit features. The output of the FBMN job associated with the example data of this protocol is publicly available at https://gnps.ucsd.edu/ProteoSAFe/status.jsp?task=b661d12ba88745639664988329c1363e and can be downloaded using the 'Download Cytoscape Data' option. The FBMN job's .graphml file, found under the folder "gnps_molecular_network_graphml", can be used to visualize the molecular network in Cytoscape software. The respective annotated files are located in the "**DB_result**" and "**DB_analog_result**" sub-folders (assuming an analog search is performed), with the former offering level 2 and the latter providing level 3 (molecular class) annotations. The analog search identifies structurally related molecules within the molecular network by applying a score threshold, such as a minimum cosine score that MS/MS spectra must achieve to be considered an annotation during spectral matching with MS/MS spectral libraries. An upper limit can be established for the mass shift between the library and potential analogs (e.g., 100 Da), thus expanding the scope of annotation.

SIRIUS[45] can predict molecular formulas, as well as structures through structure database matching using CSI:FingerID[104,210]. Furthermore, the integrated CANOPUS[121] module provides ClassyFire-based chemical class predictions. As for GNPS, the required input is a .mgf file associated with the MZmine feature quantification table with matching feature IDs across both files. However, the .mgf file exported for SIRIUS through MZmine 3 differs from the .mgf exported for GNPS in that it contains isotopic MS1 patterns for accurate molecular formula prediction.

2.3.4 Example Dataset

The example dataset is part of a previously published study[32], aimed to elucidate the effects of urbanization on organic matter chemotypes in coastal environments after a major rainfall event. Seawater samples were collected from 30 locations over seven areas along the San Diego,

California coastline: Torrey Pines, SIO La Jolla Shores (Scripps Institution of Oceanography at La Jolla Shores), La Jolla Cove, La Jolla Reefs, Pacific Beach, Mission Beach, Mission Bay, capturing both pre- (Dec 2017) and post-rainfall (Jan 2018) conditions. In our analysis, we included supplementary data from October 2018, collected from the same sites (no-rain period), for our pipeline evaluation. The dataset consisted of 180 samples from the three sample times (Dec 2017, Jan 2018, Oct 2018) and 2 PPL process blanks at each of the sample times. The datasets can be found in the MassIVE repository: MSV000082312 and MSV000085786 https://massive.ucsd.edu/ProteoSAFe/dataset.jsp?task=8a8139d9248b43e0b0fda17495387756 https://massive.ucsd.edu/ProteoSAFe/dataset.jsp?task=c8411b76f30a4f4ca5d3e42ec13998dc

While metabolomics studies are typically focused on biological investigations and the example dataset is from an environmental system, the protocol is versatile and applicable to a broad spectrum of fields and sample types. This includes combinatorial chemistry, particularly in organic synthesis, doping analysis, and trace contamination of food, pharmaceuticals, and various other industrial products. It is also suitable for biological samples from diverse sources such as microbiomes, bioreactors, or biomedical research. In general, the only requirement for the data to be processed using our protocol is compatibility with the feature table and metadata format specifications, making it a versatile tool for multi-omics studies.

Note: Seawater samples collected during October 2018 were not available in the original article yet. The .mzML files were preprocessed using MZmine 3 (version 3.3.0) (https://mzmine.github.io/) and the feature-based molecular networking workflow in GNPS (https://gnps.ucsd.edu/ProteoSAFe/status.jsp?task=b661d12ba88745639664988329c1363e).
To review the specific parameters used for building this FBMN network, readers are encouraged to visit the provided GNPS link and the SI document. Additionally, the Cytoscape files shown in the results can be found in our GitHub repository: https://github.com/Functional-Metabolomics-Lab/FBMN-STATS/tree/main/Integrating_Stats_Results_to_Molecular_Network.

2.4 Procedure

2.4.1 Preliminary Setup for the Notebook

<CRITICAL> To ensure proper execution and chronological order, please run the notebook cell-by-cell instead of running multiple cells simultaneously. The numbers assigned to each cell will help you navigate and determine if the cells have been executed correctly and in chronological

order. Additionally, refer to **Box 2** for general instructions on navigating the Jupyter Notebook, such as identifying code cells, recognizing those that require user input, and adding comments.

1| Choose a Notebook and install it. We recommend beginners use Google Colab for the R notebook due to its hassle-free setup as it requires no installations, making it accessible for those unfamiliar with the setup process. However, for regular analysis, local execution in Jupyter on a contemporary desktop computer (E.g., Intel i7, 16 core, 64 GB RAM) is typically faster and more efficient. The reported processing times here are based on our example dataset on the Colab platform. The durations other than for package installation are estimated from a beginner's viewpoint, reflecting the time typically required for someone new to complete the analysis.

To install and run Jupyter Notebook locally in R, we have provided additional installation guides in the GitHub repository (https://github.com/Functional-Metabolomics-Lab/FBMN-STATS/blob/main/Jupyter-Notebook-Installation-Guides). Windows users are advised to use Anaconda Navigator for a straightforward setup, with detailed instructions provided in our guide. Extensive documentation for Anaconda Navigator can also be found at Anaconda's official site (https://docs.anaconda.com/free/navigator/tutorials/create-r-environment/). Mac OS users may encounter difficulties with Anaconda Navigator; thus, we suggest the Homebrew-based installation, detailed in the aforementioned link. For an introductory guide and additional tips on using Jupyter Notebook, please refer to **Box 3.**

Package Installation ● Timing 15 mins

2| Install the R packages.

- The notebook utilizes R version 4.1.3 (2022-03-10)
- Begin by installing and loading the necessary R packages using the `p_load()` function from the 'pacman' (v0.5.1) package[211]. This function checks if a package is installed, if not, it installs the package from CRAN or other repositories in the pacman repository list and loads the package. It is a more efficient alternative to using `install.packages()` and `library()` functions individually for each package.

Required Packages: The following R packages are essential for this protocol:

- Data Cleanup: tidyverse[212] (v2.0.0), IRdisplay[213] (v1.1), KODAMA[214] (v2.4)

- Multivariate Statistics: factoextra[215] (v1.0.7), vegan[216] (v2.6-4), ComplexHeatmap[217] (v2.10.0), dendextend[218] (v1.17.1), NbClust[219] (v3.0.1), rfPermute[220] (v2.5.1)
- Univariate Statistics: FSA[221] (v0.9.4), matrixStats[222] (v0.63.0)
- Visualization and Plotting: ggsci[223] (v3.0.0), cowplot[224] (v1.1.1), svglite[225] (v2.1.1)

At this stage, install the packages required for Data Cleanup.

<CRITICAL STEP> Packages are installed just before their respective sections (in **steps 2, 26, 31, 51, 57**) to reduce installation time. However, please note that the packages installed initially in one section can be used for the later sections as well. For example, tidyverse (v2.0.0) can be used throughout the notebook, not just for data cleanup.

3| **Set Working Directory** *(User Input Required)*

- Set a folder as the working directory. This is where you access input files and save output files. Make sure to include all necessary input files in this folder.
- In Google Colab, click on the three dots in the upper left corner to see the notebook contents. Click on the folder icon and create a new folder by right-clicking in the empty space and selecting 'new folder'. Further details on using Google Colab are provided in **Step 4**, along with **Figure 5** for visual guidance.

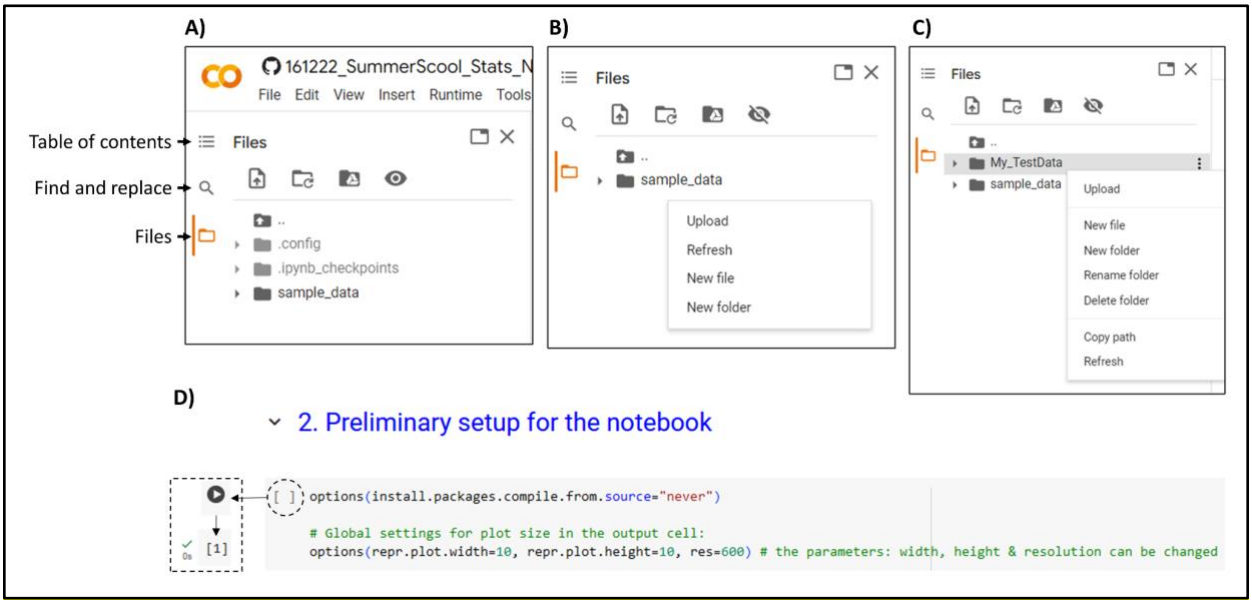


Figure 5: Google Colab Interface for Managing R Notebooks: A) Menu and left sidebar highlighting icons, 'Table of Contents,' 'Find and Replace,' and 'Files,' with 'Files' icon selected in orange. B) The action of right-clicking in the 'Files' area shows options for new folder creation, alongside the default 'sample_data' folder in the Colab. C) User-created 'My_Testdata' folder for organized data storage, demonstrating file

upload options. D) showcases the code execution process with empty square brackets before running, changing to a play button upon hover, and displaying execution order (“1”) and time (0s) after running.

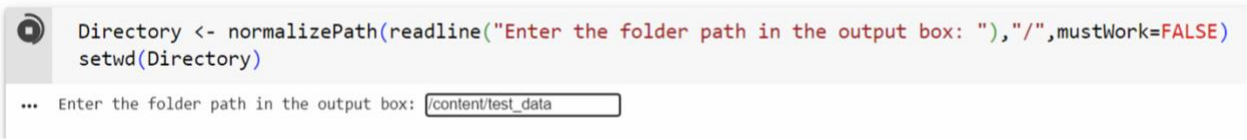


Figure 6: Screenshot of the code cell from R Google Colab Notebook to set the working directory. The image displays the code cell targeting the ‘/content/test_data’ directory. This user-created directory holds the input files for the data analysis. Note the stop symbol with the surrounding loading circle, indicating the cell awaits user input. To proceed to the next cell, provide the input (e.g., /content/test_data) and press enter.

- To set your working directory in the notebook, run the code cell (a cell or a section containing codes) as shown in **Figure 6**. A pop-up box will appear as illustrated in **Figure 6**, prompting you to enter the path to your designated folder with input files. Insert the respective path and press ‘Enter’ to proceed to the next cell.
- **(For Local Environment)** If you are running the notebook in your local environment, you can directly specify the local path of your folder to set it as your working directory. For example, if your folder is located at D:\User\Project\Test_Data, simply input this path when prompted and press ‘Enter’.

Data Import

<CRTICAL> This section guides users through the process of importing necessary data files for the Notebook. Various methods are outlined, catering to different data sourcing preferences. Make sure your metadata has the necessary attribute columns to describe the data (at least one, e.g., ATTRIBUTE_SampleType). If your FBMN metadata is insufficient, you might need to load additional metadata from a local folder for downstream statistical analysis, adding a step in the workflow.

4| **Uploading Files to Google Colab** *(User Input Required)* Some general guidelines for using Google Colab are as follows:

Kernel Selection: When setting up the Google Colab environment, ensure you have the correct kernel (R or Python) activated for the notebook. To do this, go to the ‘Runtime’ tab, and

select ‘Change runtime type’ to choose your desired kernel. For multiple R versions, like ‘R’ and ‘R.4.2.2’, we suggest choosing ‘R’ to ensure compatibility.

Interface Navigation: The Google Colab interface is user-friendly, featuring some useful icons on the left sidebar (**Figure 5A**). The ‘Table of contents’ icon facilitates quick navigation through the notebook and a ‘Find and replace’ icon for efficient content search. Managing your files is streamlined through the ‘Files’ icon, where uploading and organizing data is straightforward.

File Management:

- **Uploading:** Google Colab does not allow local file access. Hence, directly upload files from your computer to Google Colab. If you do not want to use files from your local machine, you can skip this (**Step 4)** and proceed directly to **Step 5B** (‘Loading Files from URL’) or **Step 5C** (‘Loading Files from GNPS’).
- **Organization:** For an orderly workspace and clear data arrangement, create folders for your dataset. Right-click in the ‘Files’ area to upload data, refresh the view, or create new folders (**Figure 5B**). **Figure 5C** shows the folder ‘My_TestData’, created for organizing user data. Right-click on the created folder and select ‘upload’ to transfer files from the local machine to the cloud session. Alternatively, one can also ‘drag and drop’ files directly into the folder.
- **Google Drive Access:** Linking Google Drive is simpler with Python. It is less direct with R, hence not advisable.

Executing Code:

- Google Colab distinguishes code cells (in gray) from text annotations, also called ‘Markdown cells’ (in white). Run the code by clicking the play button on each cell’s top left corner. See **Figure 5D**.
- Execution outcomes are indicated by a green check (for success) or an error message (for failures), providing immediate feedback. See **Figure 5D.**
- One can also run several code cells sequentially within the notebook. Look for an arrow icon next to the section titles in the notebook (refer to **Figure 5D**, illustrating section header ‘2. Preliminary setup for the notebook’). Clicking this icon reveals the total number of concealed cells within that section, such as ‘84 hidden cells’, including both code and markdown cells. Activating the play button situated below the section title initiates the

execution of all these cells in a collective manner. However, it is important to be cautious with these batch runs, especially if you are not familiar with the steps being executed, as some may require user input. Batch runs are most useful for sections that do not need any input from the user.

5| **Select a Data Loading Method.** There are three options. Files can be uploaded from a working folder (A), from a URL (B) or directly from a repository e.g. GNPS (C).

A Loading Files from the folder

(User Input Required)

i. Begin by viewing a table displaying a list of files in your working folder (e.g., uploaded in the previous step) as shown in **Figure 7A**. Each file will have an index number associated with it. Your task will be to import three tables by specifying the index number associated with each: the feature quantification table (`ft`), the metadata table (`md`), and optionally, annotation tables (`an`). For an example, see **Figure 7B**. To guide you through this process, there are three code blocks that require user input.

A)

A data.frame: 5 × 2

INDEX	FILENAMES
<int>	<chr>
1	20221102_SD_BeachSurvey_batchFile.xml
2	20221125_Metadata_SD_Beaches_with_injection_order.txt
3	GNPS_analog_result_FBMN.tsv
4	GNPS_result_FBMN.tsv
5	SD_BeachSurvey_GapFilled_quant.csv

B)

```
input_str <- readline('Enter the index number of the feature table and metadata separated by commas: ')
Enter the index number of the feature table and metadata separated by commas:
5,2
```

Figure 7: Screenshots illustrating loading input files from a folder: A) Table showing all the files in the working folder, where the first column, labeled **"INDEX"**, denotes the serial or **index number** of the files. B) Shows the user input interface. Upon executing the code cell, the user is prompted to enter the index numbers for the feature table and metadata. In this example, "5" and "2" are entered, where "5" corresponds to the feature table file "SD_BeachSurvey_GapFilled_quant.csv" and "2" to the metadata file "20221125_Metadata_SD_Beaches_with_injection_order.txt", as shown in (A).

ii. **Feature Quantification Table and Metadata Import:** The first code block will prompt you to enter the index numbers associated with the feature quantification table and metadata, separated by commas. Simply input the corresponding index number assigned to each of these files.

iii. **Annotation Tables Import:** The second code block will request the index numbers associated with the annotation tables. Specifically, you will be asked to enter the index numbers of the GNPS library annotation file and the analog annotation files. If you have not performed an analog search for FBMN, only provide the index number of the GNPS library annotation file.

iv. **SIRIUS Annotation File Import (Optional):** The third code block in the notebook queries if you have an additional SIRIUS annotation file (Y/N). If 'Y', you will enter the file's index number; if 'N', you move to the next cell. This file will be used to merge all annotations (e.g., GNPS library, analog hits, SIRIUS) into a single master table for easier data exploration later on. It is worth noting that this protocol does not specifically focus on SIRIUS annotations for analysis. The inclusion of SIRIUS annotations is solely for the convenience of consolidating all annotations in one place for the user.

<CRITICAL STEP> By following these prompts, one can successfully load the essential tables required for the subsequent analysis. Make sure to carefully input the correct index numbers.

B Loading Files from URL

(User Input Required)

i. We also provide an example of retrieving data from a URL. For this example, open the following URL to obtain the feature quantification table: https://raw.githubusercontent.com/Functional-Metabolomics-Lab/FBMN-STATS/main/data/SD_BeachSurvey_GapFilled_quant.csv)

ii. Access the feature quantification table, metadata, and analog result files directly from our Functional Metabolomics GitHub page.

<CRITICAL STEP> The files can be sourced from any public site where your data is stored, such as GitHub. If you have your dataset hosted on GitHub or a similar platform, simply use the file's URL.

If you are using your own dataset (or the test dataset) in Google Colab, you can get the file URL by uploading the input files to the Colab workspace, right-clicking on the file, selecting "Copy path", and then replacing the URL in the relevant cell.

C Loading Files from GNPS

(User Input Required)

i. In this step, you can load files directly from the repositories MassIVE or GNPS. If you have performed FBMN on your feature quantification table, you can access the required files by providing the task ID. To locate the task ID of your FBMN job within your GNPS account, navigate to the 'Jobs' section. Here, the 'unique ID' for each job is listed in the 'Description' column.

ii. When you run the relevant cell in the notebook, you will be prompted to enter the task ID within the notebook. Given the task ID, the notebook will retrieve the necessary files for further analysis.

Exploring the Imported Files

6| Use the `head()` and `dim()` functions to get an initial view of your imported data files.

- The `head(ft)` function displays the first six rows of the feature table by default, giving you a quick look at your data's structure.
- The `dim(ft)` function reveals the dimensions of your feature quantification table, i.e., the number of rows and columns.

<CRITICAL>▲**CRITICAL:** If you encounter an error while executing certain code cells, it is good practice to verify the correctness of your data tables using the `head()` and `dim()` functions.

7| We also provide a special summary function `InsideLevels(md)` to explore the metadata, which returns a summary table with columns for INDEX, ATTRIBUTES, LEVELS, COUNT, and ATTRIBUTE_CLASS.

- **INDEX**: Row number in the summary table
- **ATTRIBUTES**: Column name of the attribute, e.g., ATTRIBUTE_Sample_Type

- **LEVELS**: Unique groups within the attribute column, e.g., Blanks, Sample
- **COUNT**: Number of files for each category, e.g., 6, 180 indicating 6 files for "Blank" sample type and 180 for "Sample" sample type.
- **ATTRIBUTE_CLASS**: Data type of the attribute. Useful for spotting cases where a numeric attribute like ATTRIBUTE_minutes is classified as 'character'.

Merging Annotations with Feature Quantification Table

<CRITICAL> This section involves integrating various annotations, such as SIRIUS and GNPS annotations, into our feature quantification table. This process is vital as it helps identify the metabolites corresponding to the features in our feature quantification table, aiding in the interpretation of our metabolomics data.

8| **Identify Appropriate Columns for Merging.** Depending on the type of annotation to be merged, the feature quantification table's unique 'row ID' column is matched with the corresponding column in the annotation file:

- **GNPS Annotations:** The '`row ID`' is matched with the '`#Scan#`' column in the GNPS annotation file. The '`Compound_Name`' column contains the annotation information.
- **GNPS Analog Annotations:** Similar to GNPS annotations, the '`row ID`' is matched with the '`#Scan#`' column in the GNPS analog annotation file. The '`Compound_Name`' column contains the annotation information.
- **SIRIUS Summary Files:** The '`row ID`' is matched with the '`id`' column in the SIRIUS summary file. A typical feature ID in the '`id`' column might look like this: "3_ProjectName_Mzmine 3_SIRIUS_1_16", where the last string (16) represents the row ID.

9| **Ensure data compatibility before merging.** Before merging, we ensure that the classes (or data type) of the columns meant to be merged are the same. Then, we can combine feature and annotation data based on the appropriate matching columns. Any mismatch, such as one column being of character type while the other one is numeric, can cause merge errors, even if the values within the columns are identical.

10| **Merging Annotations.** To do this:

- Rename the column names of analog annotation dataframe `an_analog` with an 'Analog_' prefix and merge the modified `an_analog` dataframe with `an_gnps` based on '#Scan#'.
- For each unique '#Scan#', consolidate multiple compound names into a single row. If both the GNPS compound names (actual and library hits) for a particular '#Scan#' are identical, keep one; otherwise, combine them using a ";" separator. The output is `an_final_single`.
- Merge `an_final_single` with the feature quantification table (`ft`) using '#Scan#' and 'row ID' as matching columns respectively. Keep all rows from the feature quantification table.

Additional steps:

11| **Incorporate additional annotations (optional)**

- If SIRIUS annotations are available, follow these additional steps: Extract 'row ID' from the 'id' column in the SIRIUS dataframe, rename the columns with a 'SIRIUS_' prefix, and merge the modified SIRIUS dataframe with `an_final` data frame based on 'row ID'.
- For simplicity, we have shown here how to merge the summary file from the SIRIUS module. Given the versatility of SIRIUS, which includes other modules like ZODIAC for molecular formula predictions, CSI: FingerID for molecular structure prediction, and CANOPUS for compound class prediction, users will obtain separate summary files for each module. Our protocol shows one example of merging summary files from the SIRIUS module, the commonly used module, storing the output in a variable named 'sirius'. Users working with summary files from CANOPUS or other modules can similarly save and adapt the code using a different variable, like 'canopus', allowing for flexible adaptation as per their requirements.

12| **Export the merged annotations.** To do this, write the merged annotation table to a CSV file for further data exploration and downstream analyses.

Ensuring Metadata and Feature Quantification Table Compatibility for Downstream Analysis

<CRITICAL> This section streamlines the metadata and feature quantification tables, ensuring they align perfectly for subsequent steps in the protocol and remove discrepancies between them. By following the outlined steps, we achieve harmonized data structures. A final verification confirms that all files in the feature table are mirrored in the metadata, and vice versa. Upon successful validation, the tables are set for the next section of analysis. If there is a mismatch, often due to naming inconsistencies or missing files, the user needs to rectify these issues before moving forward. As a user, you are not required to modify any of the code within this section in the Jupyter Notebook. Simply run each code cell in turn.

13| **Creating Backup Files**. The feature quantification table (`ft`) and metadata (`md`) files are stored under different names (`new_ft`, `new_md`) to preserve the original versions.

14| **Clean up the Feature Quantification Table**

- Clean the feature quantification table by removing 'peak area' extensions from the column names, a default format included in MZmine-derived feature quantification tables.
- Check and remove any columns containing only NA values present in the feature and metadata tables.
- Check and remove any rows and columns containing only empty strings in the metadata table

15| **Update the Row Names of the Feature Quantification Table**

- In this step, we reformat the row names to consolidate essential information about each feature. By doing this, we can retain only the numeric data in the feature quantification table and remove all other columns.
- The row names are constructed by concatenating the Feature ID, *m/z*, RT, and GNPS annotations into a single string, in the following format: `XFeatureID_*m/z*_RT_GNPS_annotations`.
- An example row name is `**X92649_226.951_14.813_NA;TRYPTOPHAN**`. Here, "NA" indicates that there was no direct library hit for this feature. However, the analog annotation suggests it could be tryptophan.

- In the R environment, a dataframe's row names must be characters or strings. Thus, we add the 'X' character prefix to the numeric Feature ID to ensure compatibility.

16| **Select Relevant Columns** *(User input - Optional)*

- Retain only '.mzML' (or '.mzXML') file-relevant columns and remove extraneous information, such as additional columns added due to IIMN. Here, the features are represented as rows in the feature quantification table.
- Only when the file extensions '.mzML' or '.mzXML' are not available, the user is prompted to enter their respective file extension.
- This step ensures that the feature quantification table contains only the intensity values of the features, which is crucial for subsequent calculations. The modified row names provide basic feature information, and for a more detailed understanding, you can refer back to the original feature quantification table.

17| **Verifying File Consistency** The metadata and feature quantification tables are arranged in the same order of '.mzML' (or '.mzXML') file names. We then verify consistency between the feature and metadata tables by using the `identical(new_md$filename, colnames(new_ft))` command.

- If the result is TRUE, proceed to data cleanup.
- If FALSE, there might be missing files or discrepancies in file naming. Check the corresponding column names in the feature quantification table for potential errors like spelling mistakes or case-sensitive issues, and re-upload the correct files. Re-run all the above steps once corrected.

BOX 2 General Instructions for Navigating the Jupyter Notebook

- **Text in Red**: These sections indicate critical information or code cells that require user input within the notebook. They serve as instructions for adapting the notebook to different datasets without the need to modify the code. Further details are provided within the notebook.
- **User Prompt Guidance:** Within the notebook, when you encounter code cells with red highlights, simply execute them without changing their contents. For instance, you may come across lines such as:

```
Directory <- normalizePath(readline("Enter the folder path in
the pop-up box: "),"/",mustWork = FALSE)
```

To provide input, a pop-up box will appear in the output section. Make sure to enter your answers in the pop-up box instead of entering directly within the code cell. After entering your input, remember to press 'Enter' to proceed to the next step.

Using these prompt boxes ensures that user input is seamlessly integrated into the following operations. The position of these prompt boxes might differ depending on your system as they could appear directly below the active code cell, at the notebook's top, or even towards the upper section of your screen.

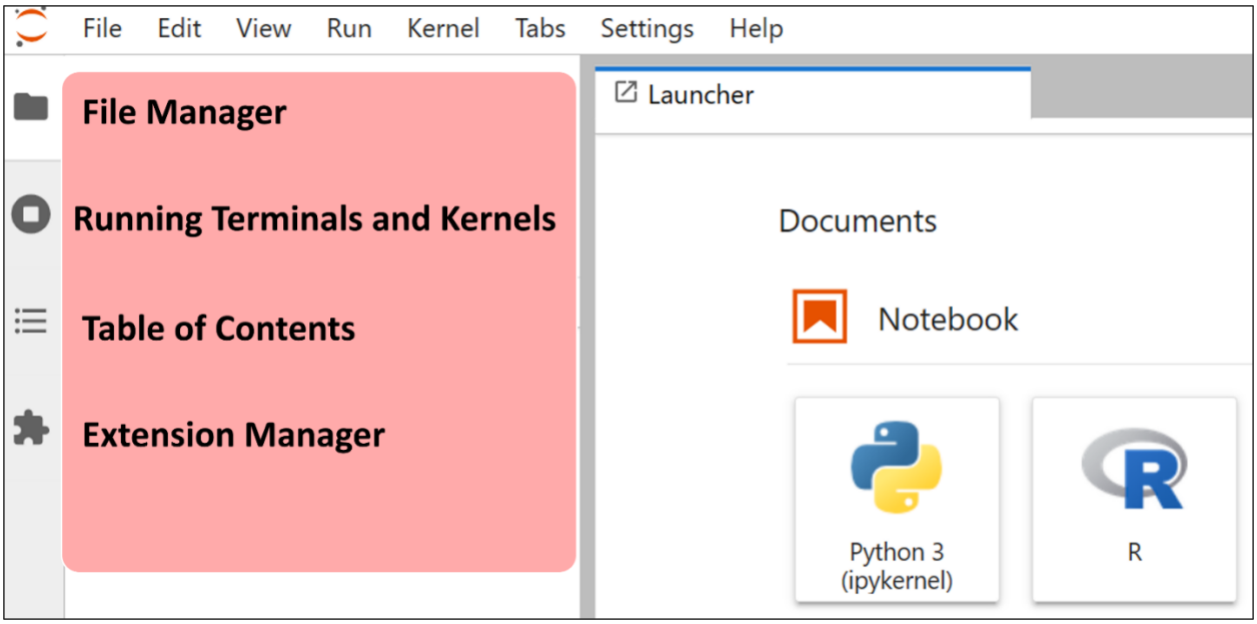
- **Text in Green**: This indicates that the following code cell in the notebook contains function definitions and will not display any visible outputs. Even though the underlying code in these cells may seem complex, its purpose is to make repetitive tasks more efficient. Readers who come across these green-highlighted code cells do not need to understand the complexities of the code.
- **Using the '#' Operator**: Lines in the code cells that start with '#' are comments explaining the code's function or purpose. These comments are "commented out" and will not be executed. To run a commented-out code, remove the '#' symbol and run the cell again.

Box 3: Using JupyterLab to manage Jupyter Notebooks locally

JupyterLab serves as an integrated development environment for managing Jupyter Notebooks. It offers a user-friendly way of working with multiple Notebooks simultaneously, sharing variables between Notebooks and saving Notebooks mid-session so that analysis can be continued at a later time.

Getting Started: For those new to running Jupyter Notebooks locally, we have included a section on how to install and open JupyterLab in our installation guidelines. The section also includes a short introduction to the JupyterLab interface.

Navigating JupyterLab: Upon launching JupyterLab a workspace will open:

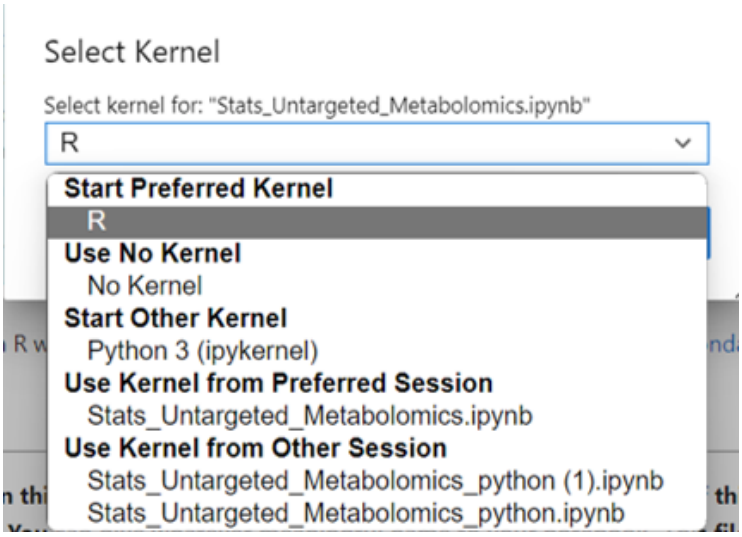


The left sidebar features four main icons: (1) **File Browser:** Access and manage your files; (2) **Running Terminals and Kernels:** View your active sessions; (3) **Table of Contents:** Quickly navigate through the sections of your current Notebook. This is particularly useful for large Notebooks; (4) **Extension Manager:** Customize JupyterLab with additional features.

Opening a Jupyter Notebook locally: There are two options to open a notebook locally: (1) Navigate to the location of your Notebook using the **File Browser** and double-clicking the notebook, or (2) Go to "*File → Open from Path"* and enter the path to your notebook (Example: `Downloads/Stats_Untargeted_Metabolomics_python.ipynb`)

Selecting a Kernel: Selecting the correct kernel is important for successful execution of a Notebook. The kernel active in the current Notebook is displayed in the top-right corner (red box). The gray circle indicates the state of the kernel (unfilled = idle, filled = running).

Click the name of the current kernel to start or switch between kernels. A dropdown menu will pop up.



Start Preferred Kernel and **Start Other Kernel** will start a *new* kernel for the Notebook. For R Notebooks, make sure to select the R kernel. If you have already started a kernel for another analysis and wish to carry over all variables and settings to the current analysis, choose a kernel from **Use Kernel from Preferred Session** or **Use Kernel from Other Session**.

Saving your work: Navigate to the *"File"* Tab found in the top left corner of the interface. Here two main options are available to save your work: (1) **Save Notebook:** Saves the state of your current Notebook, does not preserve variables. (2) **Save Workspace:** Saves the current Notebook and all variables to avoid having to re-run the entire Notebook upon return.

Further Information: For detailed installation instructions and further guidance, please refer to our installation guidelines over on GitHub (https://github.com/Functional-Metabolomics-

Lab/FBMN-STATS/blob/main/Jupyter-Notebook-Installation-Guides) and the official Jupyter documentation (https://docs.jupyter.org/en/latest/).

2.4.2 Data Cleaning: ● Timing 20-30 mins

<CRITICAL> Following the LC-MS/MS data pre-processing with MZmine, we perform the post-processing of the data (also known as data pretreatment or data clean-up) as the first crucial step in our workflow. While the 'preliminary setup for the notebook' section prepares the feature and metadata tables for analysis, actual modifications to the data commence from this section.

18| **Transposing the Feature Quantification Table**

- No user input is required other than running this code cell. The transposition and all subsequent actions will be executed automatically. As a first step, we transpose the feature quantification table. The result is a table (`ft_t`) where the row names represent the sample names, and the column names consist of concatenated feature information.
- Then, we merge this transposed feature quantification table (`ft_t`) with the metadata (`new_md`), using the sample names as the common link. This merged table, referred to as `ft_merged`, consolidates all necessary information in a single structure.
- The `ft_merged` table can also be exported to a CSV file for future use, such as batch correction or other specialized analyses.

Batch Correction (Optional)

<CRITICAL> The most common method for visualizing or identifying the presence of batch effects is through a simple PCA, guided PCA[226], or PCoA. In the PCA/PCoA scores plot, it is generally expected that all the (pooled) QCs cluster together indicating little analytical variation in the data. When the inter-batch variation gets higher, the inter-QC distances in the PCA/PCoA plot will also increase[227].

19| Batch effects can be detected and corrected using either the steps outlined in this notebook (option A) or established literature methods (option B). Option A involves steps that refer to PCoA, detailed later in the Procedure. We have deferred the resulting visualization to a later section, after data cleanup. As we delve deeper into multivariate

analyses after data cleanup, this approach avoids redundancy and ensures users can maximize the utility of this protocol.

A **Using the notebook:**

i. Execute **Step 31** to install the necessary packages for multivariate analysis.

ii. Run **Step 38** and **Step 39** to visualize the PCoA using the custom-made `plotPCoA()` function. Detailed usage instructions are provided in the respective steps.

iii. If your sample type information (description of which samples are pooled QCs, blanks, samples, etc.) is located in the 'ATTRIBUTE_Sample_Type' column of the metadata, invoke the function by typing:

```
plotPCoA(
       ft = ft_t,
       md = new_md,
       distmetric = "euclidean",
       category_permanova = "ATTRIBUTE_Sample_Type",
       pcoa_category_type = 'categorical',
       category_pcoa_colors = "ATTRIBUTE_Sample_Type")
```

B **Established Literature Methods:**

i. Determine whether there is a batch effect using analysis of variance (ANOVA) by comparing the QC mean of different batches for statistically significant differences[228].

ii. If there is a notable batch effect choose an appropriate approach to correct the effects. These include:

- **Normalization** methods such as Metabodrift[229], ComBat[230]
- **Transformation** method: waveICA[231] (wavelet transformation coupled to ICA)
- **Regression-based** approaches such as the linear least-square (LS) method[232], QC-based robust LOESS correction[163], QC-support vector regression[233] (QC-SVR)
- **ML-based** methods such as random forest-based QC-RFSC correction[234], deep learning model: NormAE (Normalization Autoencoder) algorithm[235], Regularized Adversarial Learning Preserving Similarity[236] (RALPS).

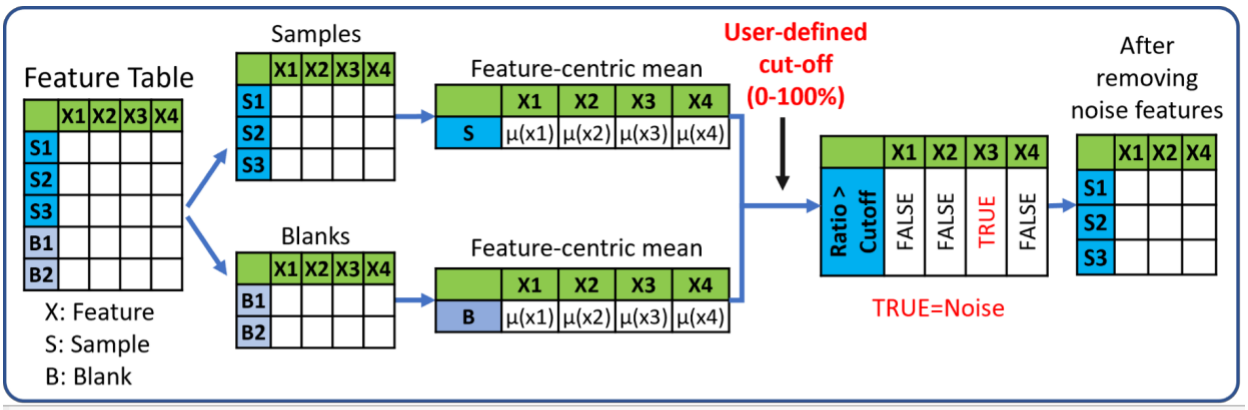
Each method has its strengths and limitations. It is important to note that, depending on the experimental design (e.g., sample size and sample diversity), pooled QCs are not always utilized. In situations where there are no QCs included in the study, normalization can be used instead to attempt to reduce most of the unwanted variations[237], at the risk of removing true biological variation. For the sake of simplicity and to cater primarily to beginners, this protocol does not elaborate on batch correction. However, for those interested in exploring batch correction in depth, we have prepared a supplementary R notebook available on our GitHub repository (https://github.com/Functional-Metabolomics-Lab/FBMN-STATS/blob/main/R/Additional_Notebooks/Batch_Correction.ipynb). In this notebook, we execute inter-batch correction similar to the method described by Qin Liu *et al.*[167] The procedure involves calculating the mean of each feature across all batches, then calculating the batch-specific feature means, and subsequently adjusting feature intensities within each batch relative to the batch-specific and overall means. For intra-batch adjustments, this additional notebook illustrates the QC-based LOESS correction method, with a prerequisite that each batch should start and end with a pooled QC injection.

Blank Removal

<CRITICAL> To prioritize or identify metabolites from our samples, we need to remove contaminants, i.e., features found in the blanks, before proceeding with statistical analysis[238]. While blank removal can be executed during pre-processing with MZmine 3, which might result in the absence of blank features and samples in both the feature table and metadata, conducting it during post-processing offers more flexibility. For a graphical overview of blank removal, see **Figure 8**, and for more insights, refer to **Box 4**.

<CRITICAL STEP> If you have performed blank removal during pre-processing, simply skip **steps21-23**. Before **Step 24**, assign your feature table and metadata to `blk_rem` and `md_Samples` respectively as in the following code block. This step prepares the data for seamless integration into subsequent steps.

```
blk_rem <- ft_t
md_Samples <- new_md
```




```r
#Getting mean for every feature in blank and Samples in a data frame named 'Avg_ft'

Avg_ft <- data.frame(Avg_blank=colMeans(Blank, na.rm= F))
# set na.rm = F to check if there are NA values. When set as T, NA values are changed to 0

# adding another column 'Avg_samples' for feature means of samples
Avg_ft$`Avg_samples` <- colMeans(Samples, na.rm= F)

#Getting the ratio of blank vs Sample
Avg_ft$`Ratio_blank_Sample` <- (Avg_ft$`Avg_blank`+1)/(Avg_ft$`Avg_samples`+1)

# Creating a bin with 1s when the ratio>Cutoff, else put 0s
Avg_ft$`Bg_bin` <- ifelse(Avg_ft$`Ratio_blank_Sample` > Cutoff, 1, 0 )

#Calculating the number of background features and features present
print(paste("Total no.of features:",nrow(Avg_ft)))
print(paste("No.of Background or noise features:",sum(Avg_ft$`Bg_bin` ==1,na.rm = T)))
print(paste("No.of features after excluding noise:",(ncol(Samples) - sum(Avg_ft$`Bg_bin` ==1,na.rm = T))))

blk_rem <- merge(as.data.frame(t(Samples)), Avg_ft, by=0) %>%
                  filter(Bg_bin == 0) %>% #picking only the features
                  select(-c(Avg_blank,Avg_samples,Ratio_blank_Sample,Bg_bin)) %>% #removing the last 4 columns
                  column_to_rownames(var="Row.names")
blk_rem <- data.frame(t(blk_rem))
```

```
[1] "Total no.of features: 11217"
[1] "No.of Background or noise features: 2125"
[1] "No.of features after excluding noise: 9092"
```

```r
head(blk_rem, 2)
dim(blk_rem)
```

	X10015_282.169_2.763_NA	X10035_325.139_2.817_NA	X10037_216.123_2.847_NA	X10047_338.159_2.845_NA	X10058_280.117_2.961_NA
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
SD_01-2018_1_a.mzXML	50907.97	196008.38	90480.91	446560.7	182757.8
SD_01-2018_1_b.mzXML	51443.73	99569.05	411595.38	239022.0	274146.0

180 · 9092

Figure 8: Blank Removal Process: Featuring a graphical representation of the blank removal followed by screenshots of the corresponding R code executed for the procedure.

20| **Examining Metadata Attributes** Run `InsideLevels(new_md)` to identify unique groups within each metadata attribute. This helps to find the attribute column containing sample type information (e.g., ‘Blanks’, ‘Samples’).

21| **Separating Blank and Sample Files** *(User Input Required)* In this step, the data is split into two groups: ‘blank’ and ‘sample’ files. It’s important to note that ‘samples’ here include all mzML (or mzXML) files except blanks, including control samples, as they might be also influenced by blank features.

- **Identify the Attribute Column:** The user will first be prompted to enter the index number of the attribute containing information about samples and blanks. Here, it is ‘`ATTRIBUTE_Sample_Type`’.

- **Display Unique Groups:** The unique groups within the chosen attribute column will be displayed. For example, in our dataset, it will show ‘Blank’ and ‘Sample’. However, your dataset might include various groups, such as ‘Blank’, ‘Samples’, ‘Control’, etc.

- **Select the Blank group:** Next, the user will be prompted to enter the index number corresponding to the blank group. If there are multiple groups representing blanks (e.g., Blank, PPL_Blank), their index numbers should be entered, separated by commas.

- **Select the Sample group:** Similarly, the user will be asked to enter the index number(s) for the sample level. If the dataset includes multiple groups for samples (e.g., Sample, Control), the corresponding index numbers should be entered, separated by commas.

- **Subset the Data:** Using the information provided, the metadata (`` `new_md` ``) will be subsetted into `` `md_Blank` `` and `` `md_Samples` ``. The corresponding feature quantification tables will be obtained and named `` `Blank` `` and `` `Samples` ``, respectively.

22| **Define Cutoff for Blank Feature Removal** *(User Input Required)* In this step, the user will need to set a cutoff value within the range of 0 to 1, with a recommended range of 0.1 to 0.3. This value will determine which features are considered to be artifacts of the blank and thus removed from the dataset. The next step will explain how the features exceeding this cutoff are identified and eliminated.

<CRITICAL STEP> Lowering the cutoff to 0.1 demands a greater contribution from the sample (90%) and limits the blank’s contribution to 10%. Raising the cutoff leads to fewer background features being identified and more analyte features being observed. Conversely, lowering the cutoff is more stringent and removes more features.

23| **Perform Blank Removal.** Calculate the blank’s contribution to each feature and eliminate those exceeding the user-defined cutoff. This is achieved by:

- Compute the mean value for each feature within the dataframes (`Blank`) and (`Samples`). This step calculates two mean values for each feature, one for blanks and one for samples. These averages are stored in a new dataframe called `Avg_ft` under the columns 'Avg_blank' and 'Avg_samples'.
- Compute the ratio of the average blank contribution to the average sample contribution for each feature.
- Generate a binary mask where entries corresponding to ratios above the user-defined cutoff are marked as 1 (TRUE), and all others are set to 0 (FALSE). This mask helps in identifying which features are significantly present in blanks as compared to samples.
- Retain only the features associated with 0s in the binary mask. Features with a ratio exceeding the cutoff (marked as 1) are considered artifacts from the blanks and are thus removed. Conversely, if the feature intensity is significantly higher in samples than in blanks, it is deemed a true feature from the samples and is retained (marked as 0).
- The final table, free from blank artifacts, is named `blk_rem`, and its corresponding metadata is `md_Samples`.

The final output is the `blk_rem` table, which excludes background or noise features. Information on the total number of features, the number of background/noise features, and the number of features after noise exclusion are also displayed. The code block used in the notebook for this **step 23** is shown in Figure 8.

Box 4 - Blank Removal

To obtain reliable and meaningful LC-MS/MS metabolomics data, it is crucial to integrate Quality Control (QC) samples and blanks throughout the measurement process, which can facilitate blank removal. This step is critical to eliminate background noise such as signals from plasticizers, solvent impurities, or sample clean-up reagents, along with cumulative carryover contamination[165,170]. Therefore, incorporating blanks during sample collection, preparation, and measurement is vital for identifying and eliminating these interferences, as detailed in **Box 1**.

Removing these non-informative features improves the quality and interpretability of the data by reducing the dataset's dimension and minimizing false correlations. Nevertheless, one should be aware of potential challenges, such as system deconditioning, which can lead to systematic

variations in the metabolomic profiles. To counteract this, careful placement of blanks within a sample batch is advised to minimize such artifacts and to maintain data integrity[170].

One of the existing methods to remove the non-informative features is creating a molecular network using the online platform, the global natural product social molecular networking (GNPS), and visualizing the network in Cytoscape. While this process is reliable, it is tedious and requires users to manually remove blank and media nodes from the molecular network.[7]. There is also the 'msPurity' R package from Lawson *et al.*[171] with a function called "SubtractMZ" to perform blank removal. Data-adaptive filtering methods have also been suggested to remove features from blanks and low abundant features from samples with undetected values[172].

Another popular feature filtering method is based on the Coefficient of Variance (CV). Also referred to as relative standard deviation (RSD) is a measure of statistical dispersion, calculated as the ratio of the standard deviation to the mean[173]. When pooled QC samples are integrated throughout a study, CV can be used to assess the stability of each feature. As a general rule of thumb, features exhibiting a CV greater than 30% are typically excluded, though the threshold is more stringent (at 20%) for FDA studies. However, it's essential to approach CV filtering with caution. Schiffman *et al.*[172] have highlighted the potential limitations of this method, pointing out that CV primarily evaluates variability across technical replicates without giving weight to biologically meaningful variability across different subjects[172]. Consequently, while CV filtering might be apt for studies focusing on homogenous samples like plasma or *Escherichia coli* cells, it might not be the best fit for diverse sample sets such as environmental or fecal samples.

The dispersion ratio or D-ratio, introduced by Broadhurst *et al*.[170], offers an alternative to a simple CV cut-off by comparing both technical and biological variance. It is calculated by dividing technical variance by the total variance, which includes both technical and biological variances. Therefore, for any feature, a 0% D-ratio signifies that the variance is entirely biological, whereas a 100% D-ratio denotes complete technical noise, without any biological information. So, when assessing D-ratios for metabolites, it is better to retain the ones with D-ratios closer to zero[170].

Imputation

<CRITICAL> Many feature extraction software programs, such as MZmine 3, often generate tables with missing values denoted as "NA", "NaN" or 0. This means that for several *m/z* and RT traces in a given sample, there may not be a peak detected and therefore no value

is available[160]. However, many statistical approaches, such as Principal Component Analysis (PCA), require numerical values for each observation. Hence, these features with missing values need to be removed or imputed. In this section, we handle the zero values in our blank-removed feature quantification table. Refer to **Box 5** for more information on imputation strategies and see the accompanied illustration for a graphical overview of imputation.

<CRITICAL> Imputation is not advised if one plans to execute a PCoA using the Jaccard distance since Jaccard transforms data into binary (0 and 1). Without zeros, it results in a table full of ones.

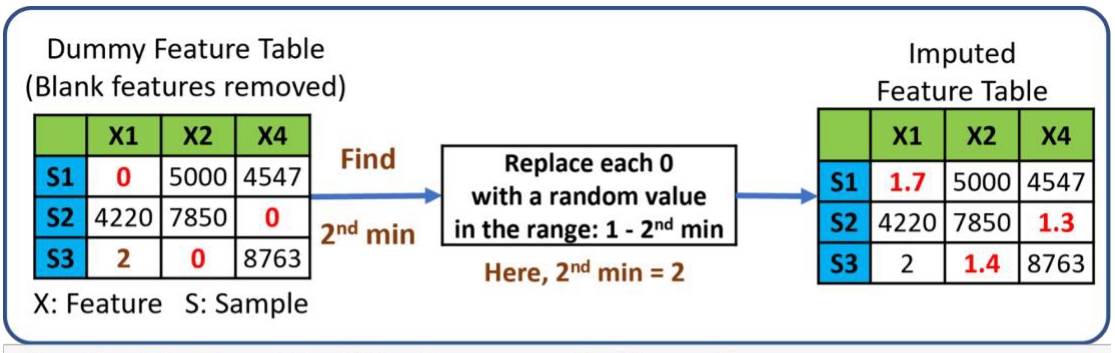
24| **Analyzing the frequency distribution of relative intensities**. Examine the distribution frequency of the relative intensities in the feature quantification table by creating a histogram. This reveals any notable gaps in the range of values, such as a large number of zeros or a lack of values within a particular range. In our example, we observed many zeros and no values in the range of 0 to 100. The smallest non-zero value in our table was between 100 and 1000.

25| **Replacing zeros with random values.** The program replaces all zero values in the dataset with the randomly generated number between 1 and the smallest non-zero value in our blank-removed table. This process, known as imputation, fills in the gaps in our data with plausible values, which can improve subsequent analyses. For the rationale behind this approach, see the first point in **Box 5**.

Box 5 - Imputation strategies

The appropriate imputation strategy depends on the nature of the missing values:

1. Below the Limit of Detection (LOD): If a value is missing because the corresponding molecule was below the analytical method's LOD, consider replacing missing values with a low value, ensuring it does not artificially lower the variance[174]. Our imputation method corresponds to this scenario.




```r
#creating bins from -1 to 10^10 using sequence function seq()
bins <- c(-1,0,(1 * 10^(seq(0,10,1))))

#cut function cuts the give table into its appropriate bins
scores_gapfilled <- cut(as.matrix(blk_rem),bins, labels = c('0','1',paste("1E",1:10,sep="")))

#transform function convert the tables into a column format: easy for visualization
FreqTable <- transform(table(scores_gapfilled)) #contains 2 columns: "scores_x1", "Freq"
FreqTable$Log_Freq <- log(FreqTable$Freq+1) #Log scaling the frequency values
colnames(FreqTable)[1] <- 'Range_Bins' #changing the 1st colname to 'Range Bins'

## GGPLOT2
ggplot(FreqTable, aes(x=Range_Bins, y=Log_Freq)) +
    geom_bar(stat = "identity", position = "dodge", width=0.3) +
    ggtitle(label = "Frequency plot - Gap Filled") +
    xlab("Range") +
    ylab("(Log)Frequency") +
    theme(plot.title = element_text(hjust = 0.5))

Cutoff_LOD <- round(min(blk_rem[blk_rem > 0]))
print(paste0("The limit of detection (LOD) is: ",Cutoff_LOD))
```

```
[1] "The limit of detection (LOD) is: 892"
```

```r
# by setting a seed, we generate the same set of random number all the time
set.seed(141222)

imp <- blk_rem

for (i in 1:ncol(imp)) {
  imp[,i] <- ifelse(imp[,i] == 0,
                    round(runif(nrow(imp), min = 1, max = Cutoff_LOD), 1),
                    imp[,i])
}
```

2. Sample Processing or Feature Extraction Artifacts: Missing values due to sample processing or extraction issues, like ion suppression or retention time shifts, may prevent accurate peak detection for certain m/z and RT traces. Furthermore, matrix effects may complicate metabolite quantification, leading to data gaps despite the presence of peaks in the raw data[160,175]. For comprehensive statistical analysis, one can consider imputing missing values with those similar to values detected in other samples. Here, machine learning methods like k-nearest neighbor (KNN) or random forest (RF) can be useful. KNN fills in multiple missing values by

identifying the k nearest data points to a given point[175]. Similarly, RF can iteratively impute missing values using a proximity matrix derived from RF classification across all metabolites[160].
However, caution is advised with imputation as it introduces data points where none existed, potentially skewing results. One needs to be aware of the risks and limitations of imputation in statistical analysis.
A visual representation of the imputation algorithm complemented by screenshots of associated R code snippets is shown below.

Normalization

<CRITICAL> Sample normalization aims to eliminate systematic bias via adjusting variations across samples[176]. In our pipeline, we show two normalization methods: Total Ion current (TIC) normalization and Probabilistic Quotient Normalization (PQN), implemented using the KODAMA library in our R Notebook. Therefore, we begin this section by installing the KODAMA package. We recommend that users run both normalization methods and scaling methods (**steps 26 to 29**), but they can choose either method for further analysis in **Step 30**. Additional information about normalization, including various methods, and guidelines for selecting the most suitable method for a given dataset, is provided in the accompanying **Box 6**. For a graphical view of the provided normalization methods, see the accompanying illustration in **Box 6**.

26| Install the KODAMA package.

27| **Run the Total Ion Current (TIC) Normalization method.** This step does not require any user input other than running the code cell. In TIC normalization, also known as total sum normalization, every feature within a sample is normalized relative to the area of the TIC chromatogram[239]. This involves dividing each feature by the sum of the peak areas of all features within a sample. The normalization function from the KODAMA[214] (v2.4) package performs row-wise sum operations; for this to work, the sample names are arranged in rows and their features in columns.

28| **Run the Probabilistic Quotient Normalization (PQN) method.** The user can simply execute this code cell without needing to input anything. PQN is another method performed on the imputed table, resulting in a PQN-normalized table with features in columns and samples in rows.

PQN is based on the comparison of a ‘test’ spectrum (the individual sample to be normalized) with a ‘reference’ or ‘control’ spectrum. The steps involved in PQN are as follows[180]:

- **Normalization of Test Spectrum:** The test spectrum is first normalized, typically using a sum normalization technique like TIC.
- **Selection of Control Spectrum:** The control spectrum acts as a standard for comparison. It could be a pre-determined standard obtained from a database or calculated as the mean or median spectrum from all samples or quality control (QC) samples.
- **Calculation of Quotients:** For each sample, quotients are calculated between the features in the test spectrum and the corresponding features in the control spectrum. This step results in a median quotient spectrum for each sample.
- **Normalization by Median Quotient Spectrum:** Each test spectrum is then normalized by dividing it by its corresponding median quotient spectrum. This process scales the test spectrum values relative to the control spectrum, ensuring an equal basis for comparison across all samples.

Box 6 - Normalization

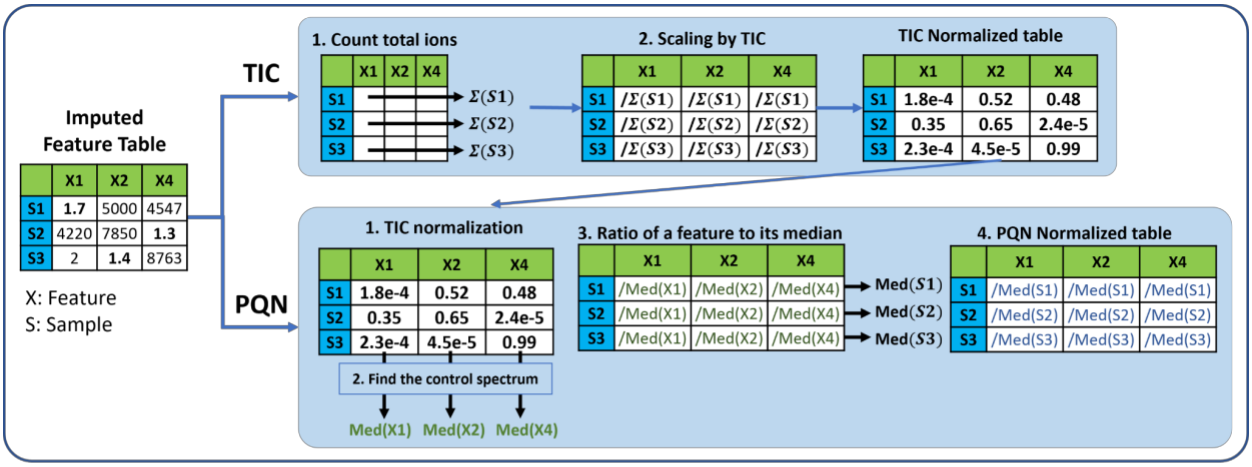
Normalization of metabolomics data can rely on either chemical or mathematical strategies. The chemical method, using internal standards and quality controls, is popular in targeted analysis as it effectively balances metabolite concentrations across sample sets and batches. However, for non-targeted metabolomics, mathematical approaches are more popular[176,177]. There are several mathematical normalization methods, each with its strengths and limitations. The selection of a normalization method depends on the specific conditions and requirements of your dataset:

1. **Unit Normalization**[178] **and TIC Normalization:** Simple and computationally efficient methods useful for large datasets. They equalize the total sum of signal intensities across each sample. They assume that the abundance of most features does not change significantly across different samples or experimental conditions and their effectiveness decreases with large global changes in metabolite levels (e.g., due to differences in metabolite level such as healthy versus diseased, sample preparation, or instrument sensitivity). TIC normalization might over-correct disease samples with lower intensity reducing the differences between healthy and diseased conditions[179].
2. **PQN**[180]**:** Recommended when significant size effects are present or when internal normalization disrupts relative peak information[176]. Among several LC/MS-based normalization methods, including Contrast Normalization, Cubic Splines, and Cyclic

Loess, PQN has been identified as the best performer in reducing sample-to-sample variations[177].

3. **Common Components and Specific Weights Analysis**[181] **(CCSWA)**: A viable alternative when QC and sample data differ.

A graphical representation of Total-ion-current (TIC) and Probabilistic Quotient normalization (PQN) methods, accompanied by corresponding R code snippets are shown below.




```
norm_TIC <- normalization(imp, #performing normalization on transformed imputed data
                          method = "sum")$newXtrain
head(norm_TIC,n=3)
dim(norm_TIC)
print(paste('No.of NA values in Normalized data:',sum(is.na(norm_TIC)== T)))
```

```
norm_pqn <- normalization(imp,
                          method = "pqn")$newXtrain
head(norm_pqn,n=3)
dim(norm_pqn)
print(paste('No.of NA values in Normalized data:',sum(is.na(norm_pqn)== T)))
```

Scaling

<CRITICAL> Scaling methods in metabolomics aim to adjust the range of peak abundances between features[176]. This is done by normalizing the intensities of each feature by a scaling factor, effectively adjusting for fold differences between features[182]. Additional information on scaling factors can be found in **Box 7** along with the graphical representation of scaling.

29| **Run the Center-Scaling method.** Simply run the code cell. The program applies center-scaling to the imputed data. This allows for a consistent spread of the data, accounting for differences in offset between high and low-abundant features.

In R, the scale function offers different options for centering and scaling data:

- When center = TRUE, centering is achieved by subtracting the column means (excluding NAs) of the data from their respective columns (each column referring to a feature). Centering ensures that the fluctuations in the data are centered around zero instead of the mean of the metabolite concentrations[182].
- If center = TRUE and scale = TRUE: then scaling is performed by dividing the centered columns by their standard deviations.
- If center = FALSE and scale = TRUE: scaling is done by dividing each column by its root mean square (RMS).
- If scale = FALSE, no scaling is performed.

<CRITICAL STEP> Since scaling introduces negative values, trying a PCoA with the Bray-Curtis difference on scaled data will trigger an error.

30| **Choosing data for further analysis** *(User Input Required).* Upon executing this step, an overview table is generated automatically, offering a list of the dataframes produced during each phase of data cleanup along with its respective metadata tables. This includes stages like the initial raw data (Raw Data), post-blank removal data (Blank Removed Data), post-imputation data (Imputed Data), and various normalization stages (TIC Normalized, PQN Normalized, Scaled Data).

To proceed after the overview table is generated, the user should select a dataset of interest by entering the corresponding index number. The chosen dataset will be stored under the `cleaned_data` variable and the corresponding metadata will be taken under the `metadata` variable. These dataframes will be used in subsequent univariate and multivariate analytical steps. This allows the user to:

- **Explore Multiple Datasets:** Easily switch between datasets to examine the effects of different processing steps.
- **Tailor Analyses to Dataset Characteristics**

Two types of analyses are meaningful:

- TIC normalized data is apt for some univariate statistical tests, especially when analyzing the relative abundance of specific features or metabolites across samples without the comparison being skewed by samples that just have overall higher or lower intensities. Also, when using normalized data for multivariate techniques like PCA, it is important to ensure that a few dominant features do not skew the overall results.
- Using scaled data in multivariate techniques like PCA prevents high variance features from dominating. Additionally, techniques relying on distance measures, like k-means or k-nearest neighbors, benefit from scaled data to ensure uniform feature influence.

However, it is important to note:

- Imputation is not advised if one plans to execute a PCoA using the Jaccard distance since Jaccard transforms data into binary (0 and 1). Without zeros, it results in a table full of ones.
- Since scaling introduces negative values, trying a PCoA with the Bray-Curtis difference on scaled data will trigger an error.

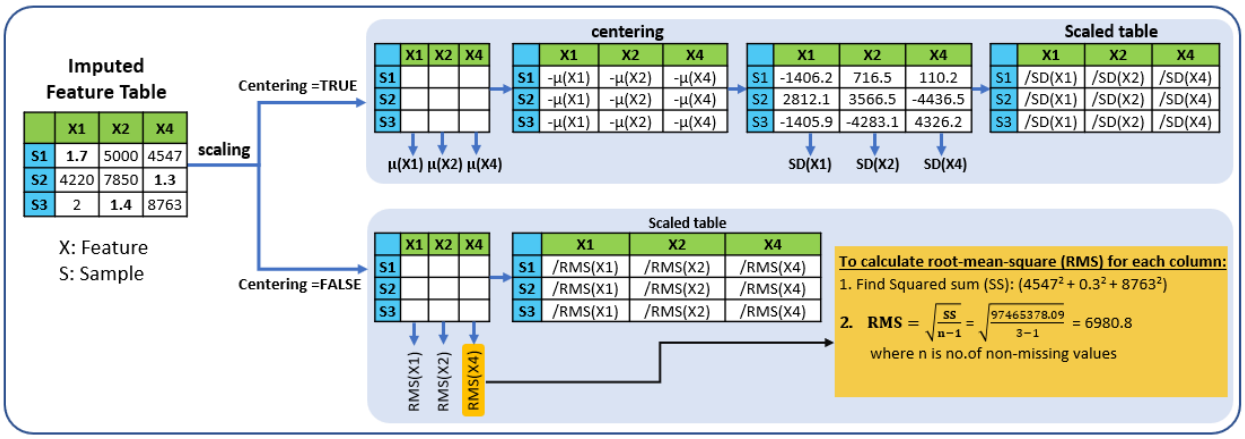
For this tutorial, we will use the `scaled_data` as our `cleaned_data` and the respective `metadata` variable is `md_Samples`. However, users are encouraged to experiment with different datasets.

Box 7 – Scaling

Scaling methods can be categorized into two subclasses based on the scaling factor used[182].

1. **Using data dispersion methods**, such as standard deviation (SD), for scaling: Examples: Autoscaling[183] and Pareto scaling[184]. Autoscaling ensures equal variance (such as SD=1) for each variable, while Pareto scaling uses the square root of SD as the scaling factor.
2. **Using size measures**, such as the mean, for scaling:
   Examples: Level scaling and Poisson scaling. Level scaling converts metabolite concentration changes relative to the mean concentration, while Poisson scaling scales each feature by the square root of the mean[182,185].

A graphical representation of the auto-scaling method, both centered and non-centered approaches, accompanied by corresponding R code snippet from the R notebook using the 'scale' function is shown below. Centering adjusts the data such as the fluctuations are centered around zero rather than the mean of the metabolite concentrations[182]




```
Imp_s <- scale(imp, center = T, scale = T) # center and scale data
head(Imp_s, n=2)
```

2.4.3 Multivariate Statistics: 🟠 Timing 50-60 mins

<CRITICAL> In this section, we will describe 4 approaches to mulitvariate analysis. We expect that users will choose one or more of these based on the information provided in the introduction.

- PCoA with PERMANOVA
- HCA
- Heatmaps
- Supervised Classification: Random Forest

31| To start the multivariate analysis, install and load the necessary R packages for this section: BiocManager[240] (v1.30.9), ComplexHeatmap[217] (v2.10.0), dendextend[218] (v1.17.1), NbClust[219] (v3.0.1), ggsci[223] (v3.0.0), and cowplot[224] (v1.1.1). This process takes 5-10 minutes.

PCoA with PERMANOVA: PCoA

<CRITICAL> Refer to **Box 8** for a detailed overview of PCoA, including a graphical illustration of the PCoA process and corresponding code snippets used to perform the following steps in the notebook.

32| **Prepare Data.** This step makes sure that the metadata (`metadata`) and the feature quantification table (`cleaned_data`) are in the same order. Also, we verify that the

sample names (row names) in both data tables are identical and in the same order using the `identical()` function. It should return TRUE.

33| **Calculate Pairwise Distances and Perform PCoA.** No user input is required. We calculate pairwise Euclidean distances across all samples in the feature quantification table using the `vegdist()` function from the 'vegan' package[99]

- Store the resulting distance or dissimilarity matrix as 'distm'.
- Apply the `cmdscale()` function from the base R 'stats' package to perform MDS on the distance matrix 'distm', considering 10 PCos (k=10).

<CRITICAL STEP> The `vegdist()` function offers various methods such as "manhattan", "euclidean", "canberra", "bray", "jaccard", "gower", "binomial", "chisq" for distance calculation. Using Euclidean distance for PCoA is equivalent to performing PCA. However, using `vegdist("euclidean")` and `cmdscale()` cannot provide loadings information. For a comprehensive PCA with both loadings and scores, use the `prcomp()` function such as `` `pca_result <- prcomp(cleaned_data, center = FALSE, scale. = FALSE)` ``. Since the `cleaned_data` we use is already centered and scaled, we set these parameters to FALSE. For loadings and PC scores, you can access `` `pca_result$rotation` `` and `` `pca_result$x` `` respectively.

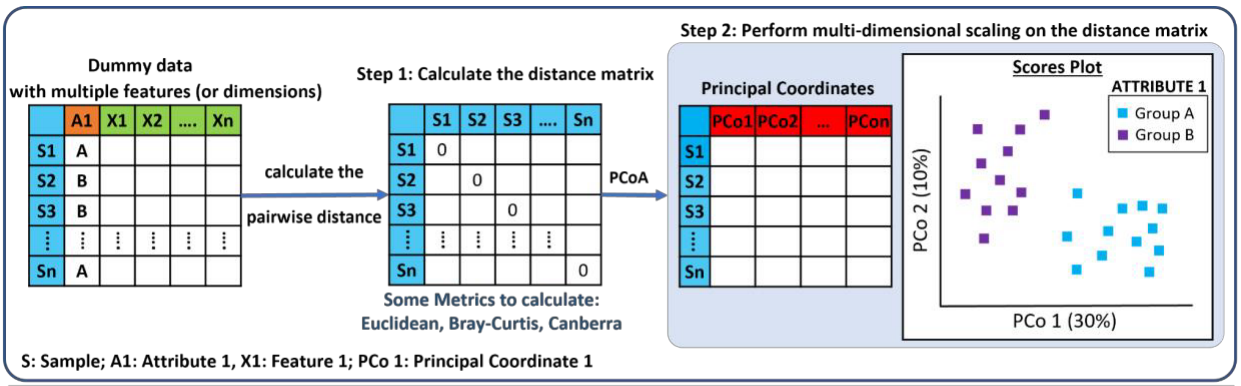
34| **Analyze PCoA Results.** No user input is required in this code cell. The user can examine the list generated by the `cmdscale()` function, which includes the following elements:

- 'points' (`PcoA$points`) represents the data matrix with the given PCos
- 'eig' (`PcoA$eig`) indicates the eigenvalues computed for the PCos, which describe the variance explained by each PCo.

35| **Plot PCoA Scores** *(User Input Required)* Using the `ggplot2` library, create a PCoA Scores Plot. Here, the samples are color-coded based on the 'ATTRIBUTE_Month' attribute. To view the sample distribution of different attributes, the user can simply adjust the line: `interested_attribute_pcoa = 'ATTRIBUTE_Month'.` Importantly, the aspect ratio of the plot's axes is maintained to ensure accurate representation, in line with recommendations by Nguyen and Holmes[126].

Box 8 - Principal Coordinate Analysis (PCoA)

PCoA offers an advantage over PCA by allowing various distance metrics beyond the Euclidean distance. This flexibility provides different insights into the data pattern based on the chosen dissimilarity measure. For example, when working with categorical data and sparse matrices containing numerous zeros, distance metrics such as Hamming distance and Jaccard distance outperform the Euclidean distance[128,186,187]. Akin to phylogenetic distance measures such as UniFrac distance[188] used in the microbial ecology field, chemical distance matrices are emerging that make use of cosine MS/MS similarity between features[189] or chemical similarity derived from CSI:FingerID[190].




```
distm <- vegdist(Imp_s, method = "euclidean") #compute distance

#computing multi-dimensional scaling on distance matrix
PcoA <- cmdscale(distm,
                 k = 10, # k=10 gets the first 10 Principal coordinates or PCOs
                 eig = T,
                 add = T)

PcoA_points <- as.data.frame(PcoA$points) #getting the PCOs into dataframe
variance <- round(PcoA$eig*100/sum(PcoA$eig),1) # getting the variance explained by each PCo
names(PcoA_points)[1:10] <- paste0('PCoA', seq(1,10))  #naming the 10 PCos

head(PcoA_points,2) # looking at the 10 PCOs
```

A data.frame: 6 × 10

	PCoA1	PCoA2	PCoA3	PCoA4	PCoA5	PCoA6	PCoA7	PCoA8	PCoA9	PCoA10
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
SD_01-2018_1_a.mzXML	-49.871362	-13.60140	-21.40328	-18.0611465	-3.155365	40.2292107	-22.4594986	14.1888608	29.504792	9.1637140
SD_01-2018_1_b.mzXML	-49.430434	-15.95842	-20.30956	-19.4388300	-3.843483	40.8474155	-22.5707000	11.4764404	31.136861	9.2803551

While PCoA effectively reveals chemical trends among samples by working with different distance matrices, it cannot provide direct information about the relationship between features and principal coordinates, unlike PCA which offers 'loadings' information[191]. To discern associated features in such contexts, it is recommended to complement PCoA with other methods like a Heatmap overview, Random Forest analysis, or any of the univariate techniques discussed in this protocol.

In addition, to assess the impact of a specific feature on the dispersion of samples along a particular PCoA axis, an indirect analysis can be performed. This involves correlating or regressing the PCoA values of the samples with the corresponding sample scores of the variable of interest[192]. For instance, in our case, to evaluate the influence of Feature 1 on PCo1, we can create a scatter plot by plotting the original values of Feature 1 (sample scores) for all samples against the PCo1 values for all samples. The points on the plot can be colored based on the sampling period. By examining any trends or correlations in the plot, we can observe how the diversity of samples changed during the sampling period.The diagram below illustrates the process of transforming feature quantification tables into score plots by calculating distance matrices, and plotting principal coordinates. The associated code demonstrates multidimensional scaling using Euclidean distance. Notably, using Euclidean with PCoA is the same as performing PCA; however, the users can adjust to other metrics, like Canberra.

PERMANOVA: Permutational multivariate ANOVA

<CRITICAL> Before performing PERMANOVA, it is important to validate the homogeneity of group dispersions, often termed 'Homoscedasticity'. This test ensures that each group exhibits approximately equal variability. Violation of this assumption might inflate the risk of Type I errors (false positives)[193,216]. If the group dispersions are homogenous, you can proceed with PERMANOVA with greater confidence. However, disparate dispersions require a more cautious interpretation of PERMANOVA results, given their higher susceptibility to Type I errors. In such cases, exploring alternative distance measures, data transformations, or delving into potential biological reasons for the dispersion differences might offer a more comprehensive analysis. To know more about multivariate dispersions, see **Box 9**. For a visual representation of assessing multivariate dispersion and conducting the PERMANOVA analysis in R, refer to the accompanying code snippets in **Box 9**.

36| **Test for Homoscedasticity** *(User Input Required)*

- Specify the attribute group for assessing group dispersions. Since we are looking for group dispersions, it is important to select a categorical metadata column (for example, '`ATTRIBUTE_Month`') and avoid choosing continuous attributes, such as '`ATTRIBUTE_Injection_order`'.

- Similar to **Step 33**, compute a distance matrix ('distm') using the feature quantification table and the selected attribute. For simplicity, we use the Euclidean distance in this instance.
- Using the `betadisper()` function from the vegan package, evaluate group dispersion against the chosen attribute group.
- Visualise the dispersion model to offer a clearer perspective.
- Perform ANOVA on the dispersion model. A significant p-value ($P < 0.05$) indicates a violation of PERMANOVA's foundational assumptions. Conversely, a non-significant result suggests that PERMANOVA is a suitable choice for the given attribute.

<CRITICAL STEP> The resulting p-value for '`ATTRIBUTE_Month`' indicates the presence of group dispersions, in this example the group dispersions are among different months. This violates the PERMANOVA assumption. When PERMANOVA is performed for this attribute, the PERMANOVA results require a more cautious interpretation.

37| **Conduct the PERMANOVA Test**. To do this:

- Use the `adonis2()` function from the `vegan` package to conduct a PERMANOVA test. The `adonis2` function allows for the analysis and partitioning of sums of squares using dissimilarity measures.
- Apply the `adonis2` function on the dissimilarity matrix ('distm') and the previously chosen metadata column '`ATTRIBUTE_Month`'. This helps in investigating if there are significant differences among the samples collected during three different months.
- Interpret the resulting p-value. In our case, we obtained a p-value of 0.001, indicating a significant difference between the samples.

38| **Define a Function for Streamlined Analysis**. To facilitate quicker analysis and avoid rewriting from **Step 33** to **Step 37** for testing different parameters, we defined a function, `plotPCoA()`. This function performs a principal coordinates analysis (PCoA) using a chosen distance metric, calculates a PERMANOVA, and plots the results in a 2-D graph. Additionally, it assesses group dispersion prior to the PERMANOVA calculation and displays the significant result in the resulting plot as well.

The function has the following parameters:

- `ft` refers to the desired feature quantification table.
- `md` refers to the respective metadata.

- `distmetric` is the distance metric of choice.
- `category_permanova` is the desired metadata group for PERMANOVA calculation.
- `pcoa_category_type` indicates whether the group type is categorical or continuous.
- `category_pcoa_colors` specifies the metadata attribute for coloring the samples.
- `cols` are the desired colors for the groups.
- `title` is the title of the plot.

Furthermore, we have created a `custom_palette` of 22 colorblind-friendly colors for coloring the groups consistently across the protocol. This color palette is used in **Steps 47, 56, 64, 68, 71, 76 and 80**. Users can customize this palette in **Step 38** according to their preferences.

Additionally, we have created another simple custom function `save_as_svg()`, to store plots in SVG format utilizing the `svglite` function. This custom function can be used as `save_as_svg(filename, desired_plot, plot_width, plot_height, plot_background)`. Throughout the notebook, you will observe this function being used after each plot creation to save the visualizations.

39| **Apply `plotPCoA()` function on different dataframes.** *(User Input Required)* In this step, the user can specify the variables as mentioned in the previous step. Here is an example of how to use the `plotPCoA()` function:

```
plotPCoA(
       ft = cleaned_data,
       md = metadata,
       distmetric = "euclidean",
       category_permanova = "ATTRIBUTE_Month",
       pcoa_category_type = 'categorical',
       category_pcoa_colors = "ATTRIBUTE_Month",
       cols = c('orange','darkgreen','red','blue','black'),
       title = 'Principal coordinates plot')
```

40| **Get PCoA plots after each data cleanup step** *(User Input Required).* Specify parameters such as the distance metric, attribute for PERMANOVA calculation, attribute

to color the PCoA scores, the category of the chosen attribute, similar to the previous plotPCoA step. These inputs will be taken to produce an overview of PCoA plots for all steps of data cleanup.

Box 9 - Dispersion Analysis

In the case of balanced sample sizes across groups, PERMANOVA identifies differences in group centroids, thus reflecting shifts in the multivariate distribution of sample units within the chosen resemblance space. Hence, the type of dissimilarity measure you choose is crucial. For example, unlike Euclidean distance, measures like Jaccard or Bray-Curtis highlight the similarity in species composition and do not focus on the central tendency such as the mean-variance relationship. On the other hand, PERMDISP is specifically tailored to detect variations in multivariate dispersions. Therefore, when analyzing your data, use PERMANOVA to understand group centroid shifts and PERMDISP to evaluate dispersion differences[193].

The following is the R code snippet for testing multivariate dispersions within the 'group', specifically referencing the 'ATTRIBUTE_Month' column (Dec, Jan, Oct) from the metadata.

```
dispersion_model <- betadisper(distm, group)
disp <- anova(dispersion_model)
disp["significant"]  <- ifelse(disp$`Pr(>F)`<0.05, "Significant", "Non-significant")
disp
```

A anova: 2 × 6

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	significant
	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>
Groups	2	37529.38	18764.6917	31.56342	1.890734e-12	Significant
Residuals	177	105227.82	594.5075	NA	NA	NA

Similarly, the R code snippet for executing PERMANOVA to analyze variations between the aforementioned groups in the previous illustration is as follows:

```
adonres <- adonis2(distm ~ group)
adonres
```

A anova.cca: 3 × 5

	Df	SumOfSqs	R2	F	Pr(>F)
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
group	2	385128.9	0.236643	27.43527	0.001
Residual	177	1242339.1	0.763357	NA	NA
Total	179	1627468.0	1.000000	NA	NA

Hierarchical Cluster Analysis

41| **Set the Plot Size.** It is important to define the size of the output plot, as dendrograms are typically larger in plot size. Adjust the plot size accordingly to ensure a clear and comprehensible visualization.

42| **Execute HCA.** No user input required. Here, we use the `hclust()` function from the 'stats' package to perform HCA. The function is applied to the distance matrix 'distm', calculated based on the feature quantification table ('cleaned_data') using a specified distance metric (e.g., Euclidean, Canberra). The 'method' argument in `hclust()` denotes the linkage method used for measuring the distance between clusters (e.g., complete, single, average). We use the default 'complete' method, which calculates the maximum distance between clusters before combining them. Once HCA is completed, a dendrogram is generated as shown in **Figure 9**. This dendrogram shows split or merge distances as 'height' along the y-axis, providing a visual representation of the cluster formation.

43| **Cut the Dendrogram** *(User Input - Optional).* Similar to k-means clustering, which seeks to establish k clusters with minimum within-cluster variation, we can cut the dendrogram into a specified number of clusters using the `cutree()` function. However, we need to initialize the clustering with random k clusters. For our sample dataset, we define `k=4` with the `cutree()` function, to create four clusters. The user can change the number of clusters. Refer to **Step 45** for more details on choosing the number of clusters.

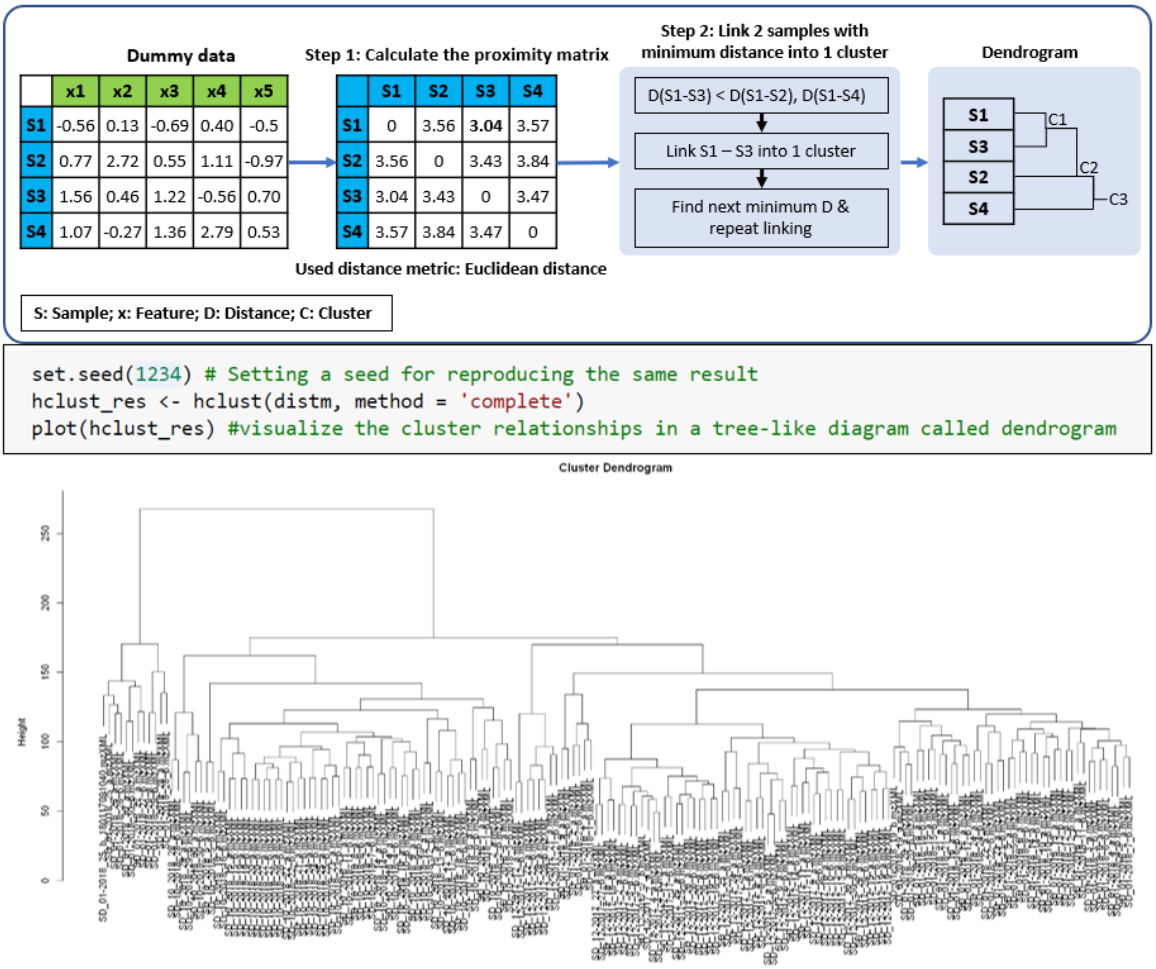


Figure 9: Dendrogram Generation and Analysis: The figure illustrates a dendrogram, as a result of applying HCA to a feature quantification table (e.g., 'cleaned_data'). From this data, a proximity matrix (or the distance matrix) is calculated (see **steps 33 and 36**), which subsequently guides the dendrogram creation. Accompanying the illustration is the related code for the cluster generation and dendrogram visualization. The distance matrix 'distm' is calculated via Euclidean distance in **step 36**, though alternative metrics can be chosen by the user. The resultant dendrogram is displayed, initially partitioning samples into two primary clusters: a smaller cluster from a subset of samples (corresponding to samples from January in our example data) and a larger subsequent cluster. Distinct sub-clusters within these main clusters are also discernible.

44| **Color the Dendrogram**. Finally, we can extract the cluster allocation information and color the dendrogram according to the clusters. For our data, the dendrogram suggests two main splits, resulting in four distinct clusters.

45| **Determine the Optimal Number of Clusters**. Here, we use heuristic methods similar to those applied in k-means clustering to determine the optimal number of clusters. For this

purpose, we use the Elbow approach and average silhouette method using the `fviz_nbclust()` function from the 'factoextra' package.

- The Elbow method calculates the total within-cluster sum of squares (WSS) for an increasing number of clusters. WSS signifies the sum of distances between data points and their corresponding centroids within each cluster. Lower WSS values indicate within-cluster variation[137].
- The resulting Elbow plot presents the WSS on the y-axis and the number of clusters on the x-axis. Lower WSS values suggest minimum within-cluster variation and better clustering. However, the 'elbow' point is considered an indicator of the optimal number of clusters, as further cluster additions do not significantly improve the clustering or decrease the WSS. For our example data, this method suggests 3 or 4 clusters. However, defining the 'elbow' can be subjective.
- An alternate approach is the average silhouette method, which assesses clustering quality by determining how well each data point fits within its assigned cluster. In our case, this method proposes two primary clusters.

Both the Elbow and Silhouette methods provide global insights without learning from the data, given their unsupervised nature. But, there are more sophisticated techniques like the gap-statistic which refines the heuristic concepts behind the Elbow and Silhouette techniques and uses a statistical procedure to estimate the optimal cluster count[137].

<CRITICAL STEP> All of the heuristic methods in **Step 45** serve as guidelines rather than definitive answers. In practice, in **Step 43**, users might choose cluster numbers based on context, for example, in our case with seven sample areas, opting for seven clusters can be insightful. Later, one can check whether these clusters correspond to known sample groups. While context-based clustering might provide initial insights, it is important to avoid biases that could arise from relying heavily on pre-existing knowledge or expectations. An objective analysis where the data substantiate the clustering choice is crucial to ensure that the results are reflective of true patterns in the data, rather than what one expects or wants to see.

Heatmaps

<CRITICAL> Heatmaps are generally used to visualize complex data or discern patterns across a high-dimensional dataset. They are commonly used in bioinformatics[241], particularly in

gene expression analysis and visualizing genomic datasets, owing to their ability to effectively represent thousands of data points[242]. This makes them equally suitable for mass spectrometry-based metabolomic experiments. Heatmaps are efficient in pattern recognition due to their color-coded matrix elements and adjacent dendrograms, which indicate functional relationships between variables and samples[138]. To see the resulting heatmap generated by the R code in the Notebook, refer to **Figure 10**. In this section, we will show how to incorporate hierarchical clustering into our heatmap.

46| **Preparing Metadata for Heatmap** *(User Input Required).* To start with, determine which metadata columns or attributes will be used to decorate the heatmap. In our case, we specified the following attributes: '`ATTRIBUTE_Year`', '`ATTRIBUTE_Month`', and '`ATTRIBUTE_Sample_Area`'. The user can select any number of attribute columns from their metadata as they see fit for the heatmap. A new dataframe is created comprising the chosen metadata.

47| **Generate annotations for Heatmap** *(User Input - Optional).* For distinct visualization, this step assigns unique colors to each category within chosen attributes from the previous step. We have created a function `generate_colors()`, which utilizes a predefined color-blind-friendly palette (from **Step 38**) to assign colors to these unique groups. Users can modify these colors if desired in **Step 38**. After assigning colors to the subset dataframe, we use this information to decorate the heatmap with annotations from the `HeatmapAnnotation()` function in the 'ComplexHeatmap' package.

48| **Creating the Heatmap** *(User Input - Optional).* To create the heatmap, apply the Heatmap function from the ComplexHeatmap package on the transposed `` `cleaned_data` `` (as previously chosen in **Step 30**). This arranges the features in rows and samples in columns.

- For the heatmap, the color intensity represents the feature intensities, with the intensity scale ranging from 0 (blue) to 1 (dark red), and 0.5 represented as white. This color coding allows for a visual comparison of feature intensity variations across samples.
- The clustering on the y-axis is based on Euclidean distance (`clustering_distance_rows = "euclidean", clustering_distance_columns = "euclidean"`). However, other distance measures such as Manhattan, Minkowski, Canberra, or even Jaccard for binary data, can be chosen based on specific needs.
- The 'complete' linkage method is used for clustering (`clustering_method_rows = "complete",clustering_method_columns = "complete"`).

49| **Refining Data Clustering with k-means.** Further refine data clustering by incorporating the built-in k-means function within the heatmap as parameters for row and column clustering (`row_km = 5, column_km = 4`). To ensure robustness, perform multiple repeats (`row_km_repeats = 100, column_km_repeats = 100`).

50| **Extracting Features from Each Cluster.** With a higher number of features, it is difficult to interpret the clustering or labeling of features on the heatmap. To address this, we extract the features from each cluster into a separate dataframe. This dataframe containing combined feature names (`XFeatureID_*m/z*_RT_GNPS_annotations) and their respective cluster assignments can be saved as a CSV file for further interpretation. For example, one could merge these cluster assignments with the feature quantification table for import into Cytoscape along with the FBMN and use these cluster assignments for coloring slices in node pie charts.

```r
# set the parameters for the type of clustering to perform. You can play with different options
set.seed(1234)
hmap <- Heatmap(
    t(cleaned_data),
    heatmap_legend_param = list(title = "Scaled/centered \n intensity"),
    col = circlize::colorRamp2(c(0, 0.5, 1),
                               colors = c("blue","white",'darkred')),
    show_row_names = FALSE, show_column_names = FALSE,
    cluster_rows = TRUE, cluster_columns = TRUE,
    show_column_dend = TRUE, show_row_dend = TRUE,
    row_dend_reorder = TRUE, column_dend_reorder = TRUE,
    clustering_distance_rows = "euclidean", # you can change the distance here
    clustering_distance_columns = "euclidean", # you can change the distance here
    clustering_method_rows = "complete",
    clustering_method_columns = "complete",
    width = unit(100, "mm"),
    top_annotation = colAnn)

ComplexHeatmap::draw(hmap, heatmap_legend_side="right", annotation_legend_side="right")
```

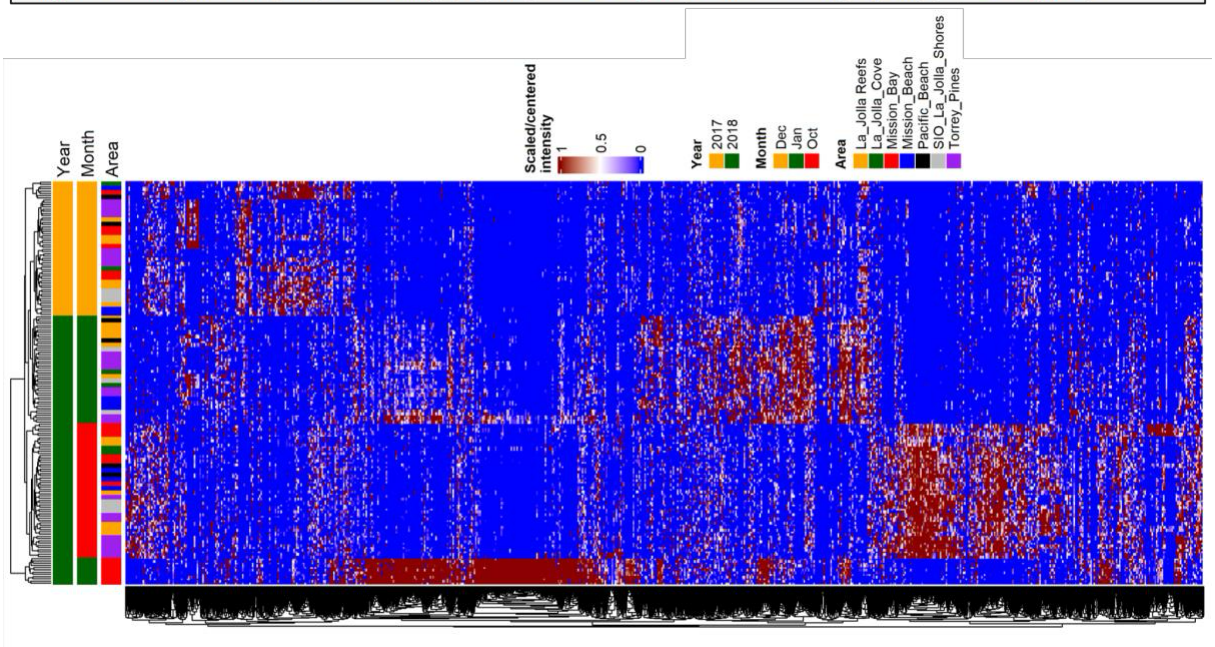


Figure 10: Heatmap Visualization and Construction: This figure presents both the R code snippet used for heatmap creation and the resultant heatmap itself. To facilitate a comprehensive view, the heatmap is oriented horizontally. The feature quantification table used here is the scaled table and feature intensities are color-coded, ranging from blue (0) to red (1). Annotations at the heatmap's top delineate clustering based on variables like year, month, and sample area.

Supervised Classification: RF

51| **Prepare the data for Random Forest.** To do this:

- First, load the `rfpermute` package.

- Start by merging the feature quantification table (in our example, `Imp_s` is chosen as the `cleaned_data` variable) and the corresponding metadata (`md_Samples`) into a dataframe named `cleaned_data_with_md`. This step ensures that the samples are correctly aligned with their corresponding attributes in the metadata, which is essential for the subsequent analyses.

52| **Select the Classification Attribute for Random Forest** *(User Input Required)* Prepare the dataset used for Random Forest classification so it only contains feature intensity information and attribute of interest for classification. Here, we are classifying the samples according to different sample areas ('ATTRIBUTE_Sample_Area'); in this step, the user is prompted to input the index number of the interested attribute to use for the classification.

53| **Balance sample sizes** If the sample size varies among the groups, balance the size of groups by using the `balancedSampsize()` function. This function helps ensure that groups with a larger number of samples do not disproportionately influence the model's outcome. It does this by selecting an equal number of samples from each group to include in each tree of the model, based on half the size of the smallest group[220]. For instance, in our example dataset, since the smallest sample count for a group is 12, thus the function selects half this number (6) of samples from every group.

This random sample selection is executed without replacement to maintain diversity in the model's training set. Samples not chosen for this process are designated as "out-of-bag" (OOB) samples and are used for individual tree's model validation, ensuring a balanced representation, with at least half of the samples from each group being utilized for testing each tree's model's accuracy[198]. This ensures that each group is equally represented in the training process. The next step will delve deeper into how these balanced samples contribute to the model-building and validation process.

54| **Run Random Forest** This step introduces the execution of Random Forest analysis using the `rfpermute()` function, designed to simplify the Random Forest modeling process[198,199]. The function, an extension of the classic `randomForest()`, requires few parameters: the feature quantification table without the target classification column (`x`), the classification labels (`y`), a balanced sample size for each group (`sampsize`), alongside the specified number of trees (in our example, `ntree=500`) and permutations (in our example, `num.rep=500`). **Box 10** provides a detailed overview of RF along with

a visualization of the RF algorithm and its implementation in R. With the `rfpermute()` function, a standard Random Forest model is initially created to calculate the variable importance. Following this, the response variable undergoes permutation a specified number of times (`num.rep`). With each permutation, a new Random Forest model is built to assess the impact of variable shuffling on the model's predictive accuracy[220].

`rfpermute()` streamlines the Random Forest modeling process without necessitating a traditional train-test data split, such as the 70-30 or 80-20 ratio[194]. This approach takes advantage of OOB samples for individual tree's model validation by using data not selected during each tree's building phase. The `balancedSampsize()` function, discussed in the previous step, ensures equal representation from each group in the model to prevent bias. As a result, a portion of the data is utilized for training while the remainder serves as OOB samples for model validation.

Once the model is run, the result displays the following:

- The number of variables tried at each split. This is the `mtry` parameter in the `randomForest` function. The value is 95 in our case. This is determined by the square root of the total feature count (9092), a default method in classification trees[197].
- The confusion matrix reveals an OOB correct classification rate (can be found under Overall `pct.correct`) of approximately 68%. This represents the model's ability to correctly classify OOB samples, thereby providing an internal measure of performance.

<CRITICAL STEP> It is important to consider the following points:

- The selection of the number of trees and permutations is crucial and should ideally be determined through hyperparameter tuning, such as with the `randomForest` package in R, to find the optimal settings. However, to keep the process straightforward for beginners, we use a value of 500 for both (see bullet below) and only the `rfpermute` package as a simplified approach. Also, `rfpermute` cannot be used for conducting traditional parameter tuning which is a limitation of this package.
- Increasing the number of trees and permutations generally enhances the model's performance but also escalates computational costs. It is advised to start with a reasonable number of trees (e.g., 500-1000) and `num.rep` (500-10000), then adjust based on performance.

- When working with large datasets, R may run out of internal memory trying to perform the random forest. To work around this, adding the "`as.factor`" to the predictor variable (y), even if the class is already a factor, will alleviate the memory error.

55| **Evaluate model performance** After getting the RF model, we need to evaluate the model's performance using several metrics such as model accuracy, the confusion matrix, trace plot, and check for potential overfitting by comparing testing versus training accuracies.

- The model accuracy we refer to here is derived from an OOB estimate, providing an approximation of OOB sample accuracy, rather than the traditional comparison between training and testing set accuracies (i.e., where the size of training and test data pools is a function of all available data instead of the result of balancing, see **step 53**). In our case, this OOB correct classification rate (Overall `pct.correct`) was found to be approximately 68%.
- The confusion matrix is the most basic summary of a Random Forest. See the figure B in **Box 10** for reference. The matrix consists of the 'original class' in rows and the 'predicted class' in columns. The diagonals represent the number of samples correctly classified in each class. The matrix also has columns that show the percent of samples that were correctly classified in a class, along with upper and lower 95% confidence intervals.
- The trace plot shows the OOB changes as trees were added to the forest. See the OOB graph **in Box 10** for reference. The model should have enough trees in it so the error rate is stable. If the error rate level increases as the number of trees increases, it may be an indication of overfitting.

56| **Interpreting RF Results** Beyond these, the RF results can be interpreted in various ways within the notebook. Users can generate the following results:

- One could plot the most impactful predictors of the RF model using violin plots. Here, we show the top 9 predictors in the notebook.
- Compare class predictions versus the actual group in a proximity plot. The group colors here are obtained from **Step 38**. This visualization is available in the notebook.
- Rank features by importance using the 'Mean Decrease Accuracy' metric. This metric helps identify features whose removal significantly impacts the model's accuracy, thus marking their importance. If a feature's removal does not affect accuracy, it may be deemed less important. Features with a 'MeanDecreaseAccuracy.pval' < .05 are

considered significant, implying that their absence would affect the model's performance significantly. This ranked list can also be exported as a CSV file for further analysis.

Box 10 - Random Forest

Random Forest (RF) is a powerful machine learning algorithm that operates by dividing data into fractions, building randomized tree predictors on each fraction, and aggregating these predictors together. The prediction could be based on either class labels (classification) or numerical values (regression). RF algorithm enhances model generalization by training each tree on a different data sample, where the sampling is done with replacement (bootstrap sampling). Typically, each tree is trained on a bootstrap sample consisting of about two-thirds of the original dataset, while the remaining one-third, not included in the bootstrap sample, forms the out-of-bag (OOB) samples for that tree. These OOB samples act as a de facto test set for validating that tree's accuracy. This internal cross-validation method contributes to RF's robustness and supports its utility without necessitating a traditional test set[194,195]. These OOB samples provide an unbiased estimate of the model error as they were not seen by the tree during training. This unique utilization of OOB samples for error estimation and variable importance assessment through permutation makes RF particularly effective for a wide range of data analysis tasks[196].

However, the above-mentioned use of approximately two-thirds of the data for training each tree in an RF model is a tunable parameter, and users may adjust it to improve model performance. For example, in the `randomForest` function[197], the parameter that controls the bootstrap sample size is `sampsize`.

The OOB error rate is a measure of prediction accuracy and helps to improve the performance of weak or unstable learners in the model. While OOB error provides a good estimate of model performance, it may not entirely replace the need for a separate test set, particularly for evaluating generalization to new data. OOB estimates, although unbiased, may overestimate the model's error rate if not run long enough to reach convergence, in other words, it is crucial to train the Random Forest with a sufficiently large number of trees until the error rates stabilize[194]. Acknowledging the diversity of practices in the field, with some using the traditional test-train split and others not[198,199], the model evaluation in RF can benefit from both OOB internal validation and testing the model on a separate test set for external generalization.

In RF, a common technique to assess the importance of each variable (or feature) in predicting the target classification is through permutation. Variable importance scores are obtained by permuting the values of each variable 'm' within the OOB samples and the tree is used to make

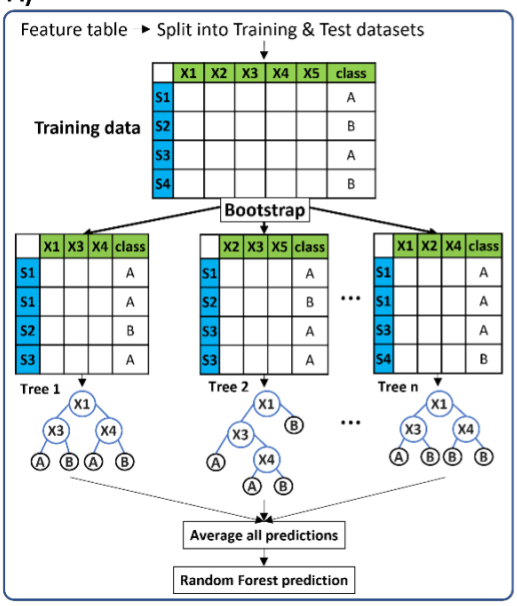
predictions on these permuted OOB samples. This essentially disrupts any relationship that variable 'm' might have with the target variable. The model then compares the prediction accuracy on the variable-m-permuted OOB samples to predict accuracy on the original (untouched) OOB samples. The average of the difference in accuracy (between permuted and original OOB) across all trees in the forest gives the raw importance score for variable "m". This raw importance score is often an average value over all trees. To determine if this importance score of variable "m" is statistically significant, a z-score can be calculated by dividing the raw score by its standard error[200].

Although permutation-based variable importance is a widely recognized method for evaluating the significance of variables in RF models, it is not a default method in all RF implementations. The `rfPermute` package specifically facilitates this process for RF models in R, automating the calculation of importance scores by permuting feature values and assessing their impact on model accuracy. This approach helps in identifying the most influential variables and determining their statistical significance. To ensure consistency, we have implemented permutation-based variable importance calculations in both our R and Python Notebooks.

In RF, there are two common metrics of variable importance used to rank features based on their predictive power: Mean Decrease Accuracy (MDA) and Mean Decrease Gini (MDG). MDA measures the decrease in model accuracy when a particular variable's values are permuted. A large decrease indicates high variable importance; MDG measures how each variable contributes to the homogeneity of the nodes and leaves in the resulting RF. A higher MDG value indicates that splitting the dataset by this variable results in purer nodes. Here, Variable Importance Projection (VIP) could be obtained by normalizing MDA, so they sum to 100, making them more interpretable on a relative scale[201].

Some of the other important parameters to keep in mind to evaluate the performance of the RF model are: model accuracy, confusion matrix (a matrix showing true vs predicted class labels), trace plot, and check for overfitting by comparing testing vs training accuracy. However, supervised models may not be suitable for all datasets, especially those with few observations or unclear class distinctions. Confounding variables, related to both the predictor and response variable, can also make these models unsuitable. For instance, age and gender in a drug study can be confounding variables, leading to erroneous results if not controlled for. In such cases, using supervised models for analysis may not be appropriate.

A)



B)

```
rp <- rfPermute(x = RF_data[,-1], #feature table without the classification info
                y = as.factor(RF_data[,1]), #classification variable
                sampsize=balanced_size, #give the balanced sample size info
                ntree=500,
                num.rep=500,
                proximity = T)
```

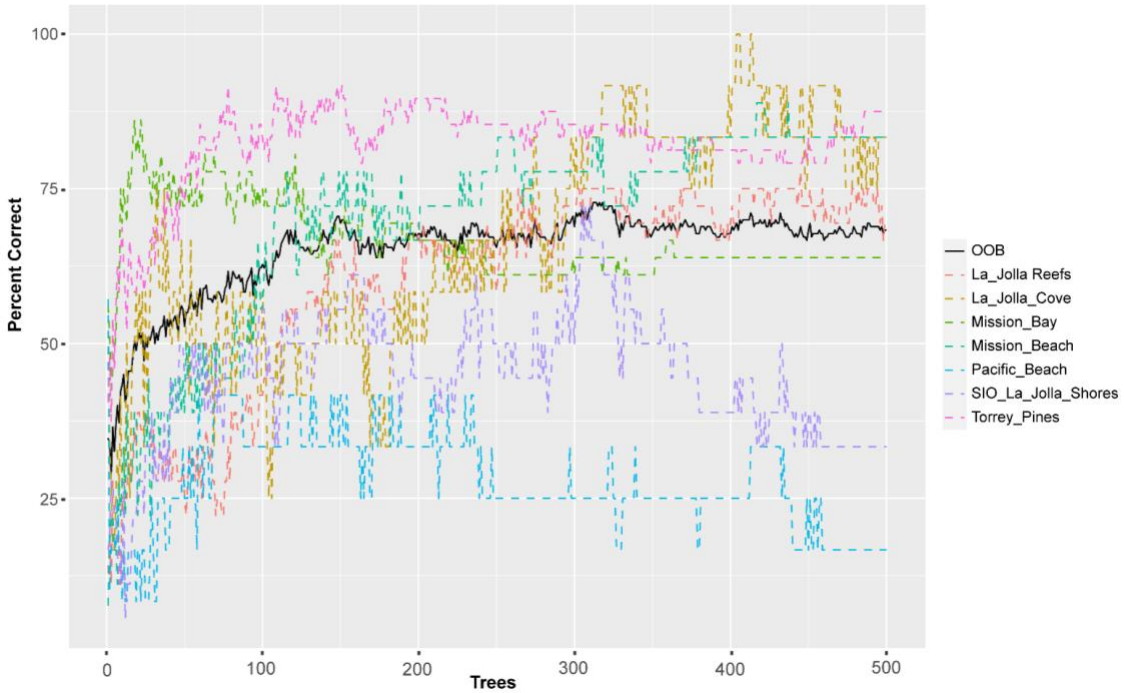
```
rp
```

```
An rfPermute model

               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 95
      No. of permutation replicates: 500
                          Start time: 2023-04-03 10:20:49
                            End time: 2023-04-03 11:11:23
                            Run time: 50.6 mins
```

	La_Jolla Reefs	La_Jolla _Cove	Mission_ Bay	Mission_ Beach	Pacific_ Beach	SIO_La_ Jolla_Shores	Torrey_ Pines	pct.correct	LCI_0.95	UCI_0.95
La_Jolla Reefs	27	1	1	1	3	3	0	75.0	57.80	87.9
La_Jolla_ Cove	0	9	0	0	0	3	0	75.0	42.81	94.5
Mission_Bay	5	0	28	2	1	0	0	77.8	60.85	89.9
Mission_ Beach	1	0	0	15	2	0	0	83.3	58.58	96.4
Pacific_ Beach	3	0	1	5	3	0	0	25.0	5.49	57.2
SIO_La_Jolla_ Shores	1	0	0	3	0	10	4	55.6	30.76	78.5
Torrey_Pines	0	0	0	0	0	8	40	83.3	69.78	92.5
Overall	NA	NA	NA	NA	NA	NA	NA	73.3	66.24	79.6

Random Forest Algorithm Visualization and Execution: A) This image shows an illustration of the Random Forest algorithm; B) The code block displayed here was used for model execution, using 500 trees and 500 permutations. Outputs of the rfpermute model, including a confusion matrix, are showcased. In addition to the above, the Out-of-Bag (OOB) error curve, another crucial model evaluation metric, is displayed as follows:



2.4.4 Univariate Statistics: ● Timing 50-60 mins

● Timing **5 mins** Start by installing the packages necessary for this section: FSA[221] (v0.9.4), matrixStats[222] (v0.63.0).

Test for Normality

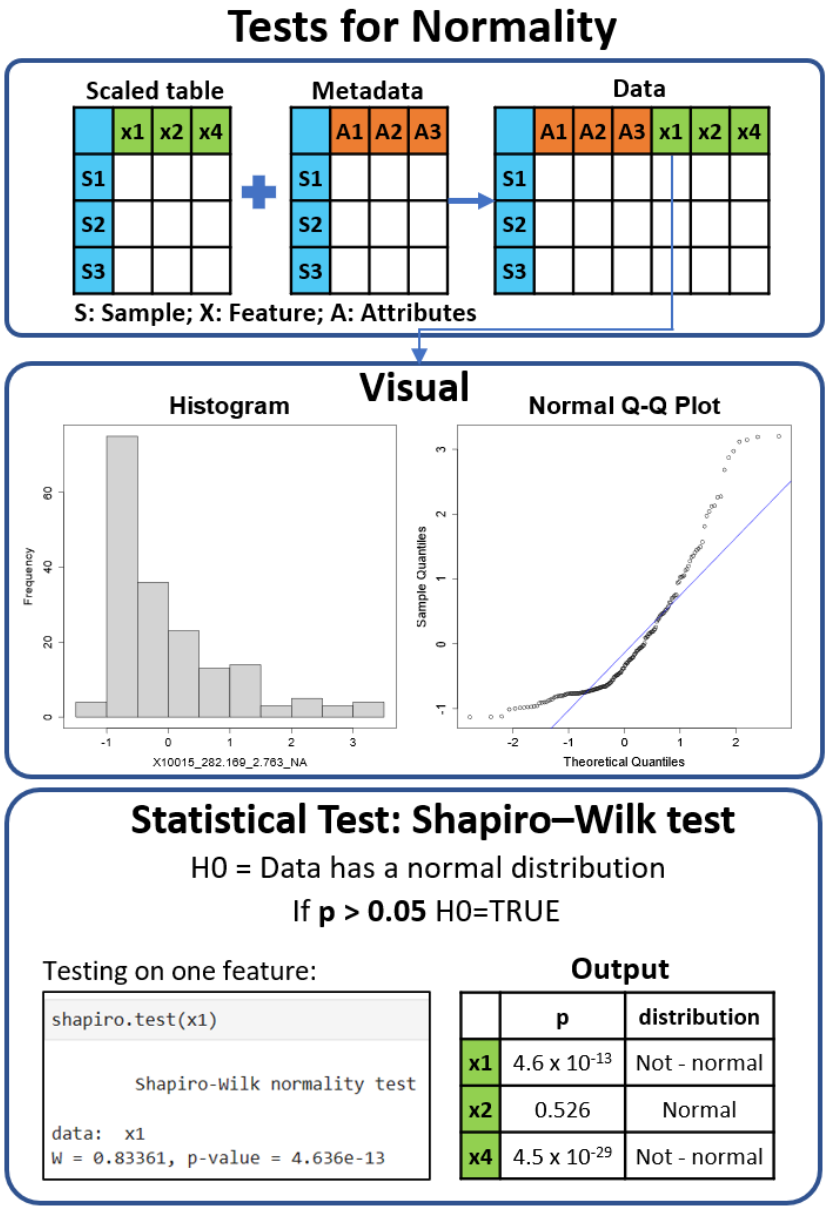


Figure 11: Assessing Normality of Features: This figure illustrates methods to assess normality for individual features. It showcases visual approaches like histograms and Q-Q plots, where deviations from normality can be visually assessed. The third segment delves into significance testing using the Shapiro-Wilk test, emphasizing that a p-value greater than 0.05 suggests a normal distribution.

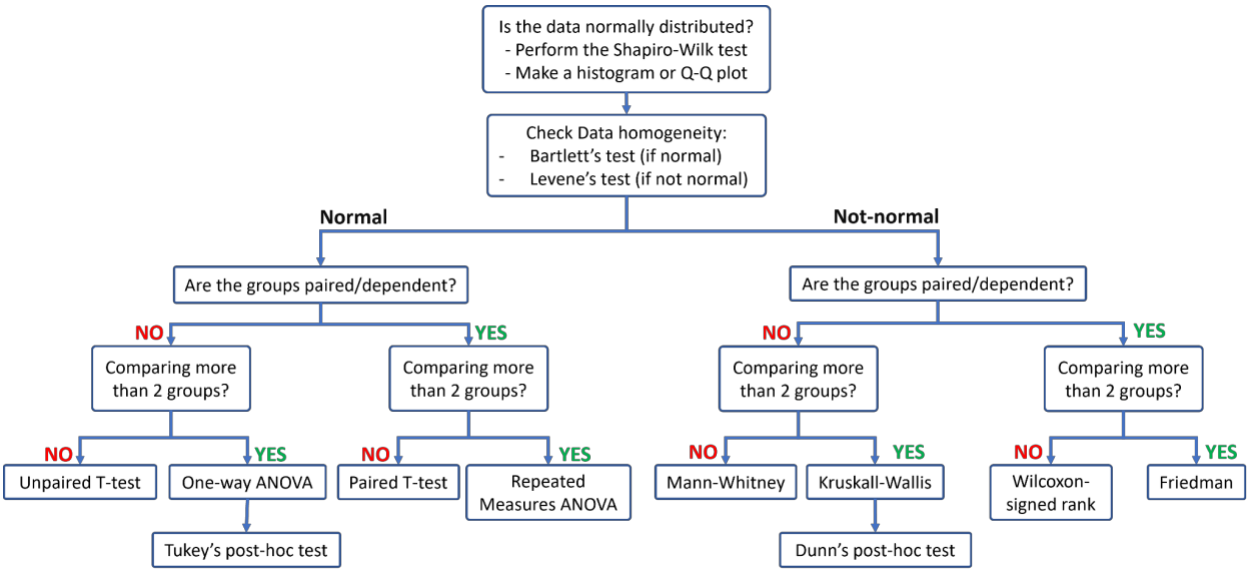


Figure 12: The flowchart guides users in choosing the appropriate univariate statistical test, starting with a normality test (for example, Shapiro-Wilk). If the data are not normally distributed ($P \leq 0.05$), nonparametric tests are recommended. Homoscedasticity, or equality of variances, can be checked using Levene's or Bartlett's tests. The flowchart then directs the choice between parametric and nonparametric tests, based on whether the data groups are paired or unpaired and on the number of groups being compared.

<CRITICAL> In our pipeline, we conduct a normality test using two approaches: visual representations such as histograms and quantile-quantile plots (Q-Q plots), and the Shapiro-Wilk statistical test. A graphical representation of testing normality of features is shown in **Figure 11**. In addition to this, to know more about normality assumptions, refer to **Box 11**. **Figure 12** provides a flowchart that guides the selection of appropriate statistical tests based on data normality and homogeneity.

57| **Normality Testing for One Feature** To illustrate how to test for normality, pick one feature and generate a Q-Q plot using the `qqnorm()` and `qqline()` functions. Then, perform a Shapiro-Wilk test using the `shapiro.test()` function. The feature for this example is chosen from the `` `cleaned_data_with_md` `` dataframe, which was prepared in **Step 51**. Additionally, **Step 58** also demonstrates how log-transforming the data can improve the normality of the data.

58| **Normality Testing for All Features** Perform a Shapiro-Wilk test for each feature and record the resulting p-values. Correct these p-values for false discovery rate (FDR) using the Benjamini & Hochberg method. If the adjusted p-value ('`p_adj`') is less than 0.05, reject the null hypothesis and consider the data to be non-normal. Tally up the features

that fall under normal and non-normal distributions. If the majority of features are non-normal, consider using non-parametric tests for further analysis. In our example data, out of the 9092 features, only 54 had a normal distribution. Thus performing non-parametric tests, such as the Kruskal-Wallis test, might make more sense for our data.

Box 11 - Normality assumptions

Besides normality, it is essential to consider two other critical assumptions when deciding between parametric and non-parametric tests: homogeneity of variances (homoscedasticity) and independence. Homoscedasticity demands that within-group variances are equal. If unequal (heteroscedasticity), it increases the chance of falsely identifying a "significant" result. Homoscedasticity can be evaluated graphically via boxplots or statistically via Levene's and Bartlett's tests. Here, the null hypothesis (H0) for these tests states that the within-group variances are equal. If the p-value is less than 0.05, it indicates a difference in population variances. The final assumption, 'independence', stipulates that the occurrence of one event does not influence the probability of another. In a metabolomic context, this implies that knowledge of one sample value does not predict another's. However, these assumptions, particularly normality, are seldom fully met in real-world metabolomics datasets[202,203].

Parametric and non-parametric tests

<CRITICAL> In this procedure, we performed both ANOVA (parametric) and Kruskal-Wallis tests (non-parametric), along with their respective post-hoc tests on the same dataset. The only reason that both tests were performed was to demonstrate to users how these tests can be applied and what the potential results might look like. However, only one of the two tests is necessary depending on whether the user's data conforms to parametric test assumptions. Therefore, it is advisable to choose the test that best suits your data's characteristics rather than employing both. The Kruskal-Wallis test is considered more appropriate for the example dataset due to the non-normal distribution of the majority of its features (See **Step 59**). For more information on which test to choose based on normality assumptions, refer to **Figure 12**.

Parametric tests

<CRITICAL> For all the tests below, the respective significance values can be saved as a CSV table, and the plots can be saved in SVG, PDF, or PNG formats for further analysis or presentation.

ANOVA test

<CRITICAL> The analysis of variance (ANOVA) is the statistical procedure used to test if there exists a significant difference in the means of a dependent variable between three or more groups. As opposed to a pair-wise comparison where we compare the means in a variable (i.e., $\mu_1=\mu_2$), in the ANOVA we compare the means of several groups[243]. For a deeper understanding of ANOVA, please refer to **Box 12**. Furthermore, the accompanying illustration offers a visual explanation of the ANOVA algorithm, detailing both the R code and a resulting plot that contrasts the F-statistic with p-values, highlighting significant features.

59| **Run ANOVA on one feature** *(User Input Required)* Here the user is prompted to enter the index number of the attribute for performing ANOVA. In the tutorial, we use '`ATTRIBUTE_Sample_Area`'. The resulting ANOVA statistics are shown in a table format.

60| **Running ANOVA on all features**

- For each metabolite feature, execute an ANOVA test within a for loop. The output for each feature is stored in a dataframe named `` `anova_out` ``. The 'for loop' passes each feature column as the first argument of the `aov()` function against the selected attribute from the previous step ('`ATTRIBUTE_Sample_Area`'). This is because we are examining how a particular feature varies across different sample areas.
- Tidy up the ANOVA output for each feature into a table using the `tidy()` function from the broom[244] package.
- Out of the two rows in the ANOVA summary table, select only the first row of this table (which contains the means, F-statistic, and p-value) and leave the second row consisting of the residuals.
- Consolidate these rows into a single dataframe which contains the features, their corresponding p-values, their BH-corrected p-values, and their significance status in several columns. Features with a BH-corrected p-value ('`anova_out$p_BH`') less than 0.05 are considered significant.

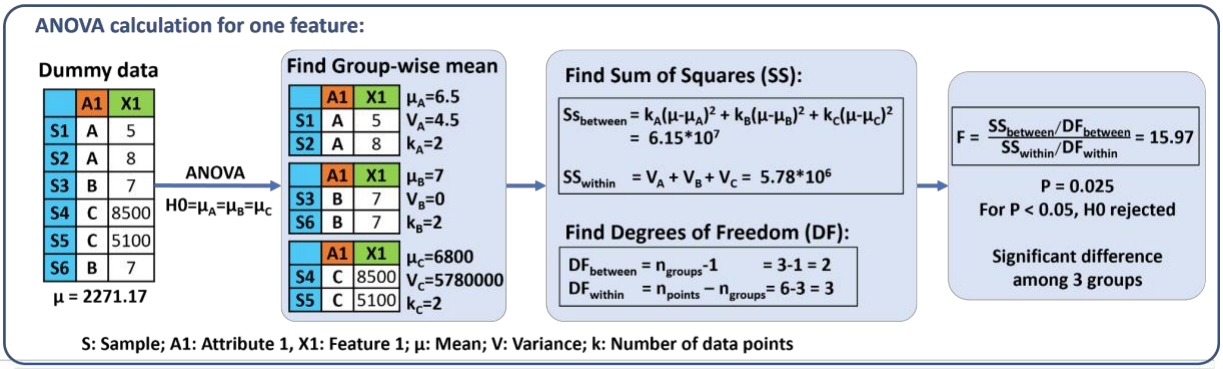
61| **Subsetting Significant Features** Filter out the significant features for further examination. Display the count of significant and non-significant features.

62| **Visualize ANOVA Results** Sort the `anova_out` results by p-value and visualize the significant features using `ggplot()`. This involves plotting log-transformed F-Statistic values on the x-axis against the negative logarithm of `p_BH` values on the y-axis. As F-Statistic and p-values can vary greatly, their log values offer easier visualization. To prevent clutter, limit the display to the names of the top 6 significant features.

63| **Visualize the Top Significant Metabolites** Generate boxplots for the top 4 significant metabolites to observe how their intensity levels differ across sampling sites. The colors for these different sampling sites are generated from **Step 38**. Extract these metabolites' data from the `uni_data` dataframe, which contains both feature intensities and metadata, and plot their intensities based on the sampling sites. In our example, the higher intensities of these features in the 'Mission Bay' sample area primarily account for the observed differences between sampling sites.

Box 12 – ANOVA

If a pairwise test is used (e.g., a t-test), an increased probability of getting a false positive difference (Type I error) would be observed just by chance due to the effects of multiple comparisons[204]. Instead, in the ANOVA test, we can perform a single test to see if the observed differences are due to randomness or due to the grouping of the samples (e.g., origin, location, type of soil, etc.). The F-statistic is calculated using the sum of squares and the degrees of freedom (seethe illustration below) and compared to a standard F-distribution to determine whether the differences among group means are greater than would be expected by chance. Importantly, the alternative hypothesis (i.e., where a difference exists between the means) is unspecific. This means that the test does not tell us where the difference(s) lie (e.g., if the difference is $\mu_A \neq \mu_B$ or $\mu_B \neq \mu_C$), it only tells us whether there exists a difference among all the means. The first assumption of the ANOVA test is the normality of population distribution and the homogeneity in their variances[202,205]. Non-parametric tests should be used if these assumptions do not hold in the data of interest.

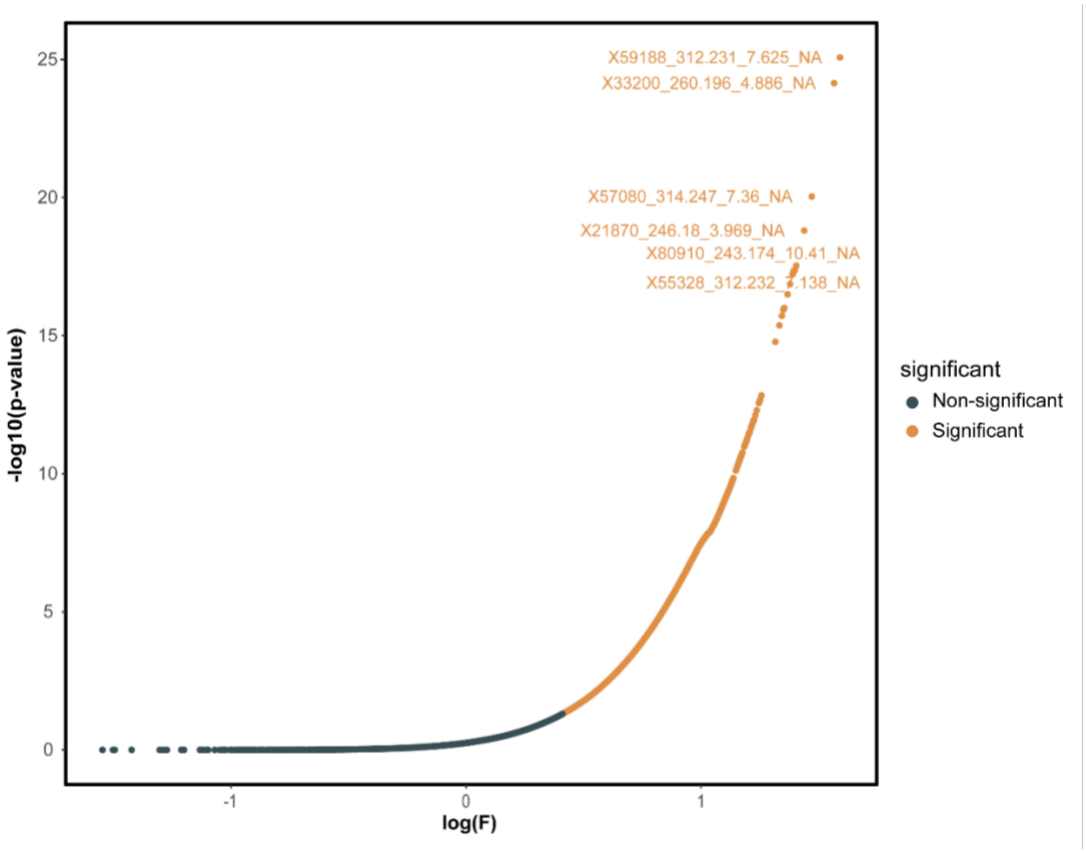



```
anova_group <- uni_metadata[,interested_attribute_anova]
anova_group <- as.factor(anova_group) # convert the attribute to 'factor' type

broom::tidy(aov(uni_data[,1] ~ anova_group)) #tidy summarizes the anova ouptut in a  tibble
```

A tibble: 2 × 6

term	df	sumsq	meansq	statistic	p.value
<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
anova_group	6	7.560749	1.2601248	1.271597	0.2728111
Residuals	173	171.439251	0.9909783	NA	NA



The illustration depicts the ANOVA process applied to a sample feature across groups A, B, and C using dummy data. Following this, the R code block used to test ANOVA on our dataset is displayed. This process involves selecting metadata for grouped information (e.g., sample areas), factorizing it for grouping, and then presenting the ANOVA outcome for one of the features in relation to various sample areas. A complementary volcano plot showcases the significance of features by mapping log(F-statistic of ANOVA) against the negative logarithm of p-values.

Tukey's Honestly Significant Difference (HSD) test

<CRITICAL> If the ANOVA test provides evidence that a difference indeed exists between the means of the groups, the next step is to find between which groups the difference or differences exist. To do this, we can conduct a Tukey HSD post hoc test used to compare multiple means in a single analysis[202]. Refer to **Box 13** for more information on Tukey's test. Additionally, the box provides a visual guide for applying the Tukey test, its implementation in R, and a resulting volcano plot that highlights significant features from our pairwise comparison.

64| **Perform Tukey HSD for a Significant Feature** First, we select a feature identified as significant in the ANOVA result, using `` `anova_sig_names` `` generated in **Step 62**. From the ANOVA output, we subset the data for this significant feature and conduct a Tukey HSD test. The output is a comprehensive table providing an assessment of every possible pairwise group difference as shown in the code snippet in **Box 13**. To conduct a Tukey HSD test for all features, consider specifying just a one-pair comparison to maintain simplicity. For instance, based on the ANOVA results, the sampling site 'Mission Bay' appeared to significantly differ from others for the top four metabolites, hence we can focus on the results from comparisons between 'Mission Bay' and another specific sampling site in the subsequent step.

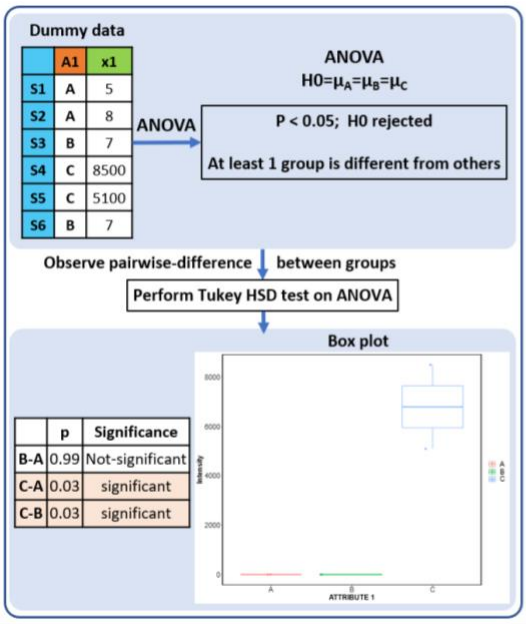
65| **Perform Tukey HSD for All Significant Features** *(User Input Required)* Carry out a Tukey HSD test for all the significant features identified in the ANOVA. Then, filter the results for the specific comparison such as 'Mission Bay vs. La Jolla Reefs'. Here, users are prompted to input the index number corresponding to their desired comparison from the 'contrast' column displayed in the previous step's output. As a result of the Tukey test of this pairwise interaction, p-values are produced for each feature. After applying the BH correction method, features with corrected p-values `(output_tukey$p_BH < 0.05)` are highlighted as significantly different between the selected sites.

66| **Count Significant Features** Determine how many features exhibit a significant difference between the chosen sites and how many do not.

67| **Visualize Results with a Volcano Plot** Create a volcano plot with '`-log(p_BH)`' on the y-axis and the group difference ('`estimate`') on the x-axis. Display the names of the top findings on the plot to highlight the most significant differences between the chosen sites. Additionally, visualize the top 2 significant metabolites as boxplots from both extremes of the volcano plot (right and left tips) to clearly represent if the significant metabolite is upregulated or downregulated among the chosen sites. The colors for the different groups in these boxplots are obtained from **Step 38**.

Box 13 - Tukey's post hoc test

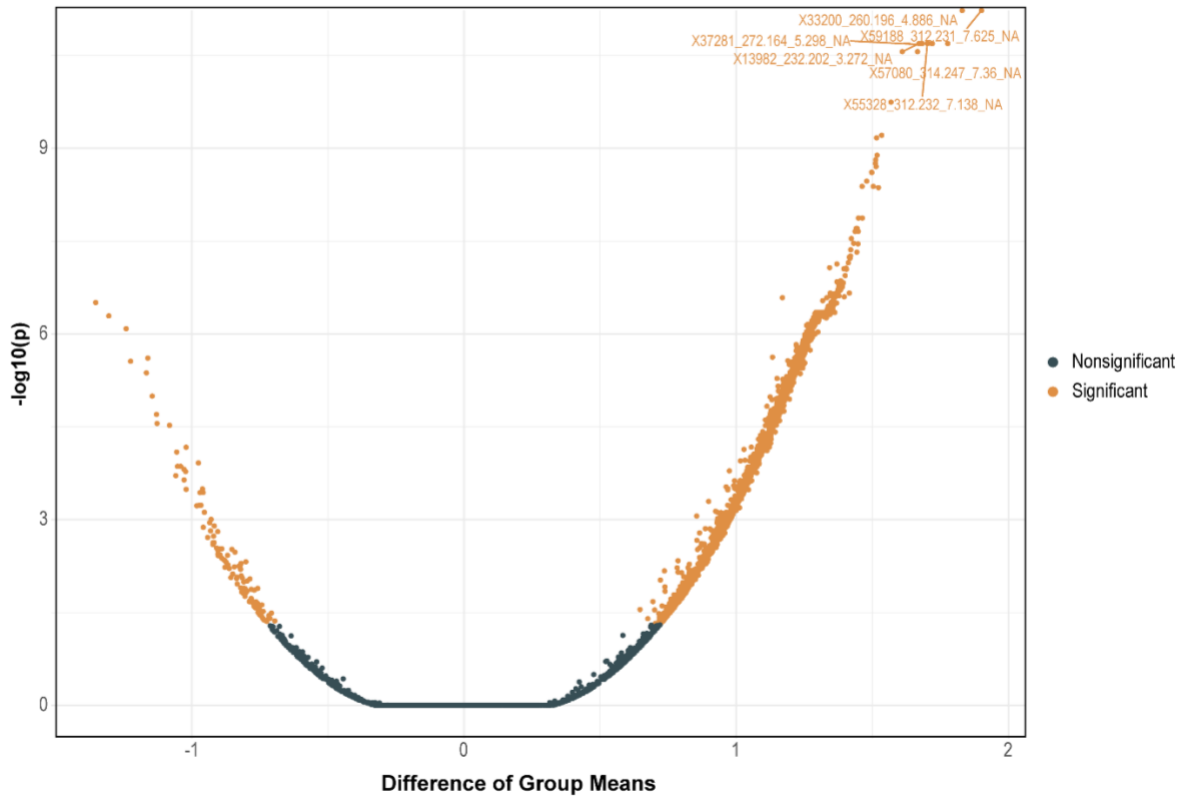
One of the goals of this test is to overcome the Type I error rate inflation of doing multiple comparisons[202]. The most used post hoc test for ANOVA is Tukey's Honestly Significant Difference (HSD). To calculate the HSD between two means, a statistical distribution defined by Student (called the q distribution) is used which takes into account the number of means being compared[206].




```
model_1 <- anova_model[[anova_sig_names[1]]] #Looking at one of the anova model that showed significant difference
broom::tidy(TukeyHSD(model_1)) # Perform Tukey HSD test on the model_1 and summarize the result
```

A tibble: 21 × 7

term	contrast	null.value	estimate	conf.low	conf.high	adj.p.value
<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Data$ATTRIBUTE_Sample_Area	La_Jolla_Cove-La_Jolla Reefs	0	-0.0146107707	-0.6740913	0.6448698	1.000000e+00
Data$ATTRIBUTE_Sample_Area	Mission_Bay-La_Jolla Reefs	0	1.9009769489	1.4346538	2.3673001	0.000000e+00
Data$ATTRIBUTE_Sample_Area	Mission_Beach-La_Jolla Reefs	0	-0.0134035939	-0.5845305	0.5577233	1.000000e+00
Data$ATTRIBUTE_Sample_Area	Pacific_Beach-La_Jolla Reefs	0	0.0642958118	-0.5951848	0.7237764	9.999494e-01
Data$ATTRIBUTE_Sample_Area	SIO_La_Jolla_Shores-La_Jolla Reefs	0	-0.0132175276	-0.5843445	0.5579094	1.000000e+00
Data$ATTRIBUTE_Sample_Area	Torrey_Pines-La_Jolla Reefs	0	0.0317892050	-0.4044162	0.4679946	9.999910e-01
Data$ATTRIBUTE_Sample_Area	Mission_Bay-La_Jolla_Cove	0	1.9155877196	1.2561071	2.5750683	1.142419e-13
Data$ATTRIBUTE_Sample_Area	Mission_Beach-La_Jolla_Cove	0	0.0012071769	-0.7361145	0.7385289	1.000000e+00
Data$ATTRIBUTE_Sample_Area	Pacific_Beach-La_Jolla_Cove	0	0.0789065825	-0.7287889	0.8866020	9.999488e-01
Data$ATTRIBUTE_Sample_Area	SIO_La_Jolla_Shores-La_Jolla_Cove	0	0.0013932431	-0.7359285	0.7387149	1.000000e+00
Data$ATTRIBUTE_Sample_Area	Torrey_Pines-La_Jolla_Cove	0	0.0463999757	-0.5921393	0.6849393	9.999911e-01
Data$ATTRIBUTE_Sample_Area	Mission_Beach-Mission_Bay	0	-1.9143805428	-2.4855075	-1.3432536	4.685141e-14
Data$ATTRIBUTE_Sample_Area	Pacific_Beach-Mission_Bay	0	-1.8366811371	-2.4961617	-1.1772006	6.189493e-13
Data$ATTRIBUTE_Sample_Area	SIO_La_Jolla_Shores-Mission_Bay	0	-1.9141944766	-2.4853214	-1.3430675	4.718448e-14
Data$ATTRIBUTE_Sample_Area	Torrey_Pines-Mission_Bay	0	-1.8691877439	-2.3053931	-1.4329823	0.000000e+00
Data$ATTRIBUTE_Sample_Area	Pacific_Beach-Mission_Beach	0	0.0776994056	-0.6596223	0.8150211	9.999201e-01
Data$ATTRIBUTE_Sample_Area	SIO_La_Jolla_Shores-Mission_Beach	0	0.0001860662	-0.6592945	0.6596666	1.000000e+00
Data$ATTRIBUTE_Sample_Area	Torrey_Pines-Mission_Beach	0	0.0451927988	-0.5016196	0.5920052	9.999809e-01
Data$ATTRIBUTE_Sample_Area	SIO_La_Jolla_Shores-Pacific_Beach	0	-0.0775133394	-0.8148350	0.6598084	9.999212e-01
Data$ATTRIBUTE_Sample_Area	Torrey_Pines-Pacific_Beach	0	-0.0325066068	-0.6710459	0.6060327	9.999989e-01
Data$ATTRIBUTE_Sample_Area	Torrey_Pines-SIO_La_Jolla_Shores	0	0.0450067326	-0.5018057	0.5918191	9.999814e-01



Overview of Tukey's HSD Test: This figure starts with an illustration that showcases how ANOVA's significance suggests that at least one group differs significantly from others, which then necessitates a further analysis through pairwise comparisons using Tukey's HSD test. Alongside this, we present the code block demonstrating the Tukey test applied to the first significant feature identified via ANOVA. Given the presence of 7 sample areas, the output presents p-values for all potential 21 pairwise comparisons. Having executed this for all ANOVA-significant features, we particularly highlighted comparisons between 'Mission Bay' and 'La Jolla Reef'. The resulting significance is visualized via a volcano plot, where right-tailed features exhibit higher prevalence in 'Mission Bay', while left-tailed features dominate in 'La Jolla Reef'.

T-tests

68| **Select Attribute for T-Test Analysis** *(User Input Required)* A t-test is suitable for comparisons involving just two groups. Therefore, users should specify the attribute for the two distinct groups by providing the corresponding index number. For our example, we explore the metabolome's response to rainfall. Hence, we introduce an '`ATTRIBUTE_rainfall`' column, designating '1' for 'Jan-2018' (a high rainfall period) and '0' for the remaining two months, Dec 2017 and Oct 2018 (without rainfall).

<CRITICAL STEP> This '`ATTRIBUTE_rainfall`' column addition caters to our dataset's context. This example aims to illustrate the concept of including samples' environmental context rather than serve as a model for this test. Users with pre-existing binary attributes can skip this addition, while others may adjust this step to align with their data.

69| **Perform T-Test** Following the same steps as ANOVA (from **Steps 60 to 63**), the `t.test()` function is used in place of `aov()` in this case. The final output is a dataframe 'ttest_output' containing the significance of each feature for the two conditions under investigation.

70| **Plot T-Test Results** Visualize the t-test results using a volcano plot, with the '`estimate`' (difference in means of the two conditions for each feature) on the x-axis and '`-log(p_BH)`' on the y-axis. Additionally, visualize the top 2 significant metabolites as boxplots from both extremes of the volcano plot (right and left tips) to clearly represent if the significant metabolite is upregulated or downregulated for the chosen attribute. The colors for the different groups in these boxplots are obtained from **Step 38**.

<CRITICAL STEP> Unlike ANOVA, post-hoc tests are not needed for t-tests as there are only two conditions to compare. In ANOVA, when a feature is found to be significant, post-hoc tests help determine which specific groups show significant differences.

Non-Parametric Tests

<CRITICAL> Non-parametric tests can be performed when parametric tests are not appropriate due to data characteristics, such as non-normality. For all of these tests, the respective significance values can be saved as a CSV table, and the plots can be saved in SVG, PDF, or PNG formats for further analysis or presentation.

Kruskal-Wallis Test

<CRITICAL> The Kruskal-Wallis (KW) test is a non-parametric statistical test used to compare three or more independent groups. It can be used when the assumptions of normality and equal variances are not met for performing an ANOVA[207]. For more information on the KW Test, refer to **Box 14**. Additionally, the accompanying figure shows a visual explanation of the KW algorithm, along with the R-code used to test a feature across various groups and determine its significance.

71| **Perform Kruskal-Wallis Test on one feature** *(User Input Required)* Begin by specifying the attribute for the KW test by entering its index number. In this tutorial, we opt for

'ATTRIBUTE_Sample_Area'. Then, apply the KW test on a single feature (the first feature in the `uni_data` dataframe) across different sample areas using the `kruskal.test()` function. Note that the `uni_data` dataframe originates from the `cleaned_data`, which we chose as the `Imp_s` scaled table (see **Step 30**). Summarize the output into a one-row table using the `tidy()` function from the broom[244] package as shown in the figure.

The steps for the KW test (**Steps 72 to 74**) are structured similarly to the ANOVA steps (**Steps 60 to 63**).

72| **Run Kruskal-Wallis Test for All Features**

- Just like in ANOVA (**Step 61**), perform the Kruskal-Wallis test for each metabolite across different sample areas. Then, tidy up the output for each feature into a table using the `tidy()` function.
- Combine these rows into a single dataframe containing features, their corresponding p-values, their BH-corrected p-values, and their significance status. Features with a BH-corrected p-value (`kruskall_out$p_BH < 0.05`) less than 0.05 are considered significant.

73| **Filter Significant Features** Display the count of significant and non-significant features. Filter out the names of significant features from the KW output dataframe for further analysis. This is done by selecting the column 'significant' in `kruskall_out` dataframe and filter rows labeled 'Significance'. Extract their row names, which include information such as unique feature ID, m/z values, RT, and annotation if available.

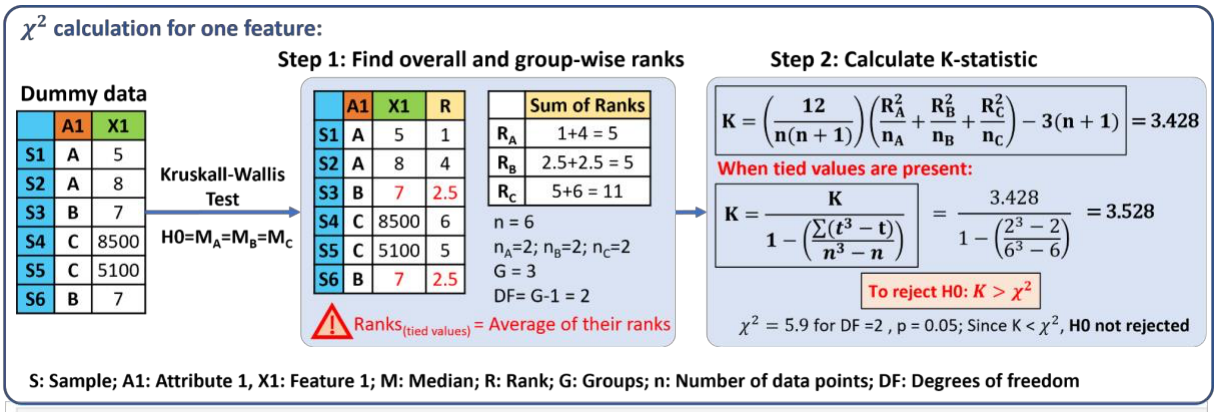
74| **Visualize Kruskal-Wallis Results** Similar to visualising ANOVA results, we first sort the `kruskall_out` dataframe results by p-value and visualize the significant features using `ggplot()`. This involves plotting log-transformed K-Statistic values on the x-axis against '`-log(p_BH)`' on the y-axis. To prevent clutter, limit the display to the names of the top 6 significant features.

75| **Visualize the Top Significant Metabolites of Kruskal-Wallis Results** Generate boxplots for the top 4 significant metabolites to observe how their intensity levels differ across sampling sites. The colors for the different sampling sites in these boxplots are obtained from **Step 38**. Extract these metabolites' data from the `uni_data` dataframe,

which contains both feature intensities and metadata, and plot their intensities based on the sampling sites.

Box 14 - Kruskal-Wallis test

Although the Kruskal-Wallis test does not assume normality, it is expected that samples are random and independent and that the observations in each group come from populations with the same shape of distribution[207]. As an extension of the Mann–Whitney U test (which is used to compare only two groups), it compares the median ranks of the groups, which are calculated by combining the ranks of all the observations across all groups and then taking their average[208]. With this information, the K statistic can be calculated and compared to the chi-square distribution to accept or reject the null hypothesis (see the illustration below). If the null hypothesis is rejected, the alternative hypothesis states that at least one group has a different median from the others.



	A1	X1
S1	A	5
S2	A	8
S3	B	7
S4	C	8500
S5	C	5100
S6	B	7

	A1	X1	R
S1	A	5	1
S2	A	8	4
S3	B	7	2.5
S4	C	8500	6
S5	C	5100	5
S6	B	7	2.5

```
broom::tidy(kruskal.test(uni_data[,1] ~ uni_metadata[,interested_attribute_kw]))
```

A tibble: 1 × 4

statistic	p.value	parameter	method
<dbl>	<dbl>	<int>	<chr>
6.007019	0.4224043	6	Kruskal-Wallis rank sum test

The illustration provides a comprehensive view of the Kruskal-Wallis test algorithm. If the test results in rejecting the null hypothesis (with $p < 0.05$), it suggests that at least one group's median deviates significantly from the others. To complement the illustration, the corresponding code snippet from the protocol is presented. Echoing the approach with ANOVA, here the Kruskal-Wallis test is executed on an individual feature in relation to the metadata column that groups information, with our primary interest being the "Sample area".

Dunn's Post Hoc Test

<CRITICAL STEP> The Dunn statistical test is a non-parametric alternative to the Tukey HSD post hoc test to make pairwise comparisons between multiple groups. The steps for Dunn's post hoc test (**Steps 77 to 80**) are structured similarly to the Tukey HSD steps (**Steps 65 to 68**). Refer to **Box 15** for more information on Dunn's post hoc test. The accompanying figure in **Box 15** shows a visual representation of applying the Dunn test and its implementation in R.

76| **Perform Dunn Test for a Significant Feature** First, we select the first feature identified as significant in the KW test result, using `` `kw_sig_names` `` generated in **Step 74**. From the KW output, we subset the data for this significant feature and conduct a Dunn test using `dunnTest()` function. The output is a comprehensive table providing an assessment of every possible pairwise group difference as shown in **Box 15**. When conducting a Dunn test on all significant features, consider specifying just one pair interaction to maintain simplicity. Similar to the Tukey HSD test, here we will focus on the results from comparisons between 'Mission Bay' and 'La Jolla Reefs' in the subsequent step.

77| **Perform Dunn Test for All Significant Features** *(User Input Required)* Carry out a Dunn test for all the significant features identified in the Kruskal-Wallis test with BH correction for p-values. Then, filter the results for the specific interaction 'Mission Bay vs La Jolla Reefs'. To perform this, the user will be prompted to enter the index number corresponding to the desired comparison. This index number can be referenced from the table produced in the preceding step. Then, the Dunn Test result for those comparisons will be filtered for each feature showing the corrected p-values. The significance is assigned based on the corrected p-values (`dunn_output$P.adj < 0.05`) to identify the features that show a significant difference between these two sites.

78| **Count Significant Features** Determine how many features exhibit a significant difference between the chosen sites and how many do not.

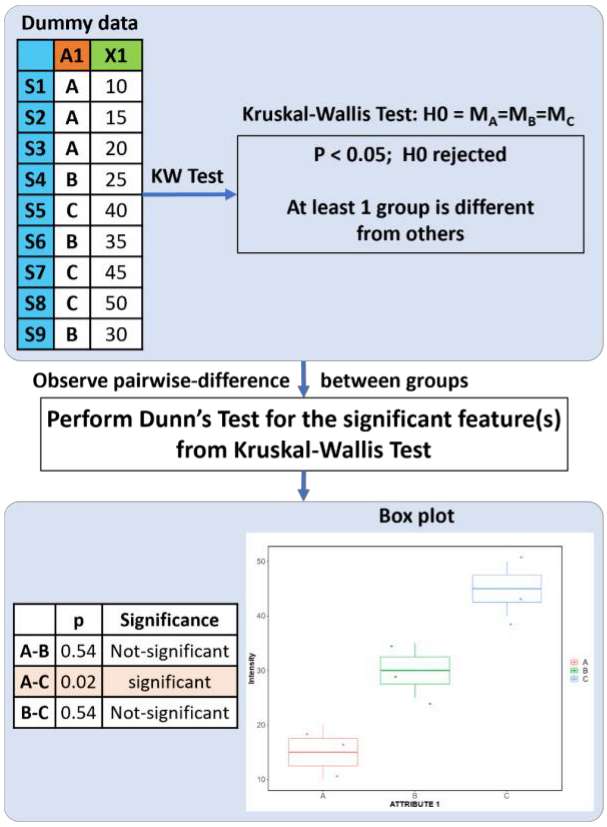
79| **Visualize Results with a Volcano Plot** Create a volcano plot with '`-log(p_BH)`' on the y-axis and the Z statistic on the x-axis. Display the names of the top findings on the plot to highlight the most significant differences between the chosen sites. Additionally, visualize the top 2 significant metabolites as boxplots from both extremes of the volcano plot (right and left tips) to clearly represent if the significant metabolite is upregulated or

downregulated for the chosen sites. The colors for the different sampling sites in these boxplots are obtained from **Step 38**.

Box 15 - Dunn test

The Dunn statistical test is a non-parametric post-hoc test following Kruskal-Wallis test similar to the Tukey HSD post hoc test for ANOVA to make pairwise comparisons between multiple groups. Dunn's z-test approximation of the exact rank-sum test statistics is calculated with the mean rankings from the preceding Kruskal–Wallis test based on the differences in mean ranks for each group and, then, the p-value is calculated using a modified version of the BH correction to account for the type I error rate increase due to multiple comparisons.[209]

The image below illustrates the Dunn test, a post hoc analysis following the Kruskal-Wallis test. After identifying significant features from the Kruskal-Wallis test, the next step is to conduct pairwise comparisons between groups. The accompanying code block demonstrates the execution of the Dunn test on the first significant feature obtained from the Kruskal-Wallis test, examining its relationship with various sample areas. The resulting display includes p-values for all potential 21 pairwise comparisons.




```
#Performing Dunn Test on the first significant feature of Kruskal Wallis output
dunn_first_sig_feature_model <- dunnTest(uni_data[,kw_sig_names[1]],
                                         as.factor(uni_metadata[,interested_attribute_kw]),
                                         method="bh")

data.frame(INDEX= 1:nrow(dunn_first_sig_feature_model$res),
           dunn_first_sig_feature_model$res)
```

A data.frame: 21 × 5

INDEX	Comparison	Z	P.unadj	P.adj
<int>	<chr>	<dbl>	<dbl>	<dbl>
1	La_Jolla Reefs - La_Jolla_Cove	1.93197175	5.336299e-02	1.018748e-01
2	La_Jolla Reefs - Mission_Bay	0.43199846	6.657425e-01	7.358207e-01
3	La_Jolla_Cove - Mission_Bay	-1.62650271	1.038428e-01	1.817248e-01
4	La_Jolla Reefs - Mission_Beach	0.99538701	3.195481e-01	4.793221e-01
5	La_Jolla_Cove - Mission_Beach	-0.95698460	3.385750e-01	4.740050e-01
6	Mission_Bay - Mission_Beach	0.64266175	5.204436e-01	6.429009e-01
7	La_Jolla Reefs - Pacific_Beach	0.36304436	7.165717e-01	7.524003e-01
8	La_Jolla_Cove - Pacific_Beach	-1.28102385	2.001853e-01	3.233762e-01
9	Mission_Bay - Pacific_Beach	0.05757532	9.540869e-01	9.540869e-01
10	Mission_Beach - Pacific_Beach	-0.44630672	6.553757e-01	7.646050e-01
11	La_Jolla Reefs - SIO_La_Jolla_Shores	5.59374261	2.222263e-08	1.166688e-07
12	La_Jolla_Cove - SIO_La_Jolla_Shores	2.60488633	9.190475e-03	1.930000e-02
13	Mission_Bay - SIO_La_Jolla_Shores	5.24101734	1.596937e-07	5.589280e-07
14	Mission_Beach - SIO_La_Jolla_Shores	3.98229276	6.825363e-05	1.791658e-04
15	Pacific_Beach - SIO_La_Jolla_Shores	4.00817764	6.118909e-05	1.835673e-04
16	La_Jolla Reefs - Torrey_Pines	8.32676723	8.307987e-17	1.744677e-15
17	La_Jolla_Cove - Torrey_Pines	3.69293308	2.216824e-04	5.172589e-04
18	Mission_Bay - Torrey_Pines	7.86494145	3.692701e-15	3.877336e-14
19	Mission_Beach - Torrey_Pines	5.60281453	2.108989e-08	1.476292e-07
20	Pacific_Beach - Torrey_Pines	5.31331432	1.076491e-07	4.521263e-07
21	SIO_La_Jolla_Shores - Torrey_Pines	0.79998875	4.237173e-01	5.561290e-01

2.4.5 Integrating Statistical Results into a Molecular Network

<CRITICAL> Statistical outcomes and generated CSV files can be integrated into a molecular network via Cytoscape to help prioritize subsequent MS annotations and full structural identifications. For instance, in this protocol, we demonstrate the integration of Random Forest (RF) analysis results with the GraphML file obtained from FBMN, which is accessible in Cytoscape. The file used for this section is the `Significant_features_RF` file generated in **Step 56** and it contains significant features with a 'MeanDecreaseAccuracy.pval' less than 0.05. The procedure for generating and interpreting this ranked list of significant features is described in **Step 56**. The file can be accessed from our GitHub Repository: https://github.com/Functional-Metabolomics-Lab/FBMN-STATS/blob/main/R/outputs_R_Notebook/Multivariate_results/2023-09-07_Top5percent_ImportantFeatures_RF_500trees_500perm.csv.

Preparing the CSV File for Integration:

80| Start by adjusting the first column of your CSV, which typically contains concatenated identifiers (e.g., X91133_907.259_12.628_NA;THEAFLAVIN DIGALLATE) into a single string. This column represents a unique ID, m/z, retention time, and annotated names.

81| Use the delimiter "_" to separate these components into 4 columns and rename the columns to 'uniqueID', 'mz', 'RT', 'annotated_names' for clarity. Retaining these unique identifiers as the first column in the CSV facilitates their use as a key for integration with the Cytoscape GraphML file.

82| Remove the "X" prefix on the first column 'uniqueID'

83| Add a column labeled "RF significant features" and fill it with "RF significant features". This can be used later in Cytoscape to mark the RF-identified significant features.

84| Once modifications are complete, save this modified CSV for subsequent integration with Cytoscape. Reference for the modified file is available at: https://github.com/Functional-Metabolomics-Lab/FBMN-STATS/tree/main/Integrating_Stats_Results_to_Molecular_Network

Integrating the Data into Cytoscape:

85| After your FBMN job is complete, download the Cytoscape data from the GNPS status page under "**Export/Download Network Files",** and click **"Download Cytoscape Data"**. This zip file contains all the necessary files for integration.

86| Launch Cytoscape and open the GraphML file from your GNPS download.

87| **Import the Modified CSV:** Use the "Import Table from File" feature in Cytoscape to bring in your modified CSV file (**Figure 13 A**; icon of a table and downward pointing arrow). In the "Import Columns from Table" dialog, ensure the settings match those depicted in the accompanying **Figure 13 B**:

- **Import Data as:** Node Table Columns
- **Key Column for Network:** Choose "**shared name**" to align with the "shared name" column in the original node table.
- In the 'Preview' section, make sure the first column of the imported table, "**uniqueID**", is assigned as the key column to merge.
- Verify the type of both these columns to confirm that they are both numeric. Different types will disturb merging.

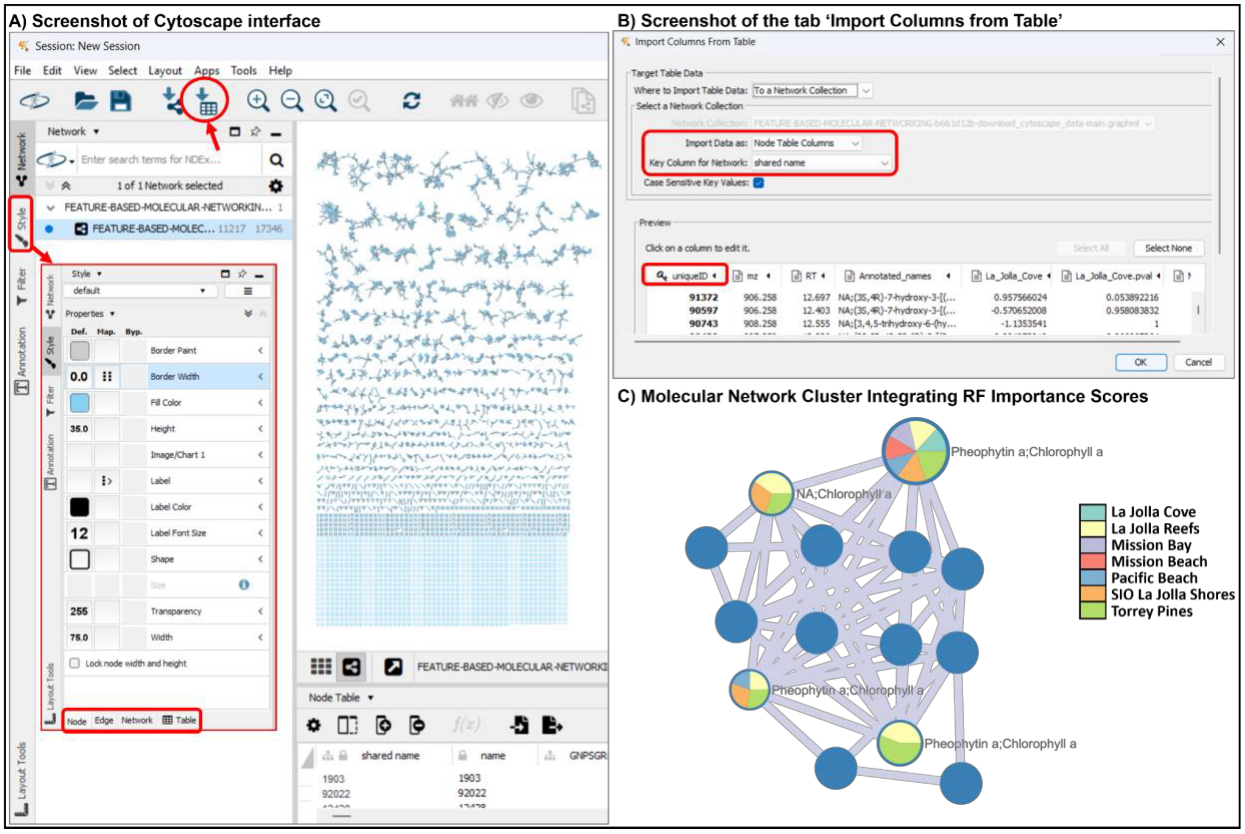


Figure 13: Visual Guide to Integrating Data into Cytoscape Networks: A) Screenshot of the Cytoscape interface showcasing the initial view upon loading a GraphML file from Feature-Based Molecular Networking results. Note that key areas are outlined in red. On the left sidebar with primary icons (Network, Style, Filter, Annotation), 'Network' is currently active (emphasized in gray) and the 'Style' icon is expanded to reveal options for customizing nodes, edges, and more. In addition, the icon 'Import Table from File' from the top menu bar is also circled; B) After selecting 'Import Table from File', the user navigates through a dialog box ('Import Columns from Table') that allows for the merging of the modified csv table into the existing network based on a unique index column present in both tables. Crucial settings such as 'Import Data as: Node Table Columns' and 'Key Column for Network: shared name' are outlined in red. Below, the Preview section shows the modified Random Forest results table, with 'uniqueID' (column containing unique feature IDs) as the key column for merging. C) The figure displays a small example network where nodes are enhanced with Random Forest importance scores. Nodes without statistical significance are depicted in blue, while significant nodes showcase a pie chart visualization. Each segment of the pie-chart represents the importance score for one of the seven locations (as an example, the seven locations share similar importance scores for the uppermost significant node, while Torrey Pines has the highest importance score for the lowermost significant node). The overall node size corresponds to the 'Mean Decrease Accuracy' (MDA) metric.

Visualizing RF Results:

89| To represent MDA scores visually, adjust node border widths under the "Style" tab in Cytoscape (**Figure 13 A**): Navigate to Node → Border Width and select your MDA column for mapping, choosing a passthrough mapping type.

90| Consider differentiating significant RF features, such as changing their shape to squares, by utilizing the "RF significant features" column. This can be done by navigating to Style → Node → Shape, selecting the "RF significant features" column under "Column", and setting "Mapping Type" to "Discrete Mapping". Assign shapes, such as rounded squares, for significant features, as shown in **Figure 13 G**. This is particularly useful if you are incorporating results from several statistical tests.

91| For a comprehensive view, pie charts can illustrate the importance score of each feature across groups (e.g., 7 locations) such as in **Figure 13 C**. This visualization can highlight features with varying significance across the groups, aiding in the prioritization process. To create a pie chart, click on the pie chart icon in the Node section's style menu, labeled "Image/Chart 1". A window will appear where all the available columns can be selected (we used the random forest importance scores for each location), and a "customize" button, where each feature can be assigned a color.

2.5 Troubleshooting

Troubleshooting advice can be found in Table 4.

Common troubleshooting tips for the R Notebook can be found in Table 4 and within each step of the main protocol as needed. Additionally, the Supplementary Information (SI) document provides insights on differences between the R steps and other notebooks, as well as the fbmn-stats app, along with further troubleshooting advice. For any unresolved issues, we recommend submitting an issue on our GitHub page.

Table 4: **Troubleshooting**

Step	**Problem**	**Possible reason**	**Possible solution**
2, 26, 31, 51, 57	Package installation fails or is slow.	'p_load' function from pacman library may not retrieve some packages well.	In such cases, install and load the packages manually: install.packages('tidyverse') library('tidyverse')
3	Incorrect path entry in code cell and it results in error	Users enter the folder path directly in the code cell.	Run the code cell first (shift + Enter in case of the JupyterLab environment or simply press the play button next to the code cell in terms of Colab environment). This will result in a pop-up box asking for path entry. Here, enter the path of the folder that contains the input files. Make sure to run this step and set a working directory as all the resulting files will be stored here
3, 4	While setting the working	Google Colab platform cannot access the local files directly. One can mount their	Make sure to create a folder in the Colab platform and upload your input files there

	directory in google colab, users enter a local path and it fails	google drive and access files from there, but that is possible only with Python and not in R	manually, then set the path of that folder as the working directory
6	Metadata table not read correctly	Metadata was uploaded in incorrect format.	The R Notebook expects the metadata to be in txt or tsv format. So make sure to save it that way and not in csv mode before uploading the file. Else, in step 5, you can change the separator sep = '\t' in the following code block to sep = ',' or ';'. md <- read.csv(file_names[input[2]], header = T, check.names = F, sep = '\t')`
6	Files not read or separated correctly.	Incorrect file formats and separators.	Ensure the feature table is '.csv' and others are '.tsv' or '.txt'. Check that '.csv' is comma-separated and not semi-colon separated because of your regional language settings
17	Filenames in the metadata are missing in the feature table and lists the filenames	Spelling errors, case sensitivity issues, and presence of whitespaces in metadata filenames are common issues, particularly as metadata is often manually entered in Excel.	Check the filenames in metadata against the corresponding column names in the feature table for spelling mistakes, case-sensitive errors. Then reupload the files.
30	Error suggesting	Missing data frames from the previous cleanup steps.	Whether the user wants to perform the data cleanup steps or not, execute all

	missing dataframe		preceding steps to make sure necessary dataframes are available to be retrieved for this step. Here, the user can choose which of these dataframes to be used for the statistical analysis.
35	PCoA plot fails due to color limitations.	In this example, the column used for coloring scores, 'ATTRIBUTE_Month', is preset based on the example dataset. Errors may occur if this column name is not updated when using different data. Additionally, plotting issues may arise if the chosen metadata column has more than the eight available colors, leading to coloring errors in the scores.	Make sure to choose a metadata column with eight or fewer groups, or customize the coloring in the code to fit the number of groups.
39	Color error in plots	Here, the colors are specified for 5 groups. If the chosen attribute has more than 5 groups, it might cause an error.	Specify colors for all groups or allow the function to use default settings by removing the cols command. The plotting function plotPCoA can handle up to 22 groups.
48, 49	Heatmap visualization issues with non-scaled data.	`t(cleaned_data), heatmap_legend_param = list(title = "Scaled/centered\nintensity"), col = circlize::colorRamp2(c(0, 0.5, 1)),`	If you chose a different dataframes (such as raw data, or just blank removed data or TIC normalized data) as your cleaned data at step 30, then adjust color ramp settings accordingly in these steps as circlize::colorRamp2(c(min(cleaned_data), 0, max(cleaned_data))).

		The above code within the heatmap function states that we are using the transposed cleaned data (which is scaled data, in the example) and the color ramp is given as 0, 0.5,1. When using non-scaled data for a heatmap, the specified color ramp may not adequately represent all intensity variations, resulting in poor visualization quality.	Consider using a color (e.g. yellow) for NA values if present by including the following within the function: `na_col = 'yellow'`.

2.6 Timing

The processing times reported here are based on using our example dataset within the Colab platform, with durations estimated from a beginner's perspective. This reflects the time typically required for someone new to complete the analysis. Running these procedures on local Jupyter Notebook systems could significantly decrease these times.

The preliminary setup for the notebook, covering steps 1-17, takes approximately 30-40 minutes, with the package installation alone taking about 10-15 minutes. The data cleaning process, spanning steps 18-30, is estimated to take 20-30 minutes. Multivariate statistical analysis, which includes steps 31-56, generally requires 50-60 minutes, with heatmap generation and Random Forest classification taking the longest time—5-10 minutes for each heatmap generation and 30-60 minutes for Random Forest.

Univariate statistics, covering steps 57-80, also take about 50-60 minutes, with specific steps 59, 61, 66, 73, and 78 each taking 5-10 minutes. Finally, integrating statistical results into a molecular network, detailed in steps 81-91, could take 1-2 hours for users who are new to Cytoscape, as they learn to navigate and utilize the software effectively.

2.7 Anticipated Results

We illustrate the types of results that could be obtained using specific worked example.

Data Refinement and Annotation Insights

In the example data, we investigated the coastal environments along the San Diego coastline from Torrey Pines State Beach to Mission Bay, USA, during different dry and wet seasons. Refer to **Figure 14A** for a spatial map of the sampling locations. The presumption was that post-rain samples from Jan 2018, influenced by runoff, would show increased pollutant levels. From FBMN analysis, we identified 5521 LC-MS/MS features, which decreased to 4384 after removing blanks. The library search against the GNPS spectral library via the FBMN workflow resulted in 92 annotated features out of the 4384 features, and an additional analog search putatively annotated 104 features. Expanding on this, we included additional data from October 2018, collected from the same sites (no-rain period) for our pipeline evaluation. The dataset contained 180 samples from seven different sites at three different time points (Dec 2017, Jan 2018, Oct 2018) and 2 PPL process blanks for each sample time. From this extended dataset, we identified 11217 features, with 260 GNPS library matches and 1991 analog matches. When focusing solely on December and January samples, the feature count surged to 10470, almost double the original count of 5521 features, 240 GNPS library hits, and 1624 analog hits.

To further expand our annotations, we used SIRIUS for in silico spectrum annotation on the extended dataset. We utilized the mgf file obtained from MZmine 3 (version 3.3.0) and extended our SIRIUS analysis using tools like CANOPUS and CSI:FingerID. The SIRIUS result provided annotations for 8255 features, with annotations or compound names available for 5001 features. All 5001 of these features were further characterized by CSI:FingerID, which predicts molecular substructures and scores them based on the likelihood that the substructure belongs to the molecule. Leveraging the predictive capabilities of both SIRIUS and CSI:FingerID, we could infer the most probable molecular formulas. SIRIUS formula identifications were generated for 8885 features, with 5411 of these having an explained intensity greater than 80%, marking them as reliable formulas. For compound class predictions, CANOPUS provided annotations for 8583 features spanning various levels such as Kingdom, Superclass, Class, Subclass, and Level 5. On the other hand, the Natural Product Classifier (NPC) was used to determine if a compound is a natural product. These compound classes can be further explored in tools like Cytoscape for network visualization based on compound classes, or sub-setting of features for subsequent statistics.

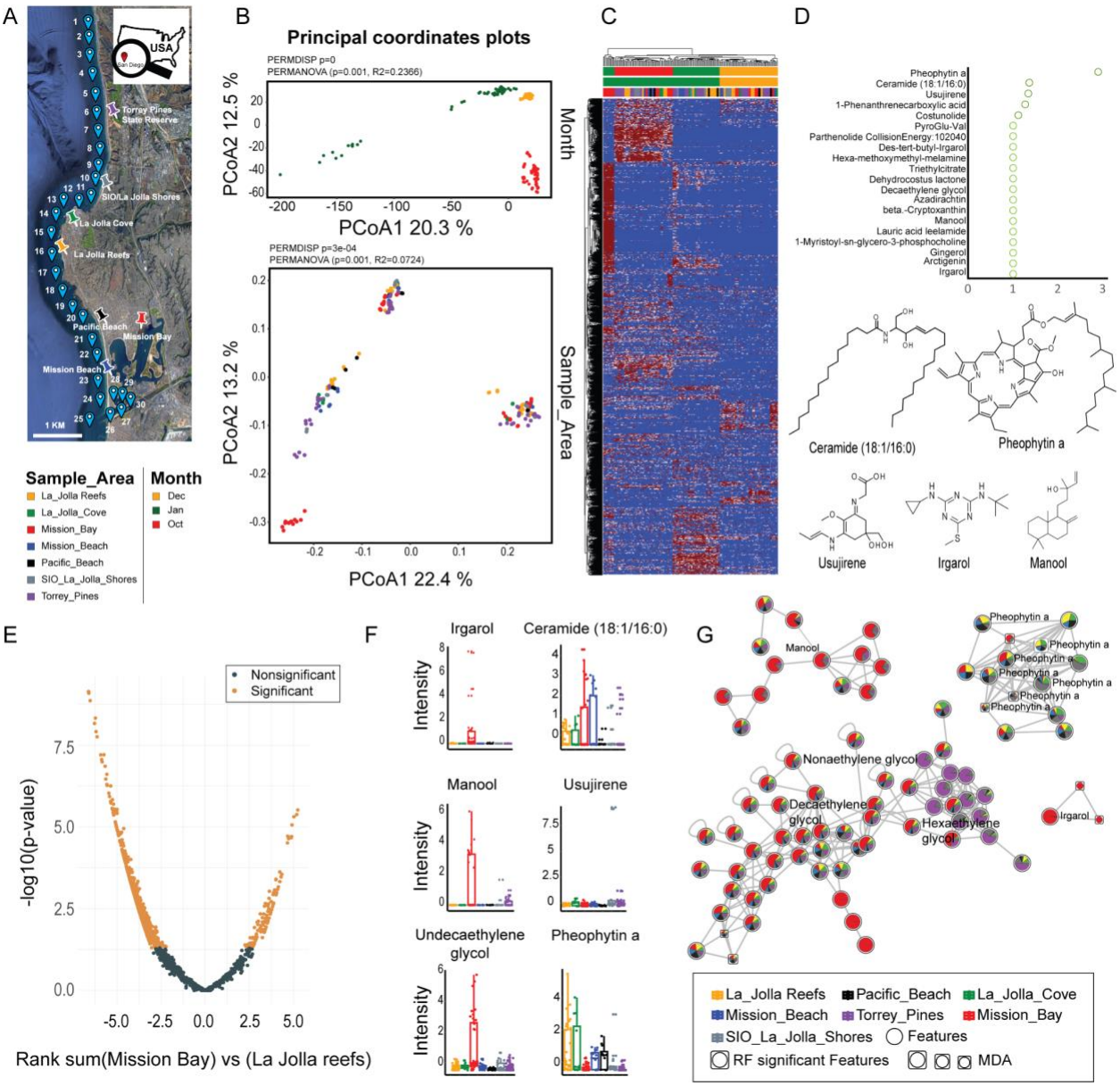


Figure 14: Anticipated results: A) Spatial map pinpointing sampling sites; B) Principal coordinate plots delineating differences by sampling month and location; C) Heatmap displaying scaled feature intensities; D) Top 20 annotated drivers for temporal changes identified via Random Forest (RF), with structures of the top 3 metabolites shown; E) Volcano plot illustrating the Dunn test comparison between Mission Bay and La Jolla Reefs samples, with features deemed significant in the Kruskal-Wallis (KW) Sample area-based test used for this post-hoc analysis; F) Box plots illustrating feature intensities across various sampling locations. The first column presents the foremost 3 annotated significant outcomes post-Dunn test, accompanied by their molecular structures. The second column highlights the top 3 significant outputs as identified by RF; G) Example molecular networks of statistically significant features identified by the KW test (e.g., Irgarol, Manool) and RF analysis (e.g., Pheophytin a, Decaethylene glycol). Each node in the network represents a feature and is visualized with a pie chart representing its intensity distribution across

seven sample locations. Nodes bordered in a square are deemed significant by the RF model. Their node size corresponds to 'Mean Decrease Accuracy' (MDA) scores from the RF analysis, where a larger size denotes a higher significance score.

Impact of Sequential Data Cleanup

Contaminant features, especially those exceeding 30% peak area relative to the sample average, were flagged and removed, leaving us with 9,092 features. Our dataset showed 32% missing values out of 1,636,560 total entries, which were imputed between 1 and the lowest feature value (892). Petras *et al*. found significant organic matter chemotype shifts between December 2017 and January 2018 samples, correlating with January's heavy rainfall[32]. Our extended dataset confirmed this, with a PCoA analysis revealing clear sample groupings by the sampling month as shown in **Figure 14B**. Post-blank removal intensified these groupings. Prior to data cleanup, no dispersion effect was apparent ($p > 0.05$), and PERMANOVA attributed 31% of the variance to sampling months. After removing blanks, however, a dispersion effect emerged. This dispersion effect and explained variance in PERMANOVA are likely due to the removal of background features, thus reflecting the true water sample chemotypes for each month. Upon examining the PCoA after imputation, individual clusters appeared closer together, though January samples exhibited some dispersion. This spread within January samples became more pronounced after normalization and scaling.

Multivariate Analysis: Diving into Site-Specific Variations

Using PERMANOVA on the scaled-imputed data, we identified a significant clustering by months, attributing 34% of variance to the sampling time ($P < 0.05$, Adonis $R2 = 0.34$). Sample locations, however, explained only 7% of the variance. Upon deeper exploration of the metabolic profiles across these sampling locations, January's variance was more prominent in Mission Bay, especially post-rainfall, due to its nutrient-rich status, potentially from increased runoff through the San Diego River. This distinction is evident in the PCoA plot in **Figure 14B**. Our data showed Mission Bay's pre-rainfall samples were similar to other sites, but post-rain samples in January diverged — a pattern absent in December 2017 and October 2018 samples. We could also observe some clear patterns in the heatmap depicted in **Figure 14C**. Color transitions from blue (0 intensity) to red (1 intensity) highlight feature intensity variations. Many features were found in higher intensities in October samples compared to December and January samples. Mission Bay samples from January (in red) and a subset from Torrey Pines (in blue) displayed increased

feature intensities. This aligns well with our initial hypothesis. Alongside this, we performed a random forest classification considering sampling sites.

Random Forest Exploration: Prioritizing Key Drivers

In our example provided in the notebook, we tried to classify surface seawater samples based on their different sampling sites using random forest. Here, the feature quantification table without metadata is the predictor variable, and the metadata group “Sampling Site” is the response variable. The figure in **Box 10** provides a visualization of the Random Forest algorithm and its implementation in R. Utilizing a Random Forest model with 500 trees and 500 permutations, we attained a 68.3% prediction accuracy for the samples. By location, accuracy ranged from 87.5% (Torrey Pines) to 16.7% (Pacific Beach). The confusion matrix in **Table 5** provides insights into these results, revealing that misclassifications were often between neighboring sites, likely due to the close 300-meter spacing between the sampling locations. Our model highlighted 438 significant features (based on the ‘Mean Decrease Accuracy p-value’). Of these, seven matched GNPS libraries and 96 were analog hits. Examining the violin plot results of RF, top features, like those with library IDs 91372 and 90597 (both sharing the same analog name), were mainly concentrated in Mission Bay and La Jolla Reefs. These concentrations began low at Torrey Pines, peaked at Cove and Reef, and saw another spike in Mission Bay. Similar patterns emerged for features like theaflavin digallatae (ID 91133). Some features, such as IDs 33200 and 53617, were notably elevated in Mission Bay alone. Certain compounds from previously reported research, such as *m/z* 1129.3145 (analog name: benzyl-tetradecyl-dimethylammonium) specific to January samples, were also detected in our study, but their significance was marginal ($p = 0.08$) and was predominantly seen in Torrey Pines. Several compounds reported in the original study such as irgarol, recognized for their pollution potential and unique spatial patterns, were also explored in our dataset. **Figure 14D** visualizes the top 20 annotated drivers for site-specific changes as identified via Random Forest, highlighting the structures of the top 5 metabolites. In summary, our extended dataset enhances the Random Forest analysis, offering a detailed understanding of chemotype differences across coastal areas and reaffirming the conclusions of the original study.

Table 5: **Confusion Matrix of Random Forest Classification**

The confusion matrix shows how many samples from each group were correctly predicted. Taking the first row as an example: out of 36 samples from La Jolla Reefs, 25 were accurately identified. The remaining samples were misclassified as follows: 1 as Mission Bay, 1 as Mission Beach, 5

as Pacific Beach, and 4 as SIO La Jolla Shores. The column labeled ‘pct.correct’ represents the percentage of samples that were correctly classified for a given group. The columns ‘LCI 0.95’ and ‘UCI 0.95’ denote the lower and upper bounds of the 95% confidence interval for each group, respectively. The ‘overall’ row at the bottom indicates the model’s total prediction accuracy, which stands at 68.3% for this dataset.

	La Jolla Reefs	**La Jolla Cove**	**Mission Bay**	**Mission Beach**	**Pacific Beach**	**SIO La Jolla Shores**	**Torrey Pines**	**pct.correct**	**LCI 0.95**	**UCI 0.95**
La Jolla Reefs	25	0	1	1	5	4	0	69.4	51.89	83.7
La Jolla Cove	0	10	0	0	0	2	0	83.3	51.59	97.9
Mission Bay	4	0	23	7	2	0	0	63.9	46.22	79.2
Mission Beach	0	0	0	15	3	0	0	83.3	58.58	96.4
Pacific Beach	6	0	1	3	2	0	0	16.7	2.09	48.4
SIO La Jolla Shores	2	0	0	1	0	6	9	33.3	13.34	59
Torrey Pines	0	0	0	0	0	6	42	87.5	74.75	95.3
Overall	NA	NA	NA	NA	NA	NA	NA	68.3	61	75.1

Univariate Analysis Insights

In our univariate analysis of 9092 features, we used both ANOVA and the Kruskal-Wallis tests to demonstrate to users how these methods can be utilized with the data and to offer insight into the potential results. However, the Kruskal-Wallis test was considered more appropriate due to the non-normal distribution of most features' relative intensity values in our dataset. The Kruskal-Wallis test highlighted 1258 significant features, including irgarol, an antifouling agent used on boats. Conversely, ANOVA pinpointed 1554 significant features, with many features having a pronounced abundance in Mission Bay compared to other sites.

Building on the Kruskal-Wallis results, Dunn's post-hoc test was used to highlight pairwise differences. Given the pronounced abundance of many features in Mission Bay, we compared it with La Jolla Reefs for further insights. The significant and non-significant features from this test are visualized in the volcano plot in **Figure 14E**. Notably, compounds like irgarol and manool were significantly higher in Mission Bay. In contrast, La Jolla Reefs had a higher presence of the natural product 'pheophytin a'. The top row in **Figure 14F** displays the intensities of the top three annotated results from the Dunn test across the sampling locations using box plots. These findings align with and reinforce the initial observations, validating the robustness of our analytical workflow.

Integration of Molecular Networking Results

After the statistical analysis of the FBMN results and prioritization of features that drive the chemical differences between the sampling sites, we further investigate related compounds, through the molecular networks. **Figure 14G** shows the networks of polyethylene glycols, indicating that many of the structurally related features of those compounds show similar spatial distribution, with the highest abundance in Mission Bay, as indicated through the pie charts on top of the network nodes. It's important to note that these are example networks selected from the larger overarching FBMN network. The resulting Cytoscape files for these and the complete network are available on our GitHub repository (https://github.com/Functional-Metabolomics-Lab/FBMN-STATS/tree/main/Integrating_Stats_Results_to_Molecular_Network) for further reference. These results show nicely how statistical prioritization and further structure-based (in our case, based on MS/MS similarity) can work hand in hand to structure the observed chemical space. Besides investigating the networks after the statistical interrogation, one can also make use of the scores obtained from the different tests and visualize those in the network. For example, the fold change and p-values from the univariate analysis or mean decreased accuracies from the supervised multivariate analysis can be imported as a new attribute to the networks with tools

such as Cytoscape to combine visual and statistical prioritization directly in the network. For guidance on this integration process, please refer to the ‘Integrating Statistical Results into a Molecular Network’ section.

2.8 Conclusion

In this protocol, we provide a comprehensive data clean-up and statistics pipeline for the analysis of non-targeted metabolomics data. Our protocol spans from initial data conversion, blank removal, imputation, and normalization/scaling to uni- and multivariate statistics and data interpretation. While our outlined workflow is as detailed and structured as possible, which should provide a comprehensive analysis solution for many scientific questions, it is important to point out that there is not a universal solution that fits every scenario. We emphasize the importance of transparency in reporting details on every step of the metabolomics pipeline, such as providing the specific normalization methods, explaining the distance metrics in multivariate analysis, or specifying parameters like the number of trees in a Random Forest model. Furthermore, in relation to our case study, the sharing of feature detection and annotation settings and batch files further augments reproducibility. Together, with open data deposition, the above steps ensure both transparency and reproducibility of metabolomics experiments.

We would also like to stress again that cataloging and identifying statistically significant compounds is just the beginning. To fully understand the relationships between metabolites, xenobiotics, and the underlying biological processes, additional experiments and orthogonal verification are typically required. Once the statistical results are studied, techniques such as pathway enrichment analyses can illuminate the multifaceted relationships between metabolites and the related biological processes. When specific compounds are of particular interest, targeted metabolomics stands as a powerful next step.

The versatility of our protocol extends to a wide range of fields and sample types, including combinatorial chemistry, doping analysis, and trace contamination of food, pharmaceuticals, and other industrial products. It is equally applicable to biological samples from diverse origins, such as microbiomes, bioreactors, or biomedical research, provided the data adheres to the feature table and metadata format specifications. Beyond sample classification and feature prioritization, our protocol facilitates the integration of statistical findings into molecular networks, allowing users to visualize complex chemical spaces across various research domains.

In summary, we anticipate that our guide to statistical analysis of FBMN results will provide both a theoretical and practical resource for scientists working with non-targeted metabolomics data.

For novices in the field, the scripts, app, and detailed step-to-step protocol provide a starting point with a set of statistical analysis solutions for many biological questions, whereas experts may accelerate parts of their statistical workflows.

2.9 Data Availability

The FBMN results are available at https://gnps.ucsd.edu/ProteoSAFe/status.jsp?task=b661d12ba88745639664988329c1363e. Raw and processed data are available through the MassIVE repository: MSV000082312 and MSV000085786 and through Zenodo (https://zenodo.org/records/10051610) with the following doi https://doi.org/10.5281/zenodo.10051610.

Code Availability

All code and software are available through GitHub under the following link https://github.com/Functional-Metabolomics-Lab/FBMN-STATS. The web application can be accessed at https://fbmn-statsguide.gnps2.org/. Downloadable Windows executables of the web app is available from https://www.functional-metabolomics.com/resources. All the codes are deposited on Zenodo with the following doi: https://doi.org/10.5281/zenodo.11350947.

Acknowledgment

We thank Greg Caporaso for guidance on preparing the QIIME2 plugins. DP, CM, and HPL were supported by the Deutsche Forschungsgemeinschaft (DFG) through the CMFI Cluster of Excellence (EXC 2124), and DP and CM, were supported by the DFG through the Collaborative Research Center CellMap (TRR 261). KD was supported by the DFG (BO 1910/23). PS was supported by the European Union's Horizon Europe research and innovation programme through a Marie Skłodowska-Curie fellowship No. 101108450 MeStaLeM. TP was supported by the Czech Science Foundation (GA CR) grant 21-11563M and by the European Union's Horizon 2020 research and innovation programme under Marie Skłodowska-Curie grant agreement No. 891397. TD was supported by the MSCA Fellowships CZ (OP JAK) grant CZ.02.01.01/00/22_010/0002733. MW was supported by the National Institutes of Health (NIH) with grants 1U24DK133658-01, NIH 1R03DE032437-01, and UC Riverside startup funding. EEK was supported by grants from the Novo Nordisk Foundation [NNF20CC0035580, NNF16OC0021746]. YW was supported by NIH 1R03DE032437-01. CB was supported by the Czech Academy of Sciences (CAS PPLZ) L200552251. FO was supported by FAPESP

2022/14603-8. JB's work was carried out as part of the German Center for Infection Research (DZIF) project 09.720. We thank Libera Lo Presti for critical reading of the manuscript.

Author Contribution

AKPS, FO, FR, ME, and DP conceptualized the protocol. YE, SZ, JS, RS advised on the concept and statistical test. AKPS, AW, FO, FR, MN, JB, EEK, JE, AP, CGM, SF, NC, YW, MD, JS, MW, and ME wrote code. AKPS, AW, and MW developed and deployed the web application. RS, ATA and DP collected the water samples. DP extracted the samples and acquired the LC-MS/MS data. AKPS, AW, FO, FR, MN, JB, JJK, EEK, JE, AP, CGM, SF, MRA, TP, NC, MP, CB, BC, AMCR, AC, Fd, KD, YE, CG, LGG, MH, SH, SK, AK, MCMK, KM, SP, PWP, TS, KSL, PS, ST, GAV, BCW, SX, MTY, SZ, Md, CB, HPL, CM, JJJvdH, TD, PCD, JS, RS, ATA, ME, and DP tested the protocol, code and app. CB, JJJvdH, TP, MW, ATA, ME, and DP supervised students and researchers. MW, AA, ME, and DP supervised the project. AKPS, MN, JB, JJK, EEK, AP, SF, TP, ATA and DP wrote the manuscript and supplemental information. FO, FR, JE, CGM, MRA, NC, MP, KD, YE, LGG, MH, SH, PS, GAV, SZ, JJJvdH, TD, TP, PCD, JS, RS, MW, and ME edited and provided critical feedback on the first draft. All authors edited and approved the final draft.

Competing Interests

JJJvdH is currently a member of the Scientific Advisory Board of Naicons Srl., Milano, Italy, and is consulting for Corteva Agriscience, Indianapolis, IN, USA. PCD is a scientific advisor and holds equity to Cybele and a Co-founder, advisor and holds equity in Ometa, Arome and Enveda with prior approval by UC-San Diego and consulted in 2023 for DSM animal health. MW is the founder of Ometa Labs. SH, TP, and RS are co-founders of mzio GmbH.

Additional information & Supplementary information

Supplemental information, including a cheat-sheet, detailed methods for the LC-MS/MS data acquisition and step-to-step guides for the Python and QIIME2 scripts as well as the web application are available in the supplemental information.

Chapter 3

A Metabolomics Framework to Track Microbiome Drug Metabolism

Abzer K. Pakkir Shah[1,2,3], Anne Griesshammer[1,2,4], Paolo Stincone[1,2], Jarmo-Charles Kalinski[3,5], Axel Walter[1,2], Mingxun Wang[6], Lisa Maier[1,2,4], Daniel Petras[1,2,3,#]

1. Interfaculty Institute for Microbiology and Infection Medicine Tübingen, University of Tübingen, Tübingen, Germany
2. Cluster of Excellence EXC 2124 Controlling Microbes to Fight Infections, University of Tübingen, Tübingen, Germany
3. Department of Biochemistry, University of California Riverside, Riverside, CA, USA
4. M3 Research Center for Malignome, Metabolome and Microbiome, University Hospital Tübingen, Tübingen, Germany
5. Rhodes University, Makhanda, South Africa
6. Department of Computer Science, University of California Riverside, Riverside, CA, USA

Correspondence should be addressed to Daniel Petras

Note: A revised version of this chapter is available as a preprint: Pakkir Shah AK *et al. A Functional Metabolomics Framework to Track Microbiome Drug Metabolism.* bioRxiv (2026). doi: 10.64898/2026.01.30.702925.

3.1 Abstract

Understanding how gut microbes transform drugs, and how this influences microbiome composition and function remains a key question to better understand human health. To accelerate the discovery of microbiome derived drug metabolites, we developed a non-targeted metabolomics workflow that combines the use of synthetic microbial communities (SynComs) with a time-series resolved molecular networking approach, and advanced metabolite annotation. We demonstrate how this chemical proportionality approach can be used to illuminate chemical transformation dynamics in a gut SynCom with 50 clinical drugs. Our results highlight a multitude of drug metabolites, including multi-step metabolic cascades, some of which correlated to shifts in microbial taxa, suggesting functional links between microbiome composition and biochemical transformations. Our chemical proportionality software is publicly available through the GNPS2

ecosystems at https://chemprop.gnps2.org/, which can be used to prioritize biotransformations and other (bio)chemical reactions in various biological and abiotic systems.

Keywords: Non-Targeted Metabolomics; Functional Metabolomics; Computational metabolomics; Gut microbiome; Drug metabolism; Microbial biotransformation; Transformation directionality.

3.2 Introduction

The human gastrointestinal (GI) tract hosts an estimated 10^{13} microbes that perform essential functions, including immune modulation, defense against pathogens, and nutrient processing[74,245–247]. Among these roles, the ability of gut microbes to chemically modify xenobiotics (e.g. pharmaceuticals, dietary molecules, and environmental pollutants) is gaining increasing recognition[74]. These microbial transformations, observed in both humans and animal models, can significantly alter the bioactivity, bioavailability, and toxicity of xenobiotics[248,249]. Unlike the host, which primarily relies on oxidative and conjugative reactions (e.g., via cytochrome P450s) to detoxify compounds, gut microbes predominantly perform hydrolysis and reduction[249]. These transformations often arise from broad substrate promiscuity rather than evolutionary selection. However, identifying the microbial contributors to these transformations remains challenging due to horizontal gene transfer and strain-level variation, and the limited predictability of function from taxonomy alone[74]. Beyond these canonical microbial reactions[250], recent studies shows oxidation, demethylation, and conjugative transformations as well, including amino acid and peptide conjugations distinct from hepatic Phase II metabolism[251–254]. Furthermore, the microbiome function is highly personalized**,** shaped by environmental factors such as diet and medication, and can influence how drugs are processed or tolerated, underscoring the need to investigate microbiome-mediated transformations at scale[247,255,256].

In parallel, there is increasing awareness of how drugs can directly affect the microbiome itself. Antibiotics, while essential for fighting infections, are known to disrupt commensal bacterial populations, leading to dysbiosis and associated health consequences such as *Clostridium difficile* infections, metabolic dysfunction, allergic, and inflammatory disorders[257–261]. Despite growing recognition of these effects, the activity spectrum of different antibiotic classes on gut bacteria remains poorly characterized, largely due to technical challenges in culturing anaerobic species and limited susceptibility data for many common or clinically relevant gut microbes[260,262]. Beyond antibiotics, recent studies have demonstrated that non-antibiotic drugs, including

antipsychotics, proton pump inhibitors (PPIs), and NSAIDs, can also inhibit the growth of gut microbes. Maier et al.[263] screened over 1,000 marketed drugs against 40 representative gut strains and found that nearly a quarter (24% of these human-targeted compounds) inhibited at least one strain. Findings from human cohort studies[54,264–266] similarly identify several non-antibiotic drugs as risk factors for microbiome disruption and infection similar to antibiotics.

Enzymatic activity from gut microbes can profoundly alter the fate of xenobiotics, enhancing or reducing their activity, generating toxic byproducts, altering stability, affecting absorption, or accelerating elimination[252]. Despite these consequences, the reaction dynamics of such microbiome-mediated chemical transformations are rarely studied. In time-series metabolomics datasets, which can contain thousands of features, it is not only the parent drug (i.e., its precursor ion $[M+H]^+$) and its final products that are relevant, but also the transient intermediates formed along the way. Inferring directionality is essential for reconstructing transformation pathways, distinguishing precursors from products, and identifying intermediates that may be bioactive or toxic[252].

To address the complexity of microbiome-drug interactions, recent efforts have leveraged multi-omics approaches[54,55,267–275], used non-targeted LC-MS to systematically profile the depletion of 271 oral drugs by 76 gut bacterial strains. For selected cases, they used LC-MS/MS to confirm microbial metabolites, revealing the widespread capacity of gut microbes to transform xenobiotics. However, most methods are tailored for known metabolic reactions and targeted enzyme assays and are not readily applicable for data-driven discovery of microbiome-associated transformations in complex metabolomics datasets.

Molecular networking offers a powerful data analysis strategy to organize and annotate non-targeted MS/MS data, which allows for the rapid prioritization of putative analogs and transformation products[124,276]. While several software tools have been developed to annotate chemical transformations in metabolomics data, most focus on structure prediction rather than transformation dynamics. For example, BioTransformer uses curated reaction rules and machine learning to predict phase I and II metabolism products[62], and MetWork integrates predicted enzymatic reactions with spectral simulation to suggest candidate structures within molecular networks[277]. While non-targeted metabolomics offer a powerful tool to capture microbial chemical activity, most workflows do not infer transformation directionality or abundance changes over full time series.

Emerging functional metabolomics tools expand the comprehensive measurement of metabolites in a biological system and link them to specific biological roles by probing their roles in phenotypes or molecular interactions[278]. Such strategies typically integrate orthogonal approaches such as perturbation, including gene knock-outs or knock-downs, protein binding, and bioactivity screens[92,279,280]. To address the challenge of prioritizing microbiome derived drug metabolism, we developed a functional metabolomics framework that integrates the use of synthetic microbial communities (SynComs), 16S amplicon sequencing, non-targeted metabolomics, and a correlation-based molecular networking workflow (ChemProp2), as well as repository-scale metabolite contextualization. By resolving transformation directionality and mapping multi-step cascades, ChemProp2 enables the identification of microbial drug metabolism and the linking of specific transformations to shifts in microbial community composition.

Here, we highlight the application of our workflow to investigate microbial metabolism of 50 clinically relevant drugs by a gut SynCom (Com20), uncovering drug-specific transformation patterns associated with microbial shifts. To further contextualize unannotated xenobiotic metabolites, we incorporated repository-scale searches with the FASST software tool[281], enabling a broader interpretation of their biological occurrence and relevance. This integrated approach supports the large-scale matching and contextualization of microbiome-mediated chemical transformations to publicly available non-targeted metabolomics studies. Together these tools provide a framework to better understand microbiome drug metabolisms, which could be leveraged in future drug discovery and personalized medicine approaches.

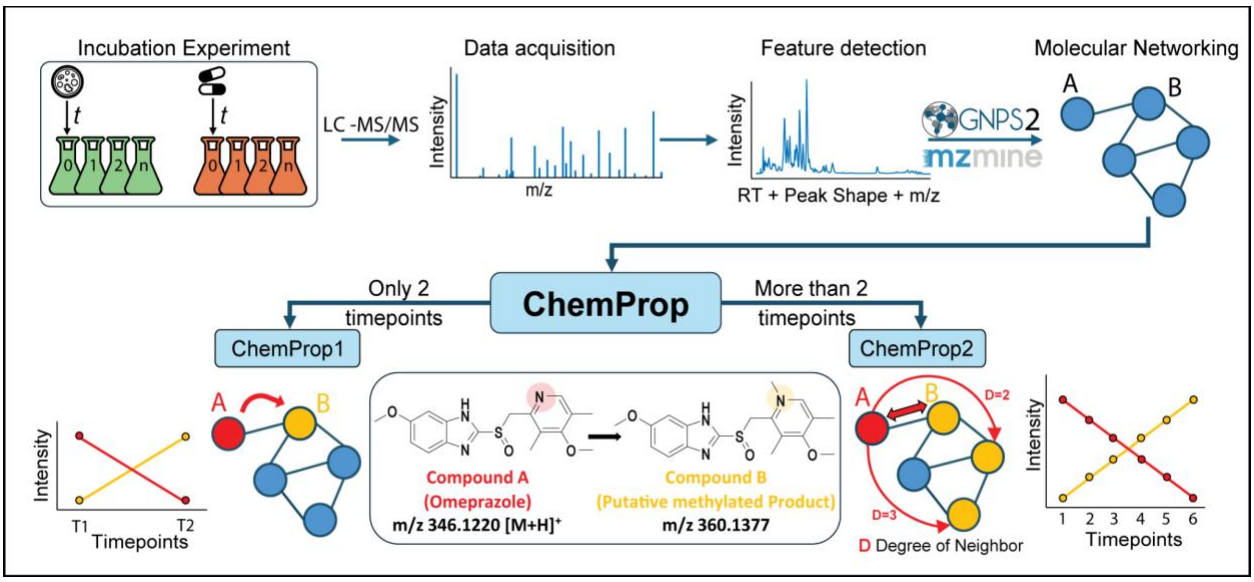


Figure 1: Schematic overview of the ChemProp workflow for detecting directional biotransformations. Starting from a microbial incubation experiment, non-targeted LC-MS/MS data are collected and processed using MZmine for feature detection. Feature-based molecular networking (FBMN) is then performed via GNPS2 to organize structurally related features. ChemProp scoring is applied to the resulting network: ChemProp1 supports two-time-timepoint designs, while ChemProp2 analyzes multi-timepoint data using correlation-based scoring. Both modules prioritize network edges based on anti-correlated intensity patterns to infer directional transformations. The middle panel shows the molecular structures of omeprazole and a putative methylated metabolite (*m/z* 360.1377), represented here as an example from our dataset. (The ChemProp web application is available at https://chemprop.gnps2.org)

3.3 Results and Discussion

3.3.1 ChemProp Infers Direction of Microbiome-Driven Drug Metabolism

To evaluate drug biotransformation within Com20, we applied the ChemProp computational pipeline (**Figure 1**). ChemProp1 computes ratio-based scores between feature pairs in molecular networks to infer putative reaction directionality and is particularly suited for endpoint designs[282] (**Figure 2A-B**). However, endpoint ratios cannot capture dynamic changes across multiple timepoints. To address this, we developed **ChemProp2**, an extension that integrates full time-series information using correlation-based scoring (**Figure 4A**) and provides an empirical false discovery rate (FDR) procedure via a decoy-based strategy[283,284]. These scores range from -1 to +1, where the magnitude reflects strength and the sign indicates the inferred direction of potential transformation. By comparing scores derived from randomized (decoy) feature tables to real networks, users can apply thresholds (e.g., 1%, 5%, 10% FDR) to prioritize high-confidence transformations. Importantly, ChemProp2 allows for analysis of primary edges (i.e., direct connections to the drug's precursor ion node) and distal cascade-level relationships, capturing subtle or multi-step transformation events.

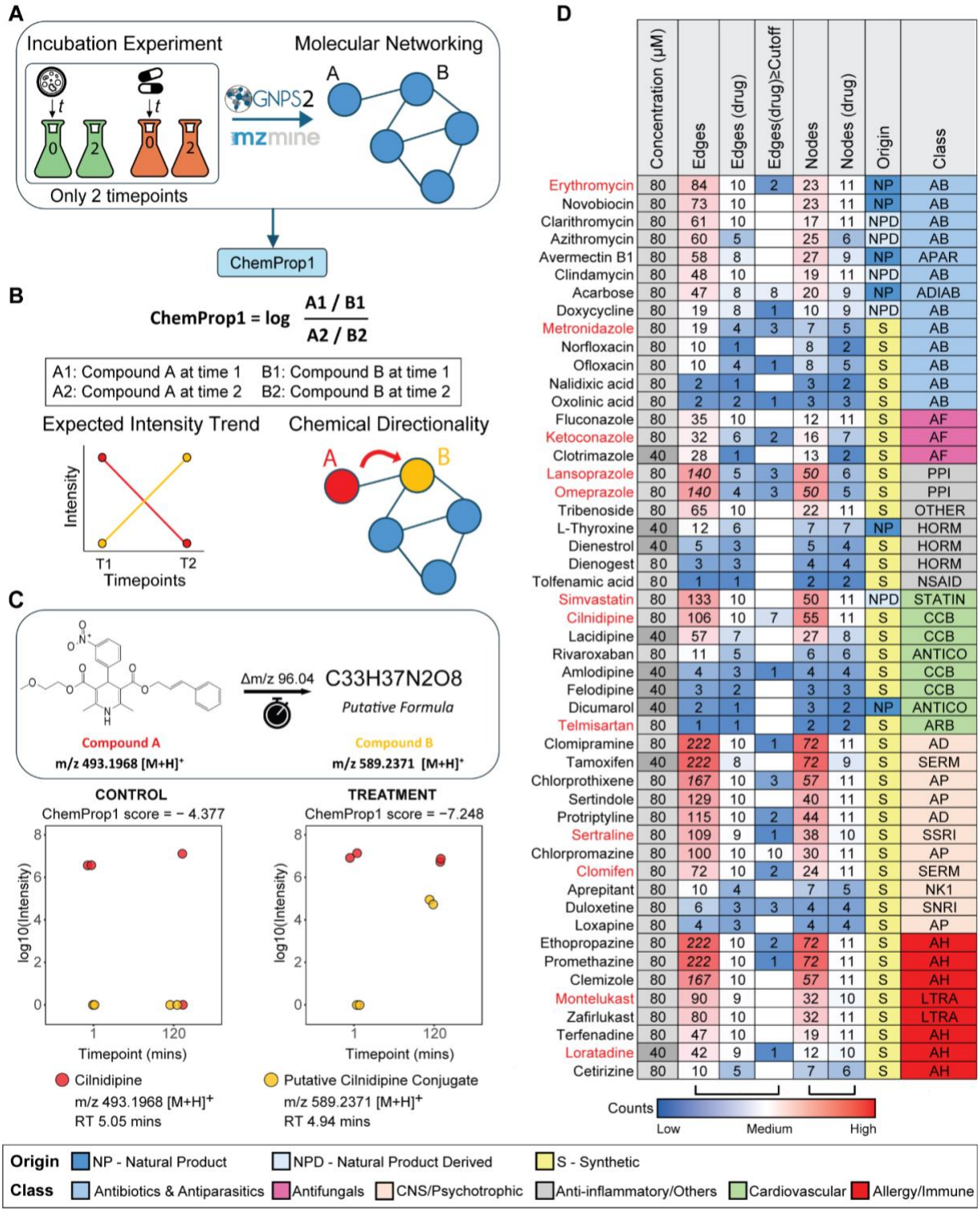
To make ChemProp broadly accessible, we developed a web application (https://chemprop.gnps2.org/) as part of the metabo-apps[285] in the GNPS2 ecosystem. The app allows users to upload feature quantification tables, metadata, and molecular network edge files. It returns scored edge tables with directional information as CSV files, as well as GraphML files for network visualization. Users can interactively filter by score range, focus on specific subnetworks, and inspect corresponding intensity trends for selected feature pairs. (See Supplementary Methods). A summary of the overall approach and decision points for using ChemProp1 versus ChemProp2 is illustrated in **Figure 1**. By integrating ChemProp2 with Feature-

Based Molecular Networking (FBMN), non-targeted metabolomics data can be coupled with time-resolved scoring to suggest potential biological directionality (e.g., drug to product) within otherwise undirected molecular networks.

3.3.2 Initial Screening and Drug Prioritization

We employed Com20, a previously established synthetic gut bacterial community comprising 20 commensal bacterial strains spanning six phyla, 11 families, and 17 genera, collectively encoding around 61 % of the metabolic pathways found in a healthy human gut microbiome[54]. The community showed stable and reproducible growth in gut-mimetic mGAM medium under anaerobic conditions[54,286]. To investigate microbiome-mediated biotransformation of small molecules, Com20 was incubated with 50 clinically relevant drugs under anaerobic conditions. Samples were collected at T = 0 h (immediately after drug addition) and T = 2 h (after incubation) and analyzed using non-targeted LC-MS/MS (**Figure 2A-B**). This two-timepoint setup, referred to as the ChemProp1 model, served as a preliminary experiment to identify drugs exhibiting measurable transformation patterns (See example at **Figure 2C**).

All 50 drugs had confidently detected precursor $[M+H]^+$ ions and surrounding network connections in FBMN. However, the number of direct edges connected to each precursor ion did not necessarily correlate with transformation likelihood as captured by the ChemProp1 scores. **Figure 2D** summarizes the ChemProp1 screen, showing the number of network connections per drug and how many suggested potential biotransformation for each drug. From this, we selected 12 compounds for time-resolved ChemProp2 analysis. Nine (Cilnidipine, Clomifen, Erythromycin, Ketoconazole, Lansoprazole, Loratadine, Metronidazole, Omeprazole, Sertraline) showed at least a two-fold change between 0 h and 2 h, suggesting potential microbial conversion and making them the strongest candidates for follow-up. To also evaluate ChemProp2's sensitivity in low-signal or borderline cases, we included three additional drugs (Montelukast, Simvastatin, and Telmisartan) that did not pass the ChemProp1 cutoff but still showed network connectivity. Together, the selected compounds span six major therapeutic categories represented in **Figure 2D**.



	Concentration (μM)	Edges	Edges (drug)	Edges(drug)≥Cutoff	Nodes	Nodes (drug)	Origin	Class
Erythromycin	80	84	10	2	23	11	NP	AB
Novobiocin	80	73	10		23	11	NP	AB
Clarithromycin	80	61	10		17	11	NPD	AB
Azithromycin	80	60	5		25	6	NPD	AB
Avermectin B1	80	58	8		27	9	NP	APAR
Clindamycin	80	48	10		19	11	NPD	AB
Acarbose	80	47	8	8	20	9	NP	ADIAB
Doxycycline	80	19	8	1	10	9	NPD	AB
Metronidazole	80	19	4	3	7	5	S	AB
Norfloxacin	80	10	1		8	2	S	AB
Ofloxacin	80	10	4	1	8	5	S	AB
Nalidixic acid	80	2	1		3	2	S	AB
Oxolinic acid	80	2	2	1	3	3	S	AB
Fluconazole	80	35	10		12	11	S	AF
Ketoconazole	80	32	6	2	16	7	S	AF
Clotrimazole	40	28	1		13	2	S	AF
Lansoprazole	80	*140*	5	3	*50*	6	S	PPI
Omeprazole	80	*140*	4	3	*50*	5	S	PPI
Tribenoside	80	65	10		22	11	S	OTHER
L-Thyroxine	40	12	6		7	7	NP	HORM
Dienestrol	40	5	3		5	4	S	HORM
Dienogest	80	3	3		4	4	S	HORM
Tolfenamic acid	80	1	1		2	2	S	NSAID
Simvastatin	80	133	10		50	11	NPD	STATIN
Cilnidipine	80	106	10	7	55	11	S	CCB
Lacidipine	40	57	7		27	8	S	CCB
Rivaroxaban	80	11	5		6	6	S	ANTICO
Amlodipine	40	4	3	1	4	4	S	CCB
Felodipine	40	3	2		3	3	S	CCB
Dicumarol	40	2	1		3	2	NP	ANTICO
Telmisartan	80	1	1		2	2	S	ARB
Clomipramine	80	222	10	1	72	11	S	AD
Tamoxifen	40	222	8		72	9	S	SERM
Chlorprothixene	80	*167*	10	3	*57*	11	S	AP
Sertindole	80	129	10		40	11	S	AP
Protriptyline	80	115	10	2	44	11	S	AD
Sertraline	80	109	9	1	38	10	S	SSRI
Chlorpromazine	80	100	10	10	30	11	S	AP
Clomifen	80	72	10	2	24	11	S	SERM
Aprepitant	80	10	4		7	5	S	NK1
Duloxetine	80	6	3	3	4	4	S	SNRI
Loxapine	80	4	3		4	4	S	AP
Ethopropazine	80	222	10	2	72	11	S	AH
Promethazine	80	222	10	1	72	11	S	AH
Clemizole	80	*167*	10		*57*	11	S	AH
Montelukast	80	90	9		32	10	S	LTRA
Zafirlukast	80	80	10		32	11	S	LTRA
Terfenadine	80	47	10		19	11	S	AH
Loratadine	40	42	9	1	12	10	S	AH
Cetirizine	80	10	5		7	6	S	AH

Figure 2: Initial Screening of 50 Drugs Using ChemProp1 (A) Experimental overview: 50 clinical drugs were incubated with the Com20 synthetic gut community under anaerobic conditions and sampled at two timepoints (T0, T2; 2 h apart). Extracts were analyzed by untargeted LC-MS/MS, processed through FBMN, and scored using ChemProp1. (B) ChemProp1 scoring formula based on

log-ratio intensity changes between timepoints to infer precursor-product directionality. (C) Example transformation: Cilnidipine (*m/z* 493.1968) to a putative downstream product (*m/z* 589.2371; *Δm/z* = 96.0403). Scatter plots show log-intensity trends in Com20 vs abiotic control. (D) Heatmap summary of ChemProp1 results across all 50 drugs. For each compound we report total nodes and edges in the main $[M+H]^+$ cluster, first-degree connections, and edges above the ChemProp1 threshold (score ≥ 1). Color scale represents edge counts (red = high, white = mid, blue = low/none). Columns additionally indicate drug origin (NP, NP-derived, synthetic) and drug class. Full class abbreviations are listed in Supplementary Table 1. Drugs marked in red were selected for ChemProp2. Italicized values (e.g., Omeprazole/Lansoprazole: 140 edges) denote that the drugs belonged to the same cluster.

3.3.3 Time-Resolved Analysis of Drug Biotransformations

To investigate microbiome-mediated transformations in greater detail, we decided to use the 12 prioritized drugs and incubated them with Com20 across a 0-8 h time course, sampled at nine timepoints (T0-T8, hourly) **(Figure 3A).** After LC-MS/MS analysis, the resulting networking and ChemProp2 results showed temporal correlations between connected features, enabling directional inference of precursor-product relationships.

Principal Coordinates Analysis (PCoA) using Bray-Curtis dissimilarity (**Figure 3B**) was performed on the metabolomics dataset (8,055 features, reduced to 5,321 after background removal, imputation and TIC normalization). PCoA showed clear separation of metabolic profiles across treatments (PERMANOVA p = 0.001, R^2 = 0.85; PERMDISP p = 0). Drug-specific clustering in PCoA was most pronounced for Ketoconazole and Telmisartan, suggesting divergent metabolic profiles over the 0-8 h time course. For Telmisartan, this divergence became apparent in the extended multi-timepoint dataset, whereas ChemProp1 captured no transformation signal at the 0-2 h range. Notably, this separation reflects shifts in global metabolite profiles captured by Bray-Curtis PCoA and is independent of cascade depth or network size, as shown by the varying FBMN cluster sizes in **Figure 3D**. While angiotensin receptor blockers have been reported to undergo microbiome-associated transformation[48] and alter microbial composition in vivo[287,288], the specific features driving Telmisartan's separation here remain unclear and warrant further investigation. The result highlights that drugs with low initial transformation signals can nonetheless produce distinct metabolic trajectories when monitored over time.

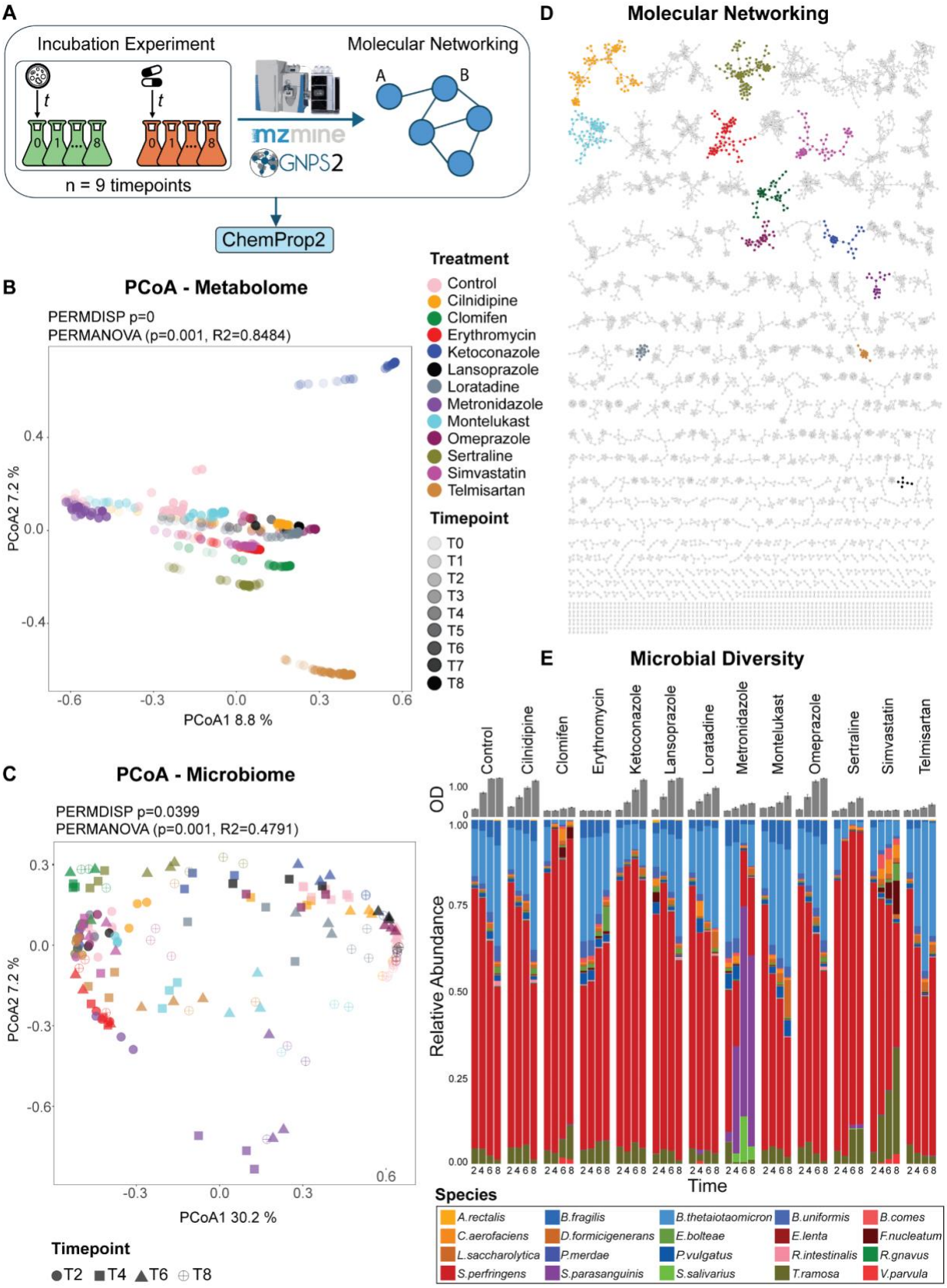

A
Incubation Experiment
Molecular Networking
t
0
1
...
8
mzmine
GNPS2
A
B
n = 9 timepoints
ChemProp2
B
PCoA - Metabolome
PERMDISP p=0
PERMANOVA (p=0.001, R2=0.8484)
PCoA2 7.2 %
PCoA1 8.8 %
Treatment
Control
Cilnidipine
Clomifen
Erythromycin
Ketoconazole
Lansoprazole
Loratadine
Metronidazole
Montelukast
Omeprazole
Sertraline
Simvastatin
Telmisartan
Timepoint
T0
T1
T2
T3
T4
T5
T6
T7
T8
C
PCoA - Microbiome
PERMDISP p=0.0399
PERMANOVA (p=0.001, R2=0.4791)
PCoA2 7.2 %
PCoA1 30.2 %
Timepoint
T2 T4 T6 T8
D
Molecular Networking
E
Microbial Diversity
OD
Relative Abundance
Time
Species
A.rectalis
B.fragilis
B.thetaiotaomicron
B.uniformis
B.comes
C.aerofaciens
D.formicigenerans
E.bolteae
E.lenta
F.nucleatum
L.saccharolytica
P.merdae
P.vulgatus
R.intestinalis
R.gnavus
S.perfringens
S.parasanguinis
S.salivarius
T.ramosa
V.parvula

Figure 3: Time-Resolved Multi-Omics Profiling of Microbiome-Drug Interactions. (A) Experimental design: 12 drugs were incubated with the Com20 microbial community and sampled at nine timepoints (T0-T8). Samples were analyzed by non-targeted LC-MS/MS and 16S rRNA sequencing. Metabolomics data were processed via FBMN and subjected to ChemProp2 scoring to identify directional biotransformations. (B) Principal Coordinates Analysis (PCoA) of metabolomics profiles based on Bray-Curtis dissimilarity across all timepoints (T0-T8), showing distinct trajectories for each drug treatment. (C) PCoA of OD-corrected microbial community composition based on 16S rRNA sequencing at four timepoints (T2, T4, T6, T8) using Bray-Curtis dissimilarity, highlighting treatment-specific differences. (D) FBMN molecular network of all detected features, with the $[M+H]^+$ ion cluster of each of the 12 drugs highlighted in distinct colors. (E) Stacked bar plots of microbial species-level composition at the four timepoints under drug treatment versus control conditions, illustrating dynamic shifts in community structure over time.

At the community level, PCoA on OD-corrected 16S rRNA data (**Figure 3C**) across timepoints T2-T8 revealed modest treatment-based separation (PERMANOVA $p = 0.001$, $R^2 = 0.48$). Variation along PCo1 reflected temporal progression, with T2 samples at one end (**SI Figure 2**), while PCo2 showed a weaker drug-associated signal, most notably for metronidazole. Superimposed on this temporal gradient, several treatments (e.g., erythromycin, clomiphene, sertraline, simvastatin, telmisartan) clustered in the upper-left region of PCo1, whereas DMSO control and DMSO-like profiles (omeprazole, lansoprazole, cilnidipine) were positioned toward the upper-right region of PCo1. Individual PCoA plots for each drug treatment and the DMSO control (**SI Figures 3-4**) further supported these trends: most treatments displayed clear temporal separation, with earlier timepoints (T2-T4) consistently diverging from later samples. In contrast, metabolomic PCoA trajectories (**Figure 3B**) showed more complex temporal patterns, while overall progression was evident, a few samples (typically at T0 or T4-T5) strongly influenced variation along PCo1.

Metronidazole, an antibiotic widely used against *Clostridioides difficile*, produced the most distinct taxonomic profile, consistent with its antimicrobial activity[260,289]. In Com20, *S. perfringens* is a dominant and stabilizing member; its reduction **(Figure 3E)** therefore led to pronounced community restructuring[290]. As *S. perfringens* declined, lower-abundance commensals (*S. parasanguinis*, *S. salivarius*) increased, aligning with reports of metronidazole tolerance or resistance[291–293]. The strong response observed here is thus expected when a keystone taxon is targeted. In addition, *E. coli* and *Streptococcus* overgrowth following metronidazole exposure has been reported previously[294,295], supporting the patterns observed in our system.

Similar effects were observed for Simvastatin, a statin, with increases in *T. ramosa* and *F. nucleatum* (**Figure 3E**). *T. ramosa* responded to several drugs but increased most consistently under simvastatin treatment, whereas *F. nucleatum* enrichment appeared unique to this condition (**SI Figure 5**). Previous work reported *E. lenta* and *B. thetaiotaomicron* as simvastatin-sensitive species showing growth inhibition and transcriptomic signatures of membrane remodeling or drug efflux[296]. Consistently, we observed a decrease in *B. thetaiotaomicron* abundance over time (**Figure 3E**).

To assess overall taxon involvement across different drug treatments, mean OD-corrected 16S rRNA abundances were visualized as heatmaps for each time point (**SI Figure 5**). Most taxa showed low but detectable abundances across treatments, underscoring the broad participation of the community in drug metabolism. This pattern is consistent with earlier reports that many drugs can be transformed by multiple taxa across phyla[48]. Together, these findings suggest that while metabolic shifts occur rapidly and are highly drug-specific, microbiome composition changes more slowly. Microbial perturbation was evident in both metabolomic and 16S datasets, but the stronger effect size in metabolomics (R^2 = 0.85 vs. 0.48) underscores its greater sensitivity in detecting early biochemical transformations. These results validate ChemProp2 as a framework for uncovering microbiome-mediated drug metabolism and highlight the importance of integrated omics approaches.

Across all 12 drugs, the summary metrics revealed substantial variation in how treatments affected metabolome and microbiome dynamics over time (**SI Figure 6**). Metabolome and microbiome trajectory lengths were positively associated (Spearman r = 0.73). Several drugs, including lansoprazole, montelukast, cilnidipine, omeprazole, and the DMSO control, showed relatively large temporal shifts in both metabolomic and microbiome PCo1 trajectories, whereas erythromycin and clomiphene exhibited minimal movement in either layer. Metronidazole displayed a distinct pattern: it induced noticeable microbiome restructuring but only minor metabolomic change. (**SI Fig. 6A**). No clear relationship was observed between the number of ChemProp2-predicted transformations and the magnitude of metabolome PCo1 change across treatments **(SI Fig. 6B)**. Biomass trends aligned with the observations in SI Fig. 6A, as treatments with larger OD increases generally also showed greater temporal change along PCo1 in both the metabolome (Spearman r = 0.74) and microbiome (Spearman r = 0.97) **(SI Fig. 6C-D)**. Shannon diversity remained stable for most treatments over the 8-hour incubation period (**SI Table 2**), which is expected given the short timeframe, initial dilution, and typical doubling times of gut bacteria. Importantly, PPIs often formed only small or fragmented FBMN clusters. This likely

reflects chemical behavior rather than biological absence of transformation, as PPIs such as omeprazole generate many structurally diverse, low-abundance degradation products with heterogeneous MS/MS fragmentation patterns[297,298]. Together, these results show that drugs can substantially reshape microbial metabolic activity without proportionally altering community structure.

3.3.4 Cascade Scoring Reveals Multi-Step Biotransformations

An inherent bottleneck of modified-cosine scoring for molecular networking is that it primarily links metabolites that differ by a single structural modification, as the algorithm compares MS/MS similarity while accounting for small mass shifts. Consequently, only one-step precursor-product relationships are typically connected. To move beyond these direct edges, we implemented a cascade scoring approach in ChemProp2 that traverses the molecular network to identify multi-step connections from each parent drug node (**Figure 4A-B**). This is particularly useful for multi-step biotransformations, in which the intermediates have short half-lifes and are only present in minor amounts. Starting from 115 first neighbor edges (D1), cascade expansion added 549 additional links to the 14 parent drug nodes, yielding a total of 664 drug-associated edges, a ~5.8-fold increase compared to D1 alone. Many distal features displayed anti-correlated intensity profiles with their parent drugs, suggestive of sequential metabolic turnover. **Figure 4C** summarizes the distribution of ChemProp2 scores by Δ*m/z*, highlighting recurring shifts corresponding to common modifications such as methylation (+14.02 Da) and oxidation (+16.00 Da). High-scoring edges with larger Δ*m/z* values often indicated potential conjugation or fragmentation events.

To benchmark the cascade scoring utility, we selected four representative drugs: Cilnidipine, Metronidazole, Omeprazole, and Simvastatin. Each showed strong ChemProp2 scores and treatment-specific features absent from abiotic controls, reinforcing their microbial origin.

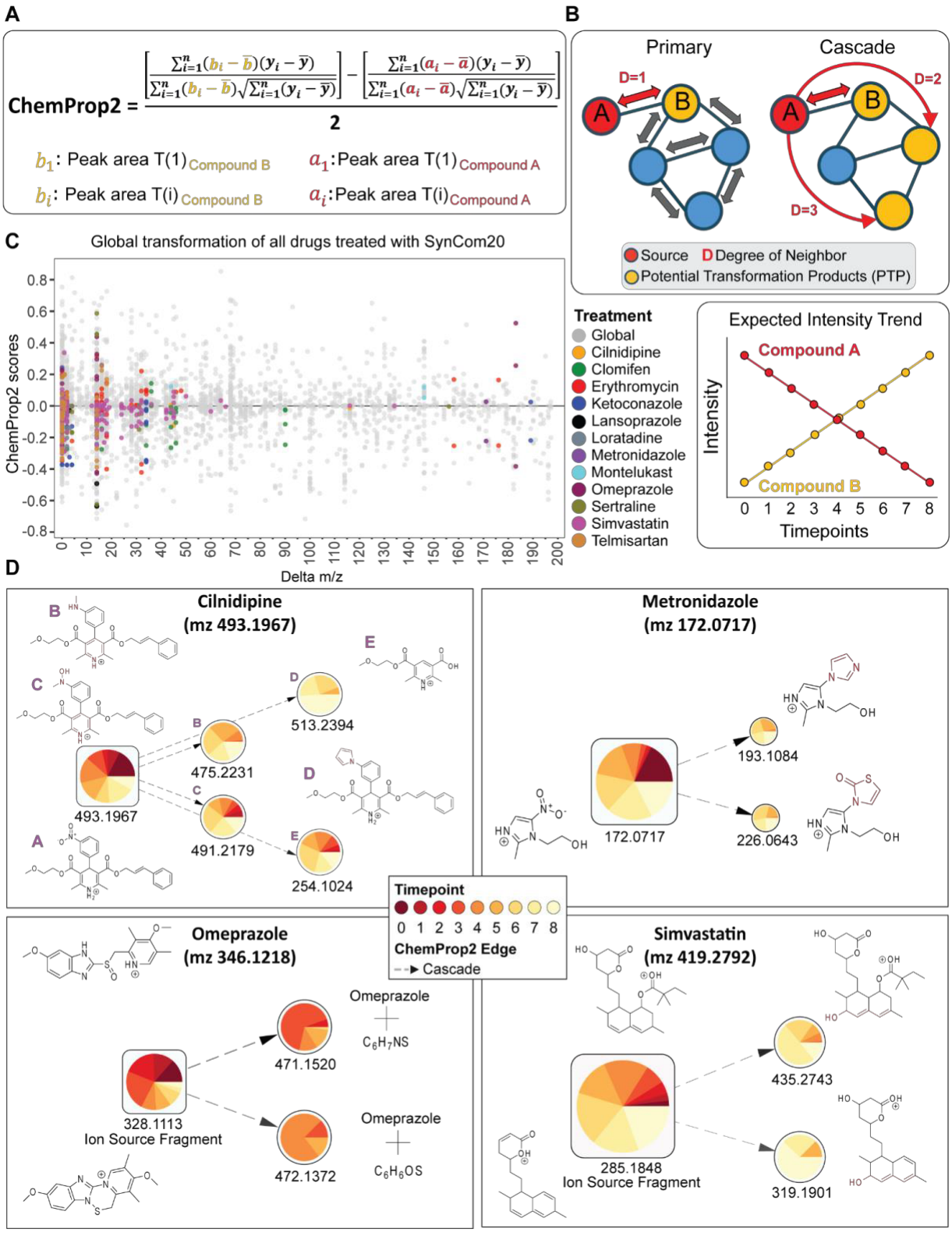


Figure 4: ChemProp2 Scoring Framework and Application to Drug Networks (A) ChemProp2 scoring formula based on Pearson correlation, the default metric used to quantify directional relationships between

feature pairs across multiple timepoints. (B) Conceptual illustration of primary edges (direct drug-feature connections) versus cascade edges (multi-step downstream connections) within a molecular network. Expected intensity trends associated with putative biotransformations are also shown. (C) Global ChemProp2 score distribution for all 12 drugs. The x-axis shows observed *m/z* differences between feature pairs, corresponding to potential chemical modifications (e.g., +14 Da for methylation), while the y-axis shows ChemProp2 correlation scores (-1 to +1). Each drug's primary-edge scores are color-coded, and all other pairwise scores in the network are shown in gray. (D) Molecular subnetworks for four representative drugs (Cilnidipine, Metronidazole, Omeprazole, and Simvastatin), with highlighted putative transformation products (PTPs) and proposed structures.

Cilnidipine (*m/z* 493.1967 $[M+H]^+$), a dihydropyridine calcium channel blocker[299,300], is metabolized hepatically via CYP3A to three major products: dehydrogenated (aromatized) form, a demethylated side-chain product, and a combined dehydrogenation/demethylation metabolite[301]. In our community experiment, 44 treatment-exclusive features were detected, of which a subset with high ChemProp2 scores and interpretable MS/MS spectra are highlighted in **Figure 4D**. MS/MS spectra of the highlighted features were manually interpreted using exact ion masses of fragments (**SI Figure 7**). In addition, pairwise MS/MS spectral comparisons between each highlighted feature and the corresponding parent drug, generated using the Spectrum Resolver in GNPS2[302], as well as predicted modification sites from ModiFinder[303], are provided in **SI Figure 8**. The feature at *m/z* 491.2179 was consistent with dehydrogenation of the dihydropyridine ring together with nitroreduction to a hydroxylamine and *N*-methylation, transformations commonly associated with microbial activity[304,305]; *m/z* 475.2231 likely represents an analogous metabolite with full reduction of the hydroxylamine. Additional features at *m/z* 254.1024 and 513.2394 may correspond to products of nitrobenzene cleavage and nitroreduction followed by addition of C_4H_4, respectively. These findings suggest that cilnidipine can undergo both hepatic-like dehydrogenation and additional microbial nitroreductive transformations consistent with previously described gut bacterial activities[250,306]. Clinical pharmacokinetic studies report cilnidipine peak plasma concentrations at ~2 h and elimination half-lives of 3-4 h[307]. In our SynCom experiment, however, the major cilnidipine-derived features appeared later (5-8 h). Several detected products such as ring dehydrogenation resembled known hepatic metabolites[301], whereas heavier products lacked hepatic analogs and likely represent microbial-specific transformations. Cascade scoring also revealed additional high-scoring mass differences consistent with demethylation, hydroxylation, or conjugation events[74,250,306], though some extreme shifts (e.g., -432 Da) remain difficult to interpret and may reflect adduct dynamics or in-source

fragments. Collectively, these findings suggest that microbial enzymes within Com20 can reproduce hepatic-like dehydrogenation while introducing distinct nitro-group modifications[250,306].

Metronidazole (*m/z* 172.0717 $[M+H]^+$) is a nitroimidazole antibiotic[308] metabolized hepatically via hydroxylation and conjugation, with 2-hydroxymetronidazole as the predominant product, whereas microbial metabolism is dominated by nitroreduction[309]. In our dataset, cascade scoring revealed two treatment-specific products at later timepoints (T4-T8): *m/z* 193.1084 (+21 Da), consistent with nitro reduction and possible formation of a nitrogen-containing imidazole derivative, and *m/z* 226.0643 (+54 Da), likely representing a conjugated product with a thiazolone-like moiety. Both metabolites showed strong residual precursor intensity and characteristic neutral loss of the ethoxy chain during fragmentation, indicating structural stability for the putative metabolites (**SI Figure 9**). Corresponding MS/MS spectral pair comparisons with the parent drug, along with predicted modification sites from ModiFinder, are provided in **SI Fig. 10A-B**. Their appearance coincided with enrichment of *Sarcina perfringens*, a known nitroreductase producer[290], underscoring their potential microbial origin. Although known hepatic metabolites such as N-(2-hydroxyethyl)-oxamic acid and acetamide[308,309] were absent, canonical nitroreduction products and several new metabolites emerged under microbial conditions at later timepoints (5-8 h). The metronidazole network also highlighted a limitation of relying solely on spectral connectivity, as several features were attributed to neighboring clusters. Among these disconnected nodes, several with larger mass shifts (+94 to +143 Da) exhibited fragmentation patterns consistent with partial ring cleavage and conjugation with amino acid- or peptide-like moieties[74,250,310]. Such transformations, though previously unreported for metronidazole, are chemically plausible given the widespread occurrence of bacterial nitroreductases and conjugating enzymes in gut microbes[306,311]. Together, these findings extend current models of metronidazole activation beyond simple nitroreduction and highlight potential downstream fates of reduced intermediates within microbial communities.

In our dataset, known metabolites such as 5-O-desmethylomeprazole and hydroxyomeprazole appeared as singletons or in small, disconnected clusters, revealing a limitation of ChemProp2 when relevant products are not networked to the parent node. Omeprazole-related features formed three distinct clusters: the protonated parent ion (*m/z* 346.1217), a sodium adduct (*m/z* 368.1039), and a dehydrated in-source fragment (*m/z* 328.1113). The latter structure likely arises from water loss involving the sulfoxide oxygen[312,313]. The parent and sodium-adduct clusters included features also present in abiotic controls, suggesting spontaneous or non-microbial processes. By contrast, the in-source-fragment cluster included treatment-enriched features (*m/z*

471.1520 and 472.1372) connected by strong ChemProp2 scores (**Figure 4D**). These exhibited mass shifts of +125.03 Da and +126.02 Da relative to the parent, corresponding to additions of C_6H_7NS and C_6H_6OS, respectively. Their MS/MS spectra yielded a dominant fragment at *m/z* 328.1113 but lacked additional diagnostic ions, preventing confident structural assignment (**SI Figure 11**). The corresponding MS/MS spectral comparisons and ModiFinder predictions are provided in **SI Figure 10C-D**. Known hepatic metabolites of omeprazole such as 5-O-desmethylomeprazole and hydroxyomeprazole were detected but appeared as singletons or small, disconnected clusters. ChemProp2 revealed treatment-enriched derivatives at *m/z* 471.1520 and *m/z* 472.1372 that likely represent previously uncharacterized conjugates. The corresponding mass shifts suggest addition of heteroatom-containing moieties, possibly through microbial conjugation reactions. Prominent neutral losses of the added atoms indicate that the mass shifts represent distinct substituents, possibly bound by a single bond to the parent structure. Although analogous additions have not been reported for omeprazole, sulfur- and nitrogen-based microbial conjugations are well documented[250,306]. These observations raise the possibility that anaerobic gut microbes can mediate conjugative transformations beyond those seen in hepatic metabolism. Moreover, as multiple microbial species are capable of transforming omeprazole[314–317], parallel cascades may occur simultaneously and complicate the reconstruction of directionality.

Simvastatin (*m/z* 419.2792, $[M+H]^+$), yielded low ChemProp1 scores, but ChemProp2 identified ten features unique to the microbial treatment. These included both singletons and connected subnetworks, with some features more abundant than the parent ion. Most appeared within the first 4 h of incubation. For example, the ion at *m/z* 435.2743, likely corresponds to the known metabolite 3´-hydroxysimvastatin[318] and was clustered with the major in-source fragment (*m/z* 285.1848). Another feature at *m/z* 319.1901 did not match reported derivative and likely represents an in-source fragment of *m/z* 435.2743 (**SI Figure 12**). The corresponding MS/MS spectral comparisons and ModiFinder predictions are provided in **SI Figure 10E-F.** Simvastatin illustrates how ChemProp2 captures microbial transformation products that were missed by the two-timepoint model. Most features in the simvastatin cluster appeared within the first four hours, outside the 0-2 h window used in ChemProp1, explaining its minimal initial scores. Cascade expansion further increased ChemProp2's sensitivity, revealing a range of simvastatin-related products, including the known metabolite 3'-hydroxysimvastatin (*m/z* 435.27) alongside in-source fragments, adducts, and statin analogs. Telmisartan similarly produced weak ChemProp1 scores but yielded detectable ChemProp2 features suggestive of methylation and charge-state shifts.

While several other drugs also showed cascade-level ChemProp2 scores, their associated features were often present in abiotic controls, suggesting non-biological or ambiguous origins. We therefore focused subsequent analyses on the above-mentioned four representative case studies, which best illustrate how ChemProp2 cascade scoring captures both known and candidate microbial-specific transformations in time-resolved metabolomics data. Compared to the earlier two-timepoint ChemProp1 model, ChemProp2 integrates temporal dynamics across multiple timepoints and applies FDR-based correction, resulting in fewer but more robust edges (**Table S3** and **Figure S13**).

3.3.5 Contextualization of Drug Metabolites against public metabolomics data

ChemProp2 suggested numerous putative drug metabolites across the SynCom time series. Once such candidate metabolites emerge from temporal analysis, a natural question arises: how broadly do these features appear beyond this experimental system, and which ones merit deeper biological investigation? Because direct experimental validation for every feature is not feasible, an intermediate step is needed to assess whether a metabolite is recurrent, condition-specific, or potentially an artifact.

To provide an additional contextual layer, we searched all prioritized MS/MS spectra using the FASST platform[281], which enables large-scale spectral similarity searches across public databases (more than 2 billion MS/MS spectra, November 2025) . Although public metadata are not uniformly curated, repository matches offer valuable orthogonal evidence, indicating whether a feature is observed across diverse biological or environmental studies, or instead appears to be unique to the microbiome-drug interaction examined here. In total, 13 databases were searched, including curated spectral libraries (e.g., GNPS Library, MassIVE-KB) and repository-scale datasets spanning MassIVE, Metabolomics Workbench, and Pan-Repository collections (**full list in SI section *FASST Inquiry***). Across 13 repositories, 1,063 of 1,202 queried features returned at least one match, spanning 1,670 unique MassIVE datasets (including the six datasets generated in this study, for more details: see **SI section *FASST Inquiry*; SI Table 4**). This broad recurrence suggests that several drug-derived features or structural analogs appear across independent human, animal, plant, or environmental studies.

Table 1: ChemProp2 Summaries.

The table provides the global subnetwork characteristics of the 12 drugs, including the number of cascade nodes, Δm/z distributions, and annotation coverage from GNPS and FASST. ChemProp2 refined the interpretation of these otherwise unannotated nodes; for example, Sertraline yielded 39 prioritized features consistently detected across multiple datasets, while many Cilnidipine features were unique to this study yet exhibited strong treatment-associated trends (Figure S14).

Drug Name	Nodes per drug	Nodes (Δ*m/z* > 0.5)	Library Matched	Unmatched Compounds	MassIVE Matches	ChemProp2 Hits (>0.1)	Annotated Hits (>0.1)	MassIVE Hits (>0.1)	Final Hits
Cilnidipine	98	98	0	98	20	76	0	14	11
Clomifen	42	36	1	37	34	9	0	6	9
Erythromycin	64	59	12	48	60	9	2	9	30
Ketoconazole	30	20	0	20	13	8	0	3	2
Lansoprazole	5	5	0	5	0	3	0	0	0
Loratadine	14	5	1	4	2	0	0	0	1
Metronidazole	18	6	0	6	3	3	0	2	2
Montelukast	70	64	0	64	27	10	0	5	10
Omeprazole	2	2	0	2	1	1	0	0	0
Sertraline	97	92	0	92	48	39	0	18	7
Simvastatin	55	53	18	38	38	4	2	7	5
Telmisartan	20	7	0	8	7	3	0	3	1
Total	**515**	**447**	**32**	**422**	**253**	**165**	**4**	**67**	**78**

To prioritize putative transformation products (PTPs), we retained cascade nodes that (i) differed from the parent $[M+H]^+$ ion by > 0.5 Da, (ii) were observed in external datasets, and (iii) had non-zero ChemProp2 treatment scores (**see SI section *Cross-Dataset Distribution of Cascade Nodes;* SI Table 5**). This filtering resulted in 78 PTPs, visualized in a heatmap (**Figure 5**), which shows their distribution across external dataset categories and contrasts ChemProp2 treatment versus control scores. Smaller Δ*m/z* shifts (e.g., −2.02, +14.02, +15.99 Da) corresponded to chemically intuitive modifications such as hydrogen loss, methylation, or oxidation, whereas larger Δ*m/z* values often lacked clear annotation but gained contextual support from repository matches. Representative examples illustrate how public data strengthen interpretation. A −17.97 Da transformation linked to Cilnidipine appeared in one plant-related and three unclassified repository datasets and showed a ChemProp2 treatment score of 0.31, consistent with a plausible biological transformation.

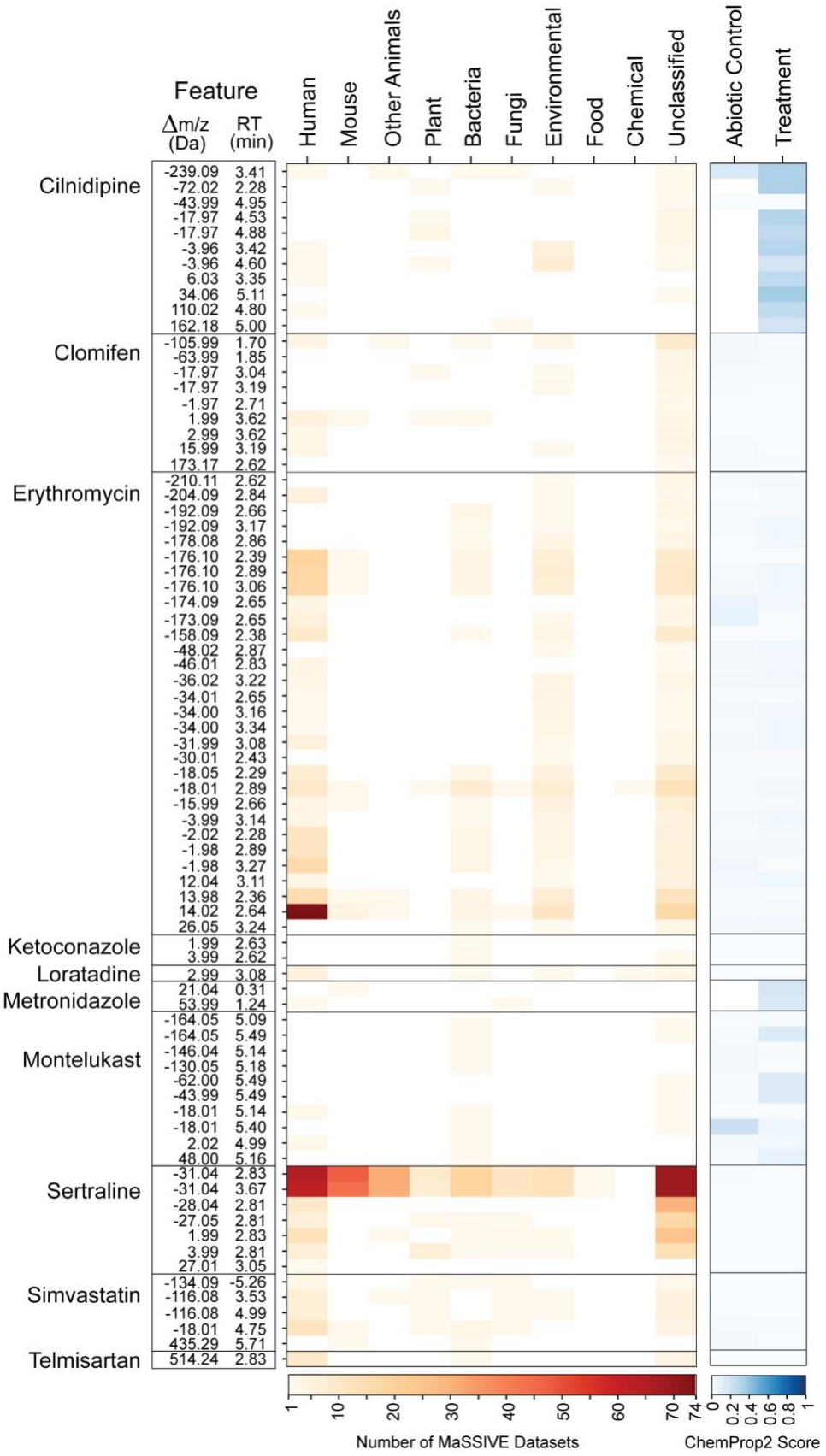

Feature
Δm/z (Da)
RT (min)
Human
Mouse
Other Animals
Plant
Bacteria
Fungi
Environmental
Food
Chemical
Unclassified
Abiotic Control
Treatment
Cilnidipine
-239.09 3.41
-72.02 2.28
-43.99 4.95
-17.97 4.53
-17.97 4.88
-3.96 3.42
-3.96 4.60
6.03 3.35
34.06 5.11
110.02 4.80
162.18 5.00
Clomifen
-105.99 1.70
-63.99 1.85
-17.97 3.04
-17.97 3.19
-1.97 2.71
1.99 3.62
2.99 3.62
15.99 3.19
173.17 2.62
Erythromycin
-210.11 2.62
-204.09 2.84
-192.09 2.66
-192.09 3.17
-178.08 2.86
-176.10 2.39
-176.10 2.89
-176.10 3.06
-174.09 2.65
-173.09 2.65
-158.09 2.38
-48.02 2.87
-46.01 2.83
-36.02 3.22
-34.01 2.65
-34.00 3.16
-34.00 3.34
-31.99 3.08
-30.01 2.43
-18.05 2.29
-18.01 2.89
-15.99 2.66
-3.99 3.14
-2.02 2.28
-1.98 2.89
-1.98 3.27
12.04 3.11
13.98 2.36
14.02 2.64
26.05 3.24
Ketoconazole
1.99 2.63
3.99 2.62
Loratadine
2.99 3.08
Metronidazole
21.04 0.31
53.99 1.24
Montelukast
-164.05 5.09
-164.05 5.49
-146.04 5.14
-130.05 5.18
-62.00 5.49
-43.99 5.49
-18.01 5.14
-18.01 5.40
2.02 4.99
48.00 5.16
Sertraline
-31.04 2.83
-31.04 3.67
-28.04 2.81
-27.05 2.81
1.99 2.83
3.99 2.81
27.01 3.05
Simvastatin
-134.09 -5.26
-116.08 3.53
-116.08 4.99
-18.01 4.75
435.29 5.71
Telmisartan
514.24 2.83
1
10
20
30
40
50
60
70
74
Number of MaSSIVE Datasets
0
0.2
0.4
0.6
0.8
1
ChemProp2 Score

Figure 5: Prioritizing ChemProp2 Features Using Public Dataset Context Heatmap showing the distribution of 78 ChemProp2-prioritized features (|ChemProp2 treatment score| > 0, |Δ*m/z*| > 0.5 Da, not unique to our dataset) across 10 high-level MassIVE dataset categories. Categories were derived from species-level metadata associated with matched GNPS entries. Corresponding ChemProp2 treatment scores for the same features, colored by score magnitude, illustrate directionality and potential transformation strength.

Conversely, a +14.02 Da shift from Erythromycin matched across 74 human datasets and was annotated in GNPS as clarithromycin, a broadly used clinical antibiotic. In our time series it showed a weak ChemProp2 score (0.02) and similar behavior in abiotic controls, suggesting a pre-existing analog rather than a microbially driven product.

Several strongly enriched features were not detected in any public repositories when queried (August 2025; **SI Figure 11**), suggesting that they may represent previously unreported metabolites unique to the biological conditions examined here. Together, this repository-level screening provides an important intermediate step between temporal inference and targeted experimental validation. By distinguishing recurrent metabolites from those unique to this system, public data help refine which candidate biotransformations warrant deeper mechanistic follow-up.

Contextualizing candidate biotransformations against public metabolomics repositories provides an important bridge between time-resolved inference and downstream biochemical validation. By examining whether features recur across independent datasets or remain exclusive to our system, we can prioritize metabolites that likely represent true microbiome-drug chemistry. For example, cilnidipine produced 98 transformation nodes; 11 were also found in external datasets and exceeded the ChemProp2 treatment threshold, while 65 were unique to our experiment (60 of these also exceeded the threshold). This absence of public matches for several strongly enriched metabolites likely reflects the unique chemical space captured in our microbiome-drug system, as well as the natural variability in what has been deposited by the community to date. Additionally, upon prioritization with ChemProp2, complementary computational tools, such as ModiFinder[303], can be used to help pinpoint the nature of the structure transformation, together providing a method combining MS/MS and time-series quantification. Combined with the appropriate computational tools to mine them, public repositories remain an invaluable resource, and as they continue to expand, some of these features may eventually find counterparts that help contextualize their biological origin.

3.4 Conclusion

This work demonstrates how *in vitro* experiments with gut synthetic communities in combination with non-targeted metabolomics and our ChemProp data analysis approach enables the systematic characterization of microbiome-mediated drug metabolism. Starting from 50 clinical drugs screened during endpoint experiments we selected 12 compounds belonging to different drug classes for a detailed timepoint analysis, spanning strong, moderate, and low-signal candidates. ChemProp uncovered diverse and multi-step transformation profiles, including for cilnidipine, metronidazole, omeprazole, and simvastatin, while cascade scoring expanded first-neighbor edges nearly six-fold, revealing distal products that tracked with microbial dynamics. Several metabolites appeared only at later timepoints and aligned with shifts in community composition; for example, late-stage metronidazole derivatives coincided with decrease of *Sarcina perfringens*, consistent with known nitroreductase activity. These results demonstrate that microbial drug metabolism is not only shaped by the intrinsic chemistry of each compound but also by the ecological dynamics and functional capabilities of the resident microbes.

To place these transformations in a broader biological landscape, repository-scale spectral searches provided an additional layer of context, distinguishing metabolites observed in diverse public datasets from those not yet represented in community submissions at the time of querying. This complementary view helps highlight candidate metabolites that may be driven by the specific conditions of synthetic community, while also situating others within widely occurring chemical space. Together, these insights show how combining synthetic communities with time-resolved non-targeted metabolomics can disentangle microbial contributions to xenobiotic metabolism and reveal a more diverse, dynamic chemical landscape than previously recognized.

3.5 Online Methods

3.5.1 Experimental designs and sample preparation

A. Endpoint screen

To investigate microbiome-mediated drug metabolism, 50 compounds were incubated with the Com20 synthetic gut community under anaerobic conditions in mGAM medium. Each drug was tested in triplicate at two timepoints (0 h and 2 h) alongside abiotic and vehicle controls. Incubations were performed in U-bottom 96-well plates (Thermo Fisher Scientific, cat. no.

Z168136) and extracted with ethyl acetate prior to LC-MS/MS analysis. The screen was conducted in three batches based on drug solubility (aqueous, DMSO, or mixed).

B. Time-series experiment

Following the initial two-timepoint screening, a longitudinal experiment was conducted to assess temporal dynamics of microbial drug biotransformations and benchmark the ChemProp2 framework. Twelve compounds were selected: nine (Cilnidipine, Clomifen, Erythromycin, Ketoconazole, Lansoprazole, Loratadine, Metronidazole, Omeprazole, and Sertraline) that exhibited at least one ChemProp1 transformation edge above the threshold (score ≥ 1), and three (Montelukast, Simvastatin, and Telmisartan) included to test ChemProp2 performance in low-signal conditions.

Incubations were sampled hourly from 0 h to 8 h, each timepoint corresponding to an individual 96-well plate containing three biological replicates and one technical replicate per biological replicate. From each well, 900 µL was reserved for 16S rRNA sequencing. All drugs, except Cilnidipine and Loratadine, were tested at 8 µM; these two were used at 4 µM due to solubility constraints. Sample preparation and incubation conditions followed those used in the endpoint screening.

3.5.2 Non-targeted Metabolomics using LC-MS/MS

Ethyl acetate extracts were dried under vacuum and resuspended in 50% methanol prior to LC-MS/MS analysis. Samples were measured on a Q Exactive HF Orbitrap mass spectrometer (Thermo Fisher Scientific) coupled to a Vanquish UHPLC system using a C18 column under a 7-minute reverse-phase gradient (0.1% formic acid in water/acetonitrile). Data were acquired in positive mode using data-dependent acquisition (DDA) with a resolution of 30,000 for MS^1 and 15,000 for MS^2. A pooled quality-control (QC) mix and an in-house six-compound QC mix were included to monitor retention time stability and batch effects. All chromatographic and MS parameters are provided in the Supplementary Methods.

3.5.3 Data processing and FBMN

Raw LC-MS/MS data were converted to the .mzML format using ProteoWizard's msconvert[108], retaining only MS/MS scans. Files were processed in MZmine v4.0.3[113] following a standardized batch workflow. Mass detection was performed separately for MS^1 and MS^2 scans using the Auto detector (noise level: 3×10^5 and 1×10^3, respectively). Chromatograms were built using the

ADAP module (minimum five scans, height ≥ 1×10^6, *m/z* tolerance = 0.002 Da or 10 ppm). Peaks were deconvoluted using the Minimum Search algorithm (duration 0.01-3 mins, minimum 5 data points, height ≥ 1×10^6). MS^2 spectra were linked to MS^1 features within 10 ppm precursor *m/z* tolerance and retention time (RT) filtering. Isotopes were grouped using the Isotope Grouper and annotated via the Isotope Finder. Alignment across samples used the Join Aligner (10 ppm, 0.15 min RT tolerance), retaining features present in ≥ 3 samples and containing ≥ 2 isotopic peaks. Gap filling employed the Multithread Peak Finder (5 ppm, 0.05 min RT tolerance), and redundant features were removed using the Duplicate Filter (5 ppm, 0.1 min RT tolerance). Final feature tables containing MS^2 scans (.mgf) and peak areas (.csv) were exported for GNPS(2) Feature-Based Molecular Networking (FBMN) and SIRIUS analysis.

FBMN was performed on GNPS(2) using default parameters unless stated otherwise. Precursor and fragment ion tolerances were both set to 0.01 Da; edges were retained for cosine > 0.7 with ≥ 6 matched peaks. Each node was connected to up to 10 most similar nodes ($\Delta m/z$ < 1999). Spectral library matching was performed against the GNPS library using identical thresholds, retaining only the top hit. A maximum component size of 100 was set. No intensity threshold or normalization was applied. The resulting edge table, together with the MZmine quantification table and sample metadata, was used as input for ChemProp analyses. GNPS job IDs, along with raw and processed data (.mzML), are listed in the Data Availability section.

3.5.4 Com20 microbiome incubations and 16S rRNA sequencing

The *Com20* synthetic gut microbial community as described in Griesshammer et al.[54], comprises 20 commensal bacterial species spanning six phyla and representing ~61% of the metabolic pathways found in a healthy human gut microbiome. All strains were originally obtained from DSMZ, BEI Resources, ATCC, or collaborating laboratories[54] and routinely cultivated in pre-reduced mGAM medium (HyServe GmbH & Co. KG, Germany) at 37 °C under anaerobic conditions (2% H_2, 12% CO_2, 86% N_2).

DNA from ChemProp2 time-series incubations (900 µL aliquots) was extracted using the DNeasy UltraClean 96 Microbial Kit (Qiagen). Amplicon sequencing of the 16S rRNA V4 region was performed on an Illumina MiSeq (2 × 250 bp) at the NGS Competence Center Tübingen following the protocol of Griesshammer et al[54]. Reads were processed with DADA2 (v1.21.0) and classified against a GTDB-based reference; Com20 members were further resolved to species level using

a custom database (≥ 98 % identity). Species-level abundances were normalized by optical density (OD_{600}) for downstream ordination (Bray-Curtis PCoA).

3.5.5 ChemProp2 analysis and cascade scoring (including FDR)

ChemProp2 scores were computed using Pearson correlations across timepoints for each connected feature pair within the FBMN network. A strong anti-correlation between two features was interpreted as a potential precursor-product relationship, where one compound decreased and the other increased over time. To assess scoring reliability, ChemProp2 implements a false discovery rate (FDR)-based approach adapted from proteomics[283,284] where peptide matches tested against real and decoy sequence databases are compared to estimate empirical thresholds. Feature tables are randomly shuffled to generate decoy datasets, which provide null distributions for comparison. Scores from original and decoy networks are then compared to estimate empirical FDR thresholds (e.g., 1%, 5%, 10%) for prioritizing high-confidence transformations. While this strategy improves interpretability, we note that shuffled datasets may still retain minor original patterns; further optimization of decoy generation will be addressed in future versions.

For cascade scoring, FBMN spectral library search returned 124 hits across the 12 analyzed drugs, including adducts and analogs. For downstream analysis, we focused on 14 features representing the parent $[M+H]^+$ ions (**SI Table 6**). Most drugs had a single $[M+H]^+$ feature, whereas Simvastatin (*m/z* 419.2791 and 419.2792; RT 4.75 and 5.01 min) and Telmisartan (*m/z* 515.2441 and 258.1256) contributed two each. The global FBMN with the 12 drugs contained 11,097 edges across 770 components, with 27 components including drug or analog nodes (2,155 edges total). Within this subset, 115 edges (D1) linked parent drug features to their immediate neighbors (**SI Table 3**). To capture downstream events, we expanded connections up to 10 cascade steps from each parent node. Cascade scoring added 3,637 edges, yielding 5,792 total scored edges (D1-D10). Of these, 664 edges were directly connected to 14 parent drug nodes, a ~5.8-fold increase relative to D1 alone. This cascade expansion highlighted distal features with drug-linked temporal trends, offering a broader view of multi-step microbial biotransformation pathways. ChemProp2 scores were also computed for drug adduct and analogs and their neighbors, but the analyses presented in the manuscript were restricted to parent drug features.

3.5.6 Selection of representative drugs for ChemProp2 analysis

From the initial drug screen, 50 compounds were selected for ChemProp1 analysis based on detectable precursor ions and network connectivity in FBMN. Each drug had a confidently observed $[M+H]^+$ feature with adjacent nodes suitable for scoring, except for acarbose, detected primarily as an $[M+NH4]^+$ adduct. ChemProp1 scores were calculated for every edge by comparing feature intensities between 0 h and 2 h, and edges with scores ≥ 1 (≥ two-fold change) were considered as candidate transformations.

Spectral library matches were found for 38 drugs (14 with class 1 annotations and 24 with class 3), providing additional confidence in compound identity. Parent ions were manually validated in the GNPS Dashboard by *m/z* and RT inspection. While some drugs displayed multiple network connections, the number of direct edges did not always correlate with transformation likelihood, highlighting the limitations of pairwise-only scoring. **Figure 2** summarizes these results, showing for each drug the number of first-degree connections and how many exceeded the ChemProp1 threshold. This initial screening established a baseline for assessing transformation potential and guided the selection of 12 representative drugs for subsequent time-resolved ChemProp2 analysis.

3.6 Data Sharing

Raw LC-MS/MS data (.raw and .mzML) for all ChemProp1 experiments (50-drug screen, two timepoints) are publicly available on MassIVE (MSV000096724, MSV000093571) and Zenodo (https://zenodo.org/records/10213654, https://zenodo.org/records/10210429). The ChemProp2 multi-timepoint dataset (12 selected drugs) is available separately on MassIVE (MSV000094899) and Zenodo (https://zenodo.org/records/15677238).

All molecular networking was conducted using the FBMN workflow on GNPS. Two GNPS1 jobs were used for the ChemProp1 screening experiment (https://gnps.ucsd.edu/ProteoSAFe/status.jsp?task=c561760343354873914a3f0bb4b03144, https://gnps.ucsd.edu/ProteoSAFe/status.jsp?task=421f28daa319440d900adfb2c9a56243). The longitudinal ChemProp2 dataset (12 drugs) was processed using the GNPS2 FBMN workflow (https://www.gnps2.org/status?task=1b5b94b4191d4223a5f57afb2aaaf0b0).

Use of Generative AI

No generative artificial intelligence (e.g. large language models) was used to generate original content of the manuscript. ChatGPT 5 (OpenAI) was used for proof reading and text editing of the manuscript. The authors take full responsibility for the content of the manuscript.

3.7 Author Contributions

AKPS and DP conceptualized the ChemProp software. AKPS, AG, LM, and DP conceptualized the SynCom experiments. AKPS, AW, and MW developed the ChemProp application. AKPS, AG, and PS performed microbial culturing and extractions. PS performed mass spectrometry experiments. AG and LM performed amplicon sequencing. AKPS, AG, JCK, LM and DP analysed data. AKPS and DP wrote the manuscript. All authors contributed to the writing and edited and approved the final manuscript.

Acknowledgement

This study was supported by the Deutsche Forschungsgemeinschaft (German Research Foundation, DFG) via the Cluster of Excellence EXC 2124: Controlling Microbes to Fight Infection (CMFI, project ID 390838134) to LM and DP. PS was supported by the European Union's Horizon Europe research and innovation programme through a Marie Skłodowska-Curie fellowship no. 101108450 MeStaLeM. We further acknowledge support by the National Institute of General Medical Sciences, GM160154 to DP, and the National Institute of Diabetes and Digestive and Kidney Diseases, 5U24DK133658-02 to MW. LM acknowledges funding from the DFG (Emmy Noether Programme MA 8164/1-1), the ERC (gutMAP 101076967). We thank the NGS Competence Center Tübingen (NCCT).

Chapter 4

CorrOmics: An Interactive Web Tool for Correlating Multi-Omics Data

Abzer K. Pakkir Shah[1,2,3], Karoline Steuer-Lodd[1,3], Mingxun Wang[4], Daniel Petras[1,2,3,#]

1. Interfaculty Institute for Microbiology and Infection Medicine Tübingen, University of Tübingen, Tübingen, Germany
2. Cluster of Excellence EXC 2124 Controlling Microbes to Fight Infections, University of Tübingen, Tübingen, Germany
3. Department of Biochemistry, University of California Riverside, Riverside, CA, USA
4. Department of Computer Science, University of California Riverside, Riverside, CA, USA

Correspondence should be addressed to Daniel Petras

Note: This manuscript is currently in draft form. The target journal is yet to be selected, and additional analyses will be incorporated. The work is intended to be submitted as a technical note in Analytical Chemistry.

4.1 Abstract

The integration of metabolomics and other omics data is essential for uncovering complex biological interactions, especially in microbial communities. We present CorrOmics, an open-source, user-friendly, web application that streamlines flexible correlation analysis between metabolomics and other omics layers such as microbiome (e.g., 16S rRNA), proteomics, or transcriptomics data. Developed in Python using Streamlit, CorrOmics allows users to interactively filter data, choose correlation metrics (Pearson, Spearman), apply FDR correction, and visualize significant associations. The tool supports hierarchical metadata selection and exports both CSV and GraphML formats for downstream network exploration in Cytoscape. A proof-of-concept case study demonstrates the utility of CorrOmics in analyzing metabolomic and microbial abundance data from a liquid-derived synthetic community (SynCom) with varying microbial compositions. By correlating features across different SynCom combinations, CorrOmics enables the identification of microbe-metabolite cooccurrences that are specific to microbial assemblages. CorrOmics is fully documented, and freely available for local installation and as a web application via the GNPS ecosystem (https://corromics.gnps2.org)

Keywords: Systems biology, Multi-omics, Omics, Correlation, Cooccurrence

4.2 Introduction

Rapid advances in bioanalytical high-throughput technologies, such as next-generation sequencing and mass spectrometry, have made it increasingly common to generate different omics datasets, capturing information across various biological layers like gene expression, protein abundance, and metabolic profiles[66,319]. Each layer offers a distinct perspective on cellular activity. To move beyond isolated insights and better capture the full complexity of living systems, it has become crucial to integrate these data types[64,319]. Such multi-omics approaches hold great promise for uncovering comprehensive mechanisms underlying diseases[320–322] and biological functions[323,324]. This interest in multi-omics integration has grown since the genomics boom of the early 2000s, evolving with technological advances that now allow holistic profiling across epigenomic, transcriptomic, proteomic, and metabolomic layers[322]. Yet, this integration is far from straightforward, as it is often hindered by the sheer scale of the data, variability across platforms, and the limited availability of biological replicates. Addressing these challenges calls for innovative computational strategies tailored to the intricacies of multi-omics analysis. As a result, developing robust methods for the integrative analysis of multi-omics datasets remains a central challenge in computational biology today[64]. Morabito et al.[325] provides a comprehensive review of current algorithms and tools for multi-omics data integration, highlighting commonly used methods tailored to specific integration strategies and study designs.

Several computational pipelines have been developed for multi-omics integration in the early 2000s, including tools like Integromics[326] and sMBPLS[327], with implementations primarily in R, C++, or MATLAB. While some tools are designed for specific combinations of omics data, others offer more general frameworks. Common analytical approaches include multi-weighted graphs, factor analysis, linear discriminant analysis, canonical correlation analysis, and partial least squares[64]. Although we do not delve into these methods in detail here, Bersanelli et al.[64] provides a helpful overview of integration strategies and tools developed up to that time, particularly those established in the early 2000s. However, many tools are no longer actively maintained, which remains a common challenge.

Multi-omics integration can be done in different ways, depending on the goal and complexity. Some methods study each omics type separately and combine the findings later, while others analyze them together to look for shared patterns. The process can also be guided by existing

biological knowledge (knowledge-driven) or by solely relying on the statistical patterns found in the data (data-driven)[325].

Our tool, CorrOmics, follows a data-driven approach to multi-omics integration. Broadly, the three commonly used strategies in multiomics integration under this data-driven approach include: (i) correlation-based methods, (ii) multivariate techniques such as PLS-DA and DIABLO, and (iii) machine learning or AI-driven approaches, such as consensus clustering[325]. Among these, correlation analysis remains one of the most fundamental and widely adopted techniques for exploring relationships between different omics layers, due to its simplicity, interpretability, and ability to reveal pairwise associations across complex datasets[328–331].

Common correlation approaches, such as Pearson and Spearman correlation, along with multivariate extensions like the RV coefficient, are widely used to identify global patterns across datasets[330,332]. More complex frameworks like WGCNA[333], xMWAS[334], and mixOmics[75] incorporate network- or model-based strategies, often relying on assumptions such as linearity or requiring dimensionality reduction through components like PLS. While powerful, these methods can be computationally intensive and less interpretable, as the resulting associations depend on latent variables, dimensionality reduction, or complex mathematical transformations that obscure direct feature-to-feature relationships.

CorrOmics adopts a straightforward, feature-level approach based on pairwise correlations between any two omics layers, such as metabolomics with microbiome, transcriptomics, or proteomics data. While it shares the linear or monotonic assumptions of Pearson and Spearman metrics, Corromics emphasizes transparency, statistical rigor, and ease of use, providing a fully interactive web interface with customizable filtering and support for flexible metadata structures. While tools such as mmvec[76] or mixOmics[75] excel in probabilistic modeling and multivariate integration, CorrOmics prioritizes accessibility, requiring no command-line knowledge, and provides a simple feature-correlation framework through an interactive web interface. An overview of the workflow is presented in **Figure 1**, illustrating data input, correlation computation, and visualization modules. It enables users to define thresholds, apply false discovery rate (FDR) correction, and interactively explore significant associations in formats suitable for exploratory analysis and publication-ready outputs. Notable features include hierarchical binning of features (e.g., taxonomic profiles) to a specified rank and FDR-based filtering of correlation scores for robust selection of significant relationships.

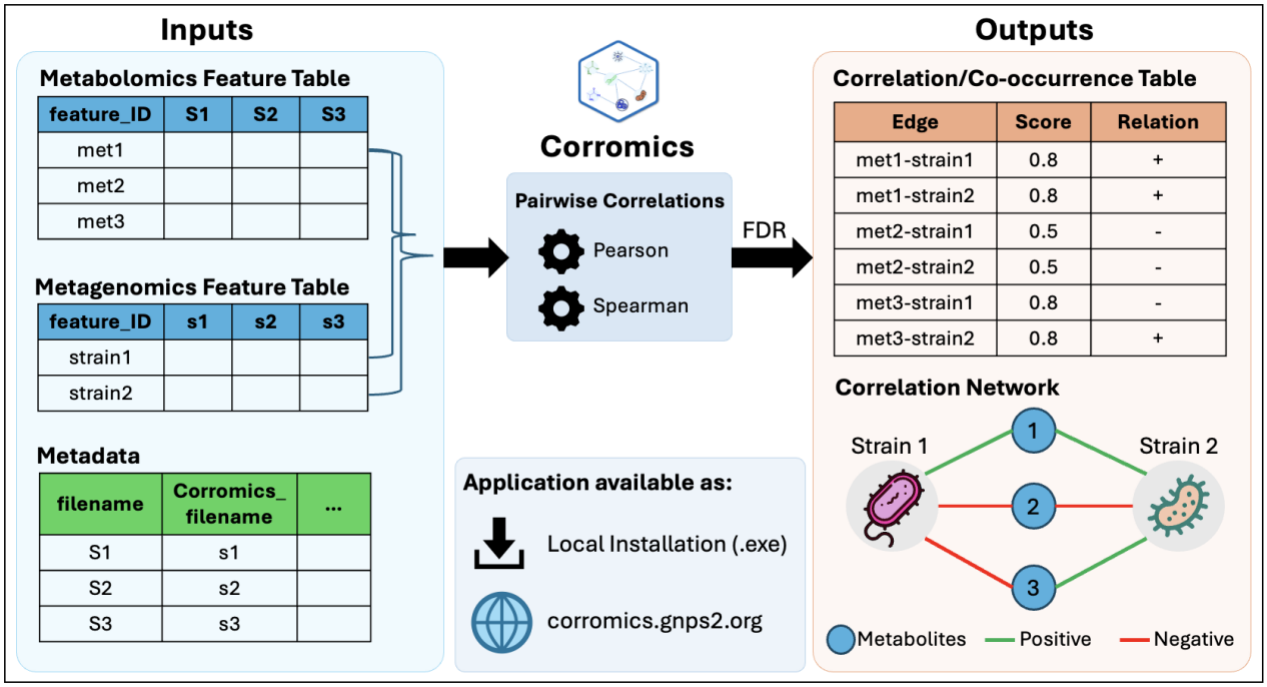


Figure 1: CorrOmics Overview / Architecture: The figure shows the schematic of input tables, the available correlation options, the resulting output formats and the app availability as local download and web application.

4.3 Implementation

CorrOmics was developed with accessibility, reproducibility, and scalability as guiding principles. The app is written entirely in Python and built with Streamlit[77], offering an interactive browser-based interface that requires no coding. It is designed for scientists with minimal computational experience while remaining robust for high-dimensional data analysis. The workflow is organized across multiple pages, with session-state management used to retain user selections and data transformations throughout the session.

All data processing is performed in-memory using pandas[335] and NumPy[336]. Correlations are computed using SciPy.stats[337], ensuring consistent statistical handling. Sample identifiers are automatically mapped between the different feature tables through the metadata columns 'filename' and 'ATTRIBUTE_Corromics_filename' **(Figure 2)**. For secondary omics datasets (e.g., proteomics, metagenomics, amplicon sequencing), users can optionally apply preprocessing such as feature abundance filtering, log transformation, and missing-value imputation prior to correlation analysis to standardize the data and reduce sparsity. In contrast, no preprocessing or transformation options are provided for the primary metabolomics dataset,

as it is assumed that users have already processed it using external tools such as FBMN-STATS[338], MZmine[113,339], or similar pipelines to generate a ready-to-use quantification table. Graph-based outputs are generated using NetworkX[340] in GraphML format. Cytoscape[341] compatible node and edge attributes are encoded directly during export, eliminating manual post-processing (**Figure 8** shows an example of the exported network).

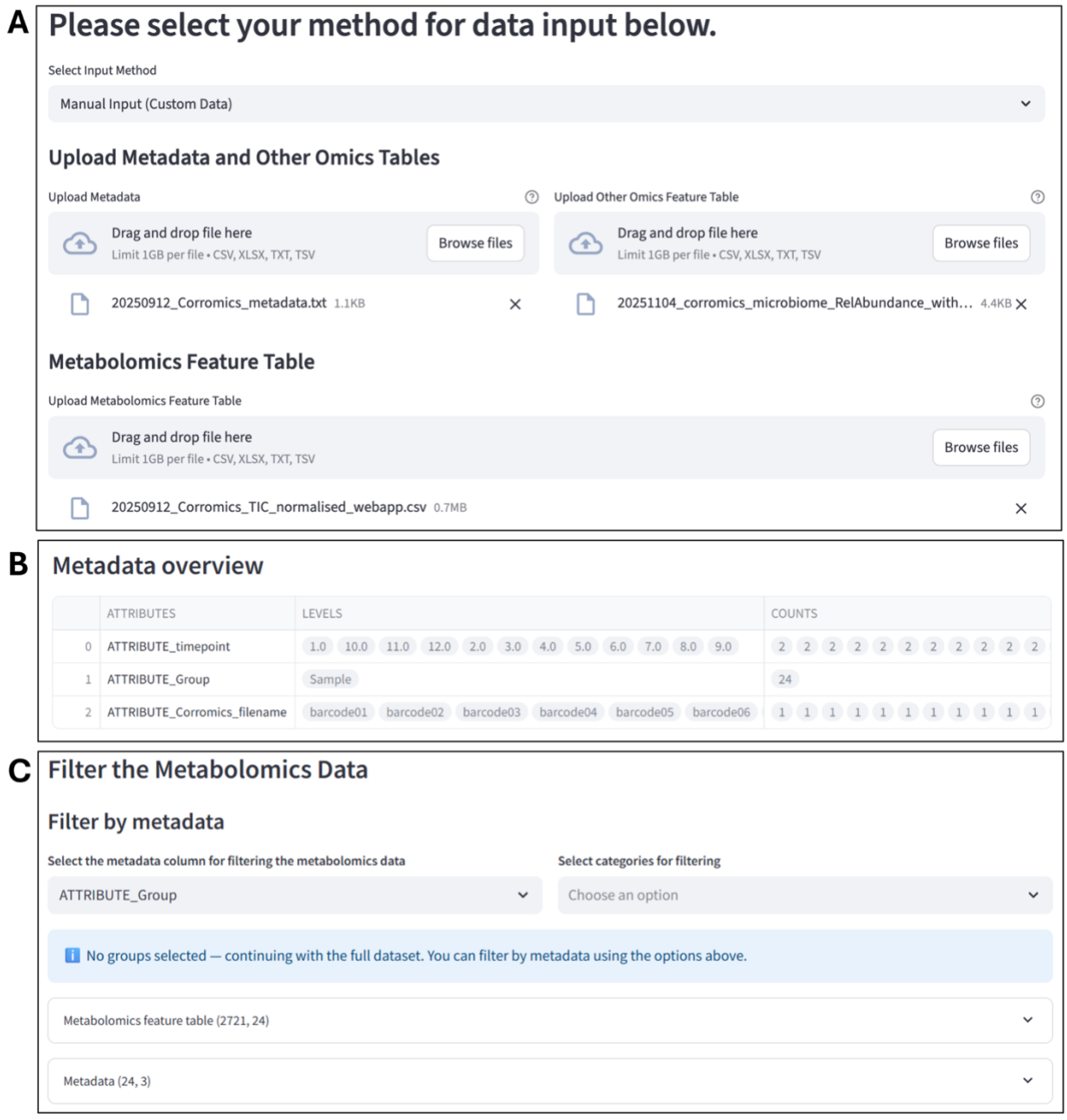


Figure 2. Data upload and filtering interface in Corromics (A) Manual upload of input files: a metadata file linking both omics datasets, a metabolomics feature table and a second omics table (e.g., proteomics or metagenomics). (B) Metadata overview summarizing columns, categories, and sample counts. The

'ATTRIBUTE_Corromics_filename' column links filenames between the two feature tables. (C) Filtering panel for metabolomics data, allowing users to include or exclude specific sample groups based on metadata categories.

CorrOmics can be accessed directly through the GNPS2[36,342] web platform or run locally from its open-source GitHub repository (https://github.com/Functional-Metabolomics-Lab/Corromics), making it fully operating-system independent. For local users, a precompiled Windows executable is also provided for convenience. On GNPS2, the app runs in a containerized environment to ensure stable deployment alongside other hosted tools. All computations are executed in-memory, with no intermediate files written to disk. No user data is stored on the GNPS2 server, and local executions process entirely within the user's own system memory, ensuring fast performance, data privacy, and reproducible analysis. To maintain performance across environments, CorrOmics dynamically adjusts the maximum number of allowable pairwise correlations (1 million on the GNPS2 server; up to 10 million locally).

4.4 Methods

4.4.1 Input Requirements

CorrOmics requires three input files in **.csv**, .**xlsx, txt,** or **.tsv** formats: a metadata table, a secondary omics feature table, and a metabolomics feature table. The workflow follows the same sequence as in the web interface.

(i) Metadata Table

The metadata table is essential for linking the metabolomics and secondary omics datasets (Table 1). It must contain two required columns:

- **filename** – sample identifiers corresponding to the metabolomics feature table (including file extensions such as '.mzML'). These must match the metabolomics sample names or their base names (e.g., *'sample_1', 'sample_1.mzML'*).
- **ATTRIBUTE_Corromics_filename** – sample identifiers corresponding to the secondary omics dataset (e.g., ASV, proteomics, transcriptomics).

Internally, CorrOmics uses the '**filename**' column to match samples to the metabolomics table and '**ATTRIBUTE_Corromics_filename**' to map them to the secondary omics table. Additional metadata columns (e.g., Treatment, Timepoint, Replicate, Batch) are optional but recommended, as they enable filtering or subsetting of samples during correlation analysis.

Table 1: Example Metadata Table

filename	ATTRIBUTE_Corromics_filename	Timepoint	Treatment	Replicate
Sample1.mzML	Sample1_proteins.tsv	T0	Drug_A	1
Sample2.mzML	Sample2_proteins.tsv	T1	Drug_A	1
Sample3.mzML		T0	Drug_A	2
Sample4.mzML	Sample3_proteins.tsv	T1	Control	1

(ii) Secondary Omics Table

The secondary omics table (e.g., ASV sequencing, proteomics, or transcriptomics) must contain features in rows and samples in columns. The first column is interpreted as the feature identifier and renamed internally to '**feature_ID'**. Non-sample columns such as taxonomy or annotation metadata are automatically detected and preserved for optional filtering and visualization.

(iii) Metabolomics Table

CorrOmics accepts three input types for the metabolomics feature table:

1. **Custom table:** Must contain features in rows and samples in columns, with a column named '**feature_ID'** as the unique identifier (See Table 2). Sample column names should match those in the metadata (including extensions such as '.mzML').

2. **FBMN-STATS output:** Directly upload the exported feature table (samples as rows, features as columns). CorrOmics automatically transposes the table, extracts feature columns (e.g., ID_mz_RT), and matches sample names with the metadata.

3. **MZmine export:** Upload the MZmine feature table containing 'row ID', 'row m/z', and 'row retention time' columns. These are automatically combined into a unique '**feature_ID'** (e.g., **45_233.12_5.02**). If column names include "Peak area," the app removes this suffix for proper metadata alignment.

Table 2: Example Feature Table (Option: Manual input - Custom Data)

feature_ID	Sample1.mzML	Sample2.mzML	Sample3.mzML
284_542.66_2.2	1000	850	720
512_631.55_3.1	234	310	198

4.4.2 Filtering and Preprocessing

By default, CorrOmics automatically filters the input data to retain only overlapping samples between the two feature tables. Additional sample-level filtering can be performed using metadata fields such as ‘treatment’, ‘timepoint’, or ‘replicate’ to create subsets for correlation analysis.

For the **secondary omics dataset** (e.g., ASV counts, OTU counts or proteomics), users can optionally perform hierarchical binning to reduce the total number of correlations. Abundance-based filtering can then be applied using the total-sum column automatically generated at the right end of the table. Thresholds on overall counts can be set to exclude features with very low or extremely high abundances.

No feature-level preprocessing is implemented for the primary metabolomics dataset (Omics 1). It is assumed that this dataset has already been processed using external tools such as FBMN-STATS or MZmine, where users can apply blank removal, imputation or total-ion-count (TIC) normalization, before importing into CorrOmics.

4.4.3 Correlation Analysis and FDR Strategy

Pairwise correlations are computed between all features in **Omics 1** (typically metabolomics) and **Omics 2** (e.g., microbiome, proteomics, or transcriptomics). Users can choose between Pearson correlation, suited for detecting linear relationships, and Spearman correlation, which captures monotonic but nonlinear trends and is more robust to non-normal distributions.

To control for false positives from multiple comparisons, CorrOmics applies a two-tiered FDR strategy. First, **Benjamini-Hochberg** adjusted p-values are calculated for each metabolite-other omics pair. Second, a **decoy-based correction** is implemented by randomly shuffling the sample labels of Omics 2 (e.g: microbial count) to generate a null distribution of “metabolite-shuffled microbe” correlations.

The resulting target and decoy score distributions are visualized as histograms, and an FDR curve is generated across correlation bins (-1 to +1) with suggested cutoffs of 1%, 5%, and 10%. Users can interactively adjust correlation thresholds (default $|r| \geq 0.5$) and examine the number of retained edges before and after FDR filtering. Filtered correlations can be exported as tables or GraphML networks compatible with Cytoscape for downstream exploration.

4.4.4 Case Study Dataset Design (Synthetic microbial community)

i) SynCom design and sample preparation

A defined synthetic leaf microbial community (SynCom) of 13 bacterial strains was assembled as a ground-truth model. The SynCom consisted of: *Aeromicrobium fastidiosum, Arthrobacter humicola, Bacillus altitudinis, Bacillus subtilis, Flavobacterium pectinovorum, Frigoribacterium faeni, Methylobacterium goesingense, Microbacterium proteolyticum, Nocardioides cavernae, Paenibacillus amylolyticus, Pseudomonas koreensis, Rhizobium skierniewicense,* and *Sphingomonas faeni.*

Each bacterial strain was cultivated individually in Nutrient Broth (NB) at 22 °C and 250 rpm overnight. Cells were washed three times in 10 mM $MgCl_2$ and adjusted to an optical density (OD_{600}) of 1.0. Twelve community combinations were created by adding different ratios of each strain while keeping the total cell count constant (1 mL per mixture). Each community was prepared in duplicate to generate two biological replicates. One mL of each mixture was used for DNA extraction and metabolite profiling.

ii) DNA Extraction and Nanopore Analysis

For the amplicon dataset, 1 mL of mixed culture was used for DNA extraction. Cells were lysed by bead-beating[363] with 0.1 mm zirconia beads in 500 µl lysis buffer (200 mM NaCl, 200 mM Tris Base, 20 mM EDTA) plus 200 µl SDS (20%), using a BeadBeater (BioSpec, Bartlesville, OK, USA) at 2,400 rpm for 5 min. The mixture was centrifuged at 6,200 rcf, 4°C for 3 min, and the aqueous layer was collected. DNA was purified twice using 500 µl Phenol:Chloroform:Isoamyl Alcohol (25:24:1), then precipitated with 600 µl ice-cold isopropanol and 60 µl of 3M Sodium acetate, incubated at -20°C for 1 hour. Following centrifugation and washing with 500 µl of 100% ethanol, the DNA was air-dried and resuspended in 30 µl of nuclease-free water.

DNA concentration was measured using NanoQuant (Tecan, Morrisville, NC, USA). Amplicon PCR was performed using 10 ng/µl DNA and 16S rRNA primer pair 341F/1378R[346,364], targeting the V3-V8 region[365] as described previously[366]. For library preparation, LongAmp Hot Start Taq DNA Polymerase (New England BioLabs, Ipswich, MA, USA) was used following the manufacturer's protocol. PCR products were purified using AMPure XP magnetic beads (Beckman Coulter, Ontario, Canada), and DNA concentration was measured using NanoQuant on a Tecan Infinite reader (Tecan U.S., Morrisville, NC, USA).

Further sequencing were performed using Oxford Nanopore Technologies with the Flongle Native Barcoding Kit 24 V14 (SQK-LSK110), following manufacturer instructions. Sequencing was run for 24 hours on a MinION device using R9.4.1 Flongle flow cells. Basecalling and barcode trimming were done a using MinKNOW software (v25.03.9) in super-accurate mode. Sub packages included MinKNOW core (v6.4.9), Dorado (v7.8.3) and Bream (v8.4.4) and ScriptConfiguration (v6.4.11). Downstream data processing was carried out with EmuWrapper[367], where the reads were filtered based on a QScore > 10 and a read length between 700 and 2000 bp.

iii) Metabolite Extraction and LC-MS/MS analysis

Cell pellets from the liquid cultures were extracted with ice-cold 80% methanol (10 mL per g dry weight), sonicated for 20 mins, concentrated, and re-dissolved to 2 mg/mL. Eight spike-in standards (aspartame, caffeine, carbamazepine, clarithromycin, coumarine 314, irgarol, N-acetylsulfamethoxazole, sulfamethoxazole) were added at 10 µg/mL to the 12 groups at varying ratios.

LC-MS/MS measurements were performed on a Thermo Orbitrap Exploris 480 coupled to a UHPLC C18 column (50 × 2.1 mm, 1.7 µm, 100 Å pore size, Phenomenex, Torrance, USA) by injecting 5 µl into a UHPLC System. The mobile phase consisted of Solvent A (water + 0.1% formic acid) and Solvent B (acetonitrile + 0.1% formic acid). A linear gradient was applied as follows: 5% B from 0-8 min, 40-99% B from 8-10 min, a washout at 99% B from 10-13 min, followed by re-equilibration at 5% B for 3 min. The detected compounds were ionized by electrospray ionization (ESI) with the following settings: sheath gas 45 L/min, auxiliary gas 5 L/min, sweep gas 1 L/min, spray voltage 3.0 kV, RF lens 75 V, capillary temperature 325 °C, and auxiliary gas temperature 300 °C. MS acquisition was performed in positive ion mode, scanning m/z 150-1500 at 120,000 resolution with one microscan, a 100 ms maximum injection time, and Automatic Gain Control (AGC) target of 100%.

MS/MS scans were acquired in data-dependent acquisition (DDA) mode. In each duty cycle, the top 5 most intense ions were selected for fragmentation at 15,000 resolution with one microscan. The maximum injection time was 50 ms, using the same AGC target as MS1. Fragmentation was performed with a 1 m/z isolation window (no offset). Dynamic exclusion was set to 10 s, ions with unassigned charge states were excluded with a mass tolerance of 10 ppm. Isotope peaks were removed using a 3 m/z exclusion window. Raw files were converted to .mzML (MSConvert[110]) and processed through GNPS2 Feature-Based Molecular Networking[42] for spectral annotation.

MZmine 4.0.3 was used to get the quant table[113]. Statistical filtering (blank removal, imputation and TIC normalization) was carried out on metabolomics data using FBMN-STATS[338].

iv) Data processing for CorrOmics analysis

Amplicon-derived ASV abundance tables and LC-MS/MS feature tables (blank-removed, imputed, TIC normalized table from FBMN-STATS app) were imported into **CorrOmics** for correlation analysis. Pairwise Pearson and Spearman correlations were computed, FDR correction applied, and significant metabolite-species associations visualized as networks in **Cytoscape**. The metabolomics dataset used in this study can be accessed from the FBMN link (https://www.gnps2.org/status?task=68ff815c6ad448d4b3bfc88db5236da8). All associated analysis files for repeating the CorrOmics analysis presented here are provided in the GitHub repository https://github.com/abzer005/Corromics_test_case.

4.5 Proof Of Concept

To demonstrate the workflow, CorrOmics was applied to an established leaf synthetic community (SynCom) derived from *Arabidopsis thaliana* composed of 13 bacterial isolates[343]. The dataset integrates microbial abundance data such as amplicon sequencing variants (ASV-based 16S rRNA counts) and metabolomics profiles (such as liquid chromatography-tandem mass spectrometry or LC-MS/MS intensities), providing a defined ground-truth system for benchmarking correlation accuracy. Controlled variation was introduced by constructing 12 community combinations in which strain abundances either increased, decreased, or remained constant, each prepared in duplicate. Eight metabolites of known identity were spiked into these samples following the same directional trends, mimicking a microbe-metabolite system that changes over time. This design allows systematic evaluation of metabolite-microbe correlations under realistic yet controlled conditions.

CorrOmics accepts data in several formats: metabolomics feature tables may be imported from MZmine**,** FBMN-STATS**,** or custom matrices, while metadata and secondary omics layers must follow a defined structure (**detailed in Methods**). Users provide three core files, a metabolomics table, a second omics table (e.g., ASV counts, proteomics), and a metadata file linking samples (**Figure 2A-B**). In the SynCom use case, metabolomics data were preprocessed externally (blank removal, imputation, total-ion-count or TIC normalization using FBMN-STATS), whereas the ASV table was used without hierarchical binning since species-level identities were known (in contrast to datasets where grouping by taxonomic rank is beneficial; **Figure 3**).

Within the interface, CorrOmics supports optional preprocessing for both metabolomics (**Figure 2C**) and secondary omics tables, including abundance-based filtering, log transformation, imputation, and TIC or centered log-ratio (CLR) normalization[344,345]. Zero-count features are excluded by default, and users may define upper or lower abundance thresholds to remove rare or dominant taxa. Pairwise metabolite-microbe correlations are then computed using Pearson or Spearman with FDR correction (**Figure 4**). For the SynCom dataset, Pearson was preferred due to the linear structure of the experimental design.

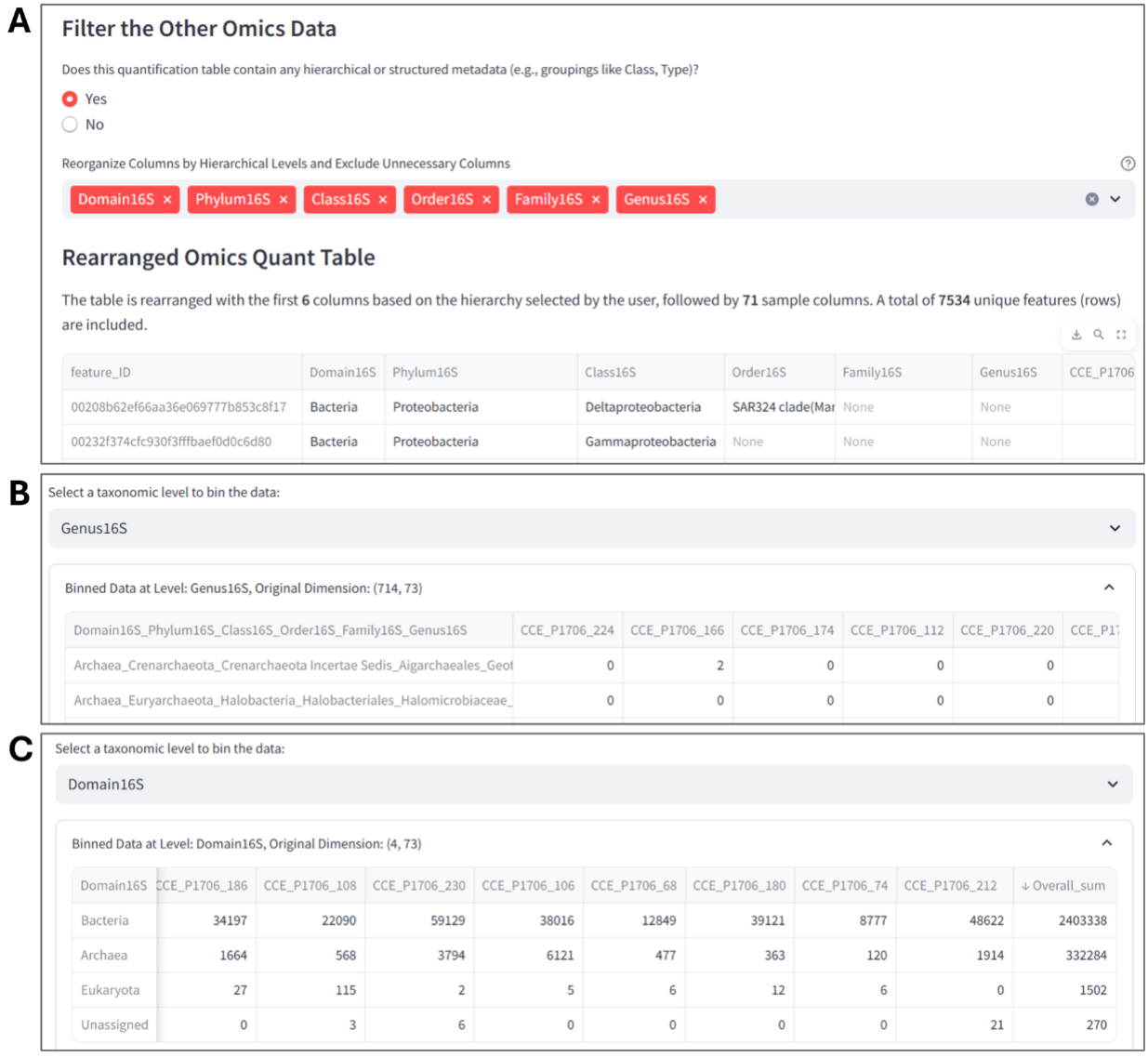
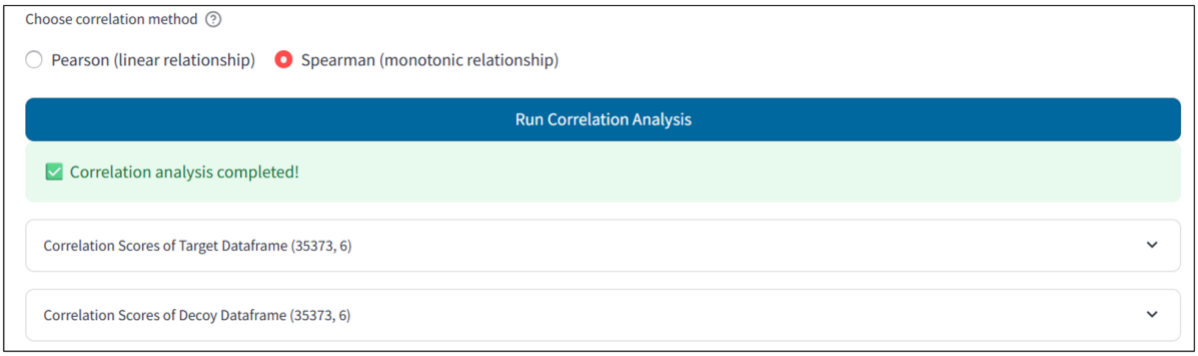


Figure 3. Hierarchical organization and binning of other omics data. (A) Interface for hierarchical data (e.g., taxonomy). Users can select columns from a dropdown, arrange them from higher to lower levels, and bin data at the desired level to simplify correlations. (B) Example dataset binned to the 'Genus level'

using ASV 16S rRNA counts (714 unique genera). (C) The same dataset is binned to the 'Domain level', showing four domains. The last column indicates total counts at each level.

A



B

Correlation Scores of Target Dataframe (35373, 6)

	Feature	Variable	Estimate	P-value	BH-Corrected P-Value	R2
0	11_150.9509_0.55	Aeromicrobium_fastidiosum	-0.7163	8.27e-05	2.76e-04	0.513
1	77_153.9976_0.56	Aeromicrobium_fastidiosum	-0.5155	9.94e-03	2.26e-02	0.2657
2	339_156.0115_3.03	Aeromicrobium_fastidiosum	-0.8594	7.53e-08	1.35e-06	0.7385
3	697_156.0587_5.43	Aeromicrobium_fastidiosum	-0.8359	3.66e-07	2.76e-06	0.6988
4	756_158.1173_5.7	Aeromicrobium_fastidiosum	-0.8504	1.43e-07	1.66e-06	0.7232
5	619_158.5413_5.3	Aeromicrobium_fastidiosum	-0.0442	8.37e-01	8.75e-01	0.002

Figure 4. Correlation metrics and output visualization. (A) Supported correlation methods (Pearson and Spearman). After computation, correlation scores are generated for both target and decoy datasets, linking each metabolite feature with each species. (B) Example output table showing correlation results for the target dataset, including the correlation coefficient (Estimate), p-value, Benjamini-Hochberg-corrected p-value, and R^2 values.

To assess robustness, CorrOmics applies a target-decoy validation strategy, in which a decoy dataset is generated by randomizing the second omics table while keeping the metabolite table fixed (**Figure S1** in Supplementary Information (SI)). Comparison of target and decoy score distributions identifies low-confidence regions (typically $-0.5 < r < +0.5$), and user-defined FDR thresholds (1%, 5%, 10%) are applied to retain high-confidence correlations (**Figure 5**).

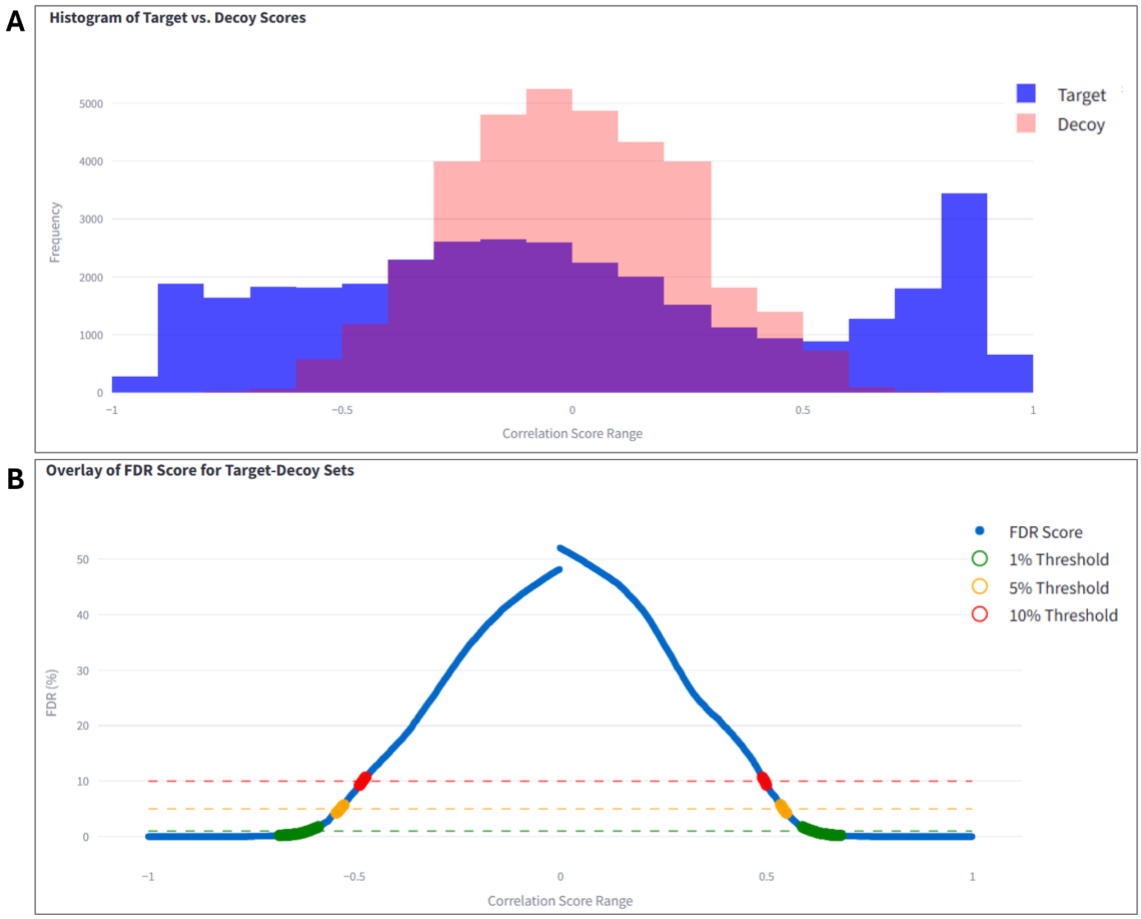


Figure 5. Evaluation of correlation scores and false-discovery-rate (FDR) thresholds. (A) Histogram comparing correlation scores from target and decoy datasets. Most overlapping scores fall between -0.5 and +0.5, while stronger correlations (|r| > 0.8) appear mainly in the target set. (B) FDR thresholds (1%, 5%, 10%) derived from target-decoy score distributions. For instance, an FDR of 1% retains correlations above ±0.6, whereas an FDR of 10% includes correlations above ±0.5.

The final output includes a correlation results table and a Cytoscape-compatible network file, where nodes represent metabolites or microbial features and edges represent correlation strength and whether the association is positive or negative. These outputs enable users to visualize multi-omics associations and prioritize biologically meaningful relationships for further analysis.

4.6 Results & Discussion

4.6.1 Validation of Synthetic Dataset Design

The designed (**SI Figure S2**) and measured (**SI Figure S3**) compositions of both datasets are provided in the Supplementary Information. Most intended abundance patterns were preserved, although certain strains exhibited lower counts, likely due to primer-binding or amplification biases in 16S amplicon sequencing, including nanopore-based methods[346–349]. In addition, sequencing and base-calling errors inherent to long-read nanopore data can contribute to compositional inaccuracy, particularly for closely related taxa[350]. Comparative studies have also reported platform-specific biases, with nanopore sequencing offering higher taxonomic resolution but reduced quantitative accuracy relative to short-read Illumina data[351].

In the metabolomics data, spiked-in compounds followed the expected trends, though some variation in signal intensity was observed, likely reflecting differences in ionization efficiency or matrix effects rather than true concentration changes[352,353]. Principal Coordinate Analysis (PCoA) of the measured microbiome and metabolome datasets (**SI Figure S4**) revealed the expected progression along the designed gradient, confirming that the synthetic community structure was largely maintained. Minor deviations in specific strains or compounds are discussed in Supplementary Note and visualized in line plots (**SI Figure S5**).

4.6.2 Evaluation of Normalization and Trend Preservation

To assess how preprocessing strategies affect trend preservation, several normalization methods were tested for both metabolomics and microbiome datasets, including raw values, TIC normalization[179], CLR transformation[354], and feature-wise mean normalization[355]. For each approach, the abundances of individual species (microbiome) and the intensities of the eight spiked-in metabolites (metabolomics) were plotted across the 12 SynCom combinations. This comparison aimed to identify methods that retained the expected increase-decrease trends while placing both datasets on comparable intensity scales, since raw metabolite intensities (10^6-10^8 range) are not directly comparable to microbial operational taxonomic unit (OUT) counts for correlation analysis.

TIC normalization preserved the relative trend shapes observed in the raw metabolomics data while compressing overall intensity values. For the ASV table, TIC normalization (i.e., per-sample total-sum normalization or relative abundance) retained the intended gradient, whereas CLR (with

zero replacement) distorted trajectories in this sparse SynCom. Accordingly, TIC-normalized metabolomics and microbiome data were used for all downstream correlation analyses (Pearson correlation, consistent with the linear study design). Comparative panels illustrating these effects are provided in SI **Figures S6-S7.**

Preprocessing and data scaling can influence the magnitude of correlations, particularly for parametric measures such as Pearson correlation, which are sensitive to compositional effects[356,357]. In contrast, non-parametric rank-based methods such as Spearman are generally less sensitive to scaling effects[358]. However, Spearman's correlation should not be overinterpreted as a direct measure of association strength, since it reflects monotonic rather than strictly linear relationships. A systematic evaluation of normalization effects across omics types is beyond the scope of this technical note. Future versions of CorrOmics may include normalization modules suited to other omics datasets, such as quantile or variance-stabilizing transformations for transcriptomics[359,360] and median-ratio or intensity-based normalization for proteomics[361,362], enabling systematic multi-omics benchmarking within the same framework.

4.6.3 Correlation Analysis and Benchmarking

The metabolomics dataset used for correlation analysis is publicly accessible on GNPS2 (https://www.gnps2.org/status?task=68ff815c6ad448d4b3bfc88db5236da8). It contains 3,371 features across 24 samples (12 experimental groups in duplicate), including eight spiked-in metabolites. Following blank filtering (30% threshold), 2,721 features remained, after which missing values were imputed and intensities were TIC-normalized using the FBMN-STATS workflow (https://fbmn-statsguide.gnps2.org/).

Correlation analysis between metabolite intensities and microbial relative abundances generated 38,094 feature-taxon pairs, of which ~11,000 met the significance criteria of $|r| \geq 0.6$ at 1% FDR (Pearson: 11,258; Spearman: 11,941 correlations). Of these associations, approximately 60% were positive and 40% negative, capturing both co-occurrence and inverse abundance patterns across taxa.

To highlight the strongest associations, the top 60 Pearson correlations (30 positive and 30 negative) were visualized, representing 59 unique metabolite features mainly linked to *Pseudomonas koreensis*, *Bacillus altitudinis*, and *Rhizobium skierniewicense* (**Figure 6A**). These included four of the eight spiked-in metabolites and other annotated compounds such as surfactins, erythromycin, terbutryn, prometryne, and irgarol. In contrast, the top 60 Spearman

correlations (46 features across four species, including *Flavobacterium pectinovorum*) contained no spike-ins, even when extended to the top 80 (63 features) (**Figure 6B**). Overall, 56 of the 59 Pearson correlations involved *P. koreensis*, whereas the top Spearman set was dominated by *P. koreensis* and *R. skierniewicense*. Together, these results suggest that Pearson better captured the designed abundance trends, whereas Spearman emphasized alternative monotonic relationships.

The overall correlation heatmap (**SI Figure S8**) extends this comparison to the non-spiked-in features (1% FDR-adjusted scores), providing a broader view of species-specific association patterns. These features were included to explore background correlations beyond the experimental spike-ins and to identify clusters of positive and negative associations across species. Spearman produced a greater number of significant correlations, primarily involving the *unassigned*, *P. amylolyticus*, and *M. goesingense* groups, although the latter two displayed very few correlations overall that passed FDR correction. While both Pearson and Spearman heatmaps appear similar, the Spearman version shows a denser distribution of significant associations (condensed for clarity). Species are represented as columns and features as rows, with hierarchical clustering applied to highlight shared correlation profiles.

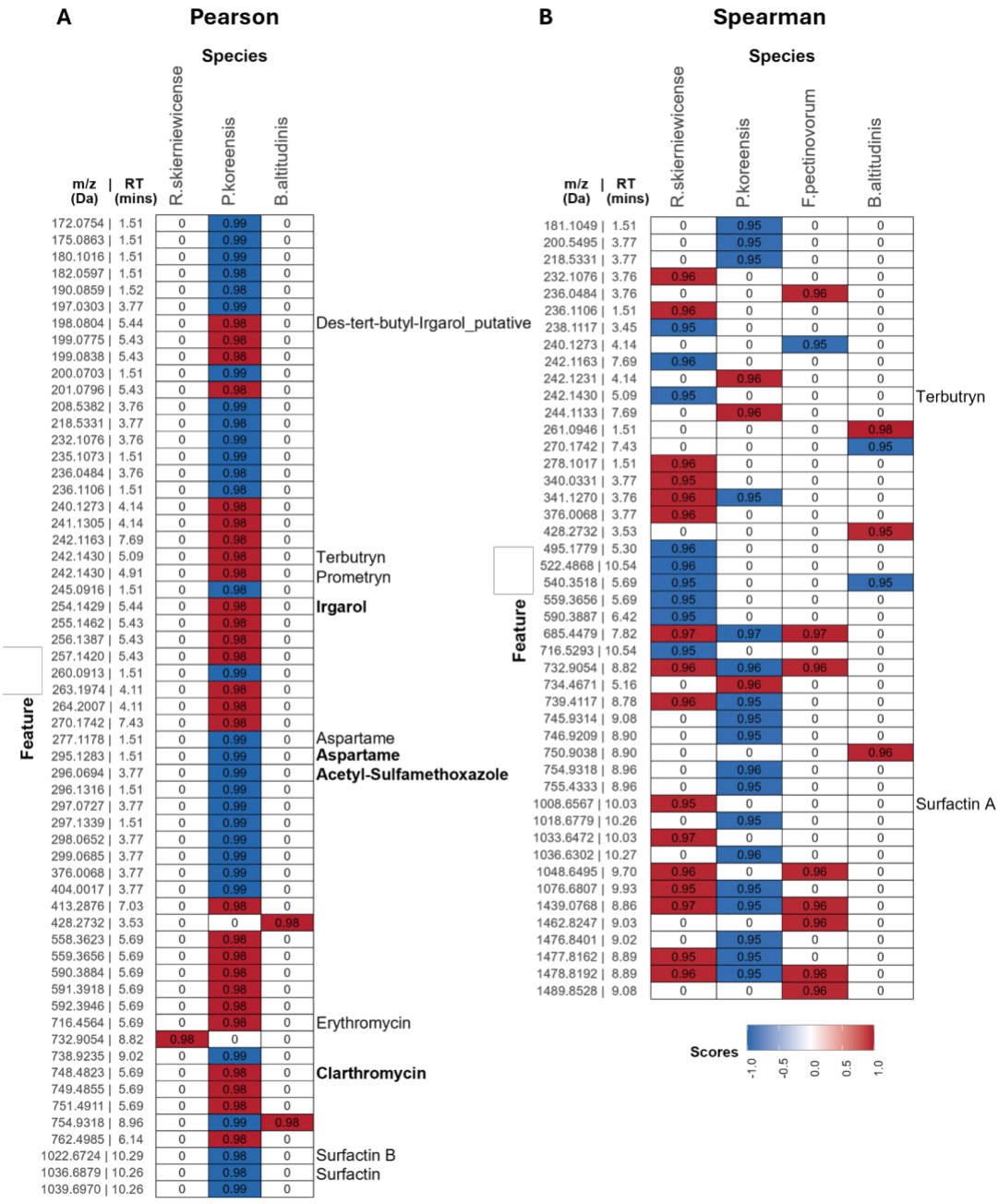


Figure 6. Heatmaps of the top correlation scores (top 60 = 30 positive + 30 negative). **A)** Pearson correlations highlight 59 unique metabolite features, predominantly associated with three species, and include 4 spiked-in compounds. **B)** Spearman correlations capture 46 features across four species but do not include any of the spike-ins. The color scale represents correlation strength from -1 (blue) to 0 (white) to +1 (red). Zero values represent correlations that did not pass the 1% FDR-adjusted significance threshold ($|r| < 0.6$), rather than true zero correlations.

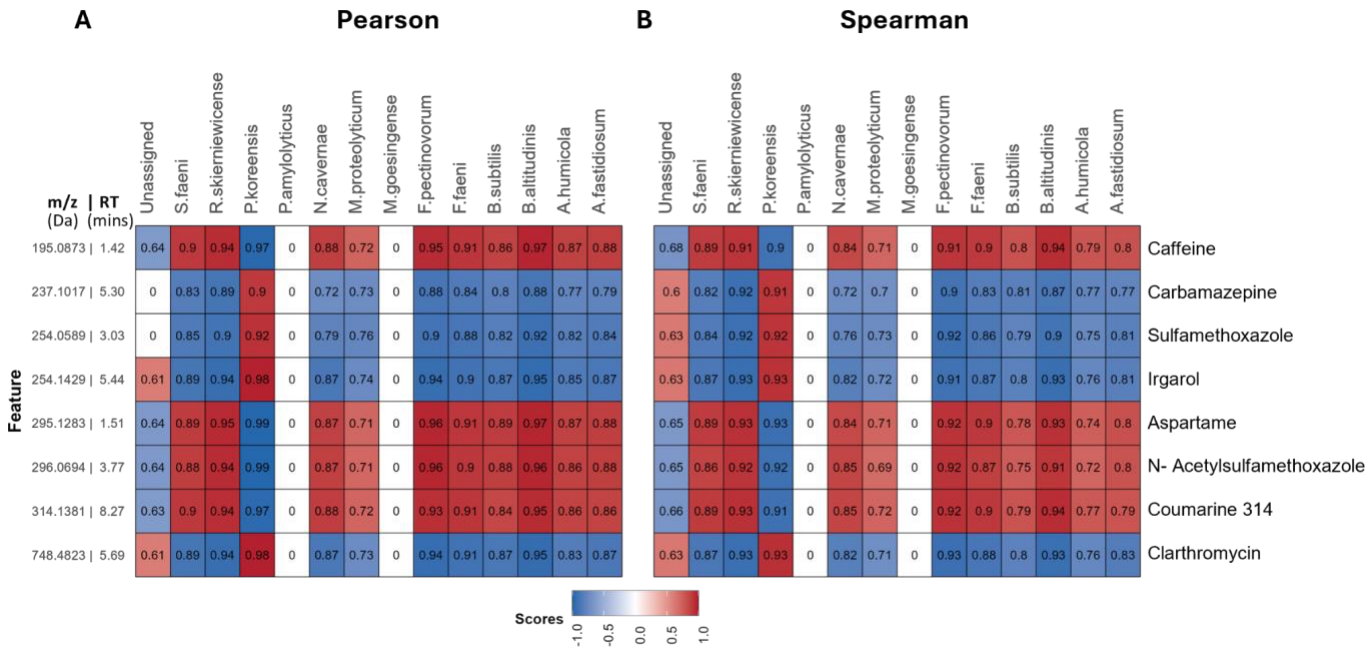


Figure 7. Comparison of (A) Pearson and (B) Spearman correlation scores for the eight spiked-in metabolites across all species. Both methods show highly similar correlation patterns. Positive spike-ins (e.g., aspartame, caffeine, coumarin 314, and acetylsulfamethoxazole) exhibited strong correlations with increasing species (shown in red), whereas negative spike-ins showed the opposite trend. Zero values indicate correlations that did not pass the 1% FDR-adjusted significance threshold ($|r| < 0.6$) rather than true zero correlations.

Figure 7 focuses specifically on the eight spiked-in metabolites, comparing their correlation signatures across species between the two methods. The overall profiles were consistent across both approaches, except for the *unassigned* group. In the experimental design, *P. koreensis* and *P. amylolyticus* were programmed to decrease, whereas two *Bacillus* species were designed to increase, with the remaining members serving as approximately constant controls. Slight deviations from these intended trends likely reflect natural variability in community growth. Neither *P. amylolyticus* nor *M. goesingense* exhibited detectable correlations with the spiked-in metabolites under the FDR-adjusted $|r| \geq 0.6$ threshold. In the full, unfiltered correlation matrices (**SI Figure S9**), weaker associations are visible, approximately $r \approx 0.6$ for *M. goesingense* and $r \approx 0.2$ for *P. amylolyticus*.

For the decreasing-trend spike-ins (carbamazepine, sulfamethoxazole, clarithromycin, and irgarol), negative correlations were observed only for *P. koreensis* and the *unassigned* group, consistent with the imposed downward abundance trend. The opposite pattern was observed for the increasing-trend spike-ins, which correlated positively with the designed *Bacillus* species.

Collectively, these results demonstrate that Pearson correlations effectively captured the intended co-variation and directionality of metabolite-species relationships, whereas Spearman correlations reflected comparable monotonic trends but with slightly attenuated magnitudes.

4.6.4 Network Representation of Multi-Omics Associations

To visualize the overall structure of species-metabolite associations, the FDR-adjusted correlation table was exported as a '.graphml' network and visualized in Cytoscape **(Figure 8)**. The Pearson-based network appeared as a dense, hairball-like structure, with a distinct peripheral cluster representing *Paenibacillus* and four associated metabolites. Edge colors denote correlation direction, red for positive and blue for negative associations. Positive values indicate co-occurrence (both increasing or decreasing), while negative values reflect opposite trends.

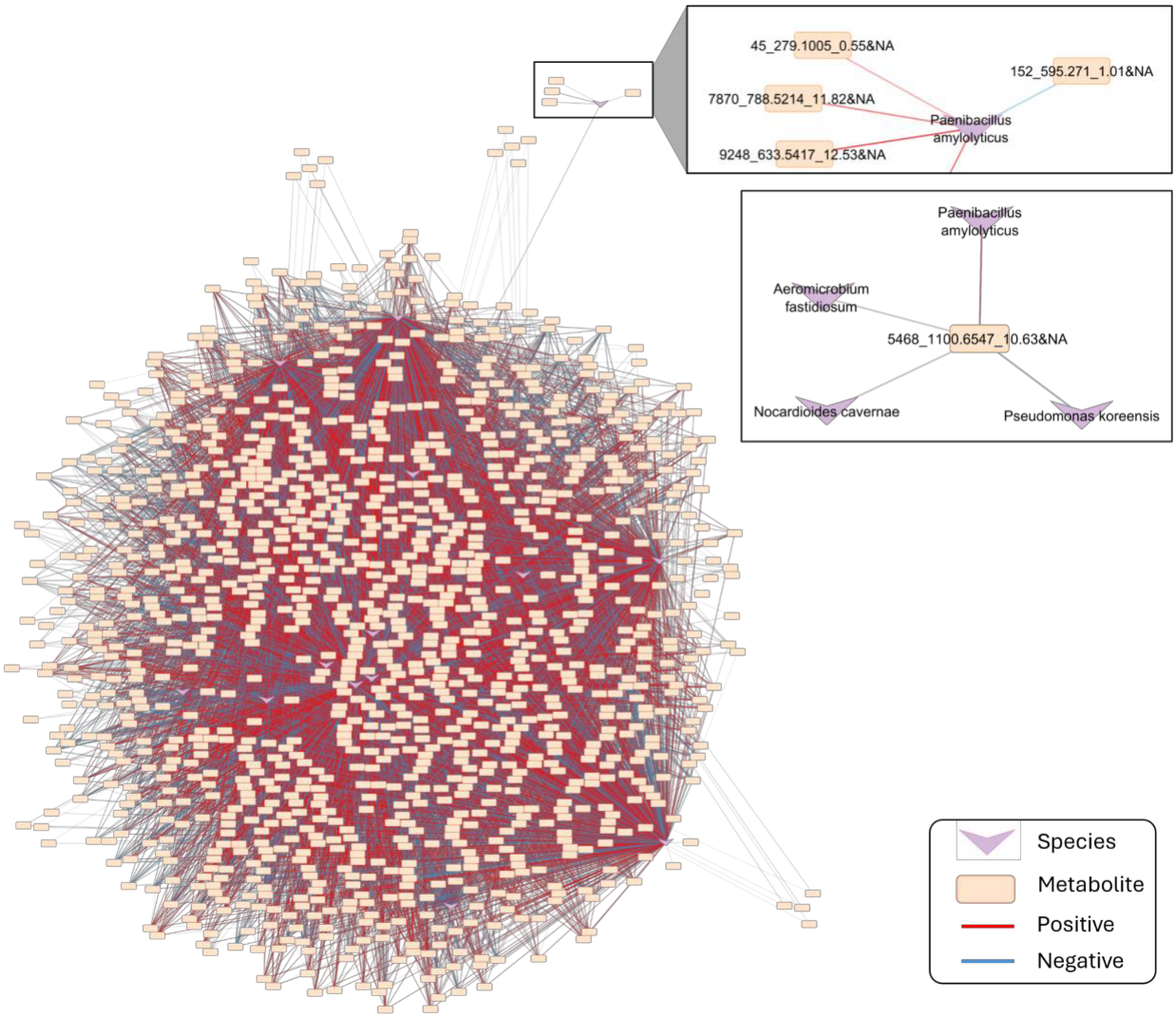


Figure 8: Pearson correlation network (FDR-adjusted scores) between species and metabolite features. Edges are colored red (positive) and blue (negative). A distinct *Paenibacillus* cluster with four correlated features is shown in the upper-right inset. The lower inset highlights *Paenibacillus's* link to a feature (with m/z 1100.6547 and RT 10.63 mins), which connects positively with *N. cavernae* and *A. fastidiosum* and negatively with *P. koreensis*, illustrating cross-species metabolic associations.

This pattern becomes clearer in **SI Figure S10**, which shows individual line plots of *Paenibacillus* and the four corresponding features across 12 experimental groups (two replicates each). *Paenibacillus* displays a gradual increase up to group 9, a sharp rise at group 10, followed by a drop, a non-linear but reproducible pattern. Features at C, D, and E exhibit similar trajectories, resulting in positive correlations (r = 0.6-0.7), whereas feature B shows the opposite trend and thus a negative correlation (r ≈ -0.6). Notably, *Paenibacillus* also connected positively to another feature (Feature ID 5648 with m/z 1100.6547 and RT 10.63 mins), which linked to *N. cavernae* and *A. fastidiosum* (positive) but *P. koreensis* (negative). These cross-links highlight potential co-metabolic or antagonistic relationships between species, illustrating how network visualization aids hypothesis generation regarding community-level metabolic interactions.

4.7 Conclusion

The CorrOmics app was developed to streamline correlation analyses between metabolomics data and other omics layers. In this study, we demonstrated its functionality using amplicon sequencing data, but the framework is readily extendable to additional omics types, including transcriptomics and proteomics. Future versions will incorporate further correlation metrics, such as SparCC, and offer transformation options tailored to secondary omics datasets to enhance interpretability.

Computation time in the web version depends on dataset size and server load. For smaller datasets (~30-40k correlations), results are typically obtained in under 30 seconds, and for medium datasets (~80k correlations) within one minute. Even large datasets (~925k correlations, as in the example dataset) complete within approximately four minutes. To ensure stable performance, analyses exceeding one million correlations are currently restricted on the GNPS2 server. Local installations, however, can achieve faster runtimes depending on the user's system and network connection.

Correlation analysis provides a complementary perspective to traditional univariate or multivariate statistics. While those methods emphasize differences between groups or features within a single

omics layer, CorrOmics enables the identification of cross-omics associations, for example, linking metabolites to microbial species, proteins, or transcripts that may drive or respond to their abundance changes. Outputs are provided as both CSV tables and GraphML network files, enabling users to explore and visualize results interactively and to generate biologically meaningful hypotheses.

CorrOmics enables users to move from raw omics data to interpretable biological associations. Depending on the question, users can apply it to diverse experimental setups. For example, in microbiome-metabolome studies, correlating bacterial abundances with metabolite intensities can reveal which taxa drive the biotransformation of a drug or nutrient compound. In paired proteomics-metabolomics experiments, correlations can highlight enzymes or transporters whose expression patterns track with specific metabolites, suggesting potential catalytic or regulatory links. Similarly, in paired transcriptomics-metabolomics studies, gene-metabolite correlations can identify co-regulated pathways or transcriptional responses underlying metabolic shifts, for instance, stress-induced metabolite accumulation matching the upregulation of biosynthetic genes. By transforming large omics matrices into feature-to-feature relationships, CorrOmics allows users to ask targeted biological questions such as which species are responsible for a metabolite change, which proteins or genes respond to it, and how these layers interact dynamically, thus bridging chemical observations with their biological context.

4.8 Availability and Requirements

Project name: CorrOmics
Project homepage: https://github.com/Functional-Metabolomics-Lab/Corromics
Operating system(s): Platform-independent (web-based and local installation supported)
Programming language: Python 3.11
Other requirements: streamlit, streamlit-extras, numpy, pandas, pandas_flavor, plotly, openpyxl, psutil, scipy, statsmodels, networkx

System requirements:

- Local installation supported on **Windows** (installer **.msi** provided) and on **macOS/Linux** via **git clone**
- For optimal performance, at least 8 GB RAM and a stable internet connection are recommended
- **License:** MIT License

- **Restrictions for non-academics:** None
- **Web app:** https://corromics.gnps2.org/

4.9 Author Contributions

AKPS and DP conceptualized the CorrOmics software. AKPS, KS, and DP designed the SynCom experiments. KS performed the SynCom experiments, mass spectrometry measurements, and amplicon sequencing. AKPS and MW developed the CorrOmics application. AKPS, KS, and DP analyzed the data. AKPS drafted the manuscript.

Acknowledgments

This study was supported by the Deutsche Forschungsgemeinschaft (German Research Foundation, DFG) via the Cluster of Excellence EXC 2124: Controlling Microbes to Fight Infection (CMFI, project ID 390838134) to DP. K.S was supported by National Science Foundation under Cooperative Agreement DBI-2400327. We further acknowledge support by the National Institute of General Medical Sciences, GM160154 to DP, and the National Institute of Diabetes and Digestive and Kidney Diseases, 5U24DK133658-02 to MW.

Chapter 5

General Discussion

5.1 Tool Development as a Driver of Biological Discovery

The central goal of this PhD work was to develop computational metabolomics approaches that enable a deeper understanding of biological interactions in microbial communities. To support this, three tools: **FBMN-STATS, ChemProp2, and Corromics**, were designed to address complementary challenges in data interpretation: from statistical exploration in chapter 2, to mechanistic inference in chapter 3, and to multi-omics integration in chapter 4. Together, these tools create a coherent workflow that transforms raw LC-MS/MS features into biologically interpretable insights.

FBMN-STATS was designed to lower the entry barrier for users who are new to untargeted metabolomics already using the large community of GNPS molecular networking user approaches. There is no dedicated pipeline to integrate structural information from Feature-Based Molecular Networking (FBMN) with robust statistical analysis. FBMN-STATS fills this gap by providing a reproducible workflow, implemented in R, Python, QIIME 2, and an interactive Streamlit application, that guides users through preprocessing, exploratory data analysis, and differential abundance statistics. This enables researchers not only to identify or annotate metabolites but also to understand how they change across conditions, relate to biological variables, and potentially influence the biological system under study.

FBMN itself simplifies interpretation by organizing thousands of features into molecular families based on spectral similarity[36,42]. This organization allows annotation propagation across related nodes, offering insight even when direct spectral matches are absent[368]. The ability to integrate this structural context with statistical results proved especially powerful for the community, as reflected in the broad uptake of FBMN-STATS since its release in 2023.

Transforming the original notebook-based workflow into a Streamlit application expanded accessibility further. The web app allows users to upload their FBMN results, explore PCA or univariate statistics, inspect resulting tables without writing code. Its development aligns with a broader shift in computational biology toward interactive, user-friendly tools that support transparent and shareable analysis. Since its launch on GNPS2[81,338,342], FBMN-STATS has undergone continuous refinement through user feedback, gaining features such as better metadata handling, optional annotation integration, customizable plots, and improved downstream visualizations.

FBMN-STATS established a foundation for rigorous and reproducible statistical exploration. However, identifying “which metabolites change” is only the first step toward understanding microbial chemistry. Addressing the mechanisms that produce these changes required a shift toward transformation-level inference, motivating the development of ChemProp2.

5.2 From Statistical Exploration to Mechanistic Interpretation: ChemProp2

5.2.1 Conceptual Motivation

ChemProp2 was developed to move beyond differential abundance patterns toward hypotheses about biochemical transformations. In time-resolved microbial metabolomics, many observed intensity shifts arise from enzymatic conversion of one compound into another. While multivariate statistics (e.g., PCA, PLS-DA) identify patterns, they cannot infer directionality. Similarly, FBMN groups structurally related molecules but does not indicate whether a compound is a precursor, product, or unrelated analog. Existing reaction-prediction frameworks such as BioTransformer[62] and MetWork[277] simulate transformations based on predefined rules, but they do not infer relationships directly from experimental time-series data. ChemProp2 bridges this gap by quantifying directional relationships between connected features in a molecular network, enabling researchers to identify putative precursor-product paths grounded in empirical data.

5.2.2 Extending molecular networking with directionality

It uses a correlation-based metric (Pearson or Spearman) to quantify temporal relationships between each connected feature pair. The score highlights anti-correlating behavior, where a strong correlation indicates potential transformation, and the sign (-1 or +1) denotes its direction between nodes (from node A to node B or vice versa). To better capture mild non-linear trends, ChemProp2 automatically computes log10- and square-root-transformed versions of the feature intensities in addition to the raw correlation. Cascade scoring further expands the interpretative power by tracing multi-step pathways beyond immediate neighbors.

Correlation-based approaches were chosen for their simplicity, interpretability, and reproducibility. Earlier studies showed that metabolite correlations often arise from systemic biochemical properties such as chemical equilibria, mass conservation, enzyme control, and gene expression variability, rather than mere pathway proximity[369]. In addition to correlation, the other common measure used in metabolomics to quantify association patterns is Mutual Information (MI). Although MI can capture non-linear dependencies, a recent study showed no consistent advantage of MI over correlation for network inference tasks[370], supporting the use of correlation as a robust first-pass metric.

Recognizing that these correlation metrics can yield false positives, ChemProp2 implements an FDR-based target-decoy approach, where shuffled decoy datasets help estimate ChemProp score cutoffs specific to each dataset. This provides a principled way to filter false positives. By

integrating with GNPS2 through a Streamlit interface, ChemProp2 makes transformation mapping accessible to a broad user base.

5.2.3 Biological Interpretation and Case Studies

The power of ChemProp2 lies in linking chemical dynamics to underlying biological processes. In the gut synthetic community experiment, ChemProp1 screening across 50 clinical drugs identified 12 candidates for deeper time-series analysis. ChemProp2 revealed multi-step transformation cascades for several drugs, including Cilnidipine, Metronidazole, Omeprazole, and Simvastatin. Cascade scoring expanded the number of drug-associated edges ~6 fold, uncovering distal products that would remain hidden in static network representations. These patterns are often aligned with microbial dynamics. Repository-scale context from FASST[281] for these ChemProp2 prioritized features helped distinguish microbially generated products from compounds common in unrelated environmental or clinical datasets, strengthening biological interpretation.

ChemProp2 has already been applied in external collaborations, including microcystin degradation pathways[371] and fungal breakdown of fengycin[372] demonstrating its utility across diverse environmental systems.

5.2.4 Limitations and Future Directions

ChemProp2 inherits assumptions from FBMN. If true metabolites are split across multiple subnetworks or appear as singletons due to spectral noise or insufficient similarity, directionality cannot be computed. This issue occurred in the omeprazole dataset, where known metabolites were disconnected from their parent. Since ChemProp2 relies on the node-pair table to compute directionality scores, singletons, having no edges, cannot be evaluated, and users must manually identify and include such features for separate inspection. A future extension will allow pairwise scoring of two specified IDs, bypassing network dependence.

In future versions, this network dependence could be bypassed by adding a pairwise comparison option, where users specify two feature IDs to compute their ChemProp score and visualize it as a line or box plot, complemented by a mirror plot of their MS/MS spectra to assess structural relatedness. While network visualization would not be possible in such isolated cases, this functionality could still provide valuable insight into potential biotransformations outside of network connectivity.

Other observed limitations relate to ion formation and spectral overlap. Certain adducts or in-source fragments may form independent clusters, breaking the link between related compounds. Ion Identity Networking (IIN) can partially address this by reuniting ions belonging to the same molecule, though it remains an external preprocessing step. In the metronidazole dataset, for instance, several features were incorrectly grouped into neighboring clusters, emphasizing the limitations of relying solely on spectral similarity. As a preliminary quality check, users can visualize intensity distributions across timepoints or conditions: for example, using pie-chart representations in Cytoscape to confirm whether features are treatment-specific or appear in both conditions. Incorporating such visual diagnostics directly into ChemProp2 could reduce manual

curation and improve interpretability, for instance, by assigning distinct colors to nodes that appear in neither condition (e.g., black for absent features), allowing users to rapidly assess condition-specific patterns.

Further improvements may incorporate alternative spectral similarity metrics (e.g., Spec2Vec[373] and MS2DeepScore[374] , MS2Query[375], MESSAR[376], DeepMASS[377], MetFrag[378], CFM-ID[379], QCxMS2[380], and ICEBERG[381]) to strengthen network connectivity and reduce false associations.

Finally, the current target-decoy strategy may occasionally reproduce patterns resembling real data. A more reliable benchmark could involve constructing decoy node pairs from features with no temporal growth, ensuring that background or constitutive signals are separated from biologically driven transformations. Developing such benchmark datasets and integrated visualization frameworks would further validate ChemProp2's robustness.

5.3 From Chemical Transformations to Biological Associations: Introducing CorrOmics

5.3.1 Conceptual Motivation and Benchmarking

While ChemProp2 assigns directionality to metabolite-metabolite pairs, understanding the biological drivers of these chemical changes requires integrating metabolomics with complementary omics data such as microbial abundances. Existing options were either too simple (R scripts) or too complex such as latent-variable models like DIABLO[75] or neural network frameworks like mmvec[76]. Many researchers require a transparent, feature-level exploratory tool. CorrOmics fills this gap by offering scalable, interpretable, correlation-based integration between LC-MS/MS intensities and OTU/ASV profiles.

CorrOmics was benchmarked using a 13-strain Arabidopsis SynCom with 12 designed mixture groups. Eight metabolites were spiked in with matching gradients, creating a mock system with known associations. The tool successfully recovered the expected correlations between spiked-in metabolites and their corresponding strain abundance trajectories. However, the benchmark also revealed challenges inherent to mock-mixture designs: each strain contributed its own metabolite background, producing many additional associations unrelated to the intentional gradients. This "mix-driven covariation" inflated correlations and produced dense networks.

Other technical factors also affected interpretation. Sequencing data showed the intended trends, but the strains intended to remain constant drifted slightly upward across groups. Spiked-in metabolites tracked their expected patterns, though with intensity variation likely driven by ionization or matrix effects[352,353].

These results emphasize that correlation strength depends heavily on experimental context, measurement scale, and normalization[67]. Relative abundances preserved trends in this dataset but are not ideal when total population sizes vary substantially. In real microbiome datasets, estimates of total microbial load (for example via OD, CFU counts) provide a more biologically

meaningful basis for omics comparison. Similarly, Pearson correlations perform best under linear gradients, whereas Spearman is more robust to monotonic but nonlinear patterns.

5.3.2 Limitations and Future Directions

CorrOmics incorporates a target-decoy FDR strategy that effectively reduces false positive correlations; however, FDR behavior remains highly dataset-dependent. In sparse datasets, such as an HIV dataset[382], most significant correlations were weak (≈±0.2), likely reflecting zero inflation and compositional noise rather than true biological signal. To address these challenges, upcoming versions of CorrOmics will explore integrating sparsity-aware correlation measures such as SparCC[383], CCLasso[384], and eventually mmvec[76].

The incorporation of hierarchical binning allows correlations to be evaluated at multiple taxonomic resolutions, mitigating instability at the ASV level, for example, collapsing ASVs to genus-level groups when species-level estimates are unstable[385,386]. Future expansions will extend CorrOmics to proteomics and transcriptomics and provide broader data transformations beyond microbiome-focused options.

As with any correlation-based tool, CorrOmics supports hypothesis generation rather than causal inference. True biological linkages require complementary evidence such as genomic context, pathway annotation, isotopic tracing, enzyme assays, or controlled perturbation experiments to distinguish true biological relationships from coincidental co-variation.

5.4 Integrating the Thesis Contributions

Across the three chapters of this thesis, the goal was to develop computational metabolomics approaches that help with the interpretation of chemical interactions in microbial communities. Untargeted LC-MS/MS is a powerful yet inherently complex technology, and the work presented here contributes to reducing this complexity through computational tool development. Within the GNPS2 ecosystem built to support modular, interoperable workflows, these tools form part of the growing MetaboApps. MetaboApps collectively advance downstream analysis through pattern-based querying (e.g., PostMN MassQL), ontology-driven contextualization (e.g., Food/Drug Readout), statistical and multi-omics integration (e.g., FBMN-STATS, ChemProp2, CorrOmics), and repository-scale investigations (e.g., MASST)[285]. The three tools developed in this thesis contribute to this ecosystem by supporting post-processing of molecular networking outputs, mechanistic interpretation, and cross-omics correlation analysis, enabling users to extract biological meaning from large-scale metabolomics experiments.

In parallel with these developments, the broader landscape of computational biology is being reshaped by artificial intelligence. Large-language-model tools such as ChatGPT and GitHub Copilot are transforming how researchers code, write, and conceptualize analyses[387,388]. Recent studies highlight both the opportunities and challenges posed by these technologies, from accelerating exploration to identifying usability gaps in real scientific workflows[389–392]. Far from replacing scientific reasoning, these tools act as amplifiers of human insight, lowering technical

barriers and enhancing reproducibility. When combined with community-driven platforms like GNPS2, LLMs will further pave the way for a future where computational and analytical innovation, and microbial ecology converge even more seamlessly. Ultimately, the work presented in this thesis underscores how interdisciplinary tool development, spanning biology, chemistry, computer science, and data science, can accelerate biological discovery. As analytical technologies and AI continue to advance, the ability to integrate, interpret, and contextualize complex datasets will only become more central to understanding chemical interactions in microbial ecosystems.

Conclusion

Untargeted metabolomics provides a powerful window into the chemistry underlying microbial ecosystems, yet its complexity demands robust computational approaches. This thesis contributed three such tools, FBMN-STATS, ChemProp2, and CorrOmics, that together support data preprocessing, mechanistic inference, and cross-omics integration within the GNPS2 MetaboApps ecosystem. By combining statistical rigor with accessible, modular interfaces, these tools lower analytical barriers and enable researchers to interrogate chemical and microbial relationships at scale. More broadly, the work demonstrates the value of interdisciplinary development, where biological questions are advanced through methods drawn from computer science, data science, and mass spectrometry. As multi-omics datasets and AI-assisted workflows continue to grow, such integrative approaches will play an increasingly central role in deciphering the chemical interactions that shape microbial communities.

References

1. Zulfiqar, M., Singh, V., Steinbeck, C. & Sorokina, M. Review on computer-assisted biosynthetic capacities elucidation to assess metabolic interactions and communication within microbial communities. *Crit. Rev. Microbiol.* **50**, 1053–1092 (2024).
2. Phelan, V. V., Liu, W.-T., Pogliano, K. & Dorrestein, P. C. Microbial metabolic exchange--the chemotype-to-phenotype link. *Nat. Chem. Biol.* **8**, 26–35 (2011).
3. Zhang, C. & Straight, P. D. Antibiotic discovery through microbial interactions. *Curr. Opin. Microbiol.* **51**, 64–71 (2019).
4. Abdel-Razek, A. S. *et al.* Microbial Natural Products in Drug Discovery. *Processes* **8**, (2020).
5. Turnbaugh, P. J. *et al.* The Human Microbiome Project. *Nature* **449**, 804–810 (2007).
6. Sommer, F. & Bäckhed, F. The gut microbiota — masters of host development and physiology. *Nat. Rev. Microbiol.* **11**, 227–238 (2013).
7. Lynch, S. V. & Pedersen, O. The Human Intestinal Microbiome in Health and Disease. *N. Engl. J. Med.* **375**, 2369–2379 (2016).
8. Gilbert, J. A. *et al.* Current understanding of the human microbiome. *Nat. Med.* **24**, 392–400 (2018).
9. Honda, K. & Littman, D. R. The microbiome in infectious disease and inflammation. *Annu. Rev. Immunol.* **30**, 759–795 (2012).
10. Safarchi, A., Al-Qadami, G., Tran, C. D. & Conlon, M. Understanding dysbiosis and resilience in the human gut microbiome: biomarkers, interventions, and challenges. *Front. Microbiol.* **16**, (2025).
11. Sogin, M. L. *et al.* Microbial diversity in the deep sea and the underexplored 'rare biosphere'. *Proc. Natl. Acad. Sci. U. S. A.* **103**, 12115–12120 (2006).
12. Buffie, C. G. & Pamer, E. G. Microbiota-mediated colonization resistance against intestinal pathogens. *Nat. Rev. Immunol.* **13**, 790–801 (2013).
13. Niehaus, L. *et al.* Microbial coexistence through chemical-mediated interactions. *Nat. Commun.* **10**, 2052 (2019).
14. Tshikantwa, T. S., Ullah, M. W., He, F. & Yang, G. Current Trends and Potential Applications of Microbial Interactions for Human Welfare. *Front. Microbiol.* **9**, (2018).
15. Keller, L. & Surette, M. G. Communication in bacteria: an ecological and evolutionary perspective. *Nat. Rev. Microbiol.* **4**, 249–258 (2006).
16. Crespi, B. J. The evolution of social behavior in microorganisms. *Trends Ecol. Evol.* **16**, 178–183 (2001).
17. Malik, V. S. Microbial secondary metabolism. *Trends Biochem. Sci.* **5**, 68–72 (1980).
18. Demain, A. L. & Fang, A. The Natural Functions of Secondary Metabolites. in *History of Modern Biotechnology I* (ed. Fiechter, A.) 1–39 (Springer, Berlin, Heidelberg, 2000). doi:10.1007/3-540-44964-7_1.
19. Martín, J.-F. & Liras, P. Engineering of regulatory cascades and networks controlling antibiotic biosynthesis in *Streptomyces*. *Curr. Opin. Microbiol.* **13**, 263–273 (2010).

20. Heul, H. U. van der, Bilyk, B. L., McDowall, K. J., Seipke, R. F. & Wezel, G. P. van. Regulation of antibiotic production in Actinobacteria: new perspectives from the post-genomic era. *Nat. Prod. Rep.* **35**, 575–604 (2018).
21. Bertrand, S. *et al.* Metabolite induction via microorganism co-culture: a potential way to enhance chemical diversity for drug discovery. *Biotechnol. Adv.* **32**, 1180–1204 (2014).
22. Konopka, A. What is microbial community ecology? *ISME J.* **3**, 1223–1230 (2009).
23. Prosser, J. I. *et al.* The role of ecological theory in microbial ecology. *Nat. Rev. Microbiol.* **5**, 384–392 (2007).
24. Qin, J. *et al.* A human gut microbial gene catalogue established by metagenomic sequencing. *Nature* **464**, 59–65 (2010).
25. Wang, Z., Gerstein, M. & Snyder, M. RNA-Seq: a revolutionary tool for transcriptomics. *Nat. Rev. Genet.* **10**, 57–63 (2009).
26. Aebersold, R. & Mann, M. Mass spectrometry-based proteomics. *Nature* **422**, 198–207 (2003).
27. Wishart, D. S. Metabolomics for Investigating Physiological and Pathophysiological Processes. *Physiol. Rev.* **99**, 1819–1875 (2019).
28. Oliver, S. G. Functional genomics: lessons from yeast. *Philos. Trans. R. Soc. B Biol. Sci.* **357**, 17–23 (2002).
29. Nicholson, J. K., Connelly, J., Lindon, J. C. & Holmes, E. Metabonomics: a platform for studying drug toxicity and gene function. *Nat. Rev. Drug Discov.* **1**, 153–161 (2002).
30. Xiao, J. F., Zhou, B. & Ressom, H. W. Metabolite identification and quantitation in LC-MS/MS-based metabolomics. *TrAC Trends Anal. Chem.* **32**, 1–14 (2012).
31. Cajka, T. & Fiehn, O. Toward Merging Untargeted and Targeted Methods in Mass Spectrometry-Based Metabolomics and Lipidomics. *Anal. Chem.* **88**, 524–545 (2016).
32. Petras, D. *et al.* Non-targeted tandem mass spectrometry enables the visualization of organic matter chemotype shifts in coastal seawater. *Chemosphere* **271**, 129450 (2021).
33. Schwaiger-Haber, M. *et al.* Using mass spectrometry imaging to map fluxes quantitatively in the tumor ecosystem. *Nat. Commun.* **14**, 2876 (2023).
34. da Silva, R. R., Dorrestein, P. C. & Quinn, R. A. Illuminating the dark matter in metabolomics. *Proc. Natl. Acad. Sci.* **112**, 12549–12550 (2015).
35. Alseekh, S. *et al.* Mass spectrometry-based metabolomics: a guide for annotation, quantification and best reporting practices. *Nat. Methods* **18**, 747–756 (2021).
36. Aron, A. T. *et al.* Reproducible molecular networking of untargeted mass spectrometry data using GNPS. *Nat. Protoc.* **15**, 1954–1991 (2020).
37. Nguyen, Q.-H., Nguyen, H., Oh, E. C. & Nguyen, T. Current approaches and outstanding challenges of functional annotation of metabolites: a comprehensive review. *Brief. Bioinform.* **25**, bbae498 (2024).
38. Peng, R. D. Reproducible Research in Computational Science. *Science* **334**, 1226–1227 (2011).
39. Sandve, G. K., Nekrutenko, A., Taylor, J. & Hovig, E. Ten Simple Rules for Reproducible Computational Research. *PLOS Comput. Biol.* **9**, e1003285 (2013).
40. Mendez, K. M., Pritchard, L., Reinke, S. N. & Broadhurst, D. I. Toward collaborative open data science in metabolomics using Jupyter Notebooks and cloud computing. *Metabolomics* **15**, 125 (2019).

41. de Jonge, N. F. *et al.* Good practices and recommendations for using and benchmarking computational metabolomics metabolite annotation tools. *Metabolomics* **18**, 103 (2022).
42. Nothias, L.-F. *et al.* Feature-based molecular networking in the GNPS analysis environment. *Nat. Methods* **17**, 905–908 (2020).
43. Wasserstein, R. L., Schirm, A. L. & Lazar, N. A. Moving to a World Beyond “p < 0.05”. *Am. Stat.* **73**, 1–19 (2019).
44. Pang, Z. *et al.* MetaboAnalyst 5.0: narrowing the gap between raw spectra and functional insights. *Nucleic Acids Res.* **49**, W388–W396 (2021).
45. Dührkop, K. *et al.* SIRIUS 4: a rapid tool for turning tandem mass spectra into metabolite structure information. *Nat. Methods* **16**, 299–302 (2019).
46. Patti, G. J., Yanes, O. & Siuzdak, G. Metabolomics: the apogee of the omics trilogy. *Nat. Rev. Mol. Cell Biol.* **13**, 263–269 (2012).
47. Chin, J. P., McGrath, J. W. & Quinn, J. P. Microbial transformations in phosphonate biosynthesis and catabolism, and their importance in nutrient cycling. *Curr. Opin. Chem. Biol.* **31**, 50–57 (2016).
48. Zimmermann, M., Zimmermann-Kogadeeva, M., Wegmann, R. & Goodman, A. L. Mapping human microbiome drug metabolism by gut bacteria and their genes. *Nature* **570**, 462–467 (2019).
49. Du, M. *et al.* Anaerobic biotransformation mechanism of marine toxin domoic acid. *J. Hazard. Mater.* **421**, 126798 (2022).
50. López de Lacey, A. M., Pérez-Santín, E., López-Caballero, M. E. & Montero, P. Biotransformation and resulting biological properties of green tea polyphenols produced by probiotic bacteria. *LWT - Food Sci. Technol.* **58**, 633–638 (2014).
51. Zhang, D. *et al.* A data-driven integrative platform for computational prediction of toxin biotransformation with a case study. *J. Hazard. Mater.* **408**, 124810 (2021).
52. Wilson, I. D. & Nicholson, J. K. Gut microbiome interactions with drug metabolism, efficacy, and toxicity. *Transl. Res.* **179**, 204–222 (2017).
53. Klünemann, M. *et al.* Bioaccumulation of therapeutic drugs by human gut bacteria. *Nature* **597**, 533–538 (2021).
54. Grießhammer, A. *et al.* Non-antibiotics disrupt colonization resistance against enteropathogens. *Nature* **644**, 497–505 (2025).
55. Subanovic, M., Frawley, D., Tierney, C., Velasco-Torrijos, T. & Walsh, F. Proteomic and metabolomic responses of priority bacterial pathogens to subinhibitory concentration of antibiotics. *Npj Antimicrob. Resist.* **3**, 80 (2025).
56. Lin, Y., Jia, Y., Zhou, C., Wang, H. & Pan, C. Impact of Pesticide Abiotic Stresses on Plant Secondary Metabolism: From Plant Individuals to Ecological Interfaces. *J. Agric. Food Chem.* **73**, 21247–21263 (2025).
57. Kuhlisch, C. *et al.* Viral infection of algal blooms leaves a unique metabolic footprint on the dissolved organic matter in the ocean. *Sci. Adv.* **7**, eabf4680 (2021).
58. Farrell, S. P. *et al.* Turf algae redefine the chemical landscape of temperate reefs, limiting kelp forest recovery. *Science* **388**, 876–880 (2025).
59. Worley, B. & Powers, R. Multivariate Analysis in Metabolomics. *Curr. Metabolomics* **1**, 92–107 (2013).

60. Chong, J. *et al.* MetaboAnalyst 4.0: towards more transparent and integrative metabolomics analysis. *Nucleic Acids Res.* **46**, W486–W494 (2018).
61. Costello, Z. & Martin, H. G. A machine learning approach to predict metabolic pathway dynamics from time-series multiomics data. *Npj Syst. Biol. Appl.* **4**, 19 (2018).
62. Djoumbou-Feunang, Y. *et al.* BioTransformer: a comprehensive computational tool for small molecule metabolism prediction and metabolite identification. *J. Cheminformatics* **11**, 2 (2019).
63. Malwe, A. S., Srivastava, G. N. & Sharma, V. K. GutBug: A Tool for Prediction of Human Gut Bacteria Mediated Biotransformation of Biotic and Xenobiotic Molecules Using Machine Learning. *J. Mol. Biol.* **435**, 168056 (2023).
64. Bersanelli, M. *et al.* Methods for the integration of multi-omics data: mathematical aspects. *BMC Bioinformatics* **17**, S15 (2016).
65. Zhalnina, K. *et al.* Dynamic root exudate chemistry and microbial substrate preferences drive patterns in rhizosphere microbial community assembly. *Nat. Microbiol.* **3**, 470–480 (2018).
66. The Integrative Human Microbiome Project: Dynamic Analysis of Microbiome-Host Omics Profiles during Periods of Human Health and Disease. *Cell Host Microbe* **16**, 276–289 (2014).
67. Knight, R. *et al.* Best practices for analysing microbiomes. *Nat. Rev. Microbiol.* **16**, 410–422 (2018).
68. Weiss, S. *et al.* Correlation detection strategies in microbial data sets vary widely in sensitivity and precision. *ISME J.* **10**, 1669–1681 (2016).
69. Shaffer, J. P. *et al.* Standardized multi-omics of Earth's microbiomes reveals microbial and metabolite diversity. *Nat. Microbiol.* **7**, 2128–2150 (2022).
70. Durham, B. P. *et al.* Sulfonate-based networks between eukaryotic phytoplankton and heterotrophic bacteria in the surface ocean. *Nat. Microbiol.* **4**, 1706–1715 (2019).
71. Nguyen, Q. P. *et al.* Associations between the gut microbiome and metabolome in early life. *BMC Microbiol.* **21**, 238 (2021).
72. D'Souza, G. *et al.* Ecology and evolution of metabolic cross-feeding interactions in bacteria. *Nat. Prod. Rep.* **35**, 455–488 (2018).
73. Roach, J. *et al.* Microbiome metabolite quantification methods enabling insights into human health and disease. *Methods* **222**, 81–99 (2024).
74. Koppel, N., Maini Rekdal, V. & Balskus, E. P. Chemical transformation of xenobiotics by the human gut microbiota. *Science* **356**, eaag2770 (2017).
75. Rohart, F., Gautier, B., Singh, A. & Lê Cao, K.-A. mixOmics: An R package for 'omics feature selection and multiple data integration. *PLOS Comput. Biol.* **13**, e1005752 (2017).
76. Morton, J. T. *et al.* Learning representations of microbe–metabolite interactions. *Nat. Methods* **16**, 1306–1314 (2019).
77. Khorasani, M., Abdou, M. & Hernández Fernández, J. Getting Started with Streamlit. in *Web Application Development with Streamlit: Develop and Deploy Secure and Scalable Web Applications to the Cloud Using a Pure Python Framework* (eds Khorasani, M., Abdou, M. & Hernández Fernández, J.) 1–30 (Apress, Berkeley, CA, 2022). doi:10.1007/978-1-4842-8111-6_1.
78. Vailati-Riboni, M., Palombo, V. & Loor, J. J. What Are Omics Sciences? in *Periparturient Diseases of Dairy Cows: A Systems Biology Approach* (ed. Ametaj, B. N.) 1–7 (Springer International Publishing, Cham, 2017). doi:10.1007/978-3-319-43033-1_1.

79. Dayalan, S., Xia, J., Spicer, R. A., Salek, R. & Roessner, U. Metabolome Analysis. in *Encyclopedia of Bioinformatics and Computational Biology* (eds Ranganathan, S., Gribskov, M., Nakai, K. & Schönbach, C.) 396–409 (Academic Press, Oxford, 2019). doi:10.1016/B978-0-12-809633-8.20251-3.
80. Tolstikov, V., Moser, A. J., Sarangarajan, R., Narain, N. R. & Kiebish, M. A. Current Status of Metabolomic Biomarker Discovery: Impact of Study Design and Demographic Characteristics. *Metabolites* **10**, 224 (2020).
81. Wang, M. *et al.* Sharing and community curation of mass spectrometry data with Global Natural Products Social Molecular Networking. *Nat. Biotechnol.* **34**, 828–837 (2016).
82. Ottosson, F. *et al.* Effects of Long-Term Storage on the Biobanked Neonatal Dried Blood Spot Metabolome. *J. Am. Soc. Mass Spectrom.* **34**, 685–694 (2023).
83. Dantas Machado, A. C. *et al.* Portosystemic shunt placement reveals blood signatures for the development of hepatic encephalopathy through mass spectrometry. *Nat. Commun.* **14**, 5303 (2023).
84. Xie, H.-F. *et al.* Feature-Based Molecular Networking Analysis of the Metabolites Produced by *In Vitro* Solid-State Fermentation Reveals Pathways for the Bioconversion of Epigallocatechin Gallate. *J. Agric. Food Chem.* **68**, 7995–8007 (2020).
85. Berlanga-Clavero, M. V. *et al.* Bacillus subtilis biofilm matrix components target seed oil bodies to promote growth and anti-fungal resistance in melon. *Nat. Microbiol.* **7**, 1001–1015 (2022).
86. Raheem, D. J., Tawfike, A. F., Abdelmohsen, U. R., Edrada-Ebel, R. & Fitzsimmons-Thoss, V. Application of metabolomics and molecular networking in investigating the chemical profile and antitrypanosomal activity of British bluebells (Hyacinthoides non-scripta). *Sci. Rep.* **9**, 2547 (2019).
87. Pendergraft, M. A. *et al.* Bacterial and Chemical Evidence of Coastal Water Pollution from the Tijuana River in Sea Spray Aerosol. *Environ. Sci. Technol.* **57**, 4071–4081 (2023).
88. Stincone, P. *et al.* Evaluation of Data-Dependent MS/MS Acquisition Parameters for Non-Targeted Metabolomics and Molecular Networking of Environmental Samples: Focus on the Q Exactive Platform. *Anal. Chem.* **95**, 12673–12682 (2023).
89. Wegley Kelly, L. *et al.* Distinguishing the molecular diversity, nutrient content, and energetic potential of exometabolomes produced by macroalgae and reef-building corals. *Proc. Natl. Acad. Sci.* **119**, e2110283119 (2022).
90. Mannochio-Russo, H. *et al.* Microbiomes and metabolomes of dominant coral reef primary producers illustrate a potential role for immunolipids in marine symbioses. *Commun. Biol.* **6**, 896 (2023).
91. Molina-Santiago, C. *et al.* Chemical interplay and complementary adaptative strategies toggle bacterial antagonism and co-existence. *Cell Rep.* **36**, 109449 (2021).
92. Reher, R. *et al.* Native metabolomics identifies the rivulariapeptolide family of protease inhibitors. *Nat. Commun.* **13**, 4619 (2022).
93. Aron, A. T. *et al.* Native mass spectrometry-based metabolomics identifies metal-binding compounds. *Nat. Chem.* **14**, 100–109 (2022).
94. Behnsen, J. *et al.* Siderophore-mediated zinc acquisition enhances enterobacterial colonization of the inflamed gut. *Nat. Commun.* **12**, 7016 (2021).

95. Pang, Z. *et al.* Using MetaboAnalyst 5.0 for LC–HRMS spectra processing, multi-omics integration and covariate adjustment of global metabolomics data. *Nat. Protoc.* **17**, 1735–1761 (2022).
96. Alder, L., Greulich, K., Kempe, G. & Vieth, B. Residue analysis of 500 high priority pesticides: Better by GC–MS or LC–MS/MS? *Mass Spectrom. Rev.* **25**, 838–865 (2006).
97. Díaz-Cruz, M. S., López de Alda, M. J., López, R. & Barceló, D. Determination of estrogens and progestogens by mass spectrometric techniques (GC/MS, LC/MS and LC/MS/MS). *J. Mass Spectrom.* **38**, 917–923 (2003).
98. Michely, J. A., Helfer, A. G., Brandt, S. D., Meyer, M. R. & Maurer, H. H. Metabolism of the new psychoactive substances N,N-diallyltryptamine (DALT) and 5-methoxy-DALT and their detectability in urine by GC–MS, LC–MSn, and LC–HR–MS–MS. *Anal. Bioanal. Chem.* **407**, 7831–7842 (2015).
99. Di Masi, S. *et al.* HPLC-MS/MS method applied to an untargeted metabolomics approach for the diagnosis of “olive quick decline syndrome”. *Anal. Bioanal. Chem.* **414**, 465–473 (2022).
100. Reveglia, P. *et al.* Untargeted and Targeted LC-MS/MS Based Metabolomics Study on In Vitro Culture of Phaeoacremonium Species. *J. Fungi* **8**, 55 (2022).
101. Baig, F., Pechlaner, R. & Mayr, M. Caveats of Untargeted Metabolomics for Biomarker Discovery∗. *J. Am. Coll. Cardiol.* **68**, 1294–1296 (2016).
102. Blaženović, I. *et al.* Comprehensive comparison of in silico MS/MS fragmentation tools of the CASMI contest: database boosting is needed to achieve 93% accuracy. *J. Cheminformatics* **9**, 32 (2017).
103. Blaženović, I., Kind, T., Ji, J. & Fiehn, O. Software Tools and Approaches for Compound Identification of LC-MS/MS Data in Metabolomics. *Metabolites* **8**, 31 (2018).
104. Dührkop, K., Shen, H., Meusel, M., Rousu, J. & Böcker, S. Searching molecular structure databases with tandem mass spectra using CSI:FingerID. *Proc. Natl. Acad. Sci.* **112**, 12580–12585 (2015).
105. Böcker, S., Letzel, M. C., Lipták, Z. & Pervukhin, A. SIRIUS: decomposing isotope patterns for metabolite identification†. *Bioinformatics* **25**, 218–224 (2009).
106. Stravs, M. A., Dührkop, K., Böcker, S. & Zamboni, N. MSNovelist: de novo structure generation from mass spectra. *Nat. Methods* **19**, 865–870 (2022).
107. Schmid, R. *et al.* Ion identity molecular networking for mass spectrometry-based metabolomics in the GNPS environment. *Nat. Commun.* **12**, 3832 (2021).
108. Kessner, D., Chambers, M., Burke, R., Agus, D. & Mallick, P. ProteoWizard: open source software for rapid proteomics tools development. *Bioinformatics* **24**, 2534–2536 (2008).
109. Hulstaert, N. *et al.* ThermoRawFileParser: Modular, Scalable, and Cross-Platform RAW File Conversion. *J. Proteome Res.* **19**, 537–542 (2020).
110. Adusumilli, R. & Mallick, P. Data Conversion with ProteoWizard msConvert. *Methods Mol. Biol. Clifton NJ* **1550**, 339–368 (2017).
111. Smith, C. A., Want, E. J., O’Maille, G., Abagyan, R. & Siuzdak, G. XCMS: Processing Mass Spectrometry Data for Metabolite Profiling Using Nonlinear Peak Alignment, Matching, and Identification. *Anal. Chem.* **78**, 779–787 (2006).
112. Kuhl, C., Tautenhahn, R., Böttcher, C., Larson, T. R. & Neumann, S. CAMERA: An Integrated Strategy for Compound Spectra Extraction and Annotation of Liquid Chromatography/Mass Spectrometry Data Sets. *Anal. Chem.* **84**, 283–289 (2012).

113. Schmid, R. *et al.* Integrative analysis of multimodal mass spectrometry data in MZmine 3. *Nat. Biotechnol.* **41**, 447–449 (2023).
114. Tsugawa, H. *et al.* A lipidome atlas in MS-DIAL 4. *Nat. Biotechnol.* **38**, 1159–1163 (2020).
115. Pfeuffer, J. *et al.* OpenMS – A platform for reproducible analysis of mass spectrometry data. *J. Biotechnol.* **261**, 142–148 (2017).
116. Gloaguen, Y., Kirwan, J. A. & Beule, D. Deep Learning-Assisted Peak Curation for Large-Scale LC-MS Metabolomics. *Anal. Chem.* **94**, 4930–4937 (2022).
117. Chetnik, K., Petrick, L. & Pandey, G. MetaClean: a machine learning-based classifier for reduced false positive peak detection in untargeted LC–MS metabolomics data. *Metabolomics* **16**, 117 (2020).
118. El Abiead, Y., Milford, M., Salek, R. M. & Koellensperger, G. mzRAPP: a tool for reliability assessment of data pre-processing in non-targeted metabolomics. *Bioinformatics* **37**, 3678–3680 (2021).
119. Damiani, T. *et al. Mass Spectrometry Data Processing in MZmine 3: Feature Detection and Annotation*. https://chemrxiv.org/engage/chemrxiv/article-details/6560961229a13c4d47e3bf51 (2023) doi:10.26434/chemrxiv-2023-98n6q.
120. Sumner, L. W. *et al.* Proposed minimum reporting standards for chemical analysis. *Metabolomics* **3**, 211–221 (2007).
121. Dührkop, K. *et al.* Systematic classification of unknown metabolites using high-resolution fragmentation mass spectra. *Nat. Biotechnol.* **39**, 462–471 (2021).
122. Liu, L.-L. *et al.* Molecular networking-based for the target discovery of potent antiproliferative polycyclic macrolactam ansamycins from Streptomyces cacaoi subsp. asoensis. *Org. Chem. Front.* **7**, 4008–4018 (2020).
123. Sedio, B. E., Boya P., C. A. & Rojas Echeverri, J. C. A protocol for high-throughput, untargeted forest community metabolomics using mass spectrometry molecular networks. *Appl. Plant Sci.* **6**, e1033 (2018).
124. Quinn, R. A. *et al.* Molecular Networking As a Drug Discovery, Drug Metabolism, and Precision Medicine Strategy. *Trends Pharmacol. Sci.* **38**, 143–154 (2017).
125. Pluskal, T., Castillo, S., Villar-Briones, A. & Orešič, M. MZmine 2: Modular framework for processing, visualizing, and analyzing mass spectrometry-based molecular profile data. *BMC Bioinformatics* **11**, 395 (2010).
126. Nguyen, L. H. & Holmes, S. Ten quick tips for effective dimensionality reduction. *PLOS Comput. Biol.* **15**, e1006907 (2019).
127. GOWER, J. C. Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika* **53**, 325–338 (1966).
128. Xu, Y. *et al.* Application of Dissimilarity Indices, Principal Coordinates Analysis, and Rank Tests to Peak Tables in Metabolomics of the Gas Chromatography/Mass Spectrometry of Human Sweat. *Anal. Chem.* **79**, 5633–5641 (2007).
129. Tian, M. *et al.* Pure Ion Chromatograms Combined with Advanced Machine Learning Methods Improve Accuracy of Discriminant Models in LC–MS-Based Untargeted Metabolomics. *Molecules* **26**, 2715 (2021).
130. Cacciatore, S., Tenori, L., Luchinat, C., Bennett, P. R. & MacIntyre, D. A. KODAMA: an R package for knowledge discovery and data mining. *Bioinformatics* **33**, 621–623 (2017).

131. Paliy, O. & Shankar, V. Application of multivariate statistical techniques in microbial ecology. *Mol. Ecol.* **25**, 1032–1057 (2016).
132. Efron, B. Bootstrap Methods: Another Look at the Jackknife. in *Breakthroughs in Statistics: Methodology and Distribution* (eds Kotz, S. & Johnson, N. L.) 569–593 (Springer, New York, NY, 1992). doi:10.1007/978-1-4612-4380-9_41.
133. Desu, M. M. & Raghavarao, D. *Nonparametric Statistical Methods For Complete and Censored Data*. (CRC Press, 2003).
134. Anderson, M. J. A new method for non-parametric multivariate analysis of variance. *Austral Ecol.* **26**, 32–46 (2001).
135. Djoumbou Feunang, Y. *et al.* ClassyFire: automated chemical classification with a comprehensive, computable taxonomy. *J. Cheminformatics* **8**, 61 (2016).
136. Kim, H. W. *et al.* NPClassifier: A Deep Neural Network-Based Structural Classification Tool for Natural Products. *J. Nat. Prod.* **84**, 2795–2807 (2021).
137. Tibshirani, R., Walther, G. & Hastie, T. Estimating the number of clusters in a data set via the gap statistic. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **63**, 411–423 (2001).
138. Benton, P. H. *et al.* An Interactive Cluster Heat Map to Visualize and Explore Multidimensional Metabolomic Data. *Metabolomics Off. J. Metabolomic Soc.* **11**, 1029–1034 (2015).
139. Ren, S., Hinzman, A. A., Kang, E. L., Szczesniak, R. D. & Lu, L. J. Computational and statistical analysis of metabolomics data. *Metabolomics* **11**, 1492–1513 (2015).
140. Liebal, U. W., Phan, A. N. T., Sudhakar, M., Raman, K. & Blank, L. M. Machine Learning Applications for Mass Spectrometry-Based Metabolomics. *Metabolites* **10**, 243 (2020).
141. Gromski, P. S. *et al.* A tutorial review: Metabolomics and partial least squares-discriminant analysis – a marriage of convenience or a shotgun wedding. *Anal. Chim. Acta* **879**, 10–23 (2015).
142. Mendez, K. M., Reinke, S. N. & Broadhurst, D. I. A comparative evaluation of the generalised predictive ability of eight machine learning algorithms across ten clinical metabolomics data sets for binary classification. *Metabolomics* **15**, 150 (2019).
143. Jafari, M. & Ansari-Pour, N. Why, When and How to Adjust Your P Values? *Cell J. Yakhteh* **20**, 604–607 (2019).
144. Korthauer, K. *et al.* A practical guide to methods controlling false discoveries in computational biology. *Genome Biol.* **20**, 118 (2019).
145. Mishra, P. *et al.* Descriptive Statistics and Normality Tests for Statistical Data. *Ann. Card. Anaesth.* **22**, 67–72 (2019).
146. Neuhaus, G. F. *et al.* Environmental metabolomics characterization of modern stromatolites and annotation of ibhayipeptolides. *PLOS ONE* **19**, e0303273 (2024).
147. Bolyen, E. *et al.* Reproducible, interactive, scalable and extensible microbiome data science using QIIME 2. *Nat. Biotechnol.* **37**, 852–857 (2019).
148. Moseley, H. N. B. ERROR ANALYSIS AND PROPAGATION IN METABOLOMICS DATA ANALYSIS. *Comput. Struct. Biotechnol. J.* **4**, e201301006 (2013).
149. Davidson, R. L., Weber, R. J. M., Liu, H., Sharma-Oates, A. & Viant, M. R. Galaxy-M: a Galaxy workflow for processing and analyzing direct infusion and liquid chromatography mass spectrometry-based metabolomics data. *GigaScience* **5**, 10 (2016).
150. Giacomoni, F. *et al.* Workflow4Metabolomics: a collaborative research infrastructure for computational metabolomics. *Bioinformatics* **31**, 1493–1495 (2015).

151. Kontou, E. E. *et al.* UmetaFlow: an untargeted metabolomics workflow for high-throughput data processing and analysis. *J. Cheminformatics* **15**, 52 (2023).
152. Chong, J. & Xia, J. MetaboAnalystR: an R package for flexible and reproducible analysis of metabolomics data. *Bioinformatics* **34**, 4313–4314 (2018).
153. Pang, Z. & Xia, J. LC-MS/MS Raw Spectral Data Processing. https://www.metaboanalyst.ca/resources/vignettes/LCMSMS_Raw_Spectral_Processing.html.
154. Tiffany, C. R. & Bäumler, A. J. omu, a Metabolomics Count Data Analysis Tool for Intuitive Figures and Convenient Metadata Collection. *Microbiol. Resour. Announc.* **8**, e00129-19 (2019).
155. Han, X. & Liang, L. metabolomicsR: a streamlined workflow to analyze metabolomic data in R. *Bioinforma. Adv.* **2**, vbac067 (2022).
156. Fernández-Albert, F., Llorach, R., Andrés-Lacueva, C. & Perera, A. An R package to analyse LC/MS metabolomic data: MAIT (Metabolite Automatic Identification Toolkit). *Bioinformatics* **30**, 1937–1939 (2014).
157. Thévenot, E. A., Roux, A., Xu, Y., Ezan, E. & Junot, C. Analysis of the Human Adult Urinary Metabolome Variations with Age, Body Mass Index, and Gender by Implementing a Comprehensive Workflow for Univariate and OPLS Statistical Analyses. *J. Proteome Res.* **14**, 3322–3335 (2015).
158. Kohler, D. *et al.* MSstats Version 4.0: Statistical Analyses of Quantitative Mass Spectrometry-Based Proteomic Experiments with Chromatography-Based Quantification at Scale. *J. Proteome Res.* **22**, 1466–1482 (2023).
159. Riquelme, G., Zabalegui, N., Marchi, P., Jones, C. M. & Monge, M. E. A Python-Based Pipeline for Preprocessing LC–MS Data for Untargeted Metabolomics Workflows. *Metabolites* **10**, 416 (2020).
160. Di Guida, R. *et al.* Non-targeted UHPLC-MS metabolomic data processing methods: a comparative investigation of normalisation, missing value imputation, transformation and scaling. *Metabolomics* **12**, 93 (2016).
161. Wilkinson, M. D. *et al.* The FAIR Guiding Principles for scientific data management and stewardship. *Sci. Data* **3**, 160018 (2016).
162. Ivanisevic, J. & Want, E. J. From Samples to Insights into Metabolism: Uncovering Biologically Relevant Information in LC-HRMS Metabolomics Data. *Metabolites* **9**, 308 (2019).
163. Dunn, W. B. *et al.* Procedures for large-scale metabolic profiling of serum and plasma using gas chromatography and liquid chromatography coupled to mass spectrometry. *Nat. Protoc.* **6**, 1060–1083 (2011).
164. Silva, A. M., Cordeiro-da-Silva, A. & Coombs, G. H. Metabolic Variation during Development in Culture of Leishmania donovani Promastigotes. *PLoS Negl. Trop. Dis.* **5**, e1451 (2011).
165. Martínez-Sena, T. *et al.* Monitoring of system conditioning after blank injections in untargeted UPLC-MS metabolomic analysis. *Sci. Rep.* **9**, 9822 (2019).
166. Raynie, D. The Vital Role of Blanks in Sample Preparation. *LCGC N. Am.* **36**, 494–497 (2018).
167. Liu, Q. *et al.* Addressing the batch effect issue for LC/MS metabolomics data in data preprocessing. *Sci. Rep.* **10**, 13856 (2020).
168. Yue, Y., Bao, X., Jiang, J. & Li, J. Evaluation and correction of injection order effects in LC-MS/MS based targeted metabolomics. *J. Chromatogr. B* **1212**, 123513 (2022).

169. Livera, A. M. D. *et al.* Statistical Methods for Handling Unwanted Variation in Metabolomics Data. *Anal. Chem.* **87**, 3606–3615 (2015).
170. Broadhurst, D. *et al.* Guidelines and considerations for the use of system suitability and quality control samples in mass spectrometry assays applied in untargeted clinical metabolomic studies. *Metabolomics* **14**, 72 (2018).
171. Lawson, T. N. *et al.* msPurity: Automated Evaluation of Precursor Ion Purity for Mass Spectrometry-Based Fragmentation in Metabolomics. *Anal. Chem.* **89**, 2432–2439 (2017).
172. Schiffman, C. *et al.* Filtering procedures for untargeted LC-MS metabolomics data. *BMC Bioinformatics* **20**, 334 (2019).
173. Carobene, A., Braga, F., Roraas, T., Sandberg, S. & Bartlett, W. A. A systematic review of data on biological variation for alanine aminotransferase, aspartate aminotransferase and γ-glutamyl transferase. *Clin. Chem. Lab. Med. CCLM* **51**, 1997–2007 (2013).
174. Wei, R. *et al.* Missing Value Imputation Approach for Mass Spectrometry-based Metabolomics Data. *Sci. Rep.* **8**, 663 (2018).
175. Do, K. T. *et al.* Characterization of missing values in untargeted MS-based metabolomics data and evaluation of missing data handling strategies. *Metabolomics* **14**, 128 (2018).
176. Gorrochategui, E., Jaumot, J., Lacorte, S. & Tauler, R. Data analysis strategies for targeted and untargeted LC-MS metabolomic studies: Overview and workflow. *TrAC Trends Anal. Chem.* **82**, 425–442 (2016).
177. Li, B. *et al.* Performance Evaluation and Online Realization of Data-driven Normalization Methods Used in LC/MS based Untargeted Metabolomics Analysis. *Sci. Rep.* **6**, 38881 (2016).
178. Scholz, M., Gatzek, S., Sterling, A., Fiehn, O. & Selbig, J. Metabolite fingerprinting: detecting biological features by independent component analysis. *Bioinformatics* **20**, 2447–2454 (2004).
179. Deininger, S.-O. *et al.* Normalization in MALDI-TOF imaging datasets of proteins: practical considerations. *Anal. Bioanal. Chem.* **401**, 167–181 (2011).
180. 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.* **78**, 4281–4290 (2006).
181. Qannari, E. M., Wakeling, I., Courcoux, P. & MacFie, H. J. H. Defining the underlying sensory dimensions. *Food Qual. Prefer.* **11**, 151–154 (2000).
182. van den Berg, R. A., Hoefsloot, H. C., Westerhuis, J. A., Smilde, A. K. & van der Werf, M. J. Centering, scaling, and transformations: improving the biological information content of metabolomics data. *BMC Genomics* **7**, 142 (2006).
183. Khalheim, O. M. Scaling of analytical data. *Anal. Chim. Acta* **177**, 71–79 (1985).
184. Kasprzak, E. M. & Lewis, K. E. Pareto analysis in multiobjective optimization using the collinearity theorem and scaling method. *Struct. Multidiscip. Optim.* **22**, 208–218 (2001).
185. Keenan, M. R. & Kotula, P. G. Accounting for Poisson noise in the multivariate analysis of ToF-SIMS spectrum images. *Surf. Interface Anal.* **36**, 203–212 (2004).
186. Jäggi, C., Wirth, T. & Baur, B. Genetic variability in subpopulations of the asp viper (Vipera aspis) in the Swiss Jura mountains: implications for a conservation strategy. *Biol. Conserv.* **94**, 69–77 (2000).
187. Pinheiro, H. P., de Souza Pinheiro, A. & Sen, P. K. Comparison of genomic sequences using the Hamming distance. *J. Stat. Plan. Inference* **130**, 325–339 (2005).

188. Lozupone, C. & Knight, R. UniFrac: a New Phylogenetic Method for Comparing Microbial Communities. *Appl. Environ. Microbiol.* **71**, 8228–8235 (2005).
189. Brejnrod, A. *et al.* Implementations of the chemical structural and compositional similarity metric in R and Python. 546150 Preprint at https://doi.org/10.1101/546150 (2019).
190. Tripathi, A. *et al.* Chemically informed analyses of metabolomics mass spectrometry data with Qemistree. *Nat. Chem. Biol.* **17**, 146–151 (2021).
191. Ramette, A. Multivariate analyses in microbial ecology. *FEMS Microbiol. Ecol.* **62**, 142–160 (2007).
192. Koenig, J. E. *et al.* Succession of microbial consortia in the developing infant gut microbiome. *Proc. Natl. Acad. Sci.* **108**, 4578–4585 (2011).
193. Anderson, M. J. & Walsh, D. C. I. PERMANOVA, ANOSIM, and the Mantel test in the face of heterogeneous dispersions: What null hypothesis are you testing? *Ecol. Monogr.* **83**, 557–574 (2013).
194. Breiman, L. Random Forests. *Mach. Learn.* **45**, 5–32 (2001).
195. Archer, F. I., Martien, K. K. & Taylor, B. L. Diagnosability of mt DNA with Random Forests: Using sequence data to delimit subspecies. *Mar. Mammal Sci.* **33**, 101–131 (2017).
196. Breiman, L. (out-of-bag estimates). (1996).
197. Liaw, A. & Wiener, M. Classification and Regression by randomForest. **2**, (2002).
198. Griffiths, E. T. *et al.* Detection and classification of narrow-band high frequency echolocation clicks from drifting recorders. *J. Acoust. Soc. Am.* **147**, 3511–3522 (2020).
199. Liu, S. *et al.* Comammox biogeography subject to anthropogenic interferences along a high-altitude river. *Water Res.* **226**, 119225 (2022).
200. Strobl, C., Boulesteix, A.-L., Kneib, T., Augustin, T. & Zeileis, A. Conditional variable importance for random forests. *BMC Bioinformatics* **9**, 307 (2008).
201. Archer, K. J. & Kimes, R. V. Empirical characterization of random forest variable importance measures. *Comput. Stat. Data Anal.* **52**, 2249–2260 (2008).
202. Vinaixa, M. *et al.* A Guideline to Univariate Statistical Analysis for LC/MS-Based Untargeted Metabolomics-Derived Data. *Metabolites* **2**, 775–795 (2012).
203. Riffenburgh, R. H. & Gillen, D. L. *Statistics in Medicine*. (Academic Press, 2020).
204. Sato, T. Type I and Type II Error in Multiple Comparisons. *J. Psychol.* **130**, 293–302 (1996).
205. Bathke, A. The ANOVA F test can still be used in some balanced designs with unequal variances and nonnormal data. *J. Stat. Plan. Inference* **126**, 413–422 (2004).
206. Abdi, H. & Williams, L. Newman-Keuls Test and Tukey Test. *Encycl. Res. Des.* (2010).
207. Ostertagová, E., Ostertag, O. & Kováč, J. Methodology and Application of the Kruskal-Wallis Test. *Appl. Mech. Mater.* **611**, 115–120 (2014).
208. Hecke, T. V. Power study of anova versus Kruskal-Wallis test. *J. Stat. Manag. Syst.* **15**, 241–247 (2012).
209. Dinno, A. Nonparametric Pairwise Multiple Comparisons in Independent Groups using Dunn's Test. *Stata J. Promot. Commun. Stat. Stata* **15**, 292–300 (2015).
210. Hoffmann, M. A. *et al.* High-confidence structural annotation of metabolites absent from spectral libraries. *Nat. Biotechnol.* **40**, 411–421 (2022).
211. Rinker, T. *et al.* pacman: Package Management Tool. (2019).
212. Wickham, H. *et al.* Welcome to the Tidyverse. *J. Open Source Softw.* **4**, 1686 (2019).

213. Kluyver, T., Angerer, P. & Schulz, J. IRdisplay: 'Jupyter' Display Machinery. (2022).
214. Cacciatore, S., Luchinat, C. & Tenori, L. Knowledge discovery by accuracy maximization. *Proc. Natl. Acad. Sci.* **111**, 5117–5122 (2014).
215. Kassambara, A. & Mundt, F. Extract and Visualize the Results of Multivariate Data Analyses [R package factoextra version 1.0.7]. in (2020).
216. Oksanen, J. *et al.* vegan: Community Ecology Package. (2022).
217. Gu, Z. Complex heatmap visualization. *iMeta* **1**, e43 (2022).
218. Galili, T. dendextend: an R package for visualizing, adjusting and comparing trees of hierarchical clustering. *Bioinforma. Oxf. Engl.* **31**, 3718–3720 (2015).
219. Charrad, M., Ghazzali, N., Boiteau, V. & Niknafs, A. NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set. *J. Stat. Softw.* **61**, 1–36 (2014).
220. Archer, E. rfPermute: Estimate Permutation p-Values for Random Forest Importance Metrics. (2023).
221. Ogle, D. H., Doll, J. C., Wheeler, A. P. & dunnTest()), A. D. (Provided base functionality of. FSA: Simple Fisheries Stock Assessment Methods. (2023).
222. Bengtsson, H. *et al.* matrixStats: Functions that Apply to Rows and Columns of Matrices (and to Vectors). (2023).
223. Xiao [aut, N., cre, Cook, J., Jégousse, C. & Li, M. ggsci: Scientific Journal and Sci-Fi Themed Color Palettes for 'ggplot2'. (2023).
224. Wilke, C. O. cowplot: Streamlined Plot Theme and Plot Annotations for 'ggplot2'. (2020).
225. Wickham, H. *et al.* svglite: An 'SVG' Graphics Device. (2023).
226. Reese, S. E. *et al.* A new statistic for identifying batch effects in high-throughput genomic data that uses guided principal component analysis. *Bioinformatics* **29**, 2877–2883 (2013).
227. Burton, L. *et al.* Instrumental and experimental effects in LC–MS-based metabolomics. *J. Chromatogr. B* **871**, 227–235 (2008).
228. Gregori, J. *et al.* Batch effects correction improves the sensitivity of significance tests in spectral counting-based comparative discovery proteomics. *J. Proteomics* **75**, 3938–3951 (2012).
229. Thonusin, C. *et al.* Evaluation of intensity drift correction strategies using MetaboDrift, a normalization tool for multi-batch metabolomics data. *J. Chromatogr. A* **1523**, 265–274 (2017).
230. Johnson, W. E., Li, C. & Rabinovic, A. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics* **8**, 118–127 (2007).
231. Deng, K. *et al.* WaveICA: A novel algorithm to remove batch effects for large-scale untargeted metabolomics data based on wavelet analysis. *Anal. Chim. Acta* **1061**, 60–69 (2019).
232. Wehrens, R. *et al.* Improved batch correction in untargeted MS-based metabolomics. *Metabolomics* **12**, 88 (2016).
233. Kuligowski, J., Sánchez-Illana, Á., Sanjuán-Herráez, D., Vento, M. & Quintás, G. Intra-batch effect correction in liquid chromatography-mass spectrometry using quality control samples and support vector regression (QC-SVRC). *The Analyst* **140**, 7810–7817 (2015).
234. Luan, H., Ji, F., Chen, Y. & Cai, Z. statTarget: A streamlined tool for signal drift correction and interpretations of quantitative mass spectrometry-based omics data. *Anal. Chim. Acta* **1036**, 66–72 (2018).
235. Rong, Z. *et al.* NormAE: Deep Adversarial Learning Model to Remove Batch Effects in Liquid Chromatography Mass Spectrometry-Based Metabolomics Data. *Anal. Chem.* **92**, 5082–5090 (2020).

236. Dmitrenko, A., Reid, M. & Zamboni, N. Regularized adversarial learning for normalization of multi-batch untargeted metabolomics data. *Bioinformatics* **39**, btad096 (2023).
237. Tokareva, A. O. *et al.* Normalization methods for reducing interbatch effect without quality control samples in liquid chromatography-mass spectrometry-based studies. *Anal. Bioanal. Chem.* **413**, 3479–3486 (2021).
238. Cleary, J. L., Luu, G. T., Pierce, E. C., Dutton, R. J. & Sanchez, L. M. BLANKA: an Algorithm for Blank Subtraction in Mass Spectrometry of Complex Biological Samples. *J. Am. Soc. Mass Spectrom.* **30**, 1426–1434 (2019).
239. Wulff, J. E. & Mitchell, M. W. A Comparison of Various Normalization Methods for LC/MS Metabolomics Data. *Adv. Biosci. Biotechnol.* **9**, 339–351 (2018).
240. Morgan, M. & Ramos, M. BiocManager: Access the Bioconductor Project Package Repository. (2023).
241. Wilkinson, L. & Friendly, M. The History of the Cluster Heat Map. *Am. Stat.* **63**, 179–184 (2009).
242. Wu, W. & Noble, W. S. Genomic data visualization on the Web. *Bioinformatics* **20**, 1804–1805 (2004).
243. Xia, Y. & Sun, J. Hypothesis Testing and Statistical Analysis of Microbiome. *Genes Dis.* **4**, 138–148 (2017).
244. Robinson, D. *et al.* broom: Convert Statistical Objects into Tidy Tibbles. (2023).
245. Koh, A., De Vadder, F., Kovatcheva-Datchary, P. & Bäckhed, F. From Dietary Fiber to Host Physiology: Short-Chain Fatty Acids as Key Bacterial Metabolites. *Cell* **165**, 1332–1345 (2016).
246. Sharon, G. *et al.* Specialized Metabolites from the Microbiome in Health and Disease. *Cell Metab.* **20**, 719–730 (2014).
247. Gilbert, J. A. *et al.* Clinical translation of microbiome research. *Nat. Med.* **31**, 1099–1113 (2025).
248. Danielsson, H. & Gustafsson, B. On serum-cholesterol levels and neutral fecal sterols in germ-free rats. Bile acids and steroids 59. *Arch. Biochem. Biophys.* **83**, 482–485 (1959).
249. Sousa, T. *et al.* The gastrointestinal microbiota as a site for the biotransformation of drugs. *Int. J. Pharm.* **363**, 1–25 (2008).
250. Spanogiannopoulos, P., Bess, E. N., Carmody, R. N. & Turnbaugh, P. J. The microbial pharmacists within us: a metagenomic view of xenobiotic metabolism. *Nat. Rev. Microbiol.* **14**, 273–287 (2016).
251. Zimmermann, M., Zimmermann-Kogadeeva, M., Wegmann, R. & Goodman, A. L. Separating host and microbiome contributions to drug pharmacokinetics and toxicity. *Science* **363**, eaat9931 (2019).
252. Pant, A., Maiti, T. K., Mahajan, D. & Das, B. Human Gut Microbiota and Drug Metabolism. *Microb. Ecol.* **86**, 97–111 (2023).
253. Fu, Y. *et al.* Balance between bile acid conjugation and hydrolysis activity can alter outcomes of gut inflammation. *Nat. Commun.* **16**, 3434 (2025).
254. Culp, E. J., Nelson, N. T., Verdegaal, A. A. & Goodman, A. L. Microbial transformation of dietary xenobiotics shapes gut microbiome composition. *Cell* **187**, 6327-6345.e20 (2024).
255. Zeevi, D. *et al.* Personalized Nutrition by Prediction of Glycemic Responses. *Cell* **163**, 1079–1094 (2015).

256. Ben-Yacov, O. *et al.* Gut microbiome modulates the effects of a personalised postprandial-targeting (PPT) diet on cardiometabolic markers: a diet intervention in pre-diabetes. *Gut* **72**, 1486–1496 (2023).
257. Blaser, M. J. Antibiotic use and its consequences for the normal microbiome. *Science* **352**, 544–545 (2016).
258. Dethlefsen, L. & Relman, D. A. Incomplete recovery and individualized responses of the human distal gut microbiota to repeated antibiotic perturbation. *Proc. Natl. Acad. Sci.* **108**, 4554–4561 (2011).
259. Cho, I. *et al.* Antibiotics in early life alter the murine colonic microbiome and adiposity. *Nature* **488**, 621–626 (2012).
260. Maier, L. *et al.* Unravelling the collateral damage of antibiotics on gut bacteria. *Nature* **599**, 120–124 (2021).
261. Uzan-Yulzari, A. *et al.* Neonatal antibiotic exposure impairs child growth during the first six years of life by perturbing intestinal microbial colonization. *Nat. Commun.* **12**, 443 (2021).
262. Nagy, E., Boyanova, L. & Justesen, U. S. How to isolate, identify and determine antimicrobial susceptibility of anaerobic bacteria in routine laboratories. *Clin. Microbiol. Infect.* **24**, 1139–1148 (2018).
263. Maier, L. *et al.* Extensive impact of non-antibiotic drugs on human gut bacteria. *Nature* **555**, 623–628 (2018).
264. Kåhrström, C. T., Pariente, N. & Weiss, U. Intestinal microbiota in health and disease. *Nature* **535**, 47–47 (2016).
265. Falony, G. *et al.* Population-level analysis of gut microbiome variation. *Science* **352**, 560–564 (2016).
266. Nagata, N. *et al.* Population-level Metagenomics Uncovers Distinct Effects of Multiple Medications on the Human Gut Microbiome. *Gastroenterology* **163**, 1038–1052 (2022).
267. Ottman, N. *et al.* Genome-Scale Model and Omics Analysis of Metabolic Capacities of Akkermansia muciniphila Reveal a Preferential Mucin-Degrading Lifestyle. *Appl. Environ. Microbiol.* **83**, e01014-17 (2017).
268. Baldini, F. *et al.* Parkinson's disease-associated alterations of the gut microbiome predict disease-relevant changes in metabolic functions. *BMC Biol.* **18**, 62 (2020).
269. Guo, P., Zhang, K., Ma, X. & He, P. Clostridium species as probiotics: potentials and challenges. *J. Anim. Sci. Biotechnol.* **11**, 24 (2020).
270. Basile, A. *et al.* Longitudinal flux balance analyses of a patient with episodic colonic inflammation reveals microbiome metabolic dynamics. *Gut Microbes* **15**, 2226921 (2023).
271. Kim, J., Jin, Y.-S. & Kim, K. H. L-Fucose is involved in human–gut microbiome interactions. *Appl. Microbiol. Biotechnol.* **107**, 3869–3875 (2023).
272. Wuyts, S. *et al.* Consistency across multi-omics layers in a drug-perturbed gut microbial community. *Mol. Syst. Biol.* **19**, MSB202311525 (2023).
273. Go, D. *et al.* Integration of metabolomics and other omics: from microbes to microbiome. *Appl. Microbiol. Biotechnol.* **108**, 538 (2024).
274. Kim, J. *et al.* Systems Metabolic Engineering to Elucidate and Enhance Intestinal Metabolic Activities of Escherichia coli Nissle 1917. *J. Agric. Food Chem.* **72**, 18234–18246 (2024).

275. Zeng, H. *et al.* Proteomic and metabolomic analyses reveal the antibacterial mechanism of Cannabidiol against gram-positive bacteria. *J. Proteomics* **315**, 105411 (2025).
276. Oberleitner, D., Schmid, R., Schulz, W., Bergmann, A. & Achten, C. Feature-based molecular networking for identification of organic micropollutants including metabolites by non-target analysis applied to riverbank filtration. *Anal. Bioanal. Chem.* **413**, 5291–5300 (2021).
277. Beauxis, Y. & Genta-Jouve, G. MetWork: a web server for natural products anticipation. *Bioinformatics* **35**, 1795–1796 (2019).
278. Vitale, G. A. *et al.* Connecting metabolome and phenotype: recent advances in functional metabolomics tools for the identification of bioactive natural products. *Nat. Prod. Rep.* **41**, 885–904 (2024).
279. Geibel, C. *et al.* High-Frequency Microfluidic Fractionation for Compound-Resolved Bioactivity-Based Metabolomics. *Anal. Chem.* **97**, 24093–24104 (2025).
280. Mülleder, M. *et al.* Functional Metabolomics Describes the Yeast Biosynthetic Regulome. *Cell* **167**, 553-565.e12 (2016).
281. Wang, M. *et al.* Mass Spectrometry Searches using MASST. *Nat. Biotechnol.* **38**, 23–26 (2020).
282. Petras, D. *et al.* Chemical Proportionality within Molecular Networks. *Anal. Chem.* **93**, 12833–12839 (2021).
283. Elias, J. E. & Gygi, S. P. Target-Decoy Search Strategy for Mass Spectrometry-Based Proteomics. *Methods Mol. Biol. Clifton NJ* **604**, 55–71 (2010).
284. Wen, B. *et al.* Assessment of false discovery rate control in tandem mass spectrometry analysis using entrapment. *Nat. Methods* **22**, 1454–1463 (2025).
285. Mannochio-Russo, H. *et al.* Bridging Complexity and Accessibility in Metabolomics with MetaboApps. Preprint at https://doi.org/10.26434/chemrxiv-2025-3nq29 (2025).
286. Müller, P. *et al.* High-throughput anaerobic screening for identifying compounds acting against gut bacteria in monocultures or communities. *Nat. Protoc.* **19**, 668–699 (2024).
287. Kyoung, J., Atluri, R. R. & Yang, T. Resistance to Antihypertensive Drugs: Is Gut Microbiota the Missing Link? *Hypertension* **79**, 2138–2147 (2022).
288. Beckmann, L. *et al.* Telmisartan induces a specific gut microbiota signature which may mediate its antiobesity effect. *Pharmacol. Res.* **170**, 105724 (2021).
289. Krutova, M., Wilcox, M. & Kuijper, E. Clostridioides difficile infection: are the three currently used antibiotic treatment options equal from pharmacological and microbiological points of view? *Int. J. Infect. Dis.* **124**, 118–123 (2022).
290. Ralph, E. D. & Kirby, W. M. M. Unique Bactericidal Action of Metronidazole Against Bacteroides fragilis and Clostridium perfringens. *Antimicrob. Agents Chemother.* **8**, 409–414 (1975).
291. Belstrøm, D. *et al.* Transcriptional Activity of Predominant Streptococcus Species at Multiple Oral Sites Associate With Periodontal Status. *Front. Cell. Infect. Microbiol.* **11**, (2021).
292. Smith, A. Metronidazole resistance: a hidden epidemic? *Br. Dent. J.* **224**, 403–404 (2018).
293. Dollas, M. N. *et al.* High prevalence of antibiotic resistance of Streptococcus species in saliva from non-hospitalized adults – a pilot study. *J. Oral Microbiol.* **17**, 2486647 (2025).
294. Pilla, R. *et al.* Effects of metronidazole on the fecal microbiome and metabolome in healthy dogs. *J. Vet. Intern. Med.* **34**, 1853–1866 (2020).

295. Belchik, S. E., Oba, P. M., Lin, C.-Y. & Swanson, K. S. Effects of a veterinary gastrointestinal diet on fecal characteristics, metabolites, and microbiota concentrations of adult cats treated with metronidazole. *J. Anim. Sci.* **102**, skae274 (2024).
296. Escalante, V. *et al.* Simvastatin induces human gut bacterial cell surface genes. *Mol. Microbiol.* **122**, 372–386 (2024).
297. Mattoli, L. *et al.* Suspect screening analysis to improve untargeted and targeted UHPLC-qToF approaches: the biodegradability of a proton pump inhibitor medicine and a natural medical device. *Sci. Rep.* **14**, 51 (2024).
298. Wang, L. *et al.* A Mechanism Study on the (+)-ESI-TOF/HRMS Fragmentation of Some PPI Prazoles and Their Related Substances. *Molecules* **28**, (2023).
299. Fujii, S., Kameyama, K., Hosono, M., Hayashi, Y. & Kitamura, K. Effect of cilnidipine, a novel dihydropyridine Ca++-channel antagonist, on N-type Ca++ channel in rat dorsal root ganglion neurons. *J. Pharmacol. Exp. Ther.* **280**, 1184–1191 (1997).
300. Das, A. *et al.* Effects of Cilnidipine on Heart Rate and Uric Acid Metabolism in Patients With Essential Hypertension. *Cardiol. Res.* **7**, 167–172 (2016).
301. Liu, X.-Q., Zhao, Y., Li, D., Qian, Z.-Y. & Wang, G.-J. Metabolism and metabolic inhibition of cilnidipine in human liver microsomes. *Acta Pharmacol. Sin.* **24**, 263–268 (2003).
302. Bittremieux, W. *et al.* Universal MS/MS Visualization and Retrieval with the Metabolomics Spectrum Resolver Web Service. 2020.05.09.086066 Preprint at https://doi.org/10.1101/2020.05.09.086066 (2020).
303. Shahneh, M. R. Z. *et al.* ModiFinder: Tandem Mass Spectral Alignment Enables Structural Modification Site Localization. *J. Am. Soc. Mass Spectrom.* **35**, 2564–2578 (2024).
304. Roldán, M. D., Pérez-Reinado, E., Castillo, F. & Moreno-Vivián, C. Reduction of polynitroaromatic compounds: the bacterial nitroreductases. *FEMS Microbiol. Rev.* **32**, 474–500 (2008).
305. Warrier, T. *et al.* N-methylation of a bactericidal compound as a resistance mechanism in Mycobacterium tuberculosis. *Proc. Natl. Acad. Sci.* **113**, E4523–E4530 (2016).
306. Haiser, H. J. & Turnbaugh, P. J. Developing a metagenomic view of xenobiotic metabolism. *Pharmacol. Res.* **69**, 21–31 (2013).
307. Lee, H.-W. *et al.* Development of a liquid chromatography/negative-ion electrospray tandem mass spectrometry assay for the determination of cilnidipine in human plasma and its application to a bioequivalence study. *J. Chromatogr. B* **862**, 246–251 (2008).
308. Zemanová, N. *et al.* Gut microbiome affects the metabolism of metronidazole in mice through regulation of hepatic cytochromes P450 expression. *PLOS ONE* **16**, e0259643 (2021).
309. Pearce, R. E., Cohen-Wolkowiez, M., Sampson, M. R. & Kearns, G. L. The Role of Human Cytochrome P450 Enzymes in the Formation of 2-Hydroxymetronidazole: CYP2A6 is the High Affinity (Low Km) Catalyst. *Drug Metab. Dispos.* **41**, 1686–1694 (2013).
310. Gupta, R. *et al.* Functionalized Nitroimidazole Scaffold Construction and Their Pharmaceutical Applications: A 1950–2021 Comprehensive Overview. *Pharmaceuticals* **15**, (2022).
311. Zhang, Q., Zhou, H., Zhai, S. & Yan, B. Natural product-inspired synthesis of thiazolidine and thiazolidinone compounds and their anticancer activities. *Curr. Pharm. Des.* **16**, 1826–1842 (2010).

312. Weidolf, L. & Castagnoli Jr, N. Study of the electrospray ionization mass spectrometry of the proton pump inhibiting drug Omeprazole. *Rapid Commun. Mass Spectrom.* **15**, 283–290 (2001).
313. Shankar, G. *et al.* Identification and structural characterization of the stress degradation products of omeprazole using Q-TOF-LC-ESI-MS/MS and NMR experiments: evaluation of the toxicity of the degradation products. *New J. Chem.* **43**, 7294–7306 (2019).
314. Watanabe, K., Yamashita, S., Furuno, K., Kawasaki, H. & Gomita, Y. Metabolism of omeprazole by gut flora in rats. *J. Pharm. Sci.* **84**, 516–517 (1995).
315. Dial, S., Delaney, J. A. C., Barkun, A. N. & Suissa, S. Use of Gastric Acid–Suppressive Agents and the Risk of Community-Acquired Clostridium difficile–Associated Disease. *JAMA* **294**, 2989–2995 (2005).
316. Seto, C. T., Jeraldo, P., Orenstein, R., Chia, N. & DiBaise, J. K. Prolonged use of a proton pump inhibitor reduces microbial diversity: implications for Clostridium difficile susceptibility. *Microbiome* **2**, 42 (2014).
317. Kostrzewska, M. *et al.* The effect of omeprazole treatment on the gut microflora and neutrophil function. *Clin. Res. Hepatol. Gastroenterol.* **41**, 575–584 (2017).
318. Prueksaritanont, T. *et al.* In vitro metabolism of simvastatin in humans [SBT]identification of metabolizing enzymes and effect of the drug on hepatic P450s. *Drug Metab. Dispos. Biol. Fate Chem.* **25**, 1191–1199 (1997).
319. Zhou, G., Ewald, J. & Xia, J. OmicsAnalyst: a comprehensive web-based platform for visual analytics of multi-omics data. *Nucleic Acids Res.* **49**, W476–W482 (2021).
320. Rantalainen, M. *et al.* Statistically Integrated Metabonomic−Proteomic Studies on a Human Prostate Cancer Xenograft Model in Mice. *J. Proteome Res.* **5**, 2642–2655 (2006).
321. Pazoki, R. *et al.* PATHWAYS UNDERLYING URINARY SODIUM AND POTASSIUM EXCRETION AND THE LINK TO BLOOD PRESSURE AND CARDIOVASCULAR DISEASE. *J. Hypertens.* **37**, e74 (2019).
322. Gurke, R. *et al.* Omics and Multi-Omics Analysis for the Early Identification and Improved Outcome of Patients with Psoriatic Arthritis. *Biomedicines* **10**, (2022).
323. Integrative Analysis of Multi-omics Data for Discovery and Functional Studies of Complex Human Diseases. in *Advances in Genetics* vol. 93 147–190 (Academic Press, 2016).
324. Dugourd, A. & Saez-Rodriguez, J. Footprint-based functional analysis of multiomic data. *Curr. Opin. Syst. Biol.* **15**, 82–90 (2019).
325. Morabito, A., De Simone, G., Pastorelli, R., Brunelli, L. & Ferrario, M. Algorithms and tools for data-driven omics integration to achieve multilayer biological insights: a narrative review. *J. Transl. Med.* **23**, 425 (2025).
326. Lê Cao, K.-A., González, I. & Déjean, S. integrOmics: an R package to unravel relationships between two omics datasets. *Bioinformatics* **25**, 2855–2856 (2009).
327. Li, W., Zhang, S., Liu, C.-C. & Zhou, X. J. Identifying multi-layer gene regulatory modules from multi-dimensional genomic data. *Bioinformatics* **28**, 2458–2466 (2012).
328. Zheng, W. *et al.* A Multi-Omics Study of Human Testis and Epididymis. *Molecules* **26**, (2021).
329. Gao, Y.-N. *et al.* Multi-Omics Reveal Additive Cytotoxicity Effects of Aflatoxin B1 and Aflatoxin M1 toward Intestinal NCM460 Cells. *Toxins* **14**, (2022).

330. Yang, F. *et al.* Quantitative proteomics and multi-omics analysis identifies potential biomarkers and the underlying pathological molecular networks in Chinese patients with multiple sclerosis. *BMC Neurol.* **24**, 423 (2024).
331. Wang, Z. *et al.* Multi-platform omics sequencing dissects the atlas of plasma-derived exosomes in rats with or without depression-like behavior after traumatic spinal cord injury. *Prog. Neuropsychopharmacol. Biol. Psychiatry* **132**, 110987 (2024).
332. Ramiro, L. *et al.* Integrative Multi-omics Analysis to Characterize Human Brain Ischemia. *Mol. Neurobiol.* **58**, 4107–4121 (2021).
333. Langfelder, P. & Horvath, S. WGCNA: an R package for weighted correlation network analysis. *BMC Bioinformatics* **9**, 559 (2008).
334. Uppal, K., Ma, C., Go, Y.-M. & Jones, D. P. xMWAS: a data-driven integration and differential network analysis tool. *Bioinformatics* **34**, 701–702 (2018).
335. McKinney, W. Data Structures for Statistical Computing in Python. in 56–61 (Austin, Texas, 2010). doi:10.25080/Majora-92bf1922-00a.
336. Oliphant, T. *Guide to NumPy*. (2006).
337. Virtanen, P. *et al.* SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods* **17**, 261–272 (2020).
338. Pakkir Shah, A. K. *et al.* Statistical analysis of feature-based molecular networking results from non-targeted metabolomics data. *Nat. Protoc.* **20**, 92–162 (2025).
339. Heuckeroth, S. *et al.* Reproducible mass spectrometry data processing and compound annotation in MZmine 3. *Nat. Protoc.* 1–45 (2024) doi:10.1038/s41596-024-00996-y.
340. Hagberg, A. A., Schult, D. A. & Swart, P. J. Exploring Network Structure, Dynamics, and Function using NetworkX. in 11–15 (Pasadena, California, 2008). doi:10.25080/TCWV9851.
341. Shannon, P. *et al.* Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks. *Genome Res.* **13**, 2498–2504 (2003).
342. Yu, J. S. *et al.* A versatile toolkit for drug metabolism studies with GNPS2: from drug development to clinical monitoring. *Nat. Protoc.* 1–35 (2025) doi:10.1038/s41596-025-01237-6.
343. Höhn, F. *et al.* Strong pairwise interactions do not drive interactions in a plant leaf associated microbial community. *ISME Commun.* **5**, ycae117 (2025).
344. McMurdie, P. J. Normalization of Microbiome Profiling Data. in *Microbiome Analysis: Methods and Protocols* (eds Beiko, R. G., Hsiao, W. & Parkinson, J.) 143–168 (Springer, New York, NY, 2018). doi:10.1007/978-1-4939-8728-3_10.
345. Austin, G. I. & Korem, T. Compositional transformations can reasonably introduce phenotype-associated values into sparse features. *mSystems* **10**, e00021-25 (2025).
346. Klindworth, A. *et al.* Evaluation of general 16S ribosomal RNA gene PCR primers for classical and next-generation sequencing-based diversity studies. *Nucleic Acids Res.* **41**, e1 (2013).
347. Tremblay, J. *et al.* Primer and platform effects on 16S rRNA tag sequencing. *Front. Microbiol.* **6**, (2015).
348. Lao, H.-Y. *et al.* The clinical utility of Nanopore 16S rRNA gene sequencing for direct bacterial identification in normally sterile body fluids. *Front. Microbiol.* **14**, (2024).
349. Chen, Z. *et al.* Biases from Oxford Nanopore library preparation kits and their effects on microbiome and genome analysis. *BMC Genomics* **26**, 504 (2025).

350. Bejaoui, S. *et al.* Comparison of Illumina and Oxford Nanopore sequencing data quality for Clostridioides difficile genome analysis and their application for epidemiological surveillance. *BMC Genomics* **26**, 92 (2025).
351. Macip, G. *et al.* Comparative analysis of illumina and oxford nanopore sequencing platforms for 16S rRNA profiling of respiratory microbial communities. *Sci. Rep.* **15**, 33688 (2025).
352. Taylor, P. J. Matrix effects: the Achilles heel of quantitative high-performance liquid chromatography–electrospray–tandem mass spectrometry. *Clin. Biochem.* **38**, 328–334 (2005).
353. Nasiri, A. *et al.* Overview, consequences, and strategies for overcoming matrix effects in LC-MS analysis: a critical review. *Analyst* **146**, 6049–6063 (2021).
354. Gloor, G. B., Macklaim, J. M., Pawlowsky-Glahn, V. & Egozcue, J. J. Microbiome Datasets Are Compositional: And This Is Not Optional. *Front. Microbiol.* **8**, (2017).
355. Bijlsma, S. *et al.* Large-scale human metabolomics studies: a strategy for data (pre-) processing and validation. *Anal. Chem.* **78**, 567–574 (2006).
356. Weiss, S. *et al.* Normalization and microbial differential abundance strategies depend upon data characteristics. *Microbiome* **5**, 27 (2017).
357. Quinn, T. P. *et al.* A field guide for the compositional analysis of any-omics data. *GigaScience* **8**, giz107 (2019).
358. Hauke, J. & Kossowski, T. Comparison of Values of Pearson's and Spearman's Correlation Coefficients on the Same Sets of Data. *Quaest. Geogr.* **30**, 87–93 (2011).
359. Love, M. I., Huber, W. & Anders, S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol.* **15**, 550 (2014).
360. Law, C. W., Chen, Y., Shi, W. & Smyth, G. K. voom: precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biol.* **15**, R29 (2014).
361. Cox, J. & Mann, M. MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification. *Nat. Biotechnol.* **26**, 1367–1372 (2008).
362. Choi, M. *et al.* MSstats: an R package for statistical analysis of quantitative mass spectrometry-based proteomic experiments. *Bioinformatics* **30**, 2524–2526 (2014).
363. Griffiths, R. I., Whiteley, A. S., O'Donnell, A. G. & Bailey, M. J. Rapid Method for Coextraction of DNA and RNA from Natural Environments for Analysis of Ribosomal DNA- and rRNA-Based Microbial Community Composition. *Appl. Environ. Microbiol.* **66**, 5488–5491 (2000).
364. Lebuhn, M. *et al.* Towards molecular biomarkers for biogas production from lignocellulose-rich substrates. *Anaerobe* **29**, 10–21 (2014).
365. Kim, K. S., Noh, J., Kim, B.-S., Koh, H. & Lee, D.-W. Refining microbiome diversity analysis by concatenating and integrating dual 16S rRNA amplicon reads. *Npj Biofilms Microbiomes* **11**, 57 (2025).
366. Seol, D. *et al.* Microbial Identification Using rRNA Operon Region: Database and Tool for Metataxonomics with Long-Read Sequence. *Microbiol. Spectr.* **10**, e02017-21.
367. Curry, K. D. *et al.* Emu: species-level microbial community profiling of full-length 16S rRNA Oxford Nanopore sequencing data. *Nat. Methods* **19**, 845–853 (2022).
368. Phelan, V. V. Feature-Based Molecular Networking for Metabolite Annotation. in *Computational Methods and Data Analysis for Metabolomics* (ed. Li, S.) 227–243 (Springer US, New York, NY, 2020). doi:10.1007/978-1-0716-0239-3_13.

369. Camacho, D., de la Fuente, A. & Mendes, P. The origin of correlations in metabolomics data. *Metabolomics* **1**, 53–63 (2005).
370. Jahagirdar, S., Saccenti, E., Jahagirdar, S. & Saccenti, E. On the Use of Correlation and MI as a Measure of Metabolite—Metabolite Association for Network Differential Connectivity Analysis. *Metabolites* **10**, (2020).
371. Iliakopoulou, S. *et al.* Elucidating Transformation Pathways of Microcystins during Advanced Oxidation/Reduction Processes for Water Treatment. Preprint at https://doi.org/10.26434/chemrxiv-2025-q4xh3 (2025).
372. Pérez-Lorente, A. I. *et al.* Deciphering the chemical dialogue between Bacillus and pathogenic fungi. https://agris.fao.org/search/en/providers/125074/records/67488fa77625988a371d8acb (2024).
373. Huber, F. *et al.* Spec2Vec: Improved mass spectral similarity scoring through learning of structural relationships. *PLOS Comput. Biol.* **17**, e1008724 (2021).
374. Huber, F., van der Burg, S., van der Hooft, J. J. J. & Ridder, L. MS2DeepScore: a novel deep learning similarity measure to compare tandem mass spectra. *J. Cheminformatics* **13**, 84 (2021).
375. de Jonge, N. F. *et al.* MS2Query: reliable and scalable MS2 mass spectra-based analogue search. *Nat. Commun.* **14**, 1752 (2023).
376. Liu, Y. *et al.* MESSAR: Automated recommendation of metabolite substructures from tandem mass spectra. *PLOS ONE* **15**, e0226770 (2020).
377. Ji, H., Xu, Y., Lu, H. & Zhang, Z. Deep MS/MS-Aided Structural-Similarity Scoring for Unknown Metabolite Identification. *Anal. Chem.* **91**, 5629–5637 (2019).
378. Ruttkies, C., Schymanski, E. L., Wolf, S., Hollender, J. & Neumann, S. MetFrag relaunched: incorporating strategies beyond in silico fragmentation. *J. Cheminformatics* **8**, 3 (2016).
379. Allen, F., Pon, A., Wilson, M., Greiner, R. & Wishart, D. CFM-ID: a web server for annotation, spectrum prediction and metabolite identification from tandem mass spectra. *Nucleic Acids Res.* **42**, W94–W99 (2014).
380. Gorges, J. & Grimme, S. QCxMS2 – a program for the calculation of electron ionization mass spectra via automated reaction network discovery. *Phys. Chem. Chem. Phys.* **27**, 6899–6911 (2025).
381. Wang, R. *et al.* Neural Spectral Prediction for Structure Elucidation with Tandem Mass Spectrometry. *bioRxiv* 2025.05.28.656653 (2025) doi:10.1101/2025.05.28.656653.
382. Mannochio-Russo, H. *et al.* The microbiome diversifies long- to short-chain fatty acid-derived N-acyl lipids. *Cell* **188**, 4154-4169.e19 (2025).
383. Friedman, J. & Alm, E. J. Inferring Correlation Networks from Genomic Survey Data. *PLOS Comput. Biol.* **8**, e1002687 (2012).
384. Fang, H., Huang, C., Zhao, H. & Deng, M. CCLasso: correlation inference for compositional data through Lasso. *Bioinformatics* **31**, 3172–3180 (2015).
385. Muller, E., Algavi, Y. M. & Borenstein, E. A meta-analysis study of the robustness and universality of gut microbiome-metabolome associations. *Microbiome* **9**, 203 (2021).
386. Jeske, J. T., Gallert, C., Jeske, J. T. & Gallert, C. Microbiome Analysis via OTU and ASV-Based Pipelines—A Comparative Interpretation of Ecological Data in WWTP Systems. *Bioengineering* **9**, (2022).

387. van Dis, E. A. M., Bollen, J., Zuidema, W., van Rooij, R. & Bockting, C. L. ChatGPT: five priorities for research. *Nature* **614**, 224–226 (2023).
388. Biswas, A., Kumari, A., Gaikwad, D. S. & Pandey, D. K. Revolutionizing Biological Science: The Synergy of Genomics in Health, Bioinformatics, Agriculture, and Artificial Intelligence. *OMICS J. Integr. Biol.* **27**, 550–569 (2023).
389. Li, Q. *et al.* Progress and opportunities of foundation models in bioinformatics. *Brief. Bioinform.* **25**, bbae548 (2024).
390. Vidanagamachchi, S. M. & Waidyarathna, K. M. G. T. R. Opportunities, challenges and future perspectives of using bioinformatics and artificial intelligence techniques on tropical disease identification using omics data. *Front. Digit. Health* **6**, (2024).
391. Lin, A. *et al.* Bridging artificial intelligence and biological sciences: a comprehensive review of large language models in bioinformatics. *Brief. Bioinform.* **26**, bbaf357 (2025).
392. Chopra, B. *et al.* Challenges in Using Conversational AI for Data Science. in *Proceedings of the Workshop on Human-In-the-Loop Data Analytics* 1–7 (Association for Computing Machinery, New York, NY, USA, 2025). doi:10.1145/3736733.3736748.
393. Stincone, P. *et al.* Evaluation of Data-Dependent MS/MS Acquisition Parameters for Non-Targeted Metabolomics and Molecular Networking of Environmental Samples: Focus on the Q Exactive Platform. *Anal. Chem.* **95**, 12673–12682 (2023).
394. Myers, O. D., Sumner, S. J., Li, S., Barnes, S. & Du, X. One Step Forward for Reducing False Positive and False Negative Compound Identifications from Mass Spectrometry Metabolomics Data: New Algorithms for Constructing Extracted Ion Chromatograms and Detecting Chromatographic Peaks. *Anal. Chem.* **89**, 8696–8703 (2017).
395. Mohimani, H. *et al.* Dereplication of peptidic natural products through database search of mass spectra. *Nat. Chem. Biol.* **13**, 30–37 (2017).
396. Brunet, S. *et al.* Nationwide multicentre study of Nanopore long-read sequencing for 16S rRNA-species identification. *Eur. J. Clin. Microbiol. Infect. Dis.* **44**, 1907–1916 (2025).

References

Appendix

Chapter 2: Supplementary Information

Supporting Methods

Data Dependent Acquisition of Example LC-MS/MS Data

The LC-MS/MS analysis was performed on a vanquish UHPLC system coupled to a Q-Exactive quadrupole orbitrap mass spectrometer (Thermo Fisher Scientific, Bremen, Germany) as previously described [32,393]. Briefly, all seawater extracts with PPL solid phase extraction (SPE) were reconstituted with 100 µL methanol/water/formic acid (80/19/1). Then, ten microliters of the extract samples were analyzed in LC-MS/MS system. For chromatographic conditions, reversed-phase C18 porous core column (Kinetex C18, 150 x 2 mm, 1.7 µm particle size, 100 Å pore size, Phenomenex, Torrance, USA) was used for separation. The mobile phase consisted of solvent A H_2O + 0.1 % formic acid (FA) and solvent B acetonitrile (ACN) + 0.1 % FA. The flow rate was set to 0.5 mL/min. The LC separation gradient started from 5% of solvent B at 0-0.5 min, then, changed %B to 59% within 7.5 min and increase to 99% within 2 min. Solvent B was hold at 99% for 2 min in washing step and switched to 5% for re-equilibration column prior to next injection. According to mass spectrometry parameters setting, MS/MS spectra was acquired in data dependent acquisition (DDA) with positive polarity. Electrospray ionization (ESI) were set to 52 AU sheath gas flow, 14 AU auxiliary gas flow, 0 AU sweep gas flow and 400 °C auxiliary gas temperature. The spray voltage was set to 3.5 kV and the inlet capillary to 320 °C. 50 V S-lens level was applied. MS scan range was set to 150-1500 *m/z* with a resolution at *m/z* 200 ($R_{m/z\ 200}$) of 70,000 with one micro-100 scan. The maximum ion injection time was set to 100 ms with automated gain control (AGC) target of 1.0E6. Up to 5 MS/MS spectra per MS1 survey scan were recorded DDA mode with $R_{m/z\ 200}$ of 17,500 with one microscan. The maximum ion injection time for MS/MS scans was set to 100 ms with a AGC target of 3.0E5 ions and minimum 5 % C-trap filling. The MS/MS precursor isolation window were set to *m/z* 1. Normalized collision energy was set to a stepwise increase from 20 to 30 to 40 % with z = 1 as default charge state. MS/MS scans were triggered at the apex of chromatographic peaks within 2 to 15 s from their first occurrence. Dynamic precursor exclusion was set to 5 s. For QC checking, three QC standards namely, dibutyl phthalate, pheophorbide A and tryptophan were used as QC and analyzed before and after sample runs.

FBMN with MZmine 3 and GNPS

The LC-MS/MS dataset was converted from ".raw" to ".mzXML" format using MSConvert (https://proteowizard.sourceforge.io). Feature detection from the DOM dataset was performed in MZmine3[113,125] (https://mzmine.github.io/). The steps of data-processing were as follows: Mass

detection was carried out at MS1 level using a noise level of 2.0E5 and at MS2 level using a noise level of 1.0E3 and feature lists were assembled using the ADAP[394] Chromatogram Builder with a minimum group size of 5 scans, a group intensity threshold of 2.0E5, a minimum highest intensity of 5.0E5, and the scan to scan accuracy set to 0.002 *m/z* or 5 ppm.

Features were resolved using local minimum search with a chromatographic threshold of 82.0 %, the range for the minimum separating two peaks was set to 0.075 min and peaks were required to have a minimum absolute height of 5.0E5 with a top/edge ratio of at least 1.4, a duration no greater than 4.00 min and a minimum of 5 data points. Additionally, MS2 scans were paired to the peaks with a retention time tolerance of 0.225 min, a precursor tolerance of 0.002 *m/z* or 10 ppm and were limited by retention time edges.

The feature list was deisotoped using the 13C isotope filter with an *m/z* tolerance of 0.0015 *m/z* or 3 ppm and a retention time tolerance of 0.1 min. Furthermore, monotopic shape was required and the most intense isotope was set as representative, while retaining all features containing MS2 spectra. Feature lists were then merged using the Join aligner with an *m/z* tolerance of 0.0015 *m/z* or 5 ppm, a retention time tolerance of 0.2 min and weights of 3 and 1 for *m/z* and retention time, respectively. The merged feature list was filtered, removing only features without a paired MS2 spectrum and that were detected for less than two samples, or had not at least two features in their isotope pattern. Gap-filling was executed using the Peak finder module with an intensity tolerance of 20 %, an *m/z* tolerance of 0.002 *m/z* or 5 ppm, a retention time tolerance of 0.05 min and a minimum of 3 data points, followed by use of the Duplicate peak filter with the filter mode set to “NEW AVERAGE”, *m/z* tolerance to 2 ppm, and retention time tolerance to 0.2 min.

The metaCorrelate module was applied with a retention time tolerance of 0.1 min and an intensity correlation threshold of 5.0E5. Correlation grouping was activated with a minimum of 5 data points with at least 2 on each edge, and a minimum feature shape correlation of 85.0 % using Pearson similarity. Feature height correlation was also activated, requiring minimum values of 2 data points and 80 % correlation for Pearson similarity. Ion identity networking[107] as implemented for all features, with an *m/z* tolerance of 0.0015 *m/z* or 3 ppm and an annotation refinement deleting smaller networks with a link threshold of 4, and networks without monomer. The ion identity library settings had maximum charge and maximum molecules/cluster values both set to 2. Adduct ions were $[M+H]^+$, $[M+Na]^+$, $[M+K]^+$, $[M+NH_4]^+$, and modifications were $[M-H_2O]$ and [M+ACN]. The merged feature list was exported for GNPS-FBMN retaining only features containing MS2 spectra, as well as for SIRIUS with an *m/z* tolerance of 0.001 *m/z* or 5 ppm. The .xml batch file is available at https://github.com/Functional-Metabolomics-Lab/FBMN-STATS/tree/main/data.

Lastly, the .mgf spectra file, the .csv feature quantification table, the tab-delimited .txt metadata file, and the IIMN supplementary pairs .csv file were uploaded to GNPS[81] (gnps.ucsd.edu) and processed using a feature-based molecular networking (FBMN) workflow[42]. The precursor and fragment ion mass tolerances were both set to 0.01 Da and the minimum cosine value to connect pairs was set 0.7. Maximum number of neighboring nodes was set to 10 and maximum number of connected nodes in a cluster to 100, while a minimum of 6 matched fragments were required and a maximum of 500 Da mass difference between precursors. Analog match search was employed with a minimum of 6 matched fragments, a minimum cosine spectral similarity threshold of 0.7 and a maximum analog search mass difference of 100 Da with only the top result being

recorded. The precursor window filter, the 50 Da window filter for peaks and the library filter were all activated, in addition to the Dereplicator function[395].

Step by step Guide QIIME2

QIIME2 (Quantitative Insights Into Microbial Ecology 2) is a software package originally developed to perform microbial community analyses but also contains various plugins amenable to the use with LC-MS/MS non-targeted metabolomics data[147]. To facilitate analysis in QIIME2, we have provided a Jupyter notebook that directly imports a feature table and metadata table from a GNPS job link (https://github.com/Functional-Metabolomics-Lab/FBMN-STATS/blob/main/QIIME2/QIIME2_Untargeted_Metabolomics_Stats.ipynb) .

1. Opening Jupyter notebooks in QIIME2/GNPS environments

- The user has two options for running the FBMN-Hitchhikers through the QIIME2 pipeline. The user can either (1) install QIIME2 and GNPS packages locally or (2) install Docker and the Docker container. Instructions can be found here: https://github.com/Functional-Metabolomics-Lab/FBMN-STATS/blob/main/QIIME2/README.md

2. Conversion of feature quantification table to QIIME2 compatible format

- The Jupyter Notebook directly imports a feature table and metadata table from a GNPS job link (https://github.com/Functional-Metabolomics-Lab/FBMN-STATS/blob/main/QIIME2/QIIME2_Untargeted_Metabolomics_Stats.ipynb - "preprocessed"). Additionally, all necessary packages to run the code and download the feature (quant), manifest, and metadata tables from GNPS are imported.
- The GNPS job ID is set in this section to specify which data to download from GNPS. Note that the GNPS job must be a feature based molecular networking job that includes sample metadata.

▲**CRITICAL:** If the user wants to upload data directly after feature finding without running a GNPS job, this can be done by adding feature table and metadata to the data folder. Then feature table and metadata can be added directly into the Jupyter Notebook. We recommend the user labels their feature table as "quant.csv" and their metadata as "unprocessed_metadata.tsv". In this case, the user should start at the step above ("Changing Metadata and Manifest Column name"). The first column of the metadata must be labeled "filename".

3. Changing "Metadata" and "Manifest Column" name

- The "unprocessed_metadata.tsv" that is downloaded is the metadata file from the GNPS job. In this step, the "filename" column name (required for GNPS) is changed to "sample id" (required for qiime2). This step creates a new file called "metadata.tsv", which is the modified version of unprocessed_metadata.tsv.

4. Preliminary Setup to Data Cleanup

- The preliminary setup and data cleanup procedures are mainly the same as the R setup. New metabolomics plugins for QIIME2 were written for blank removal, imputation, and normalization steps. Source code for these new Qiime2 Plugins can be found in the following:
    - https://github.com/Wang-Bioinformatics-Lab/qiime2_blank_removal_plugin
    - https://github.com/Wang-Bioinformatics-Lab/qiime2_normalization_plugin
    - https://github.com/Wang-Bioinformatics-Lab/qiime2_imputation_plugin

- ▲**CRITICAL:** Do not install the `scikit-learn` package on Windows, as there may be an error with building wheels. However, this package works fine on the Google Colab platform.
- The random values generated in the “imputation” step will differ from those in R, leading to a different resulting imputed table.

5. Statistical Analysis

- The next steps are recommended for statistical analysis of LC-MS/MS metabolomics data.
- This includes generating a Longitudinal ANOVA, Distance Matrix, Principal Coordinate Analysis (PCoA), Emperor Plot, Classifier Data/Heat Map, and PERMANOVA
- ▲**CRITICAL:** Please see the following links for additional information and troubleshooting: https://docs.qiime2.org/2023.5/plugins/available/longitudinal/anova/, https://docs.qiime2.org/2023.5/plugins/available/diversity/beta/, https://docs.qiime2.org/2023.5/plugins/available/diversity/pcoa/, https://docs.qiime2.org/2023.5/plugins/available/emperor/plot/, https://docs.qiime2.org/2023.5/plugins/available/sample-classifier/classify-samples/, https://docs.qiime2.org/2023.5/plugins/available/diversity/beta-group-significance/.

Step by step Guide Python

This guide provides insights into the specificities of the Python implementation, focusing on areas where differences from the R protocol may arise. While the general steps for the analysis are similar between R and Python, there are some important distinctions that need to be highlighted:

1. Dependencies:

- The libraries and packages required in Python differ from those in R.
- Unlike the R notebook, where packages are installed before each section, all necessary packages for the Python implementation are installed at the beginning of the analysis. This is generally faster in Python than in R.
- Users must ensure that all the appropriate dependencies are installed before beginning the analysis.

2. Preliminary Setup to Data Cleanup (Steps 1-27)

- The preliminary setup and data cleanup procedures are mainly the same as the R setup.

- ▲**CRITICAL:** Do not install the `scikit-bio` package on Windows, as the package is not compatible for Windows OS and will result in an error with building wheels. However, this package works fine on the Google Colab platform.
- **Imputation:** The random values generated in Python will differ from those in R, leading to a different resulting imputed table.
- **Normalization and scaling:** The results of these data manipulation steps are slightly different between the Python notebook and the R notebook. Also, due to challenges associated with the PQN package in Python, PQN normalization has been excluded from the Python notebook.

3. Data Manipulation

- **Indexing**: Python is 0-indexed, while R is 1-indexed. This difference in indexing should be considered during data manipulation. Also, the dimensions of some dataframes (e.g., ft_an, new_md) in the python notebook might differ by one column from the R notebook due to differences in indexing of the dataframes.
- **Row Names**: In Python, the equivalent of the `rownames()` function in R is referred to as the "index".

4. Multivariate analysis

- **Principal coordinates analysis:** The PCoA, like other statistical analyses, produces slightly different results in the Python notebook than in the R notebook. This is due to differences in the implementation of the respective Python or R packages. Additionally, some of the values have different signs in Python than in R, resulting in differently oriented PCoA plots. The overall patterns within the data, however, are the same.
- The distance metrics available in the `scipy` package of Python differ from the metrics available over R.
- **Permutational multivariate analysis of variance:** The PERMANOVA test also results in marginally different R2 values between the two notebooks.
- **Hierarchical clustering analysis:** The cutting of the dendrogram is differently executed in the Python notebook. Here, the dendrogram function allows us to specify the number of clusters we want to display ('lastp') and shows them as individual horizontal lines. Among those lines, the samples contained in a cluster are indicated by gray dots and their number is given on the x-axis.
- **Silhouette method:** Please note that the silhouette score in the Python `scikit-learn` package requires more than one cluster. Thus, the corresponding plot in step 38 shows a cluster range from 2 to 10.
- **Heatmaps:** The hierarchical clustering again shows slight differences between the two notebooks, but overall patterns are contained. Furthermore, the `PyComplexHeatmap` package does not provide k-means clustering. We have implemented k-means clustering in the Python notebook without a heatmap visualization and direct the user to the heatmap with k-means clustering in step 45 of the R notebook.

- **Random Forest:** The functionality of the `rfPermute` package in R differs strongly from Python scikit-learn `RandomForestClassifier`. Although the general approach of random forest is the same, the implementation, resulting values, and accessibility of those values are highly different. We therefore restricted the model evaluation in the Python notebook to the OOB error, the “Mean Decrease Accuracy” (MDA), and the interpretation of the results to the top ranked features. For more detailed evaluation curves and result plots, see the R notebook.

5. Univariate analysis

- **Normality testing:** The plotted features are different between the Python and the R notebook, due to the differently sorted dataframes. The Python notebook plots feature X2…, whereas the R notebook plots the feature X10015…
- The individual sections of the univariate analysis in the Python notebook contain more functions encompassing several cells of the corresponding sections in the R notebook. The underlying functionality and tests however, are the same in both notebooks.
- **Tukey HSD:** Please note that the volcano plot in the Python notebook is inverse along the x-axis with respect to the plot in the R notebook. This also results in the box plots of the right tail in the Python notebook corresponding to the box plots of the left tail in the R notebook. The same applies to the box plots of the left tail. Additionally, the order of the sample areas on the x-axis are different between the Python and the R notebooks.
- **T-Test:** In contrast to the R notebook, the Python notebook does not add a specific attribute corresponding to rainfall. Instead, the Python notebook performs the T-test on the ‘Attribute Month’, classifying the data into two conditions: ‘Jan-2018’ or ‘not Jan-2018’. Since the rainfall happened in January 2018, this classification corresponds to the rainfall attribute in the R notebook.
- **T-Test volcano plot:** The volcano plot in the Python notebook displays the t-statistic on the x-axis, whereas the R notebook shows the difference of the means.
- **T-Test box plots:** The boxplots in the Python notebook are based on the ‘Attribute Month’, instead of the ‘Attribute rainfall’ in the R notebook, and thus show three different categories according to the three months when the samples were acquired.

Step-to-step Guide Stats App

This section provides an overview of the steps to efficiently navigate and use the web app available at https://fbmn-statsguide.gnps2.org/. Upon accessing the homepage, titled “Statistics for Metabolomics”, users will find comprehensive guidelines such as how to effectively zoom in and out of the interactive visual representations, as well as essential details regarding the formatting of input files. Alongside the main content, an “About” tab is present at each stage of the analysis, offering brief explanations mirroring those in the protocol.

To further tailor the user experience, the left sidebar of the homepage houses few customizable settings. The default p-correction factor is set to “Bonferroni”, but we suggest selecting “Benjamini-Hochberg FDR” from the dropdown for optimal results. For exporting the figures, the

“image export” option defaults to SVG format but can be adjusted based on user preference. Although the app is self-explanatory, we mention some general information and tips in this SI section.

1. **Data Preparation**

- Two tables are required as input data: quantification table (referred as ‘feature table’ in the protocol) and metadata table. They can be imported into the app in the formats tsv, txt, csv, xlsx files by dropping them in their respective boxes.
- By clicking on the file upload box, the user can decide among different uploading modes. The user can manually import the tables by dragging and dropping them into their designated boxes, or they can select the “GNPS task ID” for uploading files. Upon selecting this latter option, the user will be prompted to paste the ID of the GNPS job of interest and load the input files directly from GNPS. If the data uploading process was successful, a dialog box will appear allowing the user to verify the input data.

2. **Data cleanup**

- The data can now be directly submitted for statistical analysis, if desired, to look at the raw data distribution using, for example, PCA or PCoA. But the protocol recommends to start with data cleanup.
- There is the option to perform blank removal on the entire dataset. To do this, check the “Blank Removal” box, additional windows will appear, allowing the user to input the necessary information to distinguish between samples and blanks.
    - In the first box named “attribute for sample selection” the user has to select an attribute from the metadata which can be used to efficiently distinguish between blanks and the samples of interest.
    - In the second box named “sample selection”, the user can select the attribute identifying the samples. The same procedure has to be repeated afterwards, this time the user has to insert the “attribute for blank selection” and for the “blank selection” table, one can choose the attribute that identifies the blanks to be subtracted from the rest of the dataset.
- Now the user has to decide how strict the blank removal has to be. This involves setting a threshold for the ratio between the blank mean and sample mean to identify which features are considered noise. This will instantly display the count of potential noise features set for removal based on the chosen threshold.
- A cutoff threshold between 0.1 and 0.95 can be selected but the recommended value is between 0.1 and 0.3.
- Following this, the user has the option to fill-in missing values by selecting the corresponding checkbox for imputation. The user can also perform normalization methods on the data prior to submitting them for statistics by clicking the red button.

3. Statistics

- Now the user can perform all the statistical analyses present in the notebook, by selecting the desired test from the left menu (PCA, PCoA, Hierarchical Clustering & Heatmap, random forest, ANOVA and Tukey´s, Kruskal Wallis & Dunn's post hoc, t-test).
- Before running each statistical test, the user needs to designate the attribute and specify the attribute options to be compared.
- There is also a guide called “Parametric assumptions evaluation?” which will lead the user to choose the appropriate univariate test (parametric or non-parametric test) for their dataset.

4. Download results

- Throughout the application, users can view the results in the form of graphical plots and tables.
- For a closer look at the plots, users can click the expansion button (⤡) located at the top-right of the graphic. This action will enlarge the plot for detailed observation.
- Users can also download the plots by clicking on the camera icon located in the top-right corner of the graphic. Upon clicking, the plot will be saved in SVG format.
- The resulting data table corresponding to each test can be accessed by clicking on the “🗀 Data” button. Users can expand these tables in a manner similar to the plots for easy readability. The data can also be downloaded by selecting the “download table” button.

Table S1: Cheat Sheet, Terms and Parameters used in the manuscript and accompanying notebooks

Terms used in Manuscript	Corresponding terms in Notebook	Definition	Relevant Parameters
Preliminary Setup			
Features, Metabolites	Compounds	Metabolites generated from chromatogram deconvolution	
Annotation, identifying	Annotation	Identification of compound from database	

Feature table	Feature table, quantification table	Table contained features e.g., retention time, *m/z*, peak area, etc.	
Metadata	Metadata	Data described the information or detail of sample	
Annotation table	Annotation table	File contained annotation compound from database	
Working Directory or folder	Working directory or folder	The file path where all necessary data and scripts are stored.	
Libraries	Libraries	Packages or functions loaded for data manipulation and analysis	**R:** library(tidyverse) **Python:** import pandas as pd, import numpy as np
Uploading Files	Data import	Function for importing csv file	**R:** read.csv() **Python:** pd.read_csv()
Exploring the Imported Files	Exploring the Imported Files	Function for viewing the header of the data and its dimensions	**R:** head(), dim() **Python:** pd.DataFrame.head(), pd.DataFrame.shape
Special summary function	Summarizing the metadata	A special summary function to get details in metadata	**R:** InsideLevels() **Python:** InsideLevels()

Count number of columns	Count number of columns	Function for count columns	**R:** ncol() **Python:** len() (general function for obtaining the length of an object; here applied to the object containing the columns e.g. md.columns)
Missing values	Missing values	Function used to check missing value	**R:** is.na() **Python:** pd.DataFrame.notna(), np.nan
Data clean up			
Dataframe	Dataframe	Changing data format to dataframe	**R:** as.data.frame() **Python:** pd.DataFrame()
Transposing	Transposing	Transposing the data	**R:** t() **Python:** pd.DataFrame.T
Merging	Merging	Merging the metadata and feature table together	**R:** merge() **Python:** pd.merge()
Clean the feature table	Cleaning the file	Removing and replacement string name in column name	**R:** gsub() **Python:** str.replace()
Filter rows	picking rows	Function for picking only the features	**R:** filter() **Python:** pd.DataFrame[pd.DataFrame[*column_number*] == *number*]
Log	Log	Function to take log	**R:** log() **Python:** np.log
Round the number	Round the number	Function to round the number	**R:** round() **Python:** round()
Normalization	Normalization	Performing normalization	**R:** normalization()

			Python: pd.DataFrame.apply(lambda x: *function*, axis=1) (the apply function is a general one that applies the specified *function* (e.g. x/np.sum(x)) to the dataframe)
Scaling, Center-Scaling	Scaling, Center-Scaling	Function to perform scaling	**R:** scale() **Python:** StandardScaler().fit_transform()
Multivariate analysis			
Perform PCoA	Perform PCoA	Function to perform PCoA	**R:** cmdscale(), plotPCoA() **Python:** skbio.stats.ordination.pcoa()
Executing HCA	Executing HCA	Function to perform HCA	**R:** hclust() **Python:** dendrogram(linkage())
Cutting the Dendrogram	Cutting the Dendrogram	Function to group dendrogram	**R:** cutree() **Python:** dendrogram(linkage(), truncate_mode='lastp', p=*number*)
Heatmap	Heatmap	Function create heatmap	**R:** Heatmap() **Python:** ClusterMapPlotter()
Supervised learning with Random Forest			
Balance sample sizes	Balance sample sizes	Function for balance sample size	**R:** balancedSampsize() **Python:** class_weight.compute_class_weight()
Run Random Forest	Run Random Forest	Function to execute random forest	**R:** rfPermute() **Python:** RandomForestClassifier()
Interpreting RF Results	Interpreting RF Results	Function to plot the top predictions	**R:** plotImpPreds() **Python:** Not plotted, but saved to a csv file
Proximity plot	Proximity plot	Function to plot class predictions vs actual group and add 95%	**R:** plotProximity() **Python:** Not available in the python notebook

		confidence ellipse	
Univariate analysis			
Test for normality	Normality Testing for One Feature	Function to plot Q-Q plot as histogram	**R:** qqnorm() **Python:** statsmodels.api.qqplot()
Test for normality	Normality Testing for One Feature	Function to plot Q-Q plot as line	**R:** qqline() **Python:** statsmodels.api.qqplot(line='s')
Non-normal distribution.	Non-normal distribution	Function to perform a Shapiro-Wilk test	**R:** shapiro.test() **Python:** scipy.stats.shapiro()
ANOVA test	Running ANOVA	Getting ANOVA output	**R:** broom::tidy(aov()) **Python:** pingouin.anova()
Tukey's post-hoc test	Tukey's post-hoc test	Function to perform Tukey HSD test and summarize the result	**R:** broom::tidy(TukeyHSD() **Python:** pingouin.pairwise_tukey()
Perform T-Test	Perform T-Test	Function to perform T-Test test	**R:** t.test() **Python:** pingouin.ttest()
Kruskal-Wallis Test	Kruskal-Wallis Test	Function to perform Kruskal-Wallis Test on the first feature	**R:** broom::tidy(kruskal.test()) **Python:** pingouin.kruskal()
Dunn's post hoc test	Dunn's post hoc test	Function to perform Dunn Test for a Significant Feature	**R:** dunnTest() **Python:** scikit_posthocs.posthoc_dunn
Getting output			

Showing the data	Showing the data	Function to show or display data	**R:** display(), print() **Python:** display(), print(), matplotlib.pyplot.show(), matplotlib.figure.Figure.show()
Export as figure	Export as figure	Function for exporting figure	**R:** svglite(), ggsave() **Python:** plotly.io.write_image(), matplotlib.pyplot.savefig()
Getting output as .csv file	Getting output as .csv file	Function to save table as .csv file	**R:** write.csv() **Python:** pandas.DataFrame.to_csv()
Getting output as zip file	Getting output as zip file	Function to export all data in folder as zip file	**R:** utils::zip() **Python:** shutil.make_archive()

Chapter 3: Supplementary Information

Endpoint drug screening

To assess microbiome-mediated drug metabolism, 50 compounds were incubated with the Com20 synthetic gut community under anaerobic conditions in mGAM medium. Experiments were conducted in three batches based on drug solubility and replication design. All incubations were performed in U-bottom 96-well plates and extracted using ethyl acetate (EtOAc) at two timepoints (t = 0 h and 2 h). *Eggerthella lenta*, a slow-growing Com20 member, was pre-cultured several days in advance, while the remaining 19 strains were freshly inoculated in mGAM and incubated anaerobically at 37 °C. The Com20 community was assembled at an initial OD_{578} of 0.01 in 5 mL mGAM and grown overnight at 37 °C under anaerobic conditions to reach OD 0.5-1.0.

Drug stocks were prepared in water or DMSO depending on solubility and diluted 1:20-1:50 in mGAM before plating. For DMSO-soluble drugs, 200 µL of diluted drug solution was dispensed into each well, followed by 200 µL of pre-grown Com20. For water-soluble drugs, 800 µL of Com20 was added directly to wells containing the diluted drug solution. Abiotic controls (drug only), vehicle controls (DMSO or water only), and at least two biological replicates per treatment were included on each plate.

Immediately after inoculation (t = 0 h), 400 µL from each well was extracted into 1 mL EtOAc, mixed, sealed, and stored at −20 °C. The remaining cultures were incubated anaerobically at 37

°C for 2 h, after which an additional 400 µL was extracted using the same procedure to generate t = 2 h samples.

Non-targeted Metabolomics using LC-MS/MS

Sample extraction and preparation

Drug-microbiome incubations were performed under anaerobic conditions at the University Hospital Tübingen. At each timepoint, samples were collected in 96-deep-well plates (2 mL capacity), sealed, and stored at -80 °C until extraction. Frozen samples were transported on ice to the Functional Metabolomics Laboratory, University of Tübingen, for LC-MS/MS analysis and stored at -20 °C upon arrival. For extraction, plates were thawed at room temperature and mixed thoroughly before adding EtOAc. Each well contained 500 µL of sample, to which 1 mL of EtOAc was added. The mixtures were sonicated for 10 min and centrifuged at 3000 × g for 5 min. The upper EtOAc layer was transferred to new plates and dried under vacuum at room temperature using a SpeedVac concentrator. Dried extracts were resuspended in 150 µL of 50% methanol in water prior to LC-MS/MS measurement.

LC-MS/MS acquisition parameters

A pooled QC sample and an in-house six-compound QC mix were injected periodically to monitor instrument performance and retention time stability across runs. LC–MS/MS analyses were performed on a Q Exactive HF Orbitrap mass spectrometer (Thermo Fisher Scientific) equipped with a heated electrospray ionization (HESI) source and coupled to a Vanquish UHPLC system. Separation was achieved on a Kinetex C18 column (2.1 × 50 mm, 1.8 µm, 100 Å; Phenomenex) using a 7 min gradient. The mobile phases consisted of solvent A (water, LC/MS grade, Fisher Scientific) with 0.1% formic acid (FA), and solvent B (acetonitrile, LC/MS grade, Fisher Scientific) with 0.1% FA. After sample injection, a 5-minute linear gradient was applied. Solvent B was increased linearly from 5% to 50% over the first 4 minutes, then ramped to 99% between 4 and 5 minutes. This was followed by a 2-minute column wash at 99% solvent B, and then a 2-minute column equilibration with 5% solvent B.

MS data were acquired in positive-ion mode using HESI (spray voltage 3.5 kV; sheath gas 50 AU; auxiliary gas 12 AU; sweep gas 1 AU; capillary 250 °C; aux heater 400 °C; S-lens RF level of 80 V). Full MS scans were collected from m/z 150-1500 *m/z* (resolution 30,000; AGC 1×10^6). Data-dependent MS/MS spectra were acquired for the top 5 intense precursor ions from each MS scan (resolution 15,000; AGC 5×10^5). An isolation window of 1 *m/z* was employed, followed by a stepped normalized collision energy (NCE) of 25, 35, 45 eV for ion fragmentation. Additional settings included a dynamic exclusion of 5.0 seconds, apex trigger enabled, and isotope exclusion enabled.

ChemProp2 Webapp

The ChemProp2 web application supports CSV, XLSX, TSV, and TXT input formats and can also import data directly from GNPS via a Feature-Based Molecular Networking (FBMN) job ID. The interface includes modules for data preprocessing, ChemProp scoring, false discovery rate (FDR)

analysis, and interactive visualization. Preprocessing options include blank removal, missing-value imputation, and total ion current (TIC) normalization.

ChemProp is available in two modes: ChemProp1, designed for two-timepoint datasets and producing unscaled scores, and ChemProp2, which supports multi-timepoint experiments and outputs normalized scores between 0 and 1. The resulting table reports node-pair information, including score magnitude (0 to 1) and transformation directionality (+1 or -1, indicating the inferred conversion between nodes).

The FDR module, available only in ChemProp2, applies a target-decoy approach to estimate reliability thresholds (e.g., 1%, 5%, 10%). Randomized decoy feature tables are used to generate a null distribution for score comparison, and cumulative FDR values are computed across score bins. Users can then apply the FDR cutoffs to filter results, downloadable as a CSV file or as a Cytoscape-compatible ZIP archive containing GraphML and style files.

Visualization tools include the (i) global transformation plot, which displays ChemProp2 scores versus m/z differences to highlight transformation trends, and (ii) an interactive edge viewer that allows filtering by score range, m/z difference, annotation name, or node ID. Selected edges are visualized in two synchronized panels: (A) intensity profiles for the node pair across timepoints, and (B) the corresponding subnetwork with directional ChemProp2 arrows indicating putative precursor–product relationships.

Performance Comparison of ChemProp1 and ChemProp2

To compare ChemProp2 with ChemProp1, we reanalyzed the multi-timepoint dataset of 12 drugs using ChemProp1, restricting it to the end timepoints (T0 vs T8). ChemProp1 was applied only to direct drug neighbors (D1 edges) from the FBMN network, as cascade expansion was not part of its original framework. ChemProp1, which reports log fold-change between pairs of timepoints, yielded more non-zero edges since it captures any change over the entire period, regardless of fluctuations in between. However, ChemProp2 integrates all timepoints and applies FDR correction, resulting in fewer but more reliable drug-metabolite relationships. Overall, ChemProp1 detected 22 edges above the threshold (≥1), while ChemProp2 identified 36 edges above threshold 0.1 and 10 above threshold 0.3, corresponding to drug-specific FDR cutoffs (~±0.35-0.4). **SI Table 3** summarizes, for each drug, the number of D1 edges retained by both methods at these thresholds, and **SI Figure 13** illustrates this comparison for edges with Δm/z values between 0 and 200 Da.

FASST Searches

The FASST search was performed against 13 GNPS-indexed databases: NORMAN, ORNL_Bioscales2, ORNL_Populus_LC_MSMS, gnpsdata_index, gnpsdata_test_index, gnpslibrary, massivedata_index, massivekb_index, metabolomicspanrepo_index_latest, metabolomicspanrepo_index_nightly, panrepo_2024_11_12, panrepo_2025_07_09, panrepo_2025_08_06, and ptfi2_index (https://wang-bioinformatics-lab.github.io/GNPS2_Documentation/masst/, https://fasst.gnps2.org/libraries.

Across the FASST searches, matches were obtained from **1,670 unique MASSIVE datasets** (MSV accessions). After excluding our six deposited repositories (MSV000093571, MSV000094899, MSV000096724, MSV000091452, MSV000095311, MSV000092059), hits remained in 1,664 external datasets.
Our FBMN analysis (GNPS job ID: 1b5b94b4191d4223a5f57afb2aaaf0b0), yielded 8,055 features, with 517 (6.4%) were annotated through GNPS spectral libraries, including all parent drug ions ($[M+H]^+$). ChemProp2 scores were calculated for all edges (cosine similarity-based and cascade). For downstream analysis, we focused on subnetworks centered around each drug's primary $[M+H]^+$ ion but also included **treatment-specific features** disconnected from the main clusters, as they may represent true biotransformation products. In total, **1,202 features** were queried via FASST, comprising (i) nodes from $[M+H]^+$ clusters, (ii) adduct clusters, and (iii) treatment-specific features. FASST searches were performed with **precursor tolerance 0.05 Da**, **fragment tolerance 0.05 Da**, and **cosine similarity ≥ 0.8**. Both library hits and repository spectral matches were considered in downstream interpretation. **SI Table 4** summarizes FASST results by drug, indicating the number of features searched, those with at least one match, and the corresponding unique datasets (including and excluding our own repositories).

While all queried features were expected to return at least one match from our deposited datasets, **139 compounds did not**. Manual verification through the FASST web interface (https://fasst.gnps2.org), confirmed that including **precursor charge state** restored matches for some cases. However, 120 features (16 from Cilnidipine, 104 from Simvastatin) still failed to return hits despite repeated manual checks. The cause remains unclear, and these 139 features were excluded from downstream analysis.

Cross-Dataset Distribution of Cascade Nodes (Heatmap Analysis)

As noted in the cascade summary, we began with 508 edges corresponding to drug-associated subnetworks and retained 432 cascade nodes with $\Delta m/z > 0.5$. To examine where these putative transformations are most prevalent, MassIVE metadata were retrieved for each feature and their species information were grouped into ten higher-level classes (e.g., Human, Mouse, Other Animals, Plant, Bacteria, Fungi, Environmental, Food, Chemical, Unclassified). Nodes were retained if they (i) had a non-zero ChemProp2 treatment score, (ii) differed from the parent drug by $\Delta m/z > 0.5$ Da, (iii) were observed in external MassIVE datasets, and (iv) represented forward transformations (Sign_ChemProp2_TRT = 1.0). This filtering resulted in 78 putative transformation nodes **(see SI Table 5)**. The heatmap (**Figure 5**) shows their distribution across MassIVE dataset categories, alongside their ChemProp2 scores, highlighting the reproducibility and directional trend of the transformations.

Cascade Node Summary Table

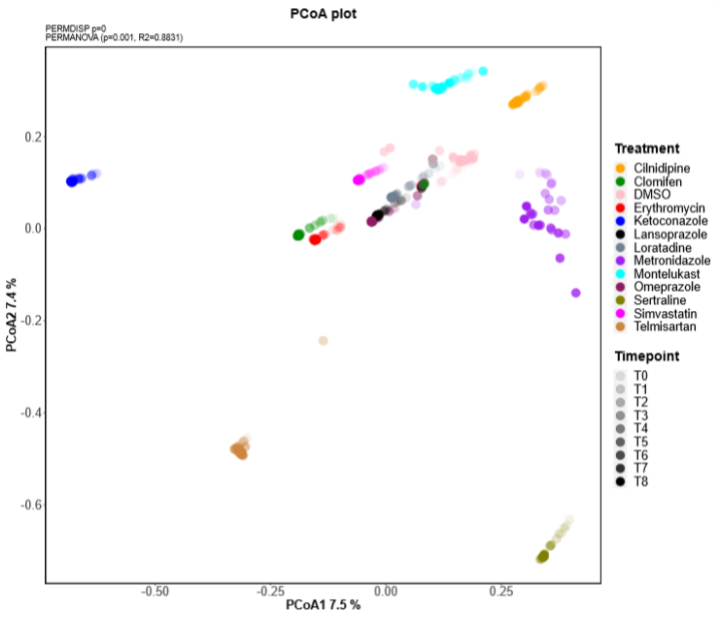
The following tables summarize drug-specific subnetworks, cascade edges, ChemProp2 hits, and whether features were annotated in GNPS libraries (FBMN) or found in external MassIVE repositories (FASST). Only cascade edges were included.

To generate these tables, we extracted all subnetworks from FBMN in which the parent drug $[M+H]^+$ ion resides (14 parent nodes in total: 12 drugs, with two $[M+H]^+$ nodes each for Simvastatin and Telmisartan, and one node for each of the remaining drugs). These 14 drug nodes all had library matches *(*See also the main manuscript section *Cascade Scoring Reveals Multi-Step Biotransformations*). For cascade scoring, we included only edges connecting each drug node to other nodes in the parent $[M+H]^+$ ion cluster, independent of spectral cosine similarity. In total, this yielded 515 drug-associated features. After removing the 14 parent drug ions, 436 features remained, connected by 508 edges with $|\Delta m/z| > 0.01$. Applying a more stringent filter of $|\Delta m/z| > 0.5$ reduced the set to 433 cascade nodes, which were used for the summary analyses below.

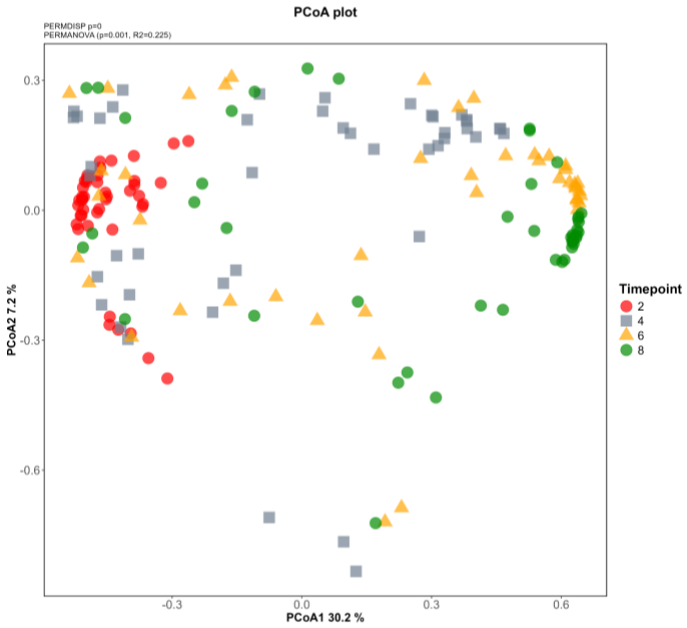
Each summary table contains the following fields:

- **Drug Name**: one of the 12 drugs.
- **Drug features**: number of parent drug nodes considered ($[M+H]^+$ only; 14 total).
- **Nodes (∆m/z > 0.5)**: number of cascade nodes differing from the drug node by more than 0.5 m/z. Nodes within ±0.5 m/z were excluded. For example, Cilnidipine yielded 98 such nodes.
- **Library Matched**: among these filtered nodes, the number of features with GNPS library matches.
- **Unmatched Compounds**: number of filtered nodes without library matches.
- **MassIVE Matches**: number of filtered nodes with ≥1 external hit in MassIVE datasets (counting nodes, not datasets).
- **ChemProp2 Hits (>0.1)**: number of filtered nodes with ChemProp2 treatment scores >0.1, indicating time-resolved intensity profiles consistent with downstream drug products.
- **Annotated Hits (>0.1)**: among ChemProp2 hits, number with GNPS library annotations.
- **MassIVE Hits (>0.1)**: among ChemProp2 hits, number also observed in external MassIVE datasets.
- **Final Hits:** Number of cascade nodes with ChemProp2 treatment score >0, m/z difference >0.5, significant treatment association (Sign_ChemProp2_TRT = 1.0) (This is represented as Heatmap in Fig 5 in the main manuscript)

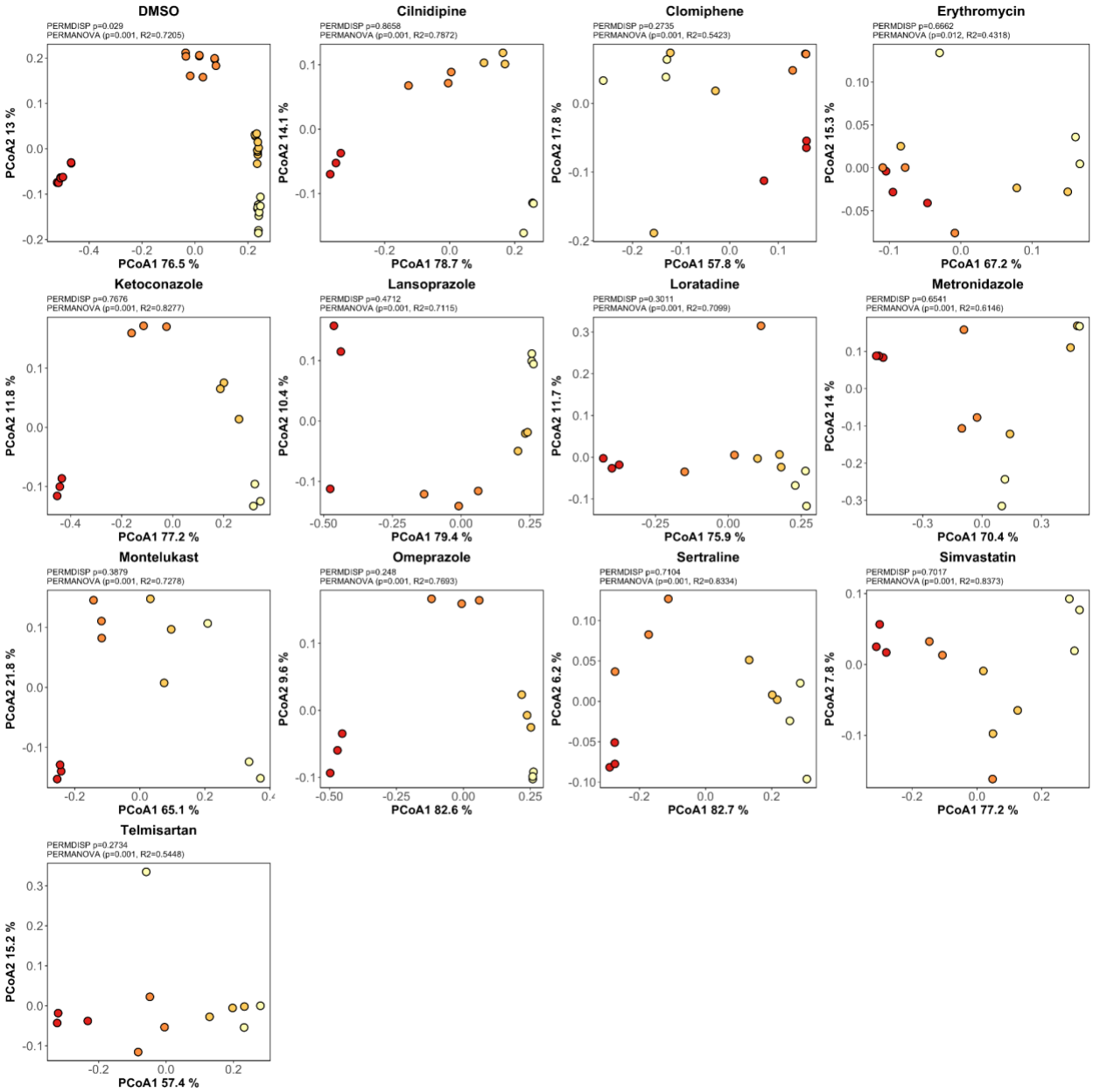
Supplementary Figures



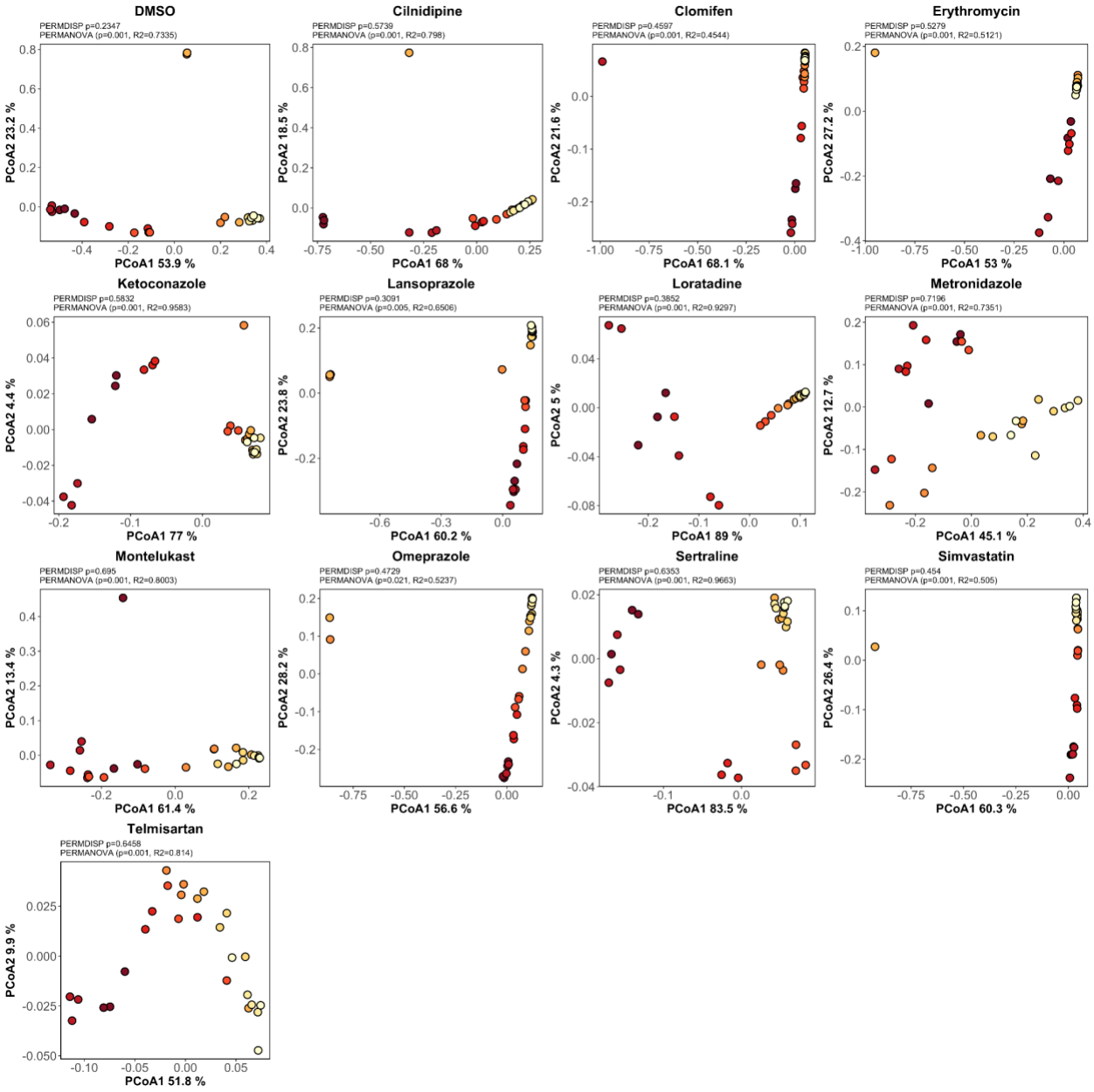
SI Figure 1: PCoA (Bray–Curtis) of metabolomic profiles for the abiotic control. PERMANOVA and PERMDISP were performed across different drug treatments. PERMANOVA p = 0.001, R^2 = 0.88; PERMDISP p = 0.



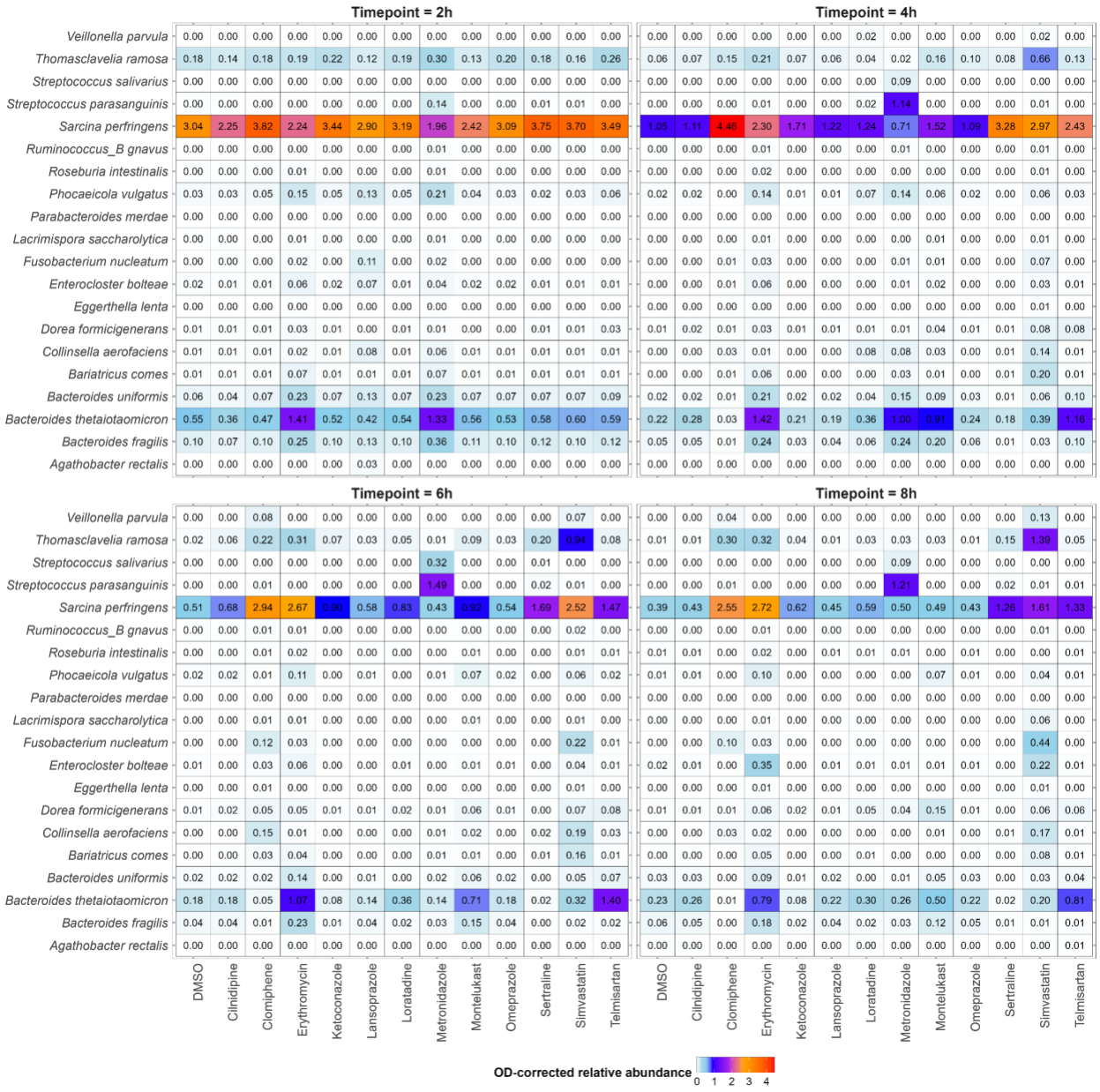
SI Figure 2: PCoA (Bray–Curtis) of 16S rRNA ASV profiles for drug-treated bacterial communities. Points are colored by timepoint. PERMANOVA and PERMDISP were performed across T2, T4, T6, and T8 samples (PERMANOVA p = 0.001, R^2 = 0.225; PERMDISP p = 0).



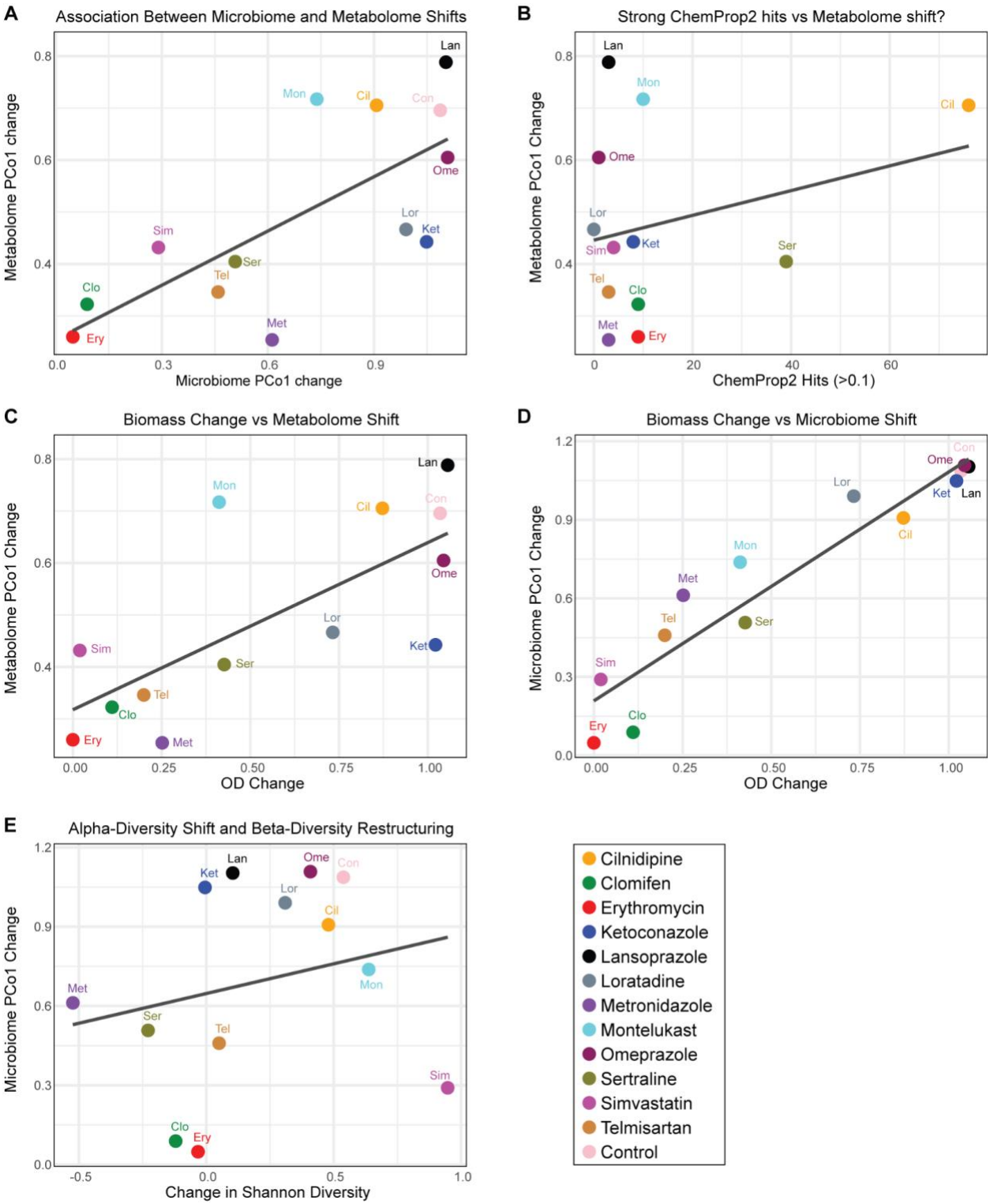
SI Figure 3. Individual PCoA (Bray–Curtis) plots of OD-corrected microbiome profiles for Com20 treated with 12 different drugs and DMSO control (13 plots total). Points are colored from yellow (T2) to dark red (T8) to represent increasing timepoints. PERMANOVA and PERMDISP statistics shown in each plot are based on timepoint differences.



SI Figure 4. Individual PCoA (Bray–Curtis) plots of metabolomic profiles for Com20 treated with 12 different drugs and DMSO control (13 plots total). Points are colored from light yellow (T0) to dark red (T8) to represent increasing timepoints. PERMANOVA and PERMDISP statistics shown in each plot are based on timepoint differences.

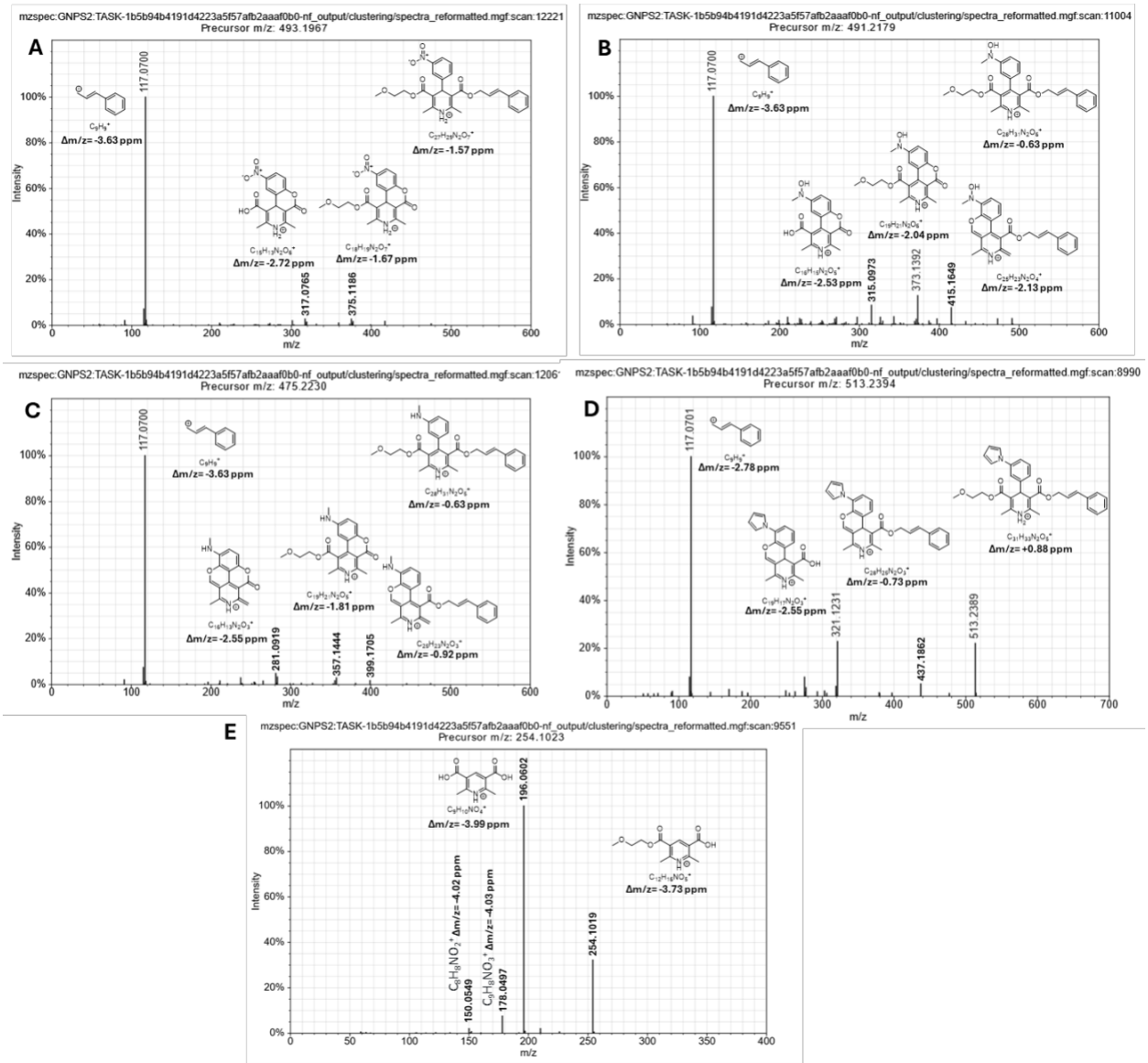


SI Figure 5. Heatmaps showing bacterial community responses across all drug treatments at each timepoint. Mean OD-corrected abundances (averaged across replicates) are displayed for all 12 treatments. Rows correspond to individual taxa and columns to treatments. Color intensity reflects the OD-corrected mean relative abundance.



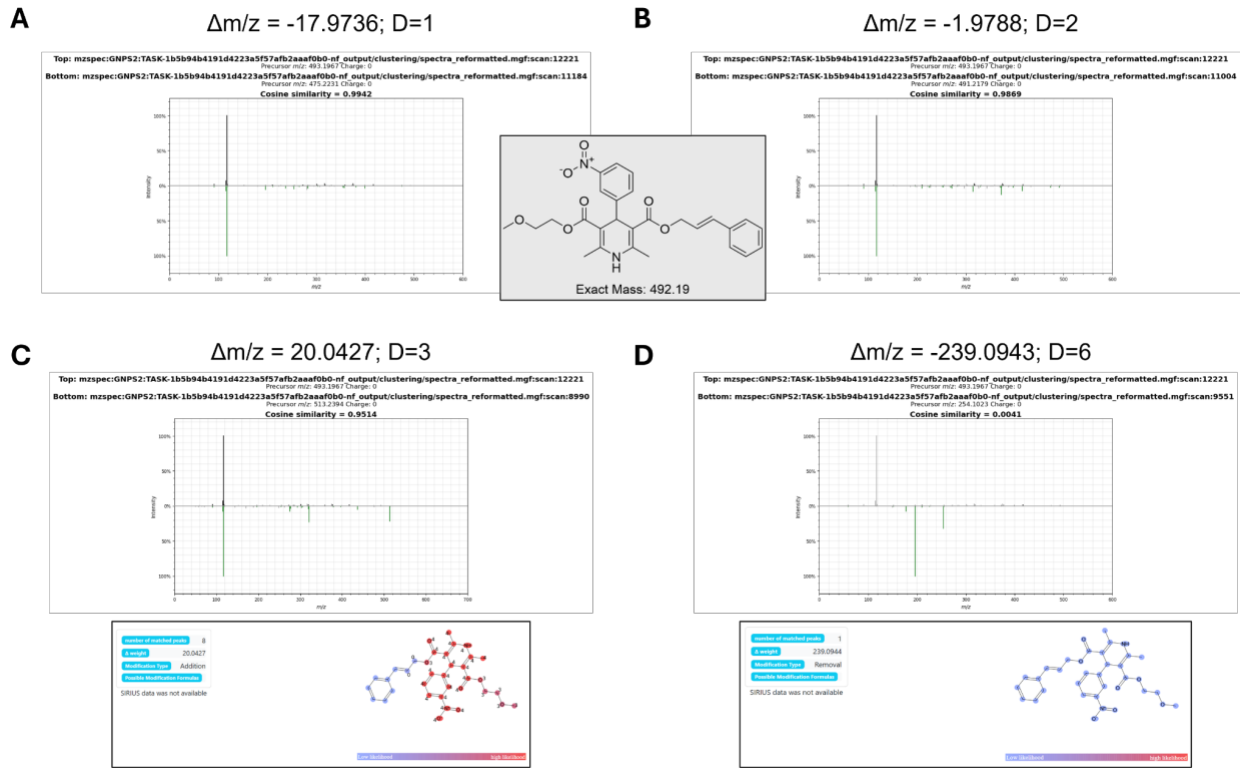
SI Figure 6. Multi-layer comparisons of drug-induced metabolome and microbiome shifts. Each point represents one drug treatment, with metrics calculated from its time-series data (T0-T8). **(A)** Relationship between metabolomic and microbiome restructuring over time. For each drug, PCo1 change reflects the total movement along the PCoA1 axis (metabolomics and 16S datasets), calculated as the cumulative change in mean PCoA1 values across consecutive

timepoints; **(B)** Association between predicted chemical transformations and metabolomic change, showing the number of ChemProp2 hits (score > 0.1) per drug versus its PCoA1 trajectory length; **(C)** Biomass change versus metabolomic restructuring, comparing ΔOD (T8 – T2) with metabolome PCoA1 trajectory lengths for each drug; **(D)** Biomass change in relation to microbiome restructuring, with restructuring measured as compositional change captured by the microbiome PCoA1 trajectory length over time; **(E)** Diversity restructuring: alpha-diversity change (within-sample diversity from T0–T8) versus beta-diversity change (microbiome PCoA1 trajectory length).

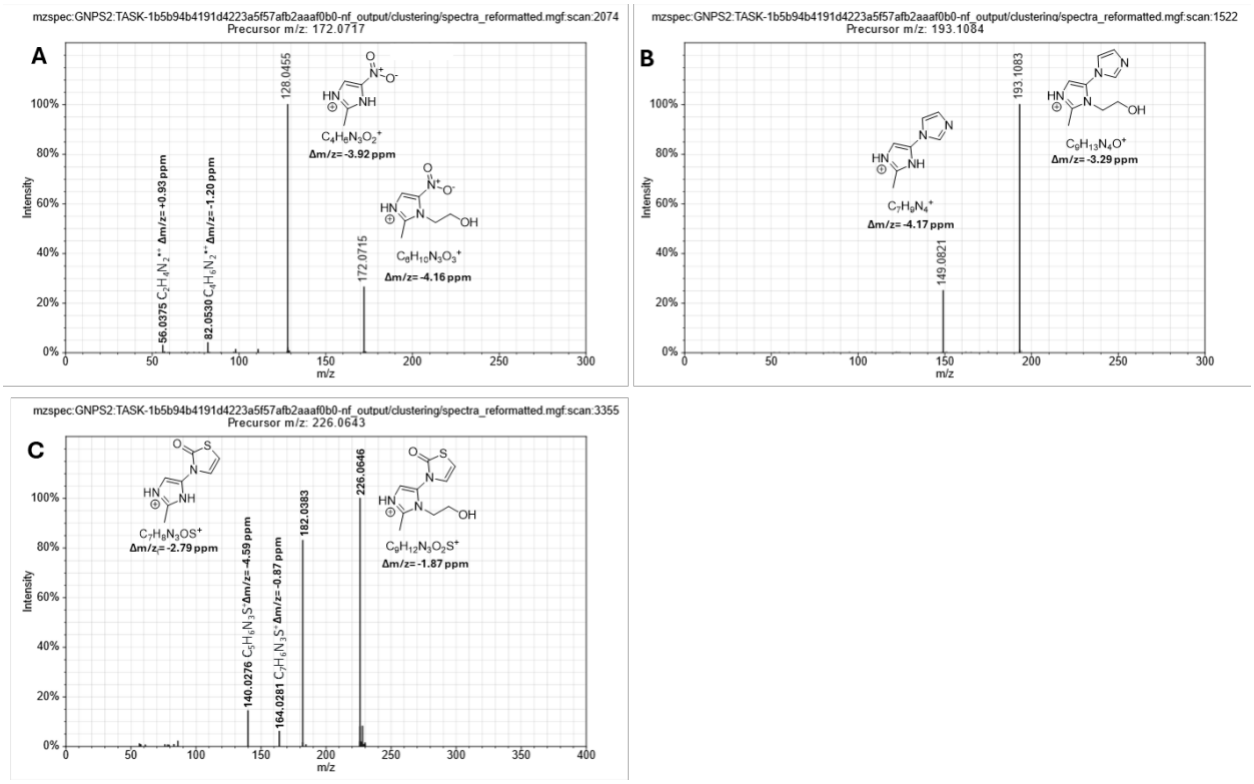


SI Figure 7. Fragmentation spectra and putative ion structure assignments for Cilnidipine and its putative microbial metabolites. A Cilnidipine feature, **B** Putative Cilnidipine metabolite resulting from nitroreduction to a hydroxylamine and *N*-methylation, **C** Putative Cilnidipine metabolite resulting from complete nitroreduction an *N*-methylation, **D** Putative Cilnidipine

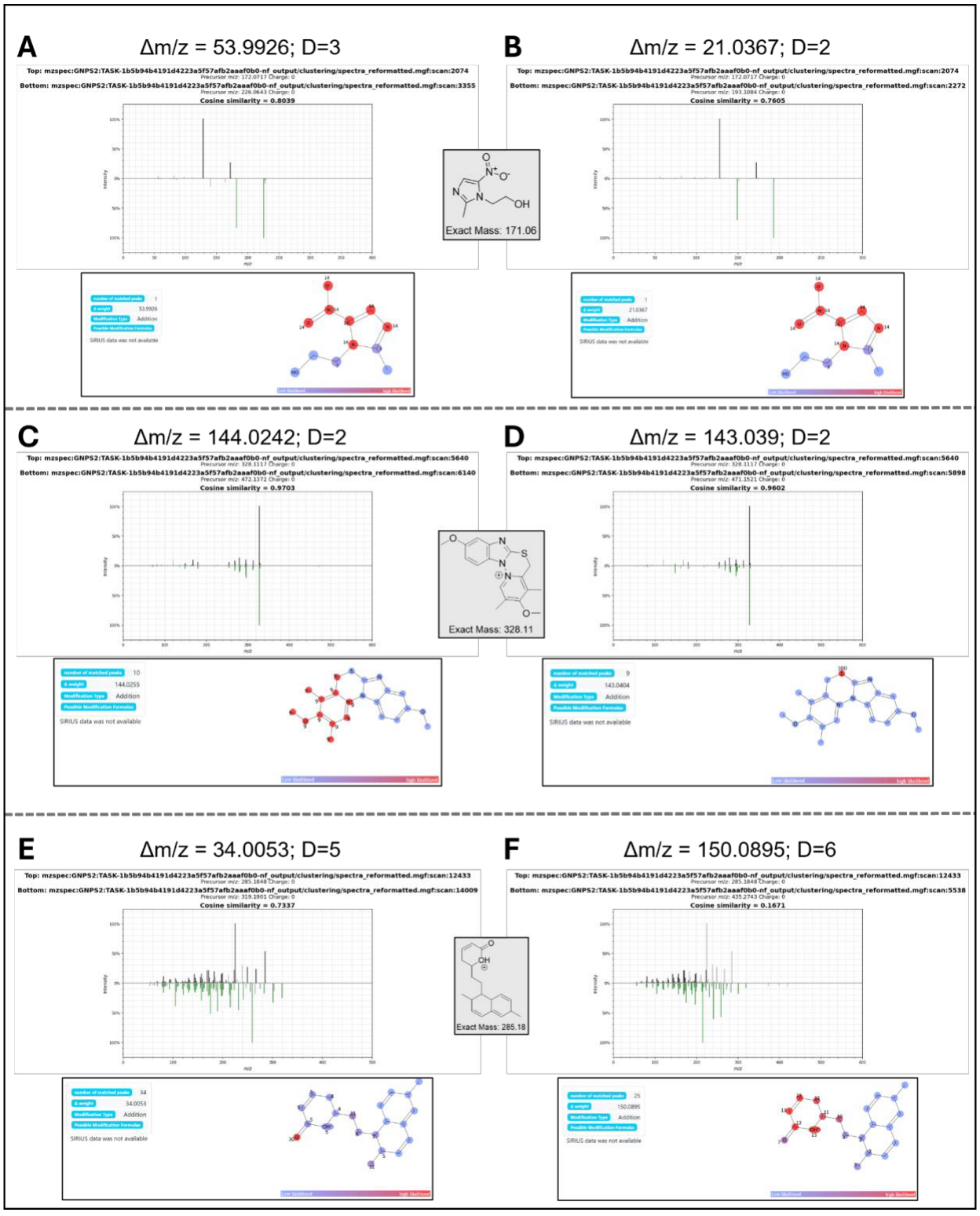
metabolite resulting from nitroreduction and formation of a pyrrole moiety. **E** Putative Cilnidipine metabolite resulting from cleavage of the nitrobenzene moiety. Δ*m/z* values correspond to theoretical values subtracted by experimental values. Ion formulas were determined by accurate mass measurements and are unambiguous within the constraints of the assigned precursor molecular formulas. During CID the Cilnidipine related ions undergo fragmentation mechanisms involving rearrangement, though the exact nature of these remains unknown and the structures proposed by us only represent one set of possible structures as do the precursor structures for **B**, **C**, and **D**.



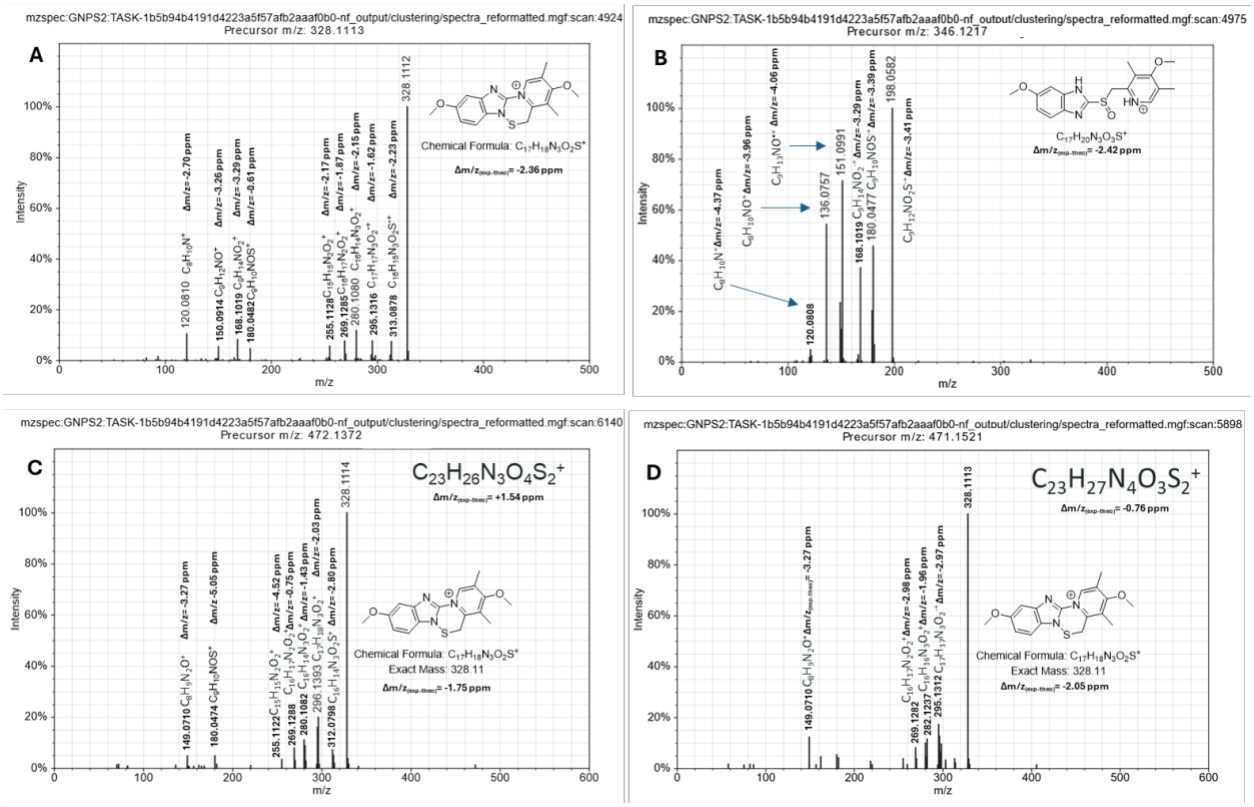
SI Figure 8. MS/MS spectral mirror matches and Modifinder predictions for ChemProp2-prioritized features of Cilnidipine. Panels A-B did not get any Modifinder predictions. The “D” values indicated in the panel titles denote the degree of neighbor within each drug cluster.



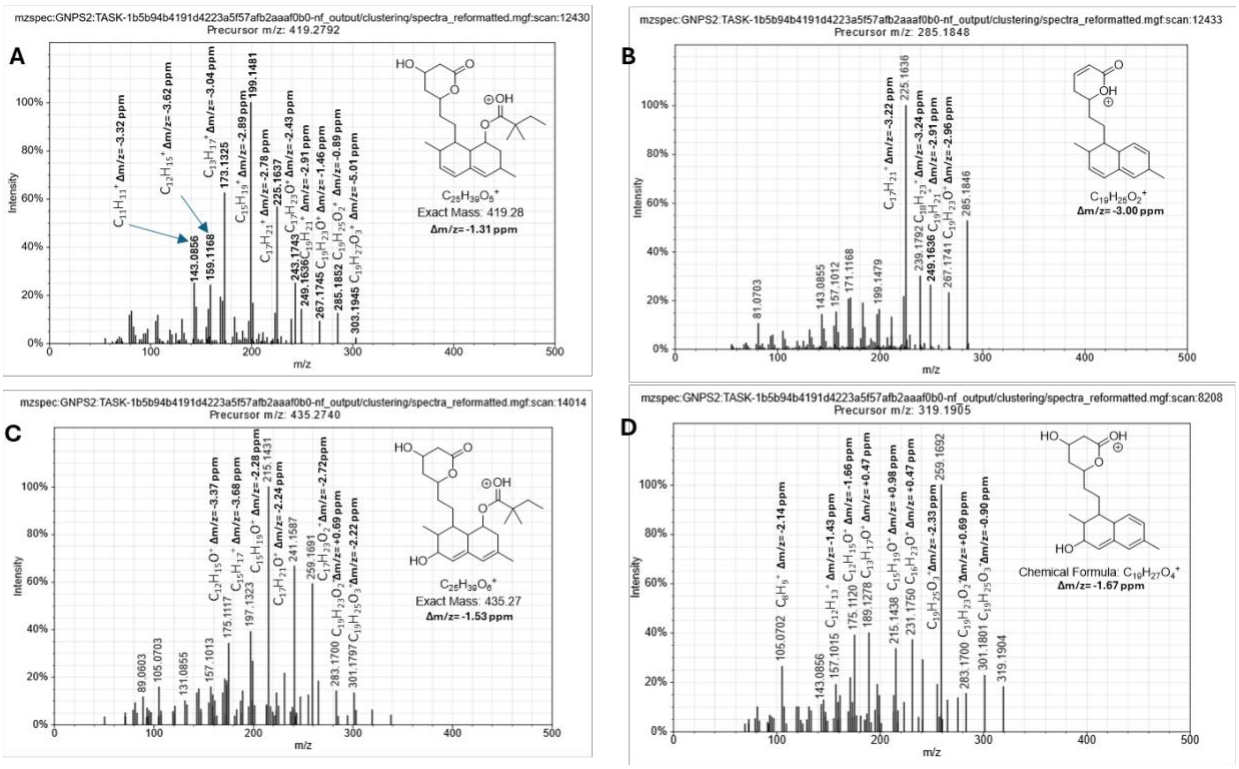
SI Figure 9. Fragmentation spectra and putative ion structure assignments for Metronidazole and its putative microbial metabolites. A Metronidazole feature, **B** Putative Metronidazole metabolite resulting from nitroreduction and potential formation of a imidazole group **C** Putative Metronidazole metabolite resulting from nitroreduction and thiazolidinone formation, Δ*m/z* values correspond to theoretical values subtracted by experimental values. Ion formulas were determined by accurate mass measurements and are unambiguous within the constraints of the assigned precursor molecular formulas. Molecular structures for fragment ions and potential microbial metabolite precursors are putative and cannot be confidently ascertained based on the current data.



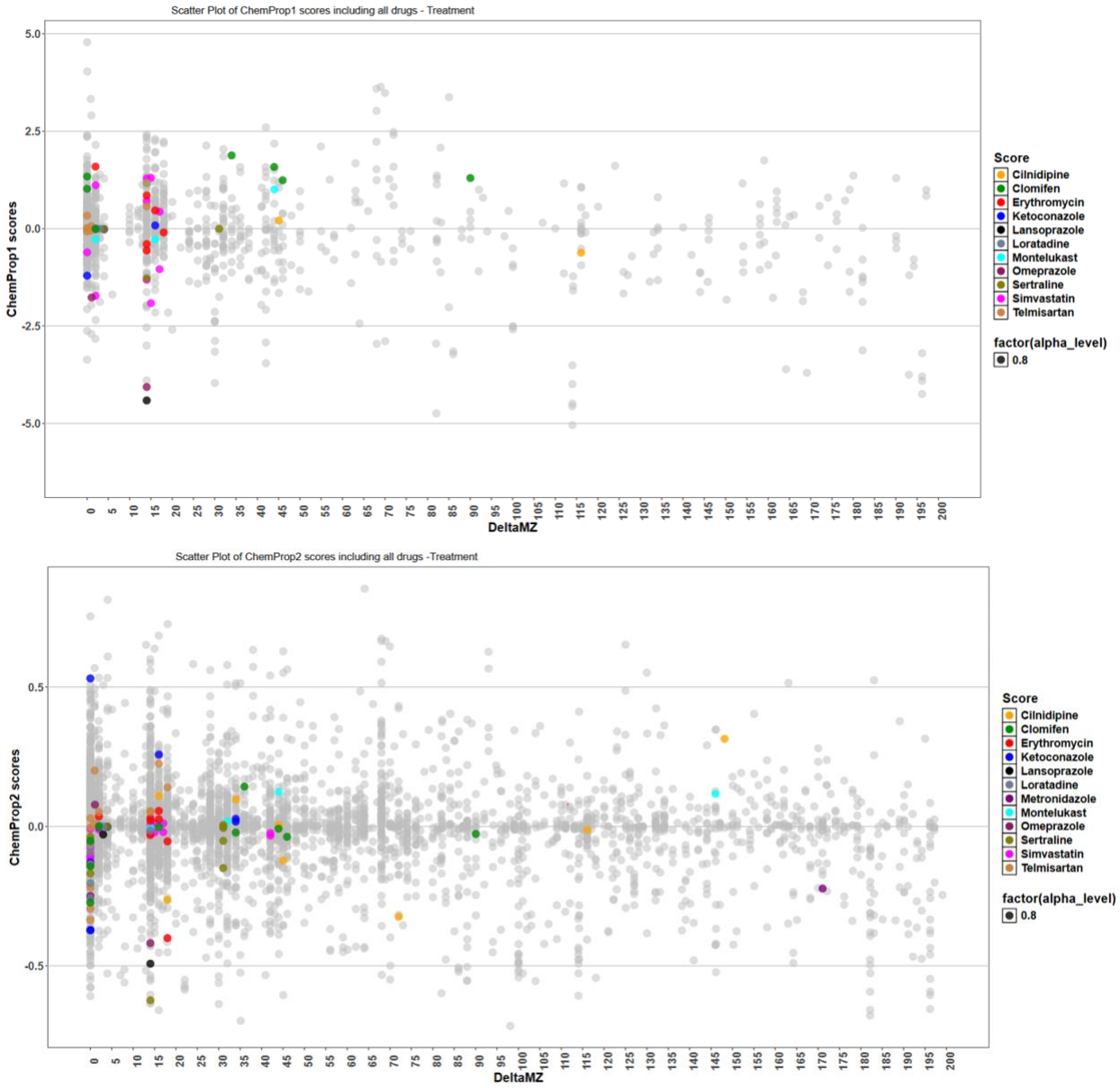
SI Figure 10. MS/MS spectral mirror matches and Modifinder predictions for ChemProp2-prioritized features. Panels A-B show metronidazole features; C-D show omeprazole (M+H-H2O); and E-F show simvastatin (ion-source fragment). The "D" values indicated in the panel titles denote the degree of neighbor within each drug cluster.



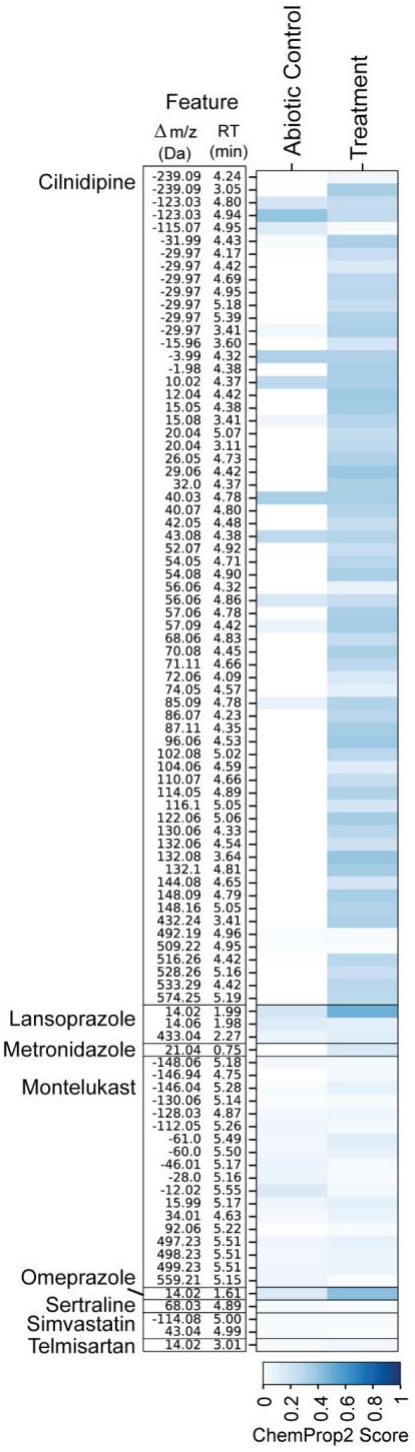
SI Figure 11. Fragmentation spectra and putative ion structure assignments for Omeprazole and its putative in-source fragment and microbial metabolites. A Omeprazole in-source fragment feature, **B** Omeprazole feature, **C** Putative Omeprazole metabolite resulting from addition of C_6H_6OS, **D** Putative Omeprazole metabolite resulting from addition of C_6H_7NS. *Δm/z* values correspond to theoretical values subtracted by experimental values. Ion formulas were determined by accurate mass measurements and are unambiguous within the constraints of the assigned precursor molecular formulas. Putative Omeparazole metabolites produce major fragment ions at *m/z* 328, matching the omeprazole in-source fragment and indicating neutral loss of all elements added during hypothetical biotransformation. During CID the Omeprazole related ions appear to undergo fragmentation mechanisms involving rearrangement, complicating evidence based assignment of fragment structures.



SI Figure 12: Fragmentation spectra and putative ion structure assignments for Simvastatin, its putative microbial metabolite and their in-source fragments. A Simvastatin feature, **B** Simvastatin in-source fragment feature, **C** Putative Simvastatin metabolite 3'-Hydroxysimvastatin **D** Putative in-source fragment of 3'-Hydroxysimvastatin. Δ*m/z* values correspond to theoretical values subtracted by experimental values. Ion formulas were determined by accurate mass measurements and are unambiguous within the constraints of the assigned precursor molecular formulas.



SI Figure 13. Scatterplot of ChemProp Scores for 12 Drugs. The plot shows ChemProp scores (y-axis) against Δm/z values (x-axis) for first-degree (D1) edges connected to each drug, with points colored by drug as indicated in the legend. The comparison between ChemProp1 and ChemProp2 was restricted to edges with Δm/z values between 0 and 200 Da.



SI Figure 14: ChemProp2-prioritized features unique to our dataset

Supplementary Tables

SI Table 1. Drug class abbreviations for the 50 compounds analyzed using ChemProp1 (corresponding to Figure 2). Classes are grouped into six major categories, and cell colors correspond to the drug classes shown in Figure 2.

Drug Class	Code	Abbreviation
Antibiotics & antiparasitics	AB	Antibiotic
	APAR	Antiparasitic
	ANTIDIAB	Antidiabetic
Antifungals	AF	Antifungal
Anti-inflammatory/Others	PPI	Proton Pump Inhibitors
	OTHER	Other / Miscellaneous Drugs
	HORM	hormonal
	NSAID	Nonsteroidal Anti-inflammatory Drugs
Cardiovascular	STATIN	statins
	CCB	calcium-channel blocker
	ANTICO	Anticoagulants
	ARB	Angiotensin II Receptor Blockers
CNS/ Psychotrophic	AD	Antidepressant
	SERM	Selective Estrogen Receptor Modulators
	AP	Antipsychotic
	SSRI	Selective Serotonin Reuptake Inhibitors
	NK1	Neurokinin-1 Receptor Antagonists
	SNRI	Serotonin–Norepinephrine Reuptake Inhibitors
Allergy/Immune	AH	Antihistamines
	LTRA	Leukotriene Receptor Antagonists

SI Table 2: Drug-specific summary of cascade composition, microbial diversity, biomass changes, and Bray-Curtis PCoA shifts.

Drug Name	Nodes per drug	Nodes (m/z > 0.5)	Library Matched	Unmatched Compounds	MassIVE Matches	ChemProp2 Hits (>0.1)	Final Hits	Microbiome α (mean)	Microbiome α (SD)	Microbiome change in α	ΔOD	OD mean	OD SD	Microbiome change in β (PCo1)	Microbiome change in β (PCo2)	Metabolome change in β (PCo1)	Metabolome change in β (PCo2)
Cilnidipine	98	98	0	98	20	76	11	1.12	0.20	0.48	0.87	0.80	0.35	0.91	0.28	0.71	0.11
Clomifen	42	36	1	37	34	9	9	0.64	0.39	-0.12	0.11	0.25	0.06	0.09	0.21	0.32	0.19
Erythromycin	64	59	12	48	60	9	30	1.46	0.15	-0.03	0.00	0.21	0.01	0.05	0.14	0.26	0.15
Ketoconazole	30	20	0	20	13	8	2	0.79	0.15	0.00	1.02	0.72	0.41	1.05	0.32	0.44	0.11
Lansoprazole	5	5	0	5	0	3	0	1.07	0.17	0.10	1.06	0.86	0.45	1.10	0.47	0.79	0.13
Loratadine	14	5	1	4	2	0	1	1.07	0.23	0.31	0.73	0.63	0.29	0.99	0.12	0.47	0.09
Metronidazole	18	6	0	6	3	3	2	1.46	0.41	-0.52	0.25	0.34	0.11	0.61	0.70	0.25	0.11
Montelukast	70	64	0	64	27	10	10	1.39	0.26	0.64	0.41	0.45	0.19	0.74	0.30	0.72	0.11
Omeprazole	2	2	0	2	1	1	0	1.12	0.17	0.41	1.04	0.85	0.44	1.11	0.34	0.60	0.10
Sertraline	97	92	0	92	48	39	7	0.61	0.17	-0.23	0.43	0.40	0.18	0.51	0.28	0.40	0.09
Simvastatin	55	38	18	55	53	4	5	1.38	0.41	0.95	0.02	0.21	0.01	0.29	0.03	0.43	0.15
Telmisartan	20	7	0	8	7	3	1	1.07	0.15	0.05	0.20	0.30	0.09	0.46	0.36	0.35	0.10
Control								1.13	0.23	0.54	1.04	0.92	0.43	1.09	0.41	0.70	0.18

SI Table 3. Comparison of ChemProp1 and ChemProp2 D1 Edges Across Thresholds for Each Drug.

D1 represents direct connections (edges) to the parent drug node in the FBMN network, whereas all other connections beyond the first degree neighbor are referred to as global edges. Cell values are color-coded from red to white to blue, indicating high, medium, and low values, respectively.

Drug	D=1 Edges	CHEMPROP 1	CHEMPROP2		
		Edges >= 1	Edges != 0	Edges > ±0.1	Edges > ±0.3
Cilnidipine	10	**0**	10	**5**	2
Clomifen	10	**6**	10	**3**	0
Erythromycin	8	**1**	8	**1**	1
Ketoconazole	10	**1**	9	**5**	3
Lansoprazole	4	**1**	3	**2**	1
Loratadine	10	**0**	8	**3**	0
Metronidazole	6	**0**	2	**2**	0
Montelukast	10	**2**	10	**2**	0
Omeprazole	2	**2**	2	**1**	1
Sertraline	10	**1**	10	**3**	1
Simvastatin	19	**7**	19	**2**	0
Telmisartan	16	**1**	15	**7**	1
Global	10982	**287**	5922	**1833**	519
Total	11097	**309**	6028	**1869**	529

SI Table 4. Summary of FASST results for ChemProp2-prioritized features.

Columns show the number of features queried per drug, features with at least one match, and those without matches. Also reported are the counts of unique datasets matched across all repositories, as well as subsets limited to MassIVE and external MassIVE datasets (excluding our six deposited repositories). “All Unique Datasets Matched” values are provided per drug but not summed in the summary row.

Drug	Features FASST input	Features with ≥ 1 Hit	Features without Hits	All Unique datasets Matched	Unique MASSIVE datasets	Unique External MassIVE Datasets
Cilnidipine	309	289	20	363	332	327
Clomifen	47	47	0	194	175	170
Erythromycin	113	113	0	221	119	194
Ketoconazole	39	37	2	52	50	47
Lansoprazole	102	101	1	290	257	251
Loratadine	16	16	0	75	66	63
Metronidazole	27	24	3	1144	1014	1010
Montelukast	83	78	5	456	400	395
Omeprazole	101	100	1	491	442	436
Sertraline	98	98	0	487	413	409
Simvastatin	245	141	104	829	772	766
Telmisartan	22	19	3	68	57	54
Total	1202	1063	139	n.c. (not calculated)	1670	1664

Supplementary Table 5. Summary of cascade node counts after ChemProp2 filtering. Counts of external and internal nodes per drug, with corresponding ChemProp2 treatment scores. External refers to features detected in external datasets, whereas Internal nodes were found only in our six in-house datasets.

	Cascade Nodes per drug	Nodes (Δm/z > 0.5)	Nodes After ChemProp2 Filtering (External)	Nodes After ChemProp2 Filtering (Internal)
Cilnidipine	98	98	11	65
Clomifen	42	36	9	0
Erythromycin	64	59	30	0
Ketoconazole	30	20	2	0
Lansoprazole	5	5	0	3
Loratadine	14	5	1	0
Metronidazole	18	6	2	1
Montelukast	70	64	10	18
Omeprazole	2	2	0	1
Sertraline	97	92	7	1

Simvastatin	55	38	5	2
Telmisartan	20	7	1	1
TOTAL	**515**	**432**	**78**	**92**

SI Table 6. Summary of $[M+H]^+$ Ions in the Feature-Based Molecular Network of the 12 Drugs.

The table lists, for each drug, the precursor m/z, retention time (min), unique feature ID, component index (CI; representing the subnetwork ID within FBMN), and the number of nodes in each subnetwork. For Simvastatin, two features at m/z 419.2791 and 419.2792, both corresponding to the $[M+H]^+$ ion, were retained, as the compound eluted as two peaks (minor and major) at 4.75 and 5.01 min, respectively. Similarly, for Telmisartan, two features were included: the parent ion at m/z 515.2443 ($[M+H]^+$) and a fragment at m/z 258.1256 (library-annotated as 'Telmisartan'), both co-eluting at the same retention time.

Drug	Precursor m/z	RT (mins)	ID	CI	Nodes
Cilnidipine	493.1967	4.95	12221	5	98
Clomifen	406.1933	3.64	9988	129	42
Erythromycin	734.4685	2.39	6524	239	64
Ketoconazole	531.1563	2.63	7741	93	30
Lansoprazole	370.083	2.28	6063	63	5
Loratadine	383.1522	3.08	8893	373	14
Metronidazole	172.0717	0.58	2074	261	18
Montelukast	586.2179	5.14	12848	188	70
Omeprazole	346.1217	1.78	4975	1278	2
Sertraline	306.0812	2.83	8073	2	97
Simvastatin	419.2791, 419.2792	4.75, 5.01	11549, 12430	295	55
Telmisartan	515.2443, 258.1256	2.84	8143, 8175	208, 263	13, 7

Chapter 4: Supplementary Information

Supplementary Figures

A

Target Dataframe (2734, 24)

feature_ID	Spiked_1	Spiked_10	Spiked_11	Spiked_12	Spiked_13	Spiked_14	Spiked_15	Spiked_16	Spiked_17	Spiked_18	Spiked_19	Spiked_2
Flavobacterium_	2.5276	3.0839	3.0095	2.8215	3.0009	3.0004	2.9375	3.1261	3.2758	3.1878	3.2281	2.7731
Frigoribacterium	1.699	2.2923	2.2529	2.1399	2.3784	2.3784	2.3243	2.3692	2.5551	2.5257	2.3802	1.9191

B

Decoy Dataframe (2734, 24)

feature_ID	Spiked_1	Spiked_10	Spiked_11	Spiked_12	Spiked_13	Spiked_14	Spiked_15	Spiked_16	Spiked_17	Spiked_18	Spiked_19	Spiked_2
Flavobacterium_	0	3.1737	1.3617	1.2304	3.026	3.0004	2.9024	1.3617	1.1761	1.4472	1.9395	2.1529
Frigoribacterium	1.8808	2.2923	1.3802	2.1759	1.2553	1.1139	1.5798	0.301	3.1638	3.1878	2.382	0

Figure S1: Target and decoy dataframes for correlation analysis. (A) Target dataframe containing metabolite features and microbial species from the soil synthetic community ground-truth dataset. As this dataset is predefined, species are retained at the species level. (B) Decoy dataframe generated by randomly shuffling species abundances to create a null reference for false-discovery-rate estimation.

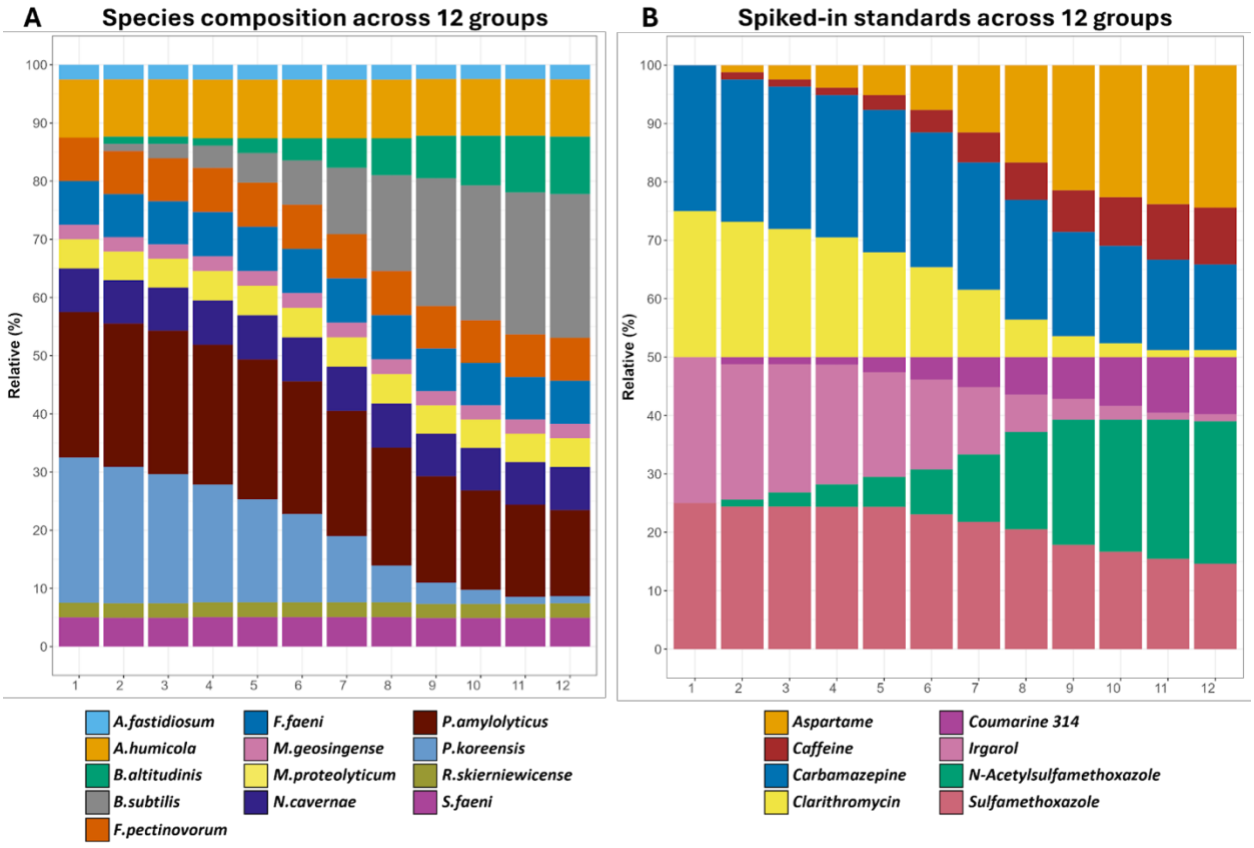


Figure S2: Designed dataset composition. (A) Relative abundance of the 13 microbial species across the 12 designed SynCom combinations. (B) Designed concentration profiles of the eight spiked-in metabolites across the same combinations.

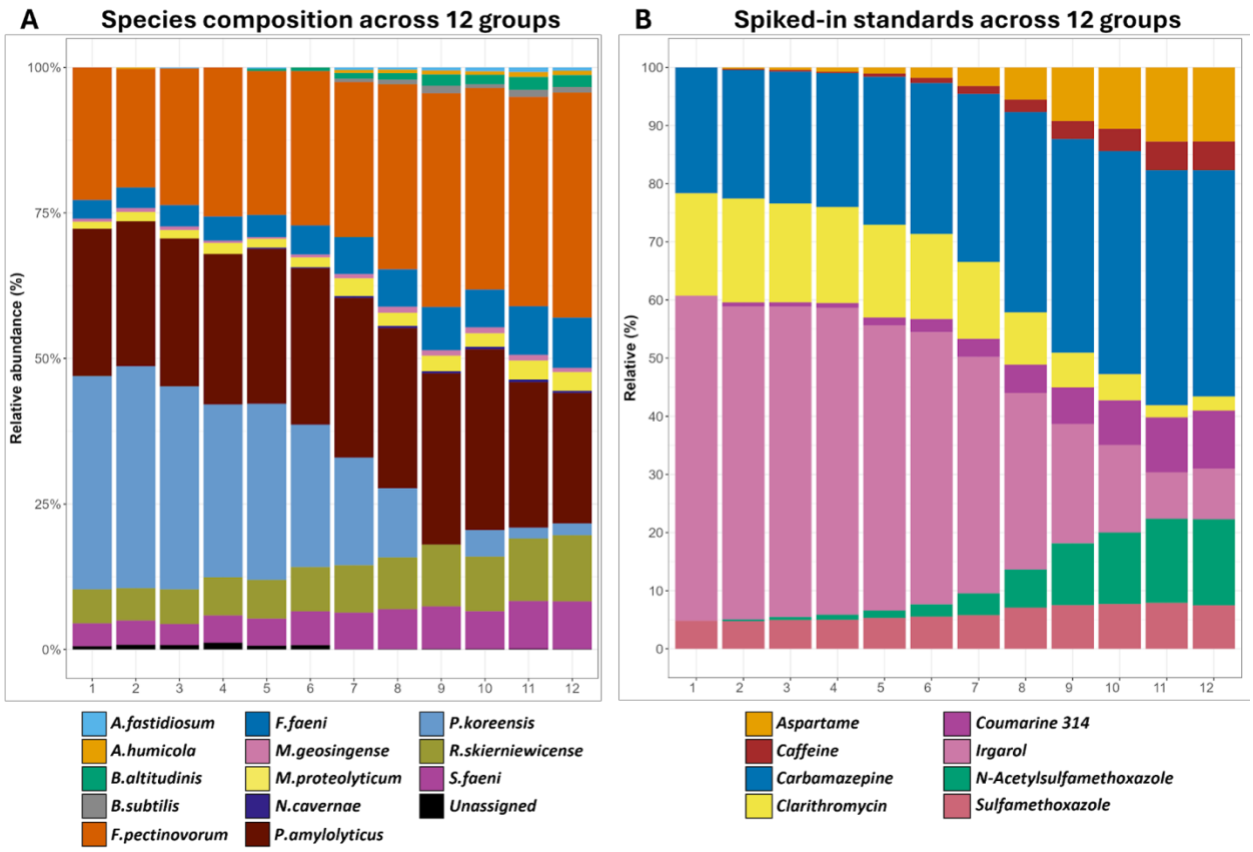


Figure S3: Measured dataset composition. (A) Relative abundance of the 13 microbial species across the 12 SynCom combinations. (B) The concentration profiles of the eight spiked-in metabolites across the same combinations.

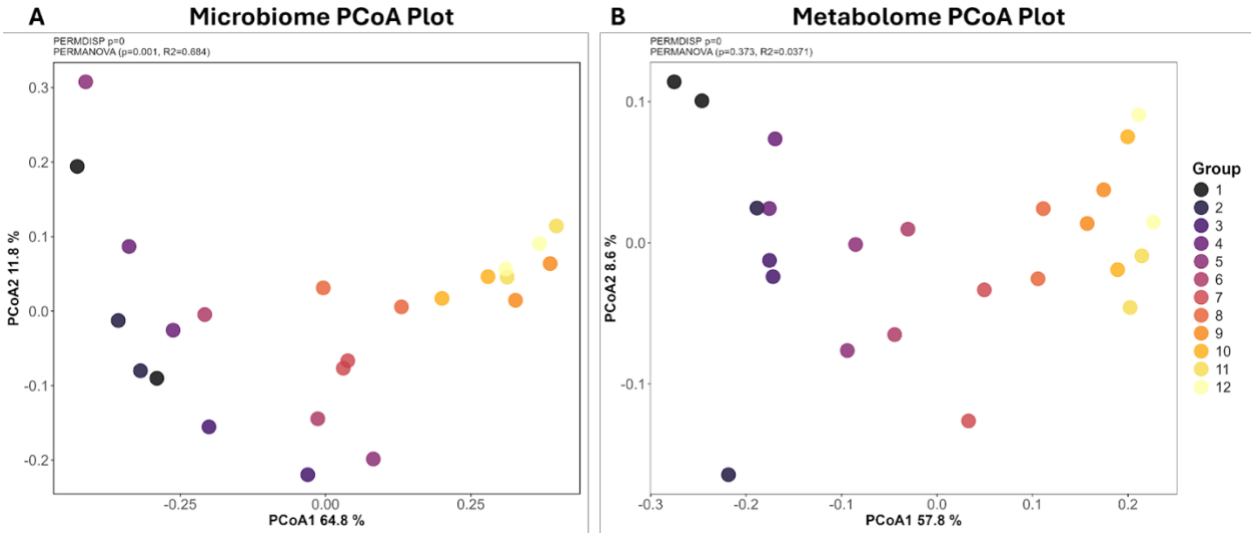


Figure S4: Principal Coordinate Analysis (PCoA) of measured datasets. PCoA was performed using Bray-Curtis distance for (A) the microbiome and (B) the metabolome datasets. The microbiome ordination was constructed from 14 species (13 defined plus one unassigned). The metabolome

ordination was based on 2,612 detected features, including the 8 spiked-in metabolites. Group labels (1-12) indicate the designed SynCom combinations, each shown with two replicate points. PERMANOVA was performed using numeric group values, and PERMDISP = 0 for both datasets.

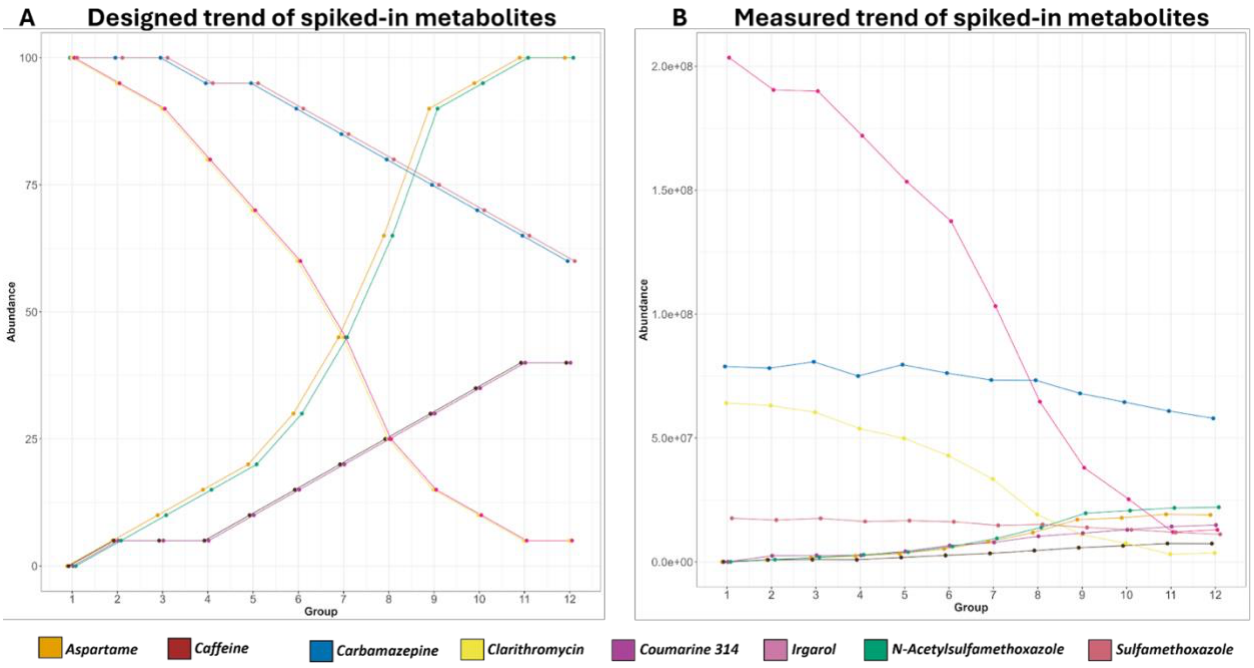


Figure S5: Comparison of designed and measured intensity trends for spiked-in metabolites. Line plots show the (A) intended and (B) experimentally measured intensity patterns of the eight spiked-in metabolites across the 12 SynCom combinations.

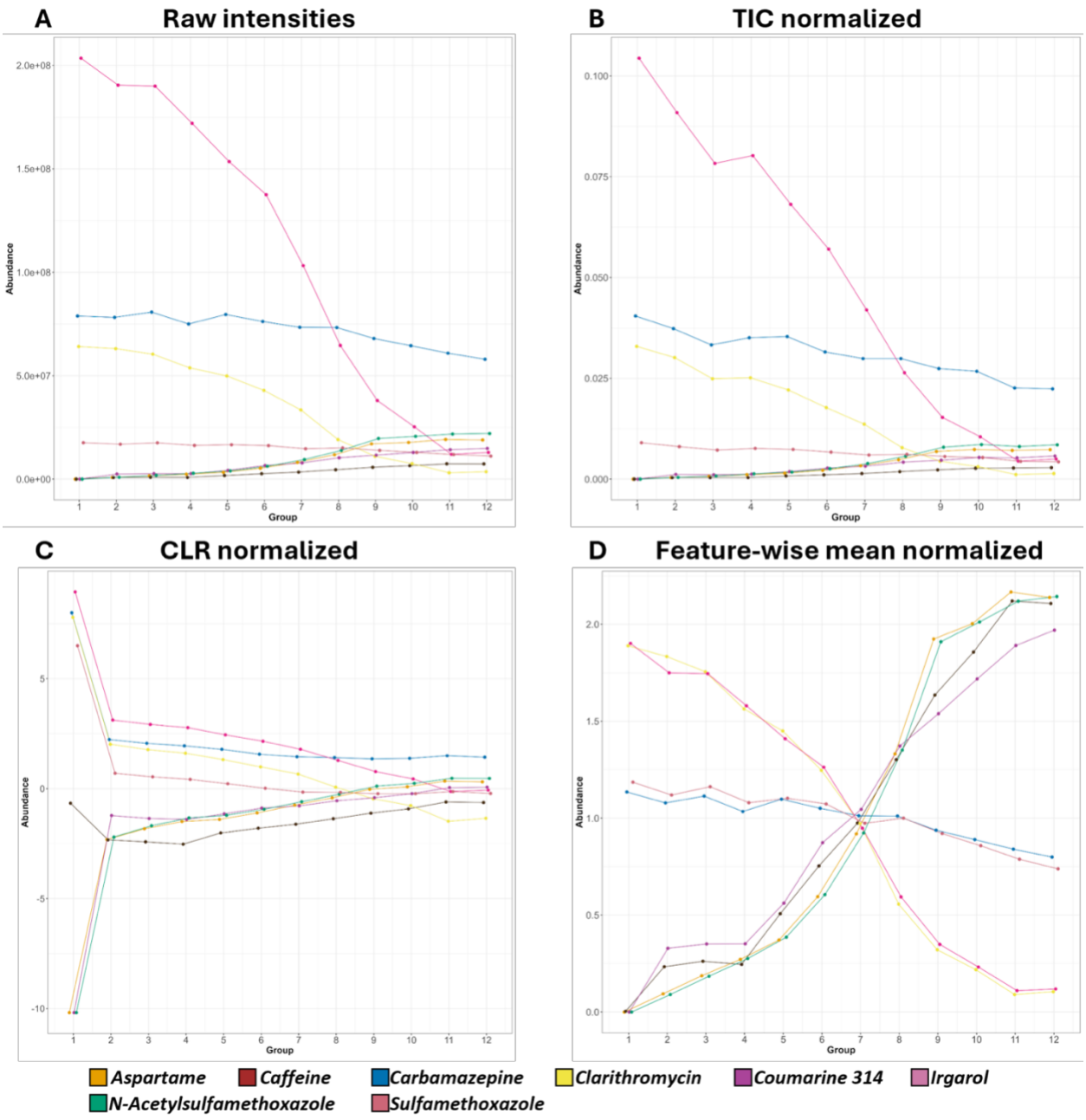


Figure S6: Normalization effects on metabolite trends. Line plots of the eight spiked-in metabolites across groups 1-12 under raw, total-ion-count (TIC) normalization (normalized to each sample), centered-log-ratio (CLR) normalization, and feature-mean scaling (normalized to each feature) methods.

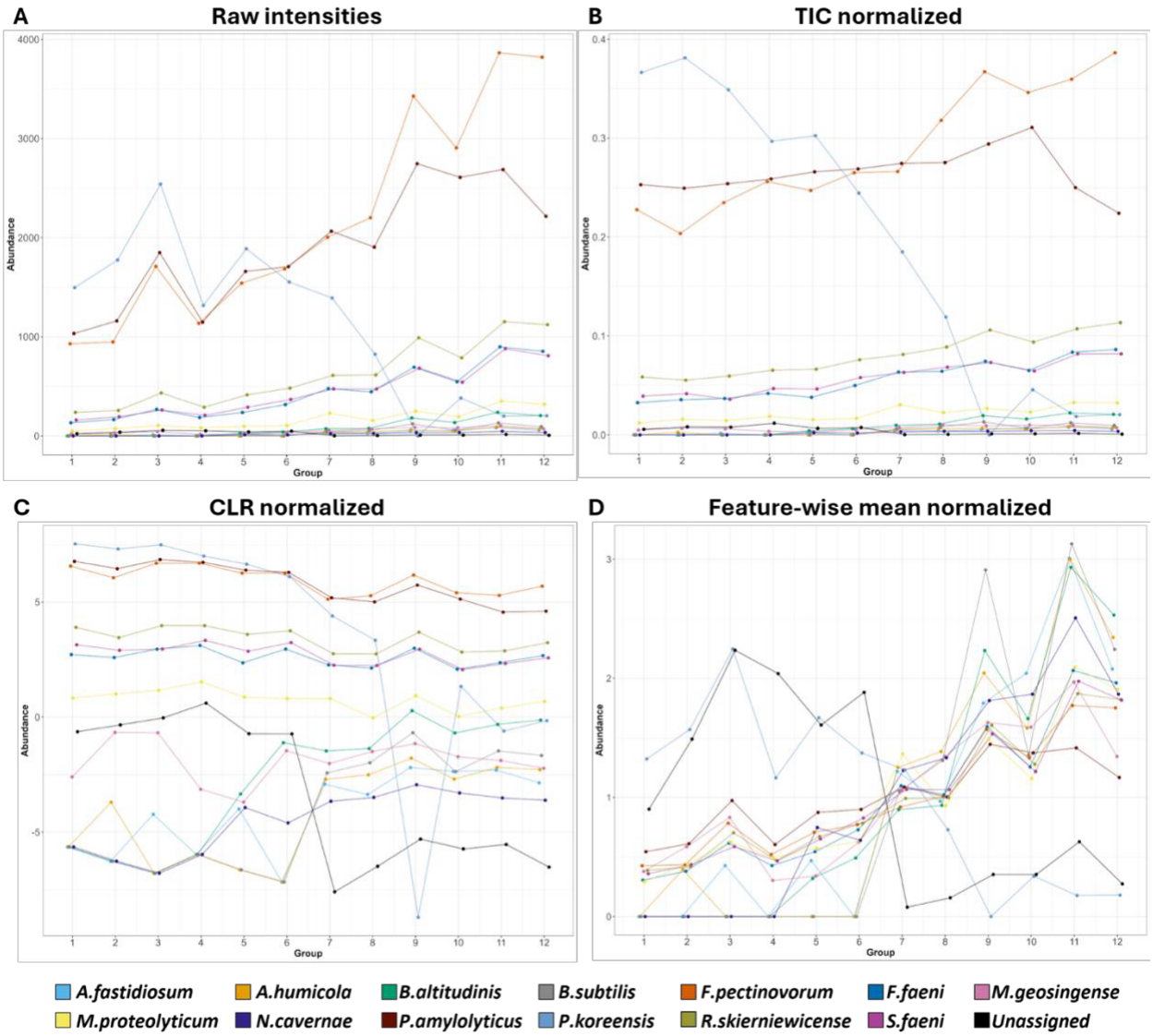


Figure S7: Normalization effects on Amplicon Sequencing Variant (ASV) trends. Line plots of representative taxa across groups 1-12 under raw, total-ion-count (TIC) normalization (relative abundance), centered-log-ratio (CLR) normalization, and feature-wise mean normalization (normalized to each feature) methods.

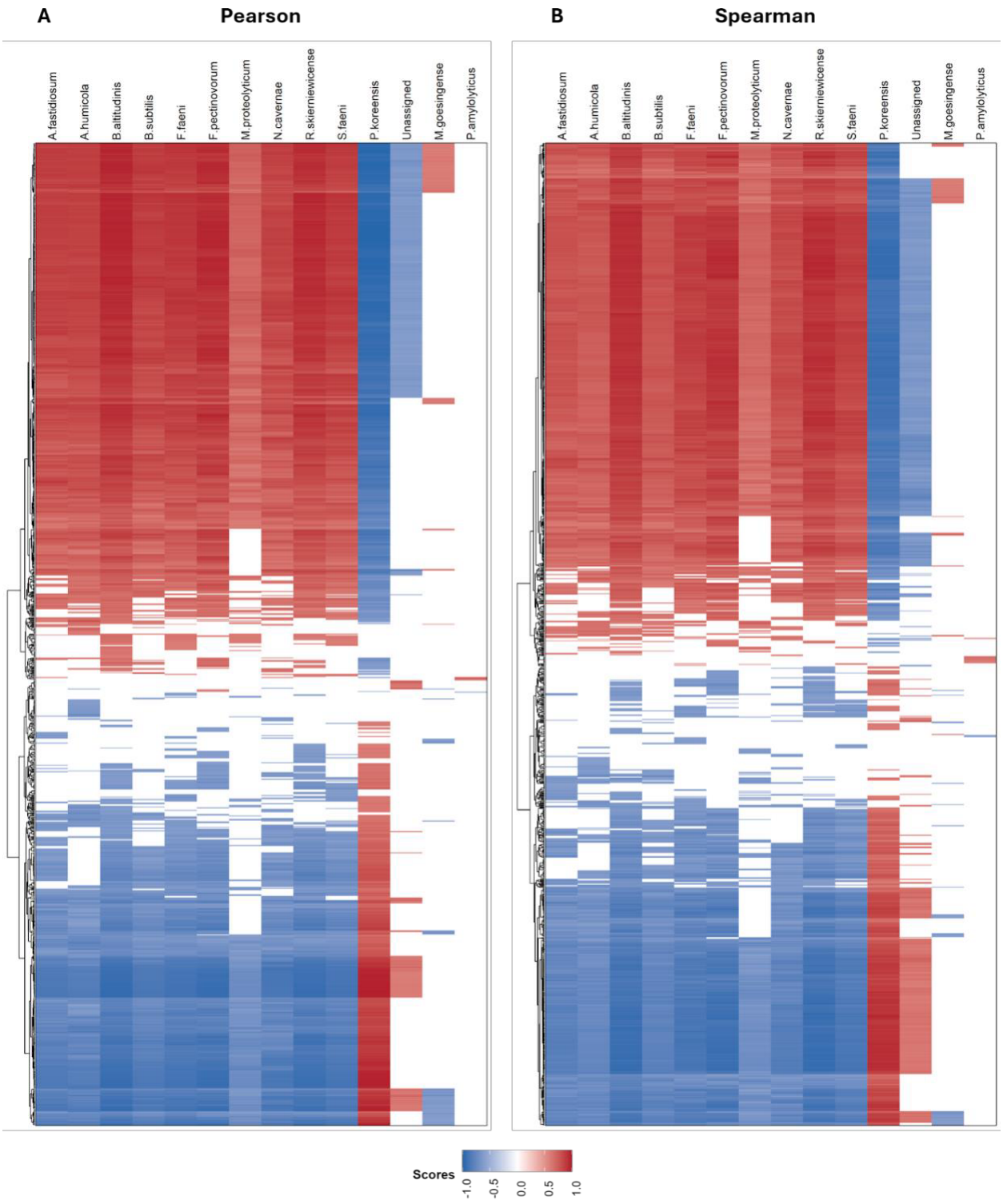


Figure S8: Overall correlation heatmaps of non-spiked-in features across all species for both (A) Pearson and (B) Spearman methods (1% FDR-adjusted scores). Species are shown as columns and features as rows, with hierarchical clustering applied to features. The color scale (-1 to +1) denotes correlation strength, revealing distinct clusters of positive (red) and negative (blue) associations. The Spearman heatmap

displays a higher density of significant correlations, particularly within the *unassigned*, *P. amylolyticus*, and *M. goesingense* groups.

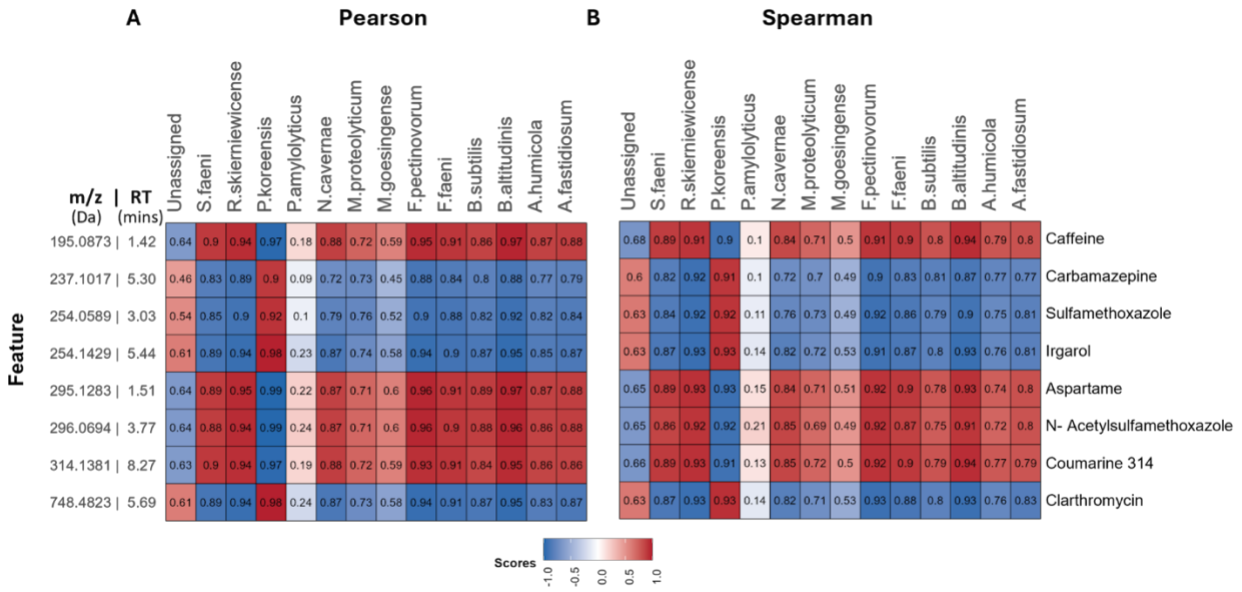
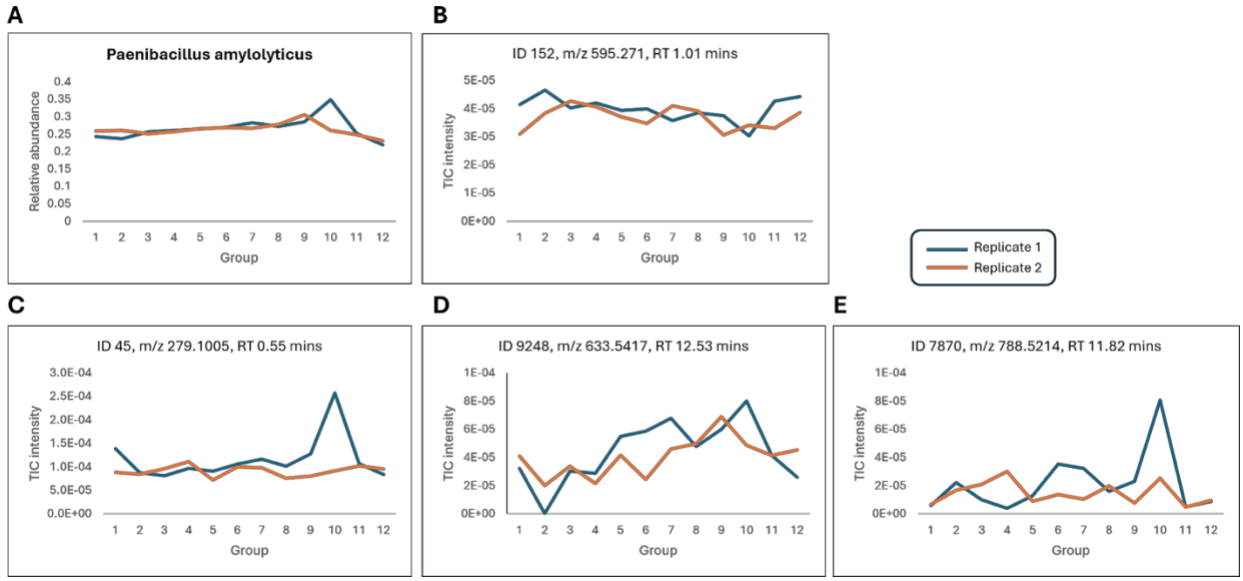


Figure S9: Correlation scores Comparison. Comparison of (A) Pearson and (B) Spearman original correlation scores (not FDR- corrected) for the eight spiked-in metabolites across all species. Both methods show highly similar correlation patterns.



SI Figure S10. ***Abundance trajectories of Paenibacillus and four metabolite features across 12 experimental groups (two replicates).*** Features C–E track the same rise-drop pattern as *Paenibacillus* (r = 0.6–0.7), while Feature B shows an inverse trend (r ≈ –0.6).

Supplementary Note

Detailed comparison of designed and measured compositions

In the designed microbiome dataset (**Figure S1**), nine of the thirteen species were kept constant, while four displayed directional trends (both *Bacillus* strains increasing; *P.koreensis* and *P.amylolyticus* decreasing). However, in the measured dataset (**Figure S2**), the observed patterns differed. Based on raw ASV counts, *F.pectinovorum* (designed to remain constant) showed an unexpected increasing trend across the twelve groups in the measured dataset. Similarly, *P.amylolyticus* also exhibited an increasing pattern. Several strains exhibited low or undetectable counts in the nanopore 16S dataset, including *A.fastidiosum*, *A.humicola*, *B.altitudinis*, *B.subtilis*, and *N.cavernae*, which were absent in early groups (1-6) despite being included in the design. Several species, including the two *Bacillus* strains, showed markedly reduced counts. Among the eight spiked-in metabolites, four were designed to increase (Caffeine, Clarithromycin, Coumarine, N-Acetylsulfamethoxazole) and four to decrease in concentration (Aspartame, Carbamazepine, Irgarol, Sulfamethoxazole). While these differences were partly visible in the stacked bar plots, they became more apparent in the line plots (**Figure S5**), where the measured intensities of the increasing compounds rose only modestly (0-5 %) compared to their designed values. Conversely, the decreasing compounds showed steeper drops, such as irgarol declining from 100 % to approximately 5 %. Thus, while the general directionality of the designed metabolite trends was captured, the dynamic range was compressed in the experimental data

Author Contributions

Thesis Chapters

Chapter 2 presented in this thesis is a peer-reviewed publication:

Statistical analysis of feature-based molecular networking results from non-targeted metabolomics data. *Nature Protocols* **20**, pages 92–162 (2025). https://doi.org/10.1038/s41596-024-01046-3

Abzer K. Pakkir Shah, Axel Walter, Filip Ottosson, Francesco Russo, Marcelo Navarro-Diaz, Judith Boldt, Jarmo-Charles J. Kalinski, Eftychia Eva Kontou, James Elofson, Alexandros Polyzois, Carolina González-Marín, Shane Farrell, Marie R. Aggerbeck, Thapanee Pruksatrakul, Nathan Chan, Yunshu Wang, Magdalena Pöchhacker, Corinna Brungs, Beatriz Cámara, Andrés Mauricio Caraballo-Rodríguez, Andres Cumsille, Fernanda de Oliveira, Kai Dührkop, Yasin El Abiead, Christian Geibel, Lana G. Graves, Martin Hansen, Steffen Heuckeroth, Simon Knoblauch, Anastasiia Kostenko, Mirte C. M. Kuijpers, Kevin Mildau, Stilianos Papadopoulos Lambidis, Paulo Wender Portal Gomes, Tilman Schramm, Karoline Steuer-Lodd, Paolo Stincone, Sibgha Tayyab, Giovanni Andrea Vitale, Berenike C. Wagner, Shipei Xing, Marquis T. Yazzie, Simone Zuffa, Martinus de Kruijff, Christine Beemelmanns, Hannes Link, Christoph Mayer, Justin J.J. van der Hooft, Tito Damiani, Tomáš Pluskal, Pieter Dorrestein, Jan Stanstrup, Robin Schmid, Mingxun Wang, Allegra Aron, Madeleine Ernst, Daniel Petras.

Personal Contribution

Together with Daniel Petras (main supervisor for the project and thesis, main corresponding author), I conceptualized the overall workflow in collaboration with Filip Ottosson, Francesco Russo, and Madeleine Ernst (co-corresponding author). I wrote substantial portions of the R and Python code and worked alongside a broad team of contributors who developed different parts of the software. Axel Walter, Filip Ottosson, Francesco Russo, Marcelo Navarro-Diaz, Judith Boldt, Jarmo-Charles J. Kalinski, Eftychia Eva Kontou, Carolina González-Marín, Nathan Chan, Yunshu Wang, Martinus de Kruijff, Jan Stanstrup, Kevin Mildau, and James Elofson contributed to code development. I also co-developed and deployed the Streamlit web application with Axel Walter and Mingxun Wang.

I designed key components of the statistical framework, integrated the molecular networking results with downstream analyses, and refined the pipeline. I coordinated communication among contributors and ensured consistent implementation of analytical decisions across the workflow. I tested all code components, including the R, Python and QIIME notebooks and the web application, and validated the workflow on multiple datasets together with Stilianos Papadopoulos Lambidis, Tilman Schramm, Karoline Steuer-Lodd, Paolo Stincone, Sibgha Tayyab, Giovanni Andrea Vitale, Berenike C. Wagner, and Christian Geibel.

I wrote major sections of the manuscript and the supplementary material and generated most of the figures. I collaborated with co-authors on figure interpretation. Corinna Brungs, Andrés Mauricio Caraballo-Rodríguez, Andres Cumsille, Fernanda de Oliveira, Kai Dührkop, Yasin El Abiead, Lana G. Graves, Martin Hansen, Steffen Heuckeroth, Simone Zuffa, Simon Knoblauch, Anastasiia Kostenko, Paulo Wender Portal Gomes, Shipei Xing, Marquis T. Yazzie, Alexandros Polyzois, Marie R. Aggerbeck, Thapanee Pruksatrakul, Magdalena Pöchhacker, Beatriz Cámara, and Tito Damiani contributed valuable information and methodological insight to both the manuscript and supplementary files. I incorporated suggestions, guidance, and testing feedback from group leaders and senior collaborators, including Christine Beemelmanns, Hannes Link, Christoph Mayer, Justin J.J. van der Hooft, Tomáš Pluskal, Pieter Dorrestein, Robin Schmid, Mingxun Wang, Allegra Aron, Madeleine Ernst, and Daniel Petras, whose expertise strengthened the final protocol. I also led the iterative revision process together with Daniel Petras, addressing reviewer and co-author feedback and ensuring consistency across all workflow implementations.

Chapter 3 presented in this thesis is based on the results of the following manuscript draft:

A Metabolomics Framework to Track Microbiome Drug Metabolism. Abzer K. Pakkir Shah, Anne Griesshammer, Paolo Stincone, Jarmo-Charles Kalinski, Axel Walter, Mingxun Wang, Lisa Maier, Daniel Petras.

A revised version of this chapter is available as a preprint: Pakkir Shah AK *et al. A Functional Metabolomics Framework to Track Microbiome Drug Metabolism.* bioRxiv (2026). doi: 10.64898/2026.01.30.702925.

Personal Contribution

I conceptualized the ChemProp software together with Daniel Petras (main supervisor for the project and thesis, main corresponding author). I co-conceptualized the incubation experiments of the gut synthetic community with clinically relevant drugs together with Anne Griesshammer, Lisa Maier, and Daniel Petras. Anne Griesshammer performed the microbial culturing and carried out the experiments in the anaerobic chamber. I collected the samples and performed the metabolite extractions together with Paolo Stincone for mass spectrometry analysis. Paolo Stincone conducted the untargeted LC-MS/MS measurements, while Anne Griesshammer and Lisa Maier carried out the amplicon sequencing. I developed the ChemProp application with the guidance of Axel Walter and Mingxun Wang. I analyzed the metabolomics and microbiome data together with Anne Griesshammer, Jarmo-Charles Kalinski, Lisa Maier, and Daniel Petras, and integrated these datasets into the combined ChemProp workflow. I wrote the manuscript, supplementary material and prepared all figures with guidance and suggestions from Daniel Petras, and all authors contributed to writing and editing the manuscript draft.

Chapter 4 presented in this thesis is based on the results of the following manuscript draft:

CorrOmics: An Interactive Web Tool for Correlating Multi-Omics Data *Manuscript in preparation to be submitted as a Technical Note.*

Abzer K. Pakkir Shah, Karoline Steuer-Lodd, Mingxun Wang, Daniel Petras.

Personal Contribution

I conceptualized the CorrOmics software together with Daniel Petras (main supervisor for the project and thesis, main corresponding author). I co-conceptualized the leaf synthetic community benchmarking experiments, where we added eight chemical standards in defined ratios to mimic correlation trends, together with Karoline Steuer-Lodd and Daniel Petras. Karoline Steuer-Lodd performed the SynCom experiments, conducted the LC-MS/MS measurements, and generated the Nanopore sequencing data. I developed the CorrOmics application with guidance from Mingxun Wang. I performed the data analysis, working together with Karoline Steuer-Lodd and Daniel Petras on data interpretation and benchmarking outcomes. I wrote the manuscript, prepared the supplementary material, and generated all figures for this chapter.

Other Publications

These publications are not part of the dissertation chapters but represent collaborative work to which I contributed during my PhD studies.

Peer-Reviewed Publications

Farrell, S.P., Petras, D., …, Pakkir Shah, A.K., …, Rasher, D.B. (2025). *Turf algae redefine the chemical landscape of temperate reefs, limiting kelp forest recovery.* **Science, 388**, 876–880. https://doi.org/10.1126/science.adt6788.

Contribution: Untargeted-metabolomics data analysis, assistance with figure preparation, visualization.

Vela-Corcia, D., Hierrezuelo, J., …, Pakkir Shah, A.K., …, Romero, D. (2024). *Cyclo(Pro-Tyr) elicits conserved cellular damage in fungi by targeting the [H+]ATPase Pma1 in plasma membrane domains.* **Communications Biology, 7**, 1253. https://doi.org/10.1038/s42003-024-06947-3

Contribution: LC-MS/MS data analysis.

Stincone, P., Pakkir Shah, A.K., …, Petras, D. (2023). *Evaluation of Data-Dependent MS/MS Acquisition Parameters for Non-Targeted Metabolomics and Molecular Networking of Environmental Samples: Focus on the Q Exactive Platform.* **Analytical Chemistry, 95(34)**, 12673–12682. https://doi.org/10.1021/acs.analchem.3c01202.

Contribution: Helped with sample collection and extraction, LC-MS/MS data acquisition and data analysis.

Publications under review:

Mannochio-Russo, H., Nunes, W.D.G., …, Pakkir Shah, A.K., …, Dorrestein, P.C. (2025). *Bridging Complexity and Accessibility in Metabolomics with MetaboApps.* **ChemRxiv.** https://doi.org/10.26434/chemrxiv-2025-3nq29. *This is a preprints and has not been peer-reviewed.*

Contribution: Development of some of the MetaboApps; all authors tested the GNPS2 web applications.

Pérez-Lorente, A.I., …, Pakkir Shah, A.K., …, Romero, D. (2025). *The offensive role of the Bacillus extracellular matrix in driving metabolite-mediated dialogue and adaptive strategies with pathogenic fungi.* **bioRxiv.** https://doi.org/10.1101/2025.02.06.636830. *(Accepted ISME J)*

Contribution: Applied ChemProp2 analysis to identify degraded fengycin products

Iliakopoulou, S., Triantis, T.M., …, Pakkir Shah, A.K., …, Kaloudis, T. (2025). *Elucidating Transformation Pathways of Microcystins during Advanced Oxidation/Reduction Processes for Water Treatment.* **ChemRxiv.** https://doi.org/10.26434/chemrxiv-2025-q4xh3**.** *This is a preprint and has not been peer-reviewed.*

Contribution: Helped with untargeted-metabolomics data analysis, ChemProp2 prioritization of degraded transformation products.

Petras, D., Torres, R., Pakkir Shah, A.K., …, Lihini Aluwihare (2025). *Metabolome diversity and oxidation state are linked to microbial networks in the surface ocean.* (To be submitted)

Contribution: Performed metabolomics and microbiome data processing, integrated the metadata, assisted with data analysis, and generated the figures.

I confirm that the above-stated is correct.

December 7th, 2025, Abzer Kelminal Pakkir Mohamed Shah
Date, Signature of the candidate

I certify that the above-stated is correct.

December 6th, 2025, Daniel Petras
Date, Signature of the supervisor