Deep Learning Frameworks for Multi-omics Analyses of the Microbiome in Disease Studies

BY

DEREK REIMAN
B.Sc. – Computer Science, University of North Texas, 2010

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Bioinformatics
in the Graduate College of the
University of Illinois at Chicago, 2021

Chicago, Illinois

Defense Committee:

Yang Dai (Chair and Advisor), Department of Biomedical Engineering
Jie Liang, Department of Biomedical Engineering
Ao Ma, Department of Biomedical Engineering
Brian Layden, College of Medicine, Department of Endocrinology/Diabetes
Aly Khan, University of Chicago, Department of Pathology

This thesis is dedicated to my wife, Sarvnaz, to my parents, Michael and Valerie, and to my younger siblings, Chloe and Brandon, for their unwavering support.

ACKNOWLEDGEMENTS

I would like to thank my thesis committee – Drs. Yang Dai, Jie Liang, Ao Ma, Brian Layden, and Aly Khan – for their unwavering support. I would like to especially thank Dr. Yang Dai for taking me on as a student. Her guidance and encouragement pushed me to accomplish my research goals and enjoy myself in the process. Lastly, I would like to further thank Dr. Aly Khan, who has become a close friend and mentor, and without whose support my path would have been much more difficult.

DJR

CONTRIBUTION OF AUTHORS

A subsection of **Chapter 2** is taken from a book chapter of which I am the first author. I developed the pipeline for the tutorial outlined in the book. Ulises Sosa tested the pipeline functionality and he and I generated the figures. Yang Dai oversaw the project. Everyone helped write the chapter manuscript.

Chapter 3 is a collection of three manuscripts (two journal publications and one conference proceedings), each of which I am the first author. Yang Dai was responsible for overseeing the projects. Together, Yang Dai and I designed the methodologies and I implemented and evaluated them. I was responsible for the illustrations in the manuscripts. All authors helped write the manuscripts.

Chapter 4 is taken from a recently published paper that I am the first author on. Yang Dai was responsible for overseeing the project. Yang Dai and I designed the method and I implemented and evaluated it. I was responsible for the illustrations in the manuscripts. All authors helped write the manuscripts.

The first part of **Chapter 5** is taken from a conference paper that I am the first author on. Yang Dai was responsible for overseeing the project. Yang Dai and I designed the method and I implemented and evaluated it. I was responsible for the illustrations in the manuscripts. All authors helped write the manuscripts. The second part of **Chapter 5** is based on ongoing unpublished work.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (CONTINUED)

LIST OF FIGURES (CONTINUED)

LIST OF ABBREVIATIONS

AE	Autoencoder
ALR	Additive log-ratio
ANN	Artificial neural network
AUC / AUC-ROC	Area under the receiver operator characteristic curve
AUC-PR	Area under the precision-recall curve
CCA	Canonical correlation analysis
CD	Crohn's Disease
CDF	Cumulative distribution function
CGAN	Conditional generative adversarial network
CLR	Centered log-ratio
CNN	Convolutional neural network
CV	Cross-validation
GCNN	Graph convolutional neural network
DBN	Dynamic Bayesian Network
DL	Deep learning
DNN	Deep neural network
ELU	Exponential linear unit
FBA	Flux balance analysis
FMT	Fecal microbiome transplant
gLV	Generalized Lotka-Volterra
HFHS	High-fat, high-sugar
IBD	Inflammatory bowel disease
KLD	Kullback-Leibler divergence
LASSO	Least absolute shrinkage and selection operator
LCFA	Long chain fatty acid
LFPP	Low-fat, high-plant-polysaccharide
MAE	Mean absolute error
MCC	Matthew's correlation coefficient

LIST OF ABBREVIATIONS (CONTINUED)

MGS	Metagenome shotgun
ML	Machine learning
MLPNN	Multilayer perceptron neural network
MoB	Mode of birth
MSE	Mean squared error
NGS	Next generation sequencing
OTU	Operational taxonomic unit
PERMANOVA	Permutational multivariate analysis of variance
PCA	Principal component analysis
PCC	Pearson correlation coefficient
PCoA	Principal coordinate analysis
PRMT	Predicted relative metabolic turnover
RA	Relative abundance
ReLU	Rectified Linear Unit
RF	Random forest
rRNA	Ribosomal RNA
SCC	Spearman correlation coefficient
SCFA	Short chain fatty acid
SVM	Support vector machine
T2D	Type 2 diabetes
UC	Ulcerative Colitis
WGCNA	Whole genome correlation network analysis

SUMMARY

The microbiome plays a vital role in development and regulation of multiple physiological functions affecting human health. In particular, the dysbiosis of the gut microbiome has been linked to multiple metabolic diseases such as obesity, type 2 diabetes, and inflammatory bowel disease. Over the last decade, sequencing technologies have allowed us to characterize the microbiome community at increasing levels of resolution. Despite advances in the characterization of the microbiome community, the mechanisms by which the microbiome community interacts with its host to drive physiological changes are not fully understood. This is due to the complex nature of how the individual microbes within the community interact with each other and the host at a metabolic level.

To handle the complex nature of the microbiome, studies have turned to machine learning approaches for tasks such as host phenotype prediction. These methods identify subjects who may be at risk for developing certain diseases and identify disease-specific microbial biomarkers. However, to obtain a better understanding of the underlying mechanisms leading to disease pathogenesis, additional data modalities, such as metabolomics, metagenomics, and host genomics, should also be considered. Not only is the integration of multiple data modalities in computational modeling a challenge in itself but doing so also greatly increases the complexity of the data to which computation models are being designed to model.

In this thesis, we will present computational methods and tools we have developed for the integration of multiple data modalities in different microbiome analyses. The first is the integration of phylogenetic information with microbial abundance for the task of host phenotype prediction in our tool, “PopPhy-CNN”. The second is the integration of metabolomic abundance with microbial abundance for the inference of microbe-metabolite interactions in our tool, “MiMeNet”. The last is the integration of patient characteristics and external factors with longitudinal microbiome abundance profiles for the

modeling of dynamic shifts in microbiome communities in our tool, “DiRLaM”. Together, these methods aim to provide a suite of robust and scalable tools, assisting researchers in predicting host disease status, identifying microbes related to disease, uncovering the metabolic function of these microbes, and identifying potential treatment routes for improving patient health through microbiome targeted therapeutics.

Chapter 1

Introduction

The microbiome consists of a community of microscopic organisms cohabitating in a shared environment and has been shown to impact both host development, normal metabolic processes, as well as the pathogenesis of various diseases. Of particular interest is the microbiome of the human gut, which has been linked to multiple metabolic diseases such as inflammatory bowel disease (IBD), obesity, and type 2 diabetes (T2D) (Franzosa et al., 2019; Kostic et al., 2015; Tilg & Kaser, 2011) as well as other non-metabolic diseases such as colorectal cancer and atherosclerotic cardiovascular disease (Ahn et al., 2013; Jie et al., 2017). The dysregulation leading to the pathogenesis of these diseases is largely suspected to occur at the metabolic level through interactions between the microbiome and the host (Kinross, Darzi, & Nicholson, 2011). Therefore, identifying microbes associated with disease status and uncovering the metabolic function of these microbes can facilitate the development of novel therapies that either alter the microbiome (probiotic/prebiotic) or target metabolic pathways (Cani & Delzenne, 2011; Helmink, Khan, Hermann, Gopalakrishnan, & Wargo, 2019; Preidis & Versalovic, 2009).

Given the microbiome's role in various diseases, one important task is the identifications of associations of microbes with host disease phenotype. One common approach for associating microbes to host disease status has been through microbiome-wide association studies (MWAS) (Gilbert et al., 2016), which use different statistical approaches such as DESeq2 (Love, Huber, & Anders, 2014) and ANCOM (Mandal et al., 2015) to identify groups of microbes associated with disease status. Recently however, machine learning (ML) and especially deep learning (DL) models demonstrated the potential of developing a microbial biomarker signature for the prediction of the host phenotype (Ditzler, Polikar, & Rosen, 2015; LaPierre, Ju, Zhou, & Wang, 2019; Pasolli, Truong, Malik, Waldron, & Segata, 2016; Wingfield, Coleman, McGinnity, & Bjourson, 2016). Due to their abilities to capture complex relationships within data, ML and

DL models are believed to be well suited for the modeling of microbiome and microbial community metabolome data, allowing for more robust identification of disease related microbial biomarkers (Geman et al., 2016; LaPierre et al., 2019). Additionally, the use of ML models to predict host phenotype allows for the construction of clinical tools for quickly diagnosing new patients based on their observed microbiome (Cammarota et al., 2020; Manandhar et al., 2021).

Once a set of disease-related microbes has been identified, there is a challenge in identifying the underlying mechanisms of the metabolic dysregulation caused by the microbial biomarkers. The identification of these microbiome-metabolome interactions contributing to the development of disease is essential for both understanding microbiome's overall effect on host health, as well as for the development of metabolic targeted clinical therapies for the prevention or management of disease (Cani & Delzenne, 2011; Helmink et al., 2019; Skelly, Sato, Kearney, & Honda, 2019). To uncover these interactions, previous methods have leveraged *a priori* knowledge (Edwards, Covert, & Palsson, 2002; Larsen et al., 2011). However, the reliance on *a priori* knowledge makes it impossible to discover novel interactions. Recently, studies generating paired microbiome and metabolome data have emerged, allowing for data-driven models to reveal the underlying microbe-metabolite interactions leading to metabolic dysregulation and disease. The emergence of these datasets has given rise to a need for data-driven computational tools and methodologies for the integration and exploration of paired microbiome-metabolome data.

In addition to the identification of a disease related set of microbial biomarkers, the development of therapeutic treatments and interventions requires an understanding of how the microbiome community changes over time when certain stimuli are applied. This requires the collection and modeling of longitudinal microbiome data combined with additional host characteristic information. To infer the dynamics of the microbiome community, studies have often used methods such as Boolean Networks, Dynamic Bayesian Networks (DBN), or generalized Lotka-Volterra equations (gLV) (Bunin, 2017; Claussen et al., 2017; Joseph, Shenhav, Xavier, Halperin, & Pe'er, 2020; Lugo-Martinez, Ruiz-Perez, Narasimhan, & Bar-Joseph, 2019; Michael J McGeachie et al., 2016; Ruiz-Perez et al.; Shenhav et al.,

2019; Steinway, Biggs, Loughran, Papin, & Albert, 2015). These methods have shown success in early disease-related microbiome datasets (Lloyd-Price et al., 2019). However, each method is computationally constrained with regards to both sample size and feature size. This is quickly becoming a challenge with the emergence of larger longitudinal microbiome datasets as well as an increased resolution in sequencing, allowing the analysis of the microbiome community at the species or even strain level. As such, the development of scalable computational models describing the interactions between the specific microbiome community components is critical for understanding the underlying function and dynamics of the microbiome (Waldor et al., 2015).

1.1 Problem Identification

For the task of predicting host phenotype, recent studies have shown that constructing abundance of features using the hierarchical structure of the taxonomic tree can lead to better classification performance (Oudah & Henschel, 2018; Qiu, Tian, & Zhang, 2015). However, there has been a lack of integration of taxonomic information in deep learning frameworks. At the time of our work, there had been only two DL models using phylogeny for the prediction of host disease. The first trains a convolutional neural network (CNN) using the phylogenetic patristic distance to group the observed microbes together (Fioravanti et al., 2018). However, this model does not consider the hierarchical features of the entire tree. More recently, a method using graph convolutional neural networks (GCNN) has been put forward (Khan & Kelly, 2020). Using multiple multi-class disease dataset, they found that the GCNN model outperformed current state-of-the-art ML methods. However, both methods apply their CNN models as a black box; neither method extracts important features used in the prediction task. Therefore, an interpretable deep learning model that can integrate multiple hierarchies of taxonomy could potentially improve host phenotype prediction as well as provide a more robust set of microbial biomarkers.

Another task in microbiome studies is uncovering the underlying metabolic function of different microbes. Previous methods have used *a priori* stoichiometric knowledge in order to model microbe-metabolite interactions (Edwards et al., 2002; Larsen et al., 2011). However, because they rely on *a priori* knowledge, these methods are unable to uncover new relationships. With the emergence of metabolomic data, it is now possible to learn microbe-metabolite relationships in a data-driven manner, which may lead to novel discoveries and downstream hypothesis generation. To our knowledge, only two tools are available for this task by using microbial features to predict metabolomic features. The first uses Elastic Net linear regression, which we believe is not well suited to capture the non-linear nature of microbe-metabolite interactions (Mallick et al., 2019). The second uses a neural encoder with positive weights to learn a latent space between the microbiome and metabolome (Le, Quinn, Tran, & Venkatesh, 2019). By only allowing for positive weights, it cannot model the microbial degradation of metabolites. In addition, the interpretability of the model diminishes through the use of the latent space. Therefore, there is a need for an interpretable model that can capture the complex interactions between the microbiome and metabolome.

In addition to identifying disease-related microbial biomarkers and their metabolic function, it is important to be able to model the dynamic structure of the microbiome under different conditions. This will empower clinicians and researchers to better understand how to treat a patient by altering their microbiome from an unhealthy composition to a healthy one. Current methods for modeling microbiome dynamics rely on probabilistic models such as Boolean or Dynamic Bayesian Networks (Äijö, Müller, & Bonneau, 2018; Michael J McGeachie et al., 2016; Shafiei et al., 2015). However, these methods cannot handle large sets of features, and as such, integrating large amounts of patient data is not feasible. By using deep learning models, multiple types of data can be integrated seamlessly, allowing for scalable modeling of microbiome dynamics.

1.2 Thesis Outline

The research in this dissertation focuses on developing novel deep learning methods for analyses of the microbiome that integrate microbiome abundance data with additional omics data in order to provide a better understanding of the microbiome's role in the development of disease. The research described in this dissertation is organized in the following manner:

Chapter 2 provides a background of microbiome research and the different physiological functions the microbiome has on the host. We discuss common analysis procedures for microbiome data used in disease-related studies. Additionally, we will discuss the training and interpretation of common ML methods used in microbiome studies. Lastly, we will discuss common deep learning frameworks and their applications across multiple domains of computational biology. This chapter is partially based on the publication:

- **Reiman, Derek**, Ulises Sosa, and Yang Dai. "Machine Learning in Identification of Disease-Associated Microbiota." Inflammation, Infection, and Microbiome in Cancers: Evidence, Mechanisms, and Implications: 431.

In **Chapter 3**, we will briefly discuss recent ML approaches for the prediction of host phenotype and discovering disease-related microbial biomarkers and their current limitations. We will then introduce three frameworks that we have developed to address the varying limitations. The first framework, "PopPhy-CNN", uses a convolutional neural network framework that integrates phylogenetic information to spatially organize microbial abundance data, improving the performance of host phenotype prediction across multiple complex disease states and allowing for the identification of disease-related microbial biomarkers and different taxonomic levels. The second framework, "Meta-Signer", further improves the identification

of disease-related microbial biomarkers by combining the microbial features identified by "PopPhy-CNN" with microbial features found across multiple ML methods. "Meta-Signer" uses a rank aggregation approach to combine the multiple ranked lists into a single, robust list of disease-related microbial biomarkers. The last framework employs a conditional adversarial network model in order to augment microbiome datasets to improve the performance of host phenotype prediction. Better prediction of host phenotype, especially in more complex disease states, will result in the identification of more robust microbial biomarkers for early diagnosis and as potential targets for downstream studies of probiotic therapy. This chapter is based on the following list of publications:

- **Reiman, Derek**, Ahmed Metwally, and Yang Dai. "Using convolutional neural networks to explore the microbiome." *2017 39th annual international conference of the IEEE engineering in medicine and biology society (EMBC)*. IEEE, 2017.
- **Reiman, Derek**, Ahmed A. Metwally, Jun Sun, and Yang Dai. "PopPhy-CNN: a phylogenetic tree embedded architecture for convolutional neural networks to predict host phenotype from metagenomic data." *IEEE journal of biomedical and health informatics* 24, no. 10 (2020): 2993-3001.
- **Reiman, Derek** and Yang Dai. Using Conditional Generative Adversarial Networks to Boost the Performance of Machine Learning in Microbiome Datasets. In *Proceedings of the 1st International Conference on Deep Learning Theory and Applications - DeLTA,* ISBN 978-989-758-441-1. 2020
- **Reiman, Derek**, Ahmed A. Metwally, Jun Sun, and Yang Dai. Meta-Signer: Metagenomic Signature Identifier based on rank aggregation of features [version 1; peer review: 1 approved with reservations, 1 not approved]. *F1000Research.* 2021; 10:194
- **Reiman, Derek**, Ali M. Farhat, and Yang Dai. "Predicting host phenotype based on gut microbiome using a convolutional neural network approach." In *Artificial Neural Networks*, pp. 249-266. Humana, New York, NY, 2021.

In **Chapter 4**, we will discuss the current state of methodologies for inferring microbe-metabolite interactions and their limitations. Then, we will introduce our novel framework "MiMeNet", one of the first data-driven approaches to leverage paired microbiome-metabolome data for identifying novel microbiome-metabolome interactions. "MiMeNet" uses a neural network approach for the prediction of the entire metabolome from the host's or environment's microbiome community. By modeling the entire metabolome at once, "MiMeNet" leverages shared information across similar metabolites to improve the overall performance of the model. Additionally, the modeling of the entire metabolome makes "MiMeNet" more scalable than the current univariate approaches. The weights of the trained neural network are then used for both the clustering the microbes and metabolites into functional modules and the identification of module-module interactions. "MiMeNet" is evaluated on three paired microbiome-metabolome datasets, one of which comprised of healthy subjects and patients with IBD. Using the IBD dataset, "MiMeNet" clustered the microbes and metabolites into modules, some of which were strongly associated to disease status. In particular, "MiMeNet" was able to group the metabolites into functional groups in which the members of a module shared similar metabolic class or pathway. By linking these functional metabolic modules with microbial modules, "MiMeNet" not only was able to identify previously validated microbe-metabolite interactions, but also provide novel microbe-metabolite interactions, allowing for future hypothesis generation and potential candidates for metabolic targeted therapy. This chapter is based on the following publication:

- **Reiman, Derek,** Brian T. Layden, and Yang Dai. "MiMeNet: Exploring microbiome-metabolome relationships using neural networks." *PLoS Computational Biology* 17, no. 5 (2021): e1009021.

Finally, in **Chapter 5**, we will discuss current methods used to model microbe-microbe and microbe-environment interactions using longitudinal microbial abundance data and their limitations. We will then introduce our work based on a conference paper to address these limitations in which we use three longitudinal mouse microbiome datasets to model the dynamics of the microbiome community based on two different diets (D. Reiman & Dai, 2019). In this work, we combine two DL frameworks: an autoencoder (AE) for reducing the microbiome community into an intrinsic latent structure and a deep neural network (DNN) for modeling the microbiome dynamics within the reduced latent space. The use of the AE allows for the reduction of noise in the data, a common challenge in microbiome data, while the use of the DNN allows for better modeling of complex relationships. Additionally, by modeling the dynamics within the latent space, the number of DNN parameters are kept to a minimum. Furthermore, we will present the extension of this work into a larger framework, "DiRLaM". "DiRLaM" extends the previous work in three ways. First, it applies a novel diversity-regularization effect on the AE model to reduce the noise in microbiome datasets. Second, it applies interpolation within the latent space rather than through traditional splining techniques Lastly, we expand it to incorporate additional environmental factors other than diet and facilitate both the identification of which factors are significant in the modeling of microbiome dynamics and how these factors impact the dynamics of the microbiome community as well. We show that DiRLaM is more stable and more robust to noise in microbiome data compared to current methods. This chapter is based on the following conference paper as well as work extending the conference paper, currently being written for journal submission:

- **Reiman, Derek**, and Yang Dai. "Using Autoencoders for Predicting Latent Microbiome Community Shifts Responding to Dietary Changes." In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 1884-1891. IEEE, 2019.

1.3 Significance of Thesis Work

The development of these therapeutics requires an understanding of which microbes are associated with disease and the underlying metabolic dysregulation caused by these microbes. The methods/tools presented in this thesis are designed to improve microbiome analyses using different DL methodologies to either augment or integrate additional data modalities in novel approaches. Specifically, "PopPhy-CNN" integrates phylogenetic spatial information to improve the prediction of host phenotype in complex multi-class disease states. "PopPhy-CNN" not only facilitates the identification of disease-related microbial biomarkers at different taxonomic levels but can also be used to develop powerful diagnostic tools for complex diseases. We show that the task of host phenotype prediction can be further improved by augmenting microbiome dataset with synthetic data generated from a CGAN model. "MiMeNet" is one of the first data-driven approaches to integrate paired microbiome-metabolome data and facilitates the clustering of microbes and metabolites into meaningful functional modules, empowering the identification of novel microbe-metabolite interactions underlying the metabolic dysregulation of disease. The last tool, "DiRLaM", is the first DL approach for modeling microbiome dynamics to our knowledge, and it combines an AE and DNN to integrate environmental factors with microbiome data in longitudinal study to accurately model longitudinal microbiome data and identify significant environmental factors affecting the microbiome community dynamics. The identification of these factors and their effect on the microbiome is critical in the identification and testing of therapeutic interventions designed to target the microbiome. Together, this work aims to provide a suite of robust and scalable tools, assisting researchers in predicting host disease status, identifying microbes related to disease, uncovering the metabolic function of these microbes, and identifying potential treatment routes for improving patient health through microbiome targeted therapeutics.

Chapter 2

Overview of Microbiome Studies in Disease and Machine Learning Methods

2.1 Microbiome Studies in Disease

The microbiome is a collection of microscopic organisms (bacteria, fungi, protozoa, and viruses) that live in s shared environment. This collection of microbes is considered to functionally interact with its host or environment. The microbiome communities on and within the human body have been shown to impact host physiology, normal metabolic processes, as well as the pathogenesis of various diseases. These microbial communities can be shaped by many factors such as host genetics and lifestyle (Turnbaugh et al., 2007; F. Xu et al., 2020), hormones (Mallott, Borries, Koenig, Amato, & Lu, 2020), diet (Carmody et al., 2015; Creswell et al., 2020), and geographical location (J. Chen et al., 2016; Gaulke & Sharpton, 2018) (**Figure 1)**.

In particular, the human gut microbiome, which is the combination of the stomach and intestinal microbiomes, has been linked to a wide range of functional roles within the human body (**Figure 2)**. Like other microbiome communities, the gut microbiome provides a protective layer against pathogens (Hansson, 2012) in addition to the production of essential vitamins and nutrients for the host (Rowland et al., 2018). Early microbiome studies were also able to quickly associate the gut microbiome to a variety of

metabolic and immune-mediated diseases such as inflammatory bowel disease (IBD), obesity, and diabetes miletus (Franzosa et al., 2019; Hartstra, Bouter, Bäckhed, & Nieuwdorp, 2015; Kostic et al., 2015; Morgan et al., 2012; Tilg & Kaser, 2011; Turnbaugh et al., 2006; Zheng, Li, & Zhou, 2018). More recently, studies have shown unique capabilities of the gut microbiome, such as its critical role in the development and modulation of the central nervous system, in what has been called the "brain-gut-microbiome" axis (Martin, Osadchiy, Kalani, & Mayer, 2018). Additionally, the gut microbiome has been shown to play a significant role in the development and modulation of the immune system (V. L. Chen & Kasper, 2014; Tomkovich & Jobin, 2016). The gut microbiome has even been implicated in having an impact on a patient's response to immunotherapy treatment for different cancer types (Gopalakrishnan, Helmink, Spencer, Reuben, & Wargo, 2018; Routy et al., 2018).

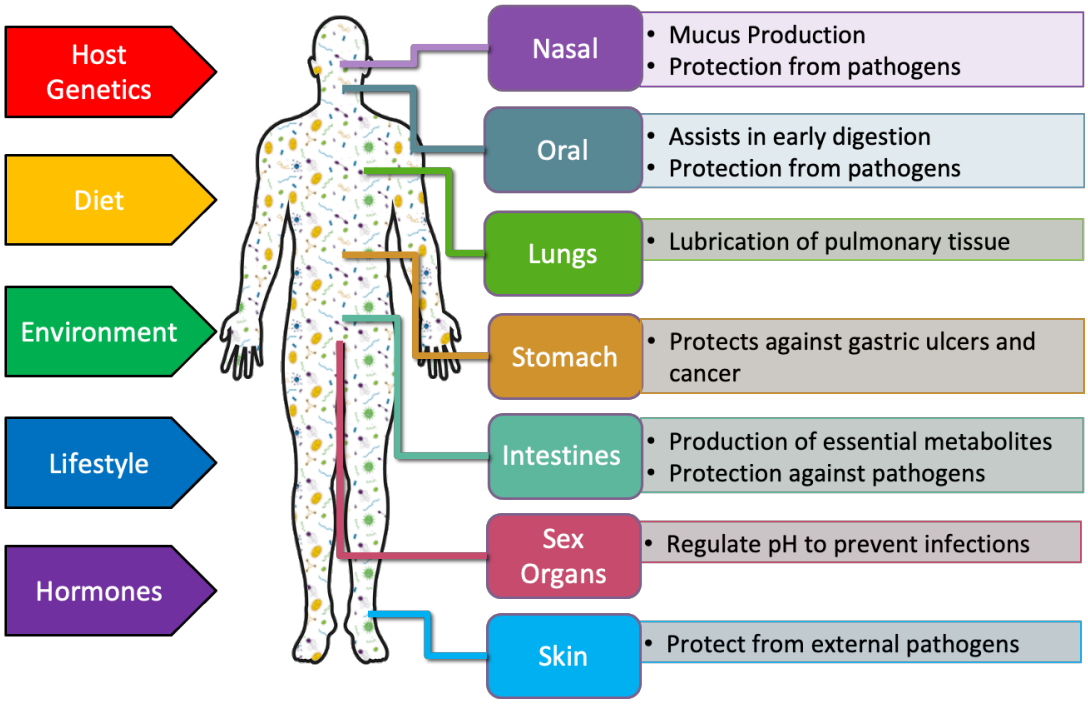


Figure 1. Common microbiome sites of study on and within the human body. The microbiome of the human body is affected by many external factors. Commonly studied sites and their function are shown here.

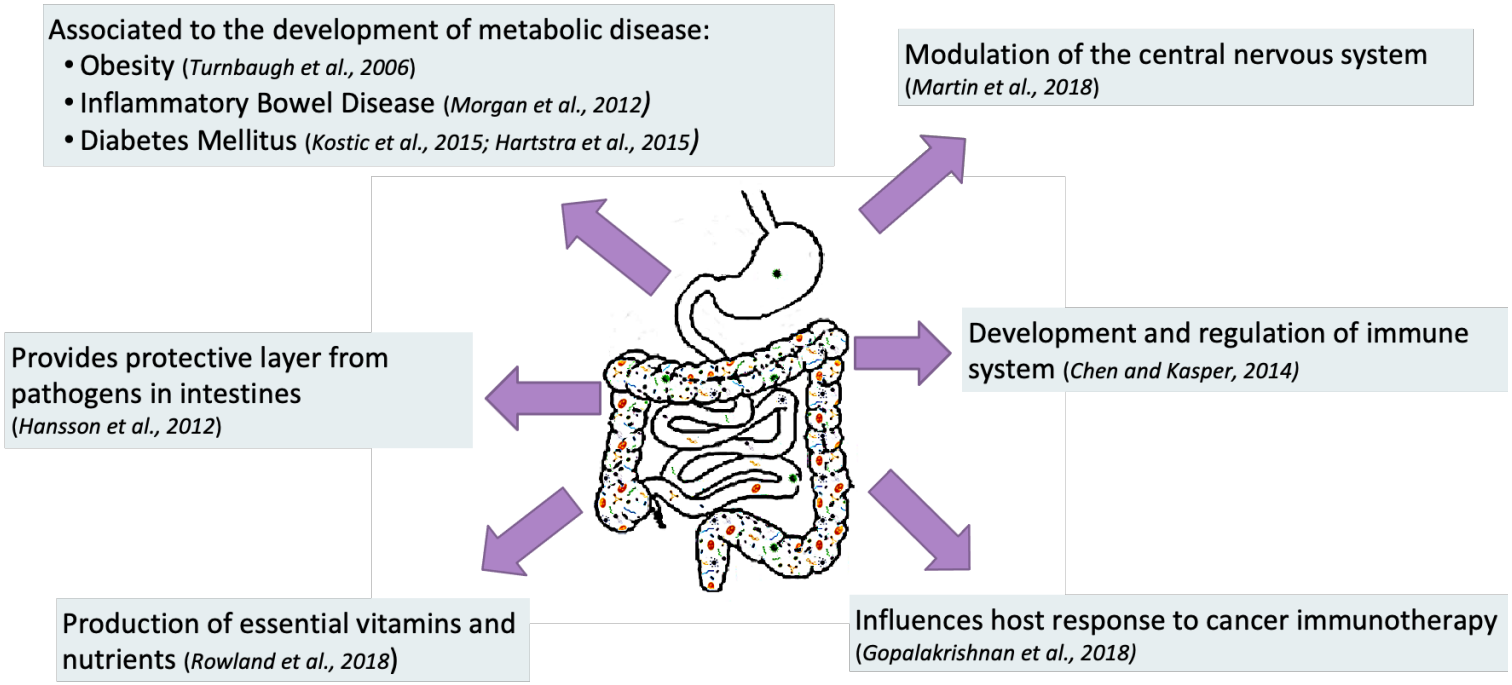


Figure 2. Functional roles of the gut microbiome. The gut microbiome has been shown to have a wide range of functional roles in human health including disease association, providing physical protection from invading pathogens, production of essential nutrients, modulation of the central nervous system and immune system, and influencing host response to cancer immunotherapy.

Therapeutic Interventions for the Microbiome

Because of the important physiological roles that the gut microbiome plays both locally and systemically, clinicians and researchers are beginning to look at this environment as a key target of therapeutic treatment and intervention. In addition to these emerging discoveries regarding the role of the gut microbiome, advances in genomic sequencing and molecular diagnostics have changed our approach to health and medicine by allowing more personalized treatments based on individual characteristics. Taken together, the microbiome is emerging as an important component on the frontier of precision medicine, as it not only is shaped by individual variability, but it is also a modifiable factor that is susceptible to targeting by therapeutics (Kashyap, Chia, Nelson, Segal, & Elinav, 2017) (**Figure 3)**.

Early therapies have used fecal microbiome transplants (FMT) from healthy subjects to reconstruct the gut microbiome of patients with diseases such as obesity or *Clostridium difficile*, a pathogenic microbe of the gut (Marotz & Zarrinpar, 2016; Mattila et al., 2012). Although FMT has demonstrated success, there have been a few reports of adverse effects and a lack of safety trials to evaluate these immediate adverse effects and long-term effects (Harsch & Konturek, 2019; Kaźmierczak-Siedlecka et al., 2020).

A more subtle approach to the modulation of the gut microbiome is with probiotics or prebiotics. Instead of reconstructing the microbiome, probiotics introduce small amounts of microbial organisms to confer a health benefit to the host. Prebiotics, on the other hand, contain fermentable, non-digestible oligosaccharides which are used to stimulate the growth of beneficial indigenous gut bacteria. These are often used together in what is referred to a synbiotic, due to their synergistic effects. Although probiotics have been shown to be effective in the treatment of gastrointestinal diseases (Ritchie & Romanuk, 2012), a recent study has found that the efficacy of probiotics is both dependent on the microbial strain used as well as the disease being treated (McFarland, Evans, & Goldstein, 2018). Therefore, for the development of more efficient probiotic therapies, there is a need to better understand not the functional role of microbes at a high resolution (e.g., species or strain) but also how microbes functionally contribute to disease status.

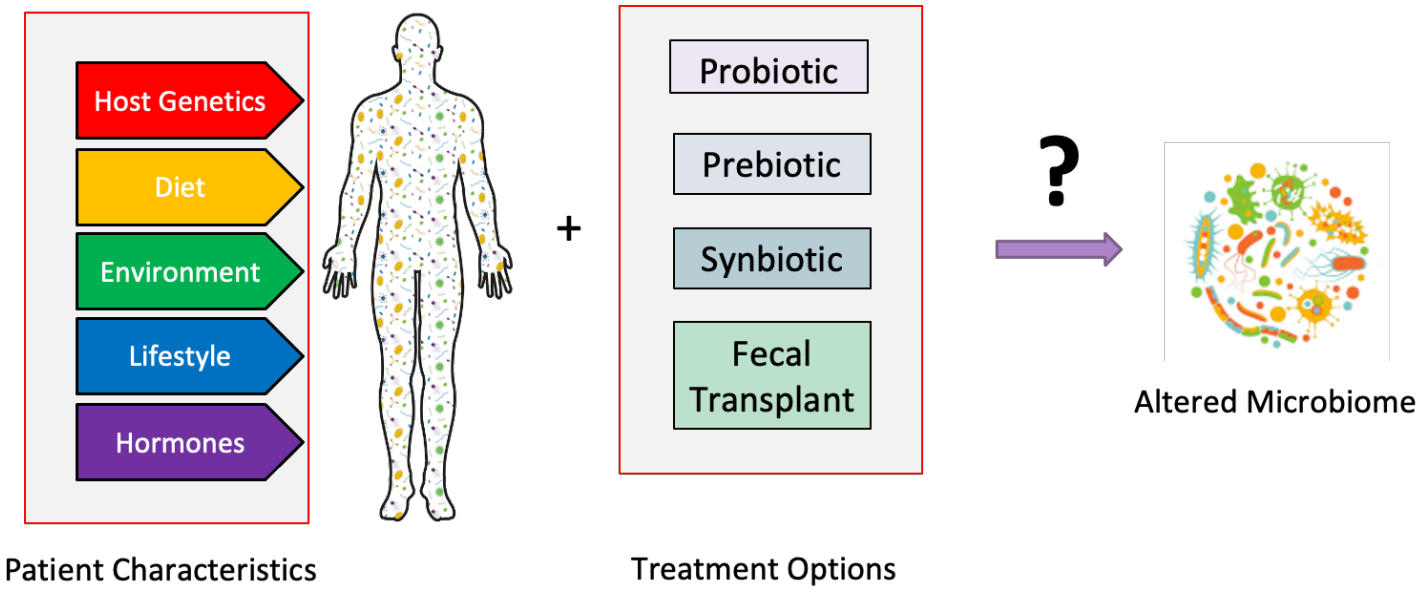


Figure 3. A personalized approach to microbiome therapeutics. Host characteristics contribute to the variability of the host microbiome. Additionally, different therapeutics can modulate the host microbiome in different ways, leading to an altered microbiome community and host phenotype.

Microbiome Sequencing and Characterization

The birth of next generation sequencing (NGS) technologies has revolutionized how we study and characterize the microbiome. The earliest sequencing approach to analyze the microbiome was the amplicon analysis of the 16S ribosomal RNA (rRNA) gene sequence (D'Amore et al., 2016; Janda & Abbott, 2007; Sanschagrin & Yergeau, 2014). The 16S rRNA gene is used since it is widely conserved among microbes containing interspersed hypervariable region. In 16S rRNA gene sequencing, the 16S rRNA region is identified using primers specific to the conserved region and amplified polymerase chain reaction. The sequences of the hypervariable regions are then used to construct operational taxonomic units (OTUs), which can then be annotated with microbial taxonomy. However, this method is limited in the level of resolution it can achieve. Due to the slow evolutionary rate of the 16S rRNA gene, the sequence is not reliable in differentiating microbes at a species level or below (Maroniche, García, Salcedo, & Creus, 2017). In addition, the effects of homologous gene recombination and horizontal gene transfer of the 16S rRNA gene can lead to the taxonomic mislabeling of OTUs (Teyssier, Marchandin, Siméon De Buochberg, Ramuz, & Jumas-Bilak, 2003; Tian, Cai, Zhang, Cao, & Qian, 2015).

More recently**,** metagenome shotgun (MGS) sequencing has been used to sequence random fragments of the microbial genomes. Gene sequences can then be mapped to reference genomes to identify the taxa present in the community. Additionally, these gene sequences can provide a functional profile of the microbiome community. One advantage that the MGS sequencing approach has over the 16S rRNA sequencing approach is that microbes can be more accurately annotated to the species or even strain level. However, even though the cost of MGS sequencing has fallen over the years, it remains more expensive than its 16S rRNA sequencing counterpart (Quince, Walker, Simpson, Loman, & Segata, 2017; Scholz, Lo, & Chain, 2012) and requires a higher coverage of sequencing (Sims, Sudbery, Ilott, Heger, & Ponting, 2014).

The sequencing of microbial communities provides us with different ways of characterizing microbes and their communities. Through the counting and annotation of sequence reads, we can identify which microbes are present in the microbiome community and how abundant each microbe is. This is most often represented as a vector of values where each position represents the abundance of a microbial features. When considering multiple samples, this is expressed as an abundance matrix where each row represents a microbial feature, and each column represents a sample. In order to compare the communities between samples, the abundance values are normalized by library size, resulting in compositional data. This is usually either done in the form of relative abundance (RA), additive log-ratio (ALR), or centered log-ratio (CLR) values. Given a microbial abundance vector $\boldsymbol{x}$ with *n* microbial features, these transformations are applied as,

$$RA(\boldsymbol{x}) = \left[\frac{x_1}{\sum_i x_i}, \frac{x_2}{\sum_i x_i}, \ldots, \frac{x_n}{\sum_i x_i}\right] \tag{2.1}$$

$$ALR(\boldsymbol{x}) = \left[\log\left(\frac{x_1}{x_D}\right), \log\left(\frac{x_2}{x_D}\right), \ldots, \log\left(\frac{x_{D-1}}{x_D}\right)\right] \tag{2.2}$$

$$CLR(\boldsymbol{x}) = \left[\log\frac{x_1}{g(\boldsymbol{x})}, \log\frac{x_2}{g(\boldsymbol{x})}, \ldots, \log\ (\frac{x_n}{g\ (\boldsymbol{x})}\right] \tag{2.3}$$

The transformation into RA values (**Equation 2.1)** divides each microbial abundance with the total abundance within the sample, resulting in values ranging between 0 and 1 and with the property that the sum of all the transformed values within a sample will add up to 1. The transformation into ALR values (**Equation 2.2**) selects a single arbitrary abundance value, x_D, and divides every other abundance by the chosen value before taking the log value. The transformation into CLR values (**Equation 2.3)** divides each abundance value by the geometric mean of the abundance vector, $g(\boldsymbol{x})$ before taking the log value.

In addition to characterizing whole microbiome communities, NGS sequencing provides us with a means to characterize the similarity and differences of microbes through phylogeny. Phylogenies are commonly estimated by using multiple sequence alignment on a set of collected gene sequences and then

applying models of mutation to infer most-likely evolutionary paths (Washburne et al., 2018). This is most commonly done using the 16S rRNA gene sequence. However, other core genes have been used as well in past studies (Matsen, 2015). The distance between microbial taxa or OTUs can then be visualized using either a rooted or unrooted tree, where each node represents a microbial taxon or OTU, and the edge length between them represents the evolutionary distance.

The combination of microbiome abundance and phylogenetic similarity facilitates the characterization of diversity in microbiome communities both locally and globally. This is commonly done by quantifying both the distribution of microbial abundances (evenness) as well as the total number of observed microbes within samples (richness). We use two types of diversity metrics to describe microbiome communities: alpha diversity and beta diversity.

Alpha diversity quantifies the diversity of a single microbiome community. It summarizes the distribution of microbial abundances in a sample into a single value that depends on the richness and evenness of the microbiome community. Two general methods for calculating alpha diversity are the Shannon Index (**Equation 2.4**) and inverse Simpson Index (**Equation 2.5**) (Lande, 1996).

$$H(\boldsymbol{x}) = -\sum_{i} x_i \log(x_i) \tag{2.4}$$

$$D(\boldsymbol{x}) = \frac{1}{\sum_i x_i^2} \tag{2.5}$$

Here $\boldsymbol{x}$ is a vector of microbiome abundance values that have been transformed to RA values (**Equation 2.1**). Neither the Shannon Index nor the inverse Simpson Index uses phylogenetic similarity information for the calculation of alpha diversity. In contrast, a widely used method that uses only phylogenetic similarity is Faith's Phylogenetic Diversity (Faith's PD) (Faith, 1992). This metric is calculated as the sum of the branch lengths of a phylogenetic tree connecting all microbial features present within a sample and

does not incorporate microbial abundance into the calculation. There are advantages to both using abundance and to using phylogenetic similarity to quantify the variation within samples. Therefore, methods have been developed that integrate the two, such as an extension of the Shannon Index to incorporate phylogenetic distance (Allen, Kon, & Bar-Yam, 2009).

Unlike alpha diversity which characterized the variation of a single sample, beta diversity is used to compare microbiome samples by quantifying similarity or dissimilarity between them. The most common measures for beta diversity are Bray-Curtis dissimilarity, Jaccard Index, and UniFrac distance. The Bray-Curtis dissimilarity is used to measure the compositional dissimilarity between two microbiome samples (Bray & Curtis, 1957).

$$BC(\boldsymbol{x_1}, \boldsymbol{x_2}) = 1 - \frac{2C}{S_1 + S_2} \tag{2.6}$$

Here $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ are microbiome abundance vectors, C is the sum of minimum values across all microbial features between $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$, and S_1 and S_2 are the total abundances across all microbial features in $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$, respectively. The Jaccard Index measures the similarity between two communities based on the presence or absence of microbes by dividing the intersection of microbial features in both samples by the union of microbial features across both samples. UniFrac uses phylogenetic similarity to calculate evolutionary distances between microbiome communities (Lozupone & Knight, 2005). For each sample, the edges of the phylogenetic tree that lead to observed microbial features are collected. The distance between two samples can then be calculated as the sum of branch lengths that are not shared divided by the sum of all edge lengths found in the phylogenetic tree. Additionally, there is a weighted version of UniFrac that weights the phylogenetic differences according to the RA values of each lineage. By pairwise comparing microbiome communities using beta-diversity measures, we can construct a similarity or dissimilarity (depending on the method used) matrix. This matrix can be used to visualize the relationships

between samples through ordination techniques such as Principal Coordinates Analysis (Borg & Groenen, 2005) .

The last data type obtained from NGS sequencing is specific to MGS sequencing. By sequencing all the microbial genes, a gene table can be constructed in a similar fashion to the microbial abundance table, however here each row represents a microbial gene. By using microbial gene abundance rather than microbial abundance, we are able to analyze the communities at a functional level. Important genes identified by analyses can then be used to identify functional enrichment and mapped to gene or metabolic pathways (De Filippo, Ramazzotti, Fontana, & Cavalieri, 2012). An overview of NGS technologies and generated data is shown in **Figure 4**.

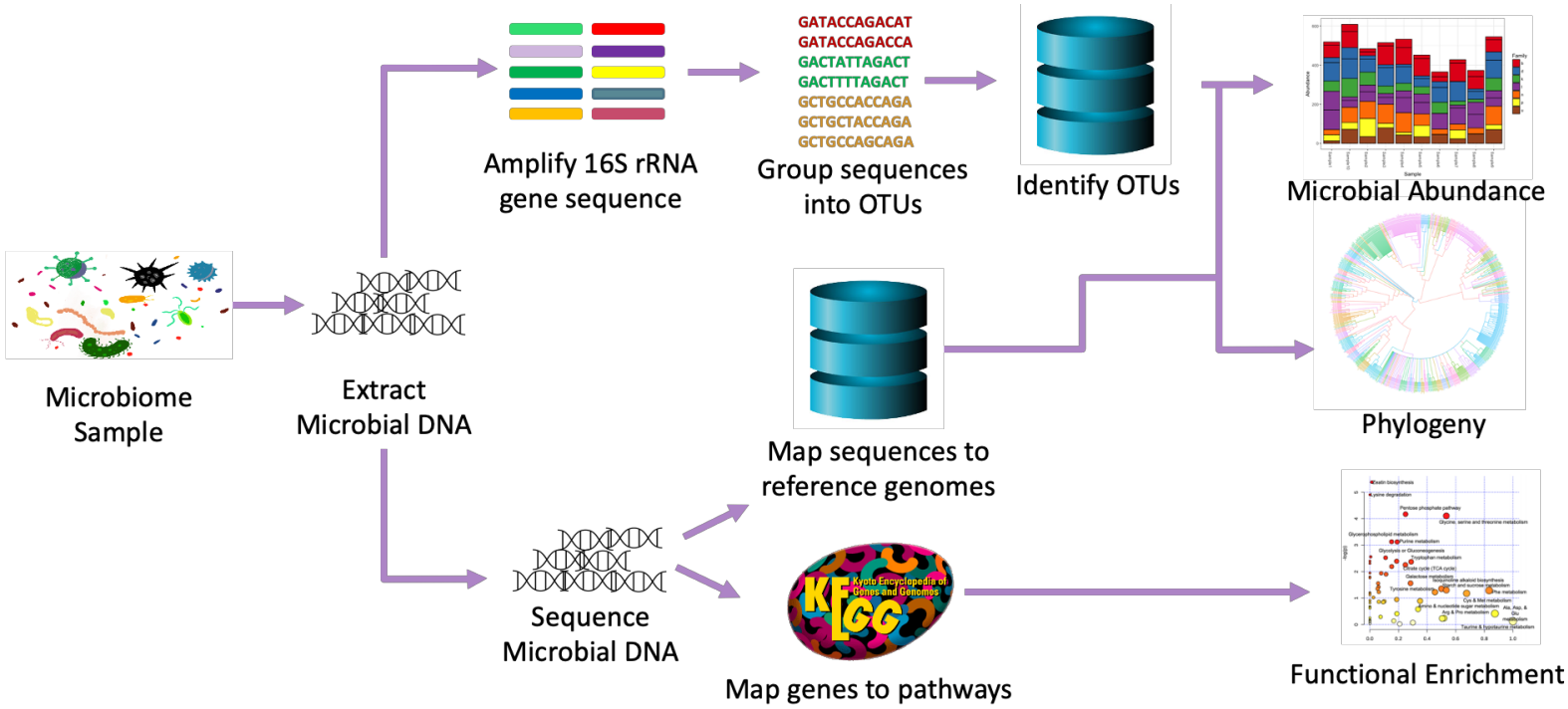


Figure 4. NGS methods for microbiome sequencing. (Top) 16S rRNA sequencing amplifies and sequences 16S rRNA genes and groups sequences into OTUs. OTUs are then annotated taxonomically, providing abundance and phylogenetic information. (Bottom) MGS sequencing amplifies and sequences all microbial genes.

2.2 Standard Microbiome Analyses

In this section, we will review common analyses of disease-related microbiome datasets based on the data generated using NGS technologies: the identification of microbial biomarkers, the prediction of host phenotype, the inference of microbe-metabolite interactions, and the modeling of the microbiome community dynamics. The work presented in the following chapters of this dissertation will address challenges in these analyses to improve upon existing methods.

2.2.1 Identification of Microbial Biomarkers

One of the major objectives of microbiome studies is the identification of specific microbes associated with changes in host phenotype. In the context of disease, knowing which of these microbial taxa or OTUs can help further the understanding of the underlying disease mechanism (Sun & Chang, 2014). In addition, it can facilitate both the development of clinical therapeutic interventions as well as the earlier diagnoses of patients. For this task, methods can be organized into two different approaches: statistical and ML.

Statistical analyses have used both parametric and non-parametric approaches to identify differentially abundant microbes between case and control groups using microbiome abundance data. Parametric methods are driven by assumptions about the underlying distribution of the data. Methods using generalized linear models, such as edgeR (Robinson, McCarthy, & Smyth, 2010) and DESeq2 (Love et al., 2014) have been adopted from gene expression analyses and widely applied to microbiome count data. Additionally, non-parametric methods have been used when handling microbiome abundance data that has undergone a compositional transformation. One method, ANCOM (Mandal et al., 2015), uses the log-ratio of all pairs of microbes to test for a difference in means. Another tool, ALDEx2 (Fernandes, Macklaim, Linn, Reid, & Gloor, 2013), uses a Dirichlet-multinomial model to infer original abundance from counts

by finding the expectation of multiple simulated instances of the data. It then uses the Wilcoxon rank-sum test to identify differentially abundant microbes across two or more groups. These statistical methods have enhanced the detection of microbial group association with respect to disease studies. However, as univariate methods, they may fail to detect complex multivariate nonlinear associations since they cannot consider the complex nature of the microbiome community as a whole.

The use of ML approaches has been motivated by the findings that a microbial signature is complex in nature, involving simultaneous over- and under-representations of multiple microbial taxa (Knights, Parfrey, Zaneveld, Lozupone, & Knight, 2011; T. Wang & Zhao, 2017). ML approaches are commonly structured as a supervised learning task of predicting the host phenotype from the microbiome community abundance, which will be introduced in the next section. Once an ML model has been trained for this task, microbial input features can be evaluated based on their importance in making predictions. Methods such as random forest (RF), Elastic Net regression, least absolute shrinkage and selection operator (LASSO) regression, and support vector machines (SVMs) have been applied successfully for identifying a microbial biomarker signature (Pasolli et al., 2016; Wingfield et al., 2016; Zhang et al., 2015). These methods will be further discussed in Section 2.3. Additionally, the use of a deep neural networks (DNNs) has been applied in the hope that DNNs could identify more complex relationships for host phenotype prediction. However, the evaluation of DNNs compared to other ML methodologies when using microbiome abundance data is still incomplete (Ditzler et al., 2015; LaPierre et al., 2019).

There are a couple of challenges in the identification of microbes associated to disease status. Parametric statistical models are based on assumptions of the underlying distribution of the data being modeled, and this can cause the statistical model to be biased if the microbiome data does not support that assumption. Non-parametric methods for compositional abundance data, on the other hand, do not have any underlying assumptions or biases. However, individual compositional abundance values are no longer independent of each other as the abundance of one feature is dependent on the abundance of all other features. This can lead to spurious correlations and larger model variances, especially when using univariate

methods (Xia & Sun, 2017). On the other hand, ML methods are believed to be able to model the complex nature of the microbiome community as a whole, however these methods have their own challenges. ML methods, and especially DL methods, require larger amounts of samples compared to statistical methods in order to generate robust and generalizable models. This is often a challenge in microbiome data since generated datasets are often relatively small in sample size, and without regularization and careful training, ML methods can quickly overfit the data leading to trained models with high variance.

2.2.2 Prediction of host phenotype

The prediction of host phenotype from microbial community abundance is often performed together with the identification of microbial biomarkers. Models with the ability to predict disease status from a subject's microbiome can facilitate the development of diagnostic tools, allowing for early detection of disease development or susceptibility in subjects. Many recent studies have employed traditional ML approaches such as RFs, LASSO regression, and SVMs and shown success in the prediction host phenotype (Pasolli et al., 2016; Wingfield et al., 2016; Zhang et al., 2015). Additionally, there is growing interest in the use of DL models due to the ability of deep architectures to better model more complex systems, allowing for the better identification of the interactions of microbial taxa in disease prediction.

One challenge in the task of predicting host phenotype is the selection of an ML model. Varying levels of predictive performance have been reported across different ML models and disease studies, and therefore it is impossible to know the best suited model for a new dataset without extensive evaluation of multiple methods. Additionally, the preprocessing of data and compositional transformations can have different effects on different ML models. Another challenge, discussed in the previous section as well, is that ML and DL methods require a relatively large amounts of samples. This is often a challenge in microbiome data since generated datasets are often relatively small in sample size, and without

regularization and careful training, ML and DL methods can quickly overfit the data leading to models that are not generalizable and perform poorly on new and unseen data.

2.2.3 Inference of Microbe-Metabolite Interactions

While previous studies have uncovered various microbe-disease associations, more recent work has further revealed the central role of bacterial metabolites in host health (Feng et al., 2016; McHardy et al., 2013; Parker, Lawson, Vaux, & Pin, 2018). Based on these findings, the identification of microbiome-metabolome interactions has become of great interest. Understanding how different microbes contribute to the overall metabolic activity of the host is essential not only for understanding the underlying physiological mechanisms of metabolic dysregulation leading to disease onset or severity, but also for the development of therapeutic interventions designed at modulating the microbiome community for the prevention or management of chronic metabolic disease (Cani & Delzenne, 2011; Helmink et al., 2019; Skelly et al., 2019).

Previous studies investigating the interactions between the microbiome and metabolites have often relied on *a priori* annotations of microbial enzymes and metabolic pathways. One prominent method, Predicted Relative Metabolic Turnover (PRMT) (Larsen et al., 2011), uses microbial genome annotations to first predict the abundance of specific microbial enzymes from the overall microbial community. Annotated metabolic pathways containing stoichiometric coefficients of the inferred enzymes are used to then predict the overall change in each metabolite within the system. The second method is constraint-based stoichiometric modeling using flux balance analysis (FBA) to learn the flux rate of metabolites in the microbiome community (Biggs, Medlock, Kolling, & Papin, 2015; Edwards et al., 2002; Gottstein, Olivier, Bruggeman, & Teusink, 2016). FBA calculates the flow of metabolites through a predefined metabolic network, making it possible to predict rate of production or consumption given metabolites in the microbiome community. However, since both PRMT and FBA rely on *a priori* information regarding the

structure and stoichiometry of metabolic pathways, they are limited from identifying novel metabolic findings.

More recently, however, paired microboime-metabolome datasets have emerged and allowed for the identification of microbiome-metabolome interactions in a data-driven approach (i.e.. these methods no longer require *a priori* annotated knowledge). As data-driven methods, they do not suffer from the limitation as the previously discussed methods and are able to identify novel microbiome-metabolome interactions. A recently developed method, MelonnPan, uses linear Elastic Net regression to predict a metabolite abundance from microbial relative abundance data (Mallick et al., 2019). Specifically, the abundance of a single metabolite y_i is modeled as,

$$y_i = \beta_0 + \sum \beta_j x_j \tag{2.7}$$

Since MelonnPan is using an Elastic Net model, it is constrained by

$$(1-\alpha)\sum|\beta_j| + \alpha\sum\beta_j^2 \leq t \quad s.t. \;\; 0 < \alpha < 1 \tag{2.8}$$

Here t is the threshold ceiling of the Elatic Net penalty and α is the ratio of L_1 and L_2 regularizations. In another study, the authors develop a neural encoder-decoder to encode the microbial community abundance into a reduced space, and then decoded the reduced space to the paired metabolome (Le et al., 2019).

2.2.4 Longitudinal Modeling of the Microbiome Community Structure

The last analysis of microbiome data we will discuss is the longitudinal modeling of the microbiome community structure. The understanding of how the microbiome community changes over

time, especially with regards to subject characteristics and various external stimuli, is critical in the development of therapeutic interventions designed at modulating the microbiome.

The modeling of longitudinal microbiome data faces many challenges, especially with regards to data collection. Microbiome data is confounded by noise, often coming from factors such as the dropout of microbial features in time-points, missing and unaligned time-points, and the varying speed of microbial dynamics between subjects. To address these challenges, many methods use smoothing splines for the interpolation missing data and correction of noise (Lugo-Martinez et al., 2019; Shafiei et al., 2015). This interpolation helps smooth out noise and facilitates the sampling and evaluation of equal time points across subjects.

There have been three widely used methods for modeling the dynamics of the microbiome community which have been successful in previous studies: Boolean networks (Claussen et al., 2017; Steinway et al., 2015), Dynamic Bayesian networks (Michael J McGeachie et al., 2016; Ruiz-Perez et al.; Shafiei et al., 2015), and generalized Lotka-Volterra (gLV) equations (Joseph et al., 2020; Kuntal, Gadgil, & Mande, 2019; Stein et al., 2013).

The network-based methods represent microbial abundance features as nodes in a network. Boolean networks have been used to model different biological processes including microbiome dynamics, gene regulatory networks, and cell cycle dynamics (Chai et al., 2014; Claussen et al., 2017; Davidich & Bornholdt, 2008; Steinway et al., 2015). Boolean networks represent nodes as binary states of either “on” or “off” and the state of each node is updated over time using Boolean functions. Nodes are connected by directed edges, and node states are updated in discrete time transitions (e.g. t to $t + 1$) based on Boolean combinations (AND, OR, NOT, etc.) of their parent node states (Schwab, Kühlwein, Ikonomi, Kühl, & Kestler, 2020).

Dynamic Bayesian networks (DBNs) are an extension of Bayesian networks. In a Bayesian network, each node represents a microbial feature as a random variable and each edge represents the

conditional probability of the target node. DBNs can extend this network to a first-order Markov process by structuring the network in a way to contain a set of nodes for time t and a set of nodes for the same microbial features at time $t+1$ with directed edges connecting nodes in time t to nodes in time $t+1$. In addition, discrete clinical variables can be present at time *t* with directed edges to microbial features at time $t+1$. In this way, the DBN is composed of a directed acyclic graph *G*. Variables in G can either be discrete variables Δ or continuous variables Ψ. We will use $\pi(X)$ to represent the set of parents of variable *X* in *G*. The DBN specifies a set of conditional probability distributions *P* over Δ and a set of conditional linear Gaussian density functions *F* over Ψ (Michael J McGeachie et al., 2016). We can then write the multivariate normal mixture density over all variables as,

$$P(\Delta)\ \mathrm{F}(\Psi|\Delta) = \prod_{x\in\Delta} p(x|\pi(x)) \prod_{y\in\Psi} f(y|\pi(y)) \tag{2.9}$$

Since we may have both continuous and discrete nodes in the DBN model, we can model continuous variables using a Gaussian regression model based on continuous parents (v) and discrete parents (u).

$$f(y|u,v) \sim N(\lambda_0 + \sum \lambda_i^{(u)} u_i + \sum \lambda_j^{(v)} v_j\,, \sigma^2) \tag{2.10}$$

Here λ_0 is the intercept, $\lambda_i^{(u)}$ is the set of regression coefficient for discrete parents, $\lambda_j^{(v)}$ is the set of regression coefficients for the continuous parents, and σ^2 is the variance. Then, using the data set, *D,* and graph structure *G*, we can directly infer the parameters *Θ* with maximum likelihood estimation,

$$\max_{\theta,G} P(D|\theta,G)P(\theta,G) = P(D,\theta|G)P(G) \tag{2.11}$$

While maximizing the likelihood, we also penalize overly complicated graph structures. A common way is to use the Bayesian information criterion score to penalize larger number of total parameters in *G*.

$$BIC = d \log(N) - 2 \log\big(P(D|\theta, G)\big) \tag{2.12}$$

Here d represents the total number of parameters ($|\theta|$) and N is the number of time points in the dataset D. The network with the lowest Bayesian information criterion score is selected as the final network model. Once the network structure and parameters are learned, the DBN can be used to infer the transition of the microbiome community from time t to time $t + 1$ (Z. Ghahramani, 1997).

The Lotka-Volterra equations were originally designed to model predator-prey ecological systems. Unlike the original Lotka-Volterra equations (Wangersky, 1978), which only consider predator-prey interactions, gLV equations allow for all possible combinations of interspecies interactions, such as the commensalism and competition interactions observed in microbiome communities (Bunin, 2017; Wangersky, 1978). This approach models the community as a set of ordinary differential equations where each equation represents the change of a single microbe's abundance considering an intrinsic growth rate and microbe-microbe interactions.

$$\frac{dx_i}{dt} = x_i \left(r_i + \sum_{j=1}^{n} \propto_{ij} x_j \right) \tag{2.13}$$

Here x_i is the abundance of the i^{th} microbial feature, r_i is the growth rate of the microbe, and $\propto_{ij}$ is the interaction coefficient between the i^{th} and j^{th} microbial feature. To solve these equations, we can estimate it as a log-lagged differences in abundances

$$\frac{1}{x_i}\frac{dx_i}{dt} = \left(r_i + \sum_{j=1}^{n} \propto_{ij} x_j \right) \tag{2.14}$$

$$\frac{d\ln\left(x_i(t)\right)}{dt} = x_i\left(r_i + \sum_{j=1}^{n} \propto_{ij} x_j\right) \tag{2.15}$$

$$\ln\left(x_i(t_{k+1})\right) - \ln\left(x_i(t_k)\right) \approx \left(r_i + \sum_{j=1}^{n} \propto_{ij} \left(\frac{x_j(t_{k+1}) + x_j(t_k)}{2}\right)\right)\Delta t \tag{2.16}$$

We use the trapezoid rule to approximate **Equation 2.15** as **Equation 2.16**. Regression models can then be used on the log-lagged difference approximation using absolute abundance values in order to estimate the model coefficients (r and $\propto$). A recent study has put forth a gLV model designed to handle compositional microbiome data in which they showed improvement in dynamic modeling compared to standard gLV models (Joseph et al., 2020).

Although these methods have shown success in past studies, these methods all share similar constraints in that they are heavily constrained by the number of samples and features. This makes them well suited for smaller datasets, but as NGS technology continues to improve, we are not only seeing larger microbiome studies with more samples, but we are able to identify microbial features at higher resolutions. This results in datasets that are increasing not only in sample size but also feature size. Additionally, the integration of host characteristics and external stimuli further increases the number of inputs considered. Therefore, to address these limitations, there is a need for scalable computational models that can use larger datasets to accurately model microbiome community dynamics.

2.3 Machine Learning Methodologies

In this section we will review some of the common ML methods used in microbiome studies (RF, SVMs, logistic regression, and neural networks).

Random Forest

RFs are decision tree learning models trained in an ensemble fashion, taking the average of the ensemble to give a robust decision tree (Ho, 1995). Given a set of samples = $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ with k classes, the model trains a set of decision trees and takes the average of the trees to give a single robust decision tree. Each tree is trained using a bootstrapped subset of the training data. While growing each tree, a decision rule is made at each node by selecting the best feature from a random subset of features that best splits the data into two subsets. Decision rules are evaluated using entropy or the Gini impurity metric. In our tutorial, we will use the Gini impurity for making decisions. For a set of samples with k classes, let p_i be the proportion of samples of class i for $i \in \{1 \ldots k\}$. The Gini impurity of the set is calculated as

$$I_G(p) = 1 - \sum_{i=1}^{k} p_i^2 \tag{2.17}$$

Once a RF model is trained, features can then be evaluated using the mean decrease impurity. For each node, an importance score for the feature being split upon is calculated as the decrease in the Gini impurity from before and after the split weighted by the proportion of total samples that were split. A feature's overall importance is then calculated by averaging the weighted scores of that feature over all the decision trees in the ensemble.

Another characteristic of the RF models that we will look at is to determine how generalizable the model is. Since each tree is trained with a bootstrapped set of the training data, we will have a subset of samples that were not used for building the tree. These samples are called the out-of-bag (OOB) samples, and they can be used to evaluate the accuracy of their respective tree, giving an OOB score. This score tells us how well each decision trees predicted its OOB samples and can give us a sense of how generalizable the model is.

Support Vector Machines

SVMs are supervised ML models that learn the best hyper-plane to separate two classes of data (Cortes & Vapnik, 1995). The orientation and position of this hyperplane are determined by a subset of data points, called support vectors, which lie close to the hyperplane. The hyperplane will be determined by a set of weights ($\boldsymbol{w}$) and an intercept (b) through model training. The class of a microbiome sample $\boldsymbol{x}_i$ represented by m features can then be predicted as

$$\hat{y} = sign\,(\boldsymbol{w}^T \boldsymbol{x}_i + b) \tag{2.18}$$

SVM models can use different types of kernels in order to transform the data to higher dimensions in order to better split the data. The simplest kernel is the linear kernel which considers the distance between two points as the inner product, $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle = \boldsymbol{x}_i^T \boldsymbol{x}_j$. Although simple, the benefit of the linear kernel is that it allows for direct interpretations of the weight values. For example, under the linear kernel, each weight w_j in **Equation 2.18** represents the importance of a feature j in determining the class label.

Logistic Regression

Logistic regression is a ML model that uses a logistic function to model a binary-dependent variable (John Lu, 2010). Given a set of samples $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$, a logistic regression model predicts the class of a sample by using a threshold value (e.g., 0.5) on the value,

$$\hat{y}_i = \frac{1}{1 + e^{-(\boldsymbol{\beta} \boldsymbol{x}_i + \beta_0)}} \tag{2.19}$$

Here β_0 is a bias value, and $\boldsymbol{\beta}$ represents the vector of weights to be multiplied by the vector of features. During training, we can penalize these weights in order to regularize the model, helping to prevent overfitting. The two most common forms of regularization are the L_1 and L_2 regularizations,

$$L_1(\boldsymbol{\beta}) = \sum_j |\beta_j| \tag{2.20}$$

$$L_2(\boldsymbol{\beta}) = \sum_j \beta_j^2 \tag{2.21}$$

The L_1 regularization technique will penalize the weights in such a way that many weights will become 0, effectively removing the respective feature from the predictive model. On the other hand, the L_2 regularization technique will penalize the weights in order to prevent any large weights, which could lead to unstable predictions. A third regularization technique, Elastic Net regularization, uses both L_1 and L_2 regularizations when fitting the model (H. Zou & Hastie, 2005). These regularizations are used in other linear models as well. Least absolute shrinkage and selection operator (LASSO) regression models use least squares regression in conjunction with L_1 regularization (Tibshirani, 1996). Ridge regression is a least squares regression model that uses L_2 regularization (Hoerl & Kennard, 1970).

Neural Network

An artificial neural network (ANN) or multilayer perceptron neural network (MLPNN) is a fully connected network composed of hidden layers of nodes, most commonly as perceptrons. The value of a perceptron is a linear combination of the values from the previous layer that is then passed to a nonlinear activation function (Dreyfus, 1990). This allows neural network models to uncover nonlinear relationships within data. More explicitly, the values of the l^{th} hidden layer $\boldsymbol{h}_l$ is calculated as

$$h_l = \varphi\left(\boldsymbol{W}_l^T \boldsymbol{h}_{l-1} + \boldsymbol{b}_l\right) \tag{2.22}$$

where h_{l-1} are the values from the previous hidden layer, $\boldsymbol{W}_l$ are the weights connecting $\boldsymbol{h}_{l-1}$ to $\boldsymbol{h}_l$, $\boldsymbol{b}_l$ is the bias values to the nodes of layer l, and φ is a non-linear activation function. The non-linear activation functions applied at each hidden layer are what give the neural network the capabilities of learning complex non-linear patterns. The entire network can be trained using a loss function, using the loss error and back-propagation to tune the network weight parameters until the network has been trained.

A DNN is defined as a fully connected neural network with at least two hidden layers between the input and output layer (W. Liu et al., 2017; Nielsen, 2015). DNNs have recently become the standard tool for solving a variety of different problems, ranging from computer vision, natural language processing, and computational biology (Deng & Liu, 2018; W. Liu et al., 2017; Razzak, Naz, & Zaib, 2018). By providing multiple hidden layers between the input and output, the network can capture more complex patterns. The DNN model is the foundation of which other DL models come from. In the next section I will present variations of the DNN framework specialized for different tasks.

The interpretation of neural network models is not as direct as other ML approaches. In the frameworks presented in this dissertation, we will use a method by Olden et al. to evaluate each feature by looking at its cumulative impact on prediction (Olden, Joy, & Death, 2004). This is done by multiplying the weight matrices of the of every layer in the set of layers L, resulting in a matrix of scores where each row represents an input feature and each column represents an output feature.

$$S = \prod_{l \in L} \boldsymbol{W}_l \tag{2.23}$$

The numerical score at a position represents how much the respective input's value attributes to the respective output's value, where a positive value represents that the output increases as the input increases and a negative value represents that the output will decrease as the input increases.

2.4 Deep Learning Methodologies

Deep learning is an extension of ML, and of neural networks in particular. The methods used in DL are built upon neural networks of many layers, resulting in the name deep neural network. In this section, I will discuss some of the common DL frameworks which will be used as the foundation of the works presented in this dissertation.

Convolutional Neural Network

Convolutional neural networks are DNN models connected in a way as to capture local spatial patterns (Albawi, Mohammed, & Al-Zawi, 2017). This is done by using kernels of shared weights that slides across multi-dimensional input data. As such, these models are specifically powerful when using visual or sequential data. As a kernel slides across the multi-dimensional image, it calculates a discrete convolution to generate a new multi-dimensional matrix called a feature map. This results in a single feature map for each kernel where each kernel tries to find a unique spatial pattern in the input matrix. Each generated feature map then undergoes pooling, usually in the form of max pooling. The reduced feature maps can either go through additional convolutional layers followed by additional pooling, or they can be flattened and passed through fully connected layers before finally going to the output layer. An overview of a standard CNN architecture as well as examples of the convolution and max pooling process are shown in **Figure 5**.

Convolutional neural networks have seen a great amount of success in computational biology, largely due to the sequential nature of DNA, RNA, and protein data. A study by Kelley et al. designed a CNN framework to learn the functional activity of DNA via chromatin accessibility from using DNA sequence data (Kelley, Snoek, & Rinn, 2016). Their tool, "Basset", showed good predictive performance, but more importantly, allows researchers to perform saturation mutagenesis very quickly *in silico*. Using the nature of protein sequences, CNN models have been applied to tasks such as the prediction of protein secondary structure (S. Wang, Peng, Ma, & Xu, 2016) and the prediction of protein-protein interactions (Hashemifar, Neyshabur, Khan, & Xu, 2018). With regards to medical image data, CNNs have been widely utilized to facilitate medical tasks including the identification and quantification of immune cells in immunohistochemistry staining (Ting Chen & Chefd'Hotel, 2014) as well the diagnosis and classification of tumors across different cancer types (L. Li et al., 2020; L. Zou et al., 2019).

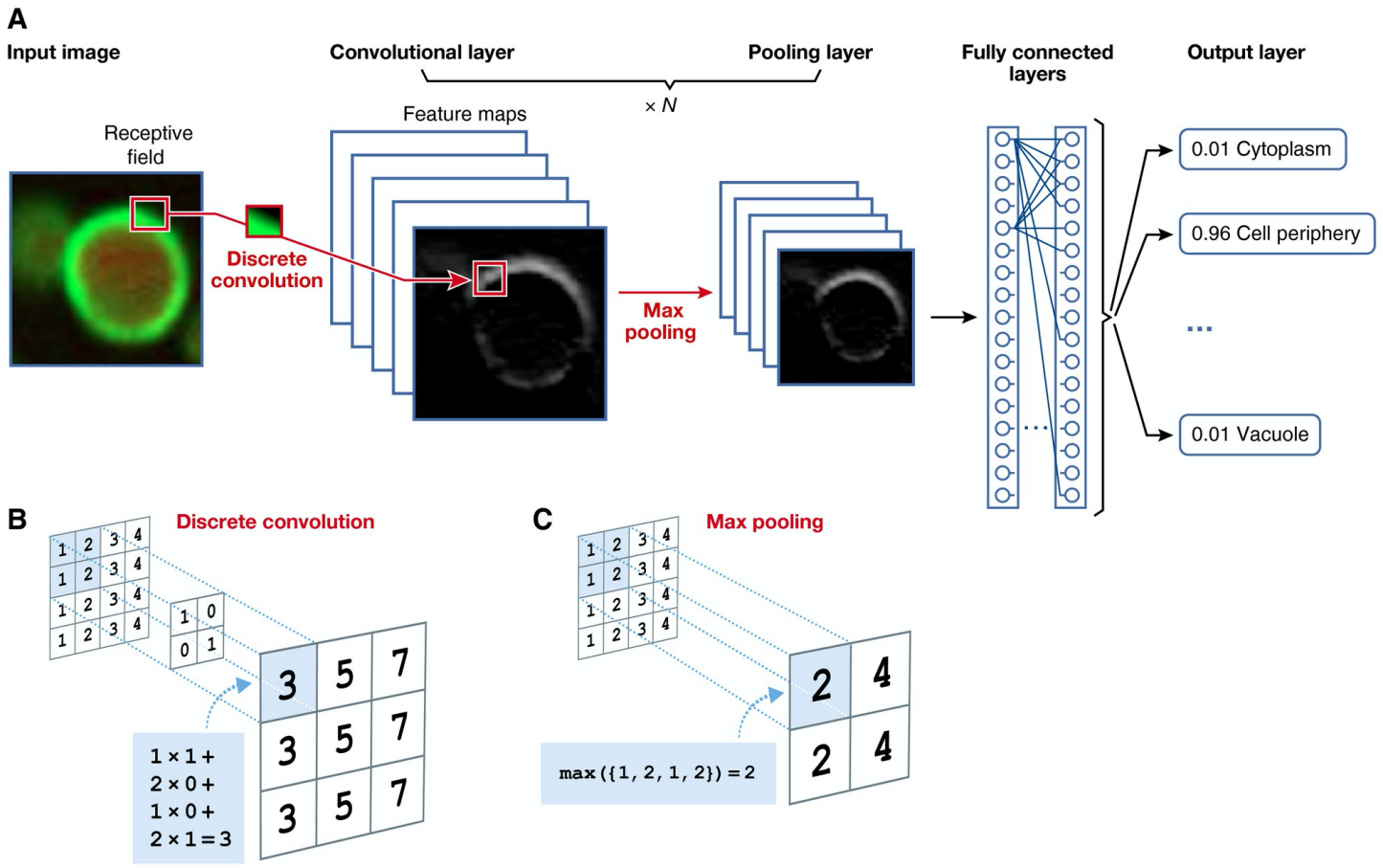


Figure 5. Architecture of a CNN model. (A) Multiple kernels of weights (receptive fields) slide across an input image to generate respective feature maps through. Max-pooling is performed on feature maps for dimension reduction. The resulting feature maps after *N* convolutional and pooling layers are flattened and passed through fully connected layers before the output layer. (B) An example of a 4x4 input and 2x2 kernel to generate the respective 3x3 feature map. (C) An example of max pooling on a 4x4 feature map using 2x2 pooling. (Angermueller, Pärnamaa,

Autoencoder

Autoencoders are DNN models that can be broken up into an encoder function and a decoder function and is used in DL for dimension reduction (Kramer, 1991). The encoder consists of the input layer, followed by a set of hidden layers that reduce in size with each layer, and finally a latent layer. The latent layer is the smallest hidden layer of the network and represents the feature reduced space of a given input. The decoder consists of the latent layer, followed by a set of hidden layers that grow in size with each layer, and finally the output layer. The network is trained such that the output tries to reconstruct the input as close as possible. This forces the network to compress the input into a meaningful way such that it can be decoded efficiently to retain as much of the original information as possible. Because of this, not only are AEs used for dimension reduction, but they are also useful for reducing the noise in data. An example of a standard AE model is shown in **Figure 6**.

With advancements in DL frameworks, many variations of the standard AE have been developed. One common variation of the AE is the variational autoencoder in which the data is encoded into a latent multivariate distribution (D. P. Kingma & Welling, 2013). By representing the latent space as a multivariate distribution, variational autoencoders not only provide feature reduction, but they can also be used as generative models through sampling of the latent space. Another variation is the sparse autoencoder, where a penalty is applied to the latent layer to enforce a predefined level of sparsity (Ng, 2011). By enforcing sparsity within the latent layer, many of the latent nodes are forced to be inactive. This has been shown to improve the overall representation learning of the latent space, which can further improve downstream tasks utilizing the latent representations. Denoising autoencoders are another common framework in which the original input is corrupted by noise and the model learns to reconstruct the corrupted data back to

the original values (Goodfellow, Bengio, & Courville, 2016). By injecting noise into the input, denoising autoencoders have been shown to be even more robust to the noise within data.

Autoencoders have been widely used in computational biology for denoising, imputation, and feature reduction. Recent studies have used AE frameworks in order to impute single-cell RNA-Seq data (Talwar, Mongia, Sengupta, & Majumdar, 2018; Trong et al., 2020), which suffers from a high level of dropout noise. A study by Chen et al. constructed an AE model that was shaped using *a priori* knowledge of gene sets to mask weights by only connect genes as inputs to their respective gene sets in the first hidden layer, resulting in improved classification of tumor subtype on TCGA data (H.-I. H. Chen et al., 2018). Another group recently used a variational autoencoder trained on protein sequences to generate new sequences representing functional protein variants (Hawkins-Hooker et al., 2021).

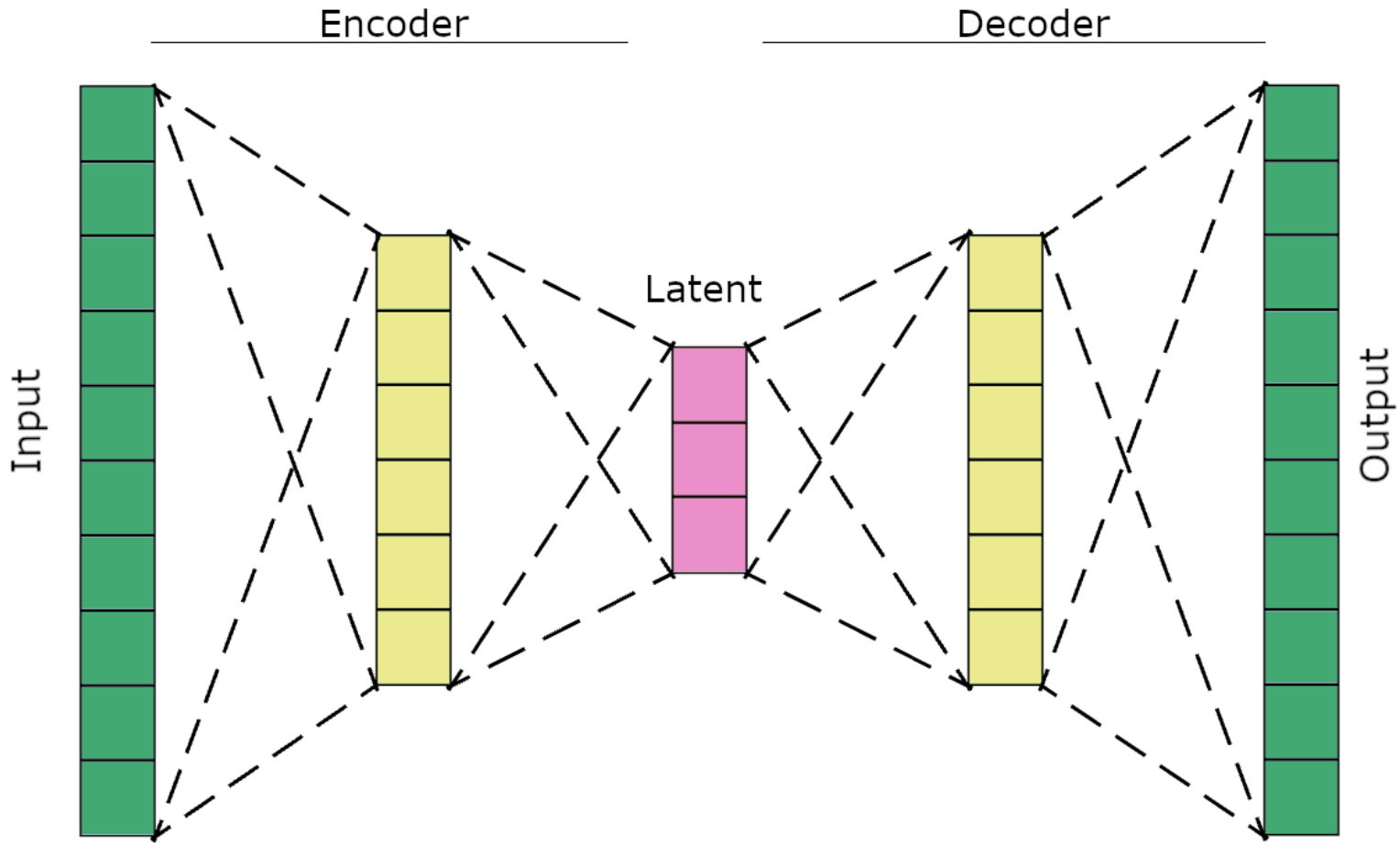


Figure 6. Architecture of an AE model. An input vector is passed through the encoder part of the neural network to a reduced latent space representation. The latent representation is then passed through the decoder part of the neural network to the output. An AE is trained to minimize the distance between the input and output values.

Generative Adversarial Networks

Generative adversarial networks (GANs) are a set of two DNNs that compete against each other (Goodfellow et al., 2014). The first DNN is a generator network that takes in a set of random prior values as an input and generates synthetic data close to the observed data of the used dataset. The second network is the discriminator network which takes in either a real or synthetic sample and predicts the probability that the input data is real. The networks are trained in an adversarial fashion, such that the generator's task is to try and fool the discriminator and the discriminator's task is to predict if an observed sample is real or synthetic. Both generator and discriminator networks are trained in an iterative fashion such that in each epoch, the discriminator is first trained on the generated and real samples and the network weights are updated. After the discriminator has been updated, the generator is updated using information on how the discriminator made its predictions. The cost functions for the discriminator and generator are shown below.

$$C_D = \frac{1}{n}\sum_{i}^{n} -\log[D(\boldsymbol{x_i})] - \log\left[1 - D(G\ (\boldsymbol{z_i})\right] \tag{2.24}$$

$$C_G = \frac{1}{n}\sum_{i}^{n} \log\left[D\left(G(\boldsymbol{z}_i)\right)\right] \tag{2.25}$$

Here n represents the number of real samples, z_i represents a vector of priors for the generator, x_i is a vector of a real data. $D(\boldsymbol{x_i})$ is the discriminator's prediction that x_i a real sample and $G(\boldsymbol{z_i})$ is the generated synthetic sample given the vector of prior noise $\boldsymbol{z}_i$. Once fully trained, the generator network can be used to generate large amounts of realistic data. An example of a standard GAN framework is shown in **Figure 7**. A common extension of GANs is the conditional generative adversarial network (CGAN) (Mirza & Osindero, 2014). In the CGAN architecture, additional side information is passed to both the generator and discriminator networks, allowing for the construction of synthetic data under different conditions.

Generative adversarial networks have become utilized in computational biology for tasks such as denoising and data generation. Using single-cell RNA-Seq, Ghahramani et al. used a GAN framework to generate synthetic expression data across multiple cell subtypes (A. Ghahramani, Watt, & Luscombe, 2018). Another study was able to model the class-switch recombination process in B-cells at a single-cell resolution using a CGAN model (Derek Reiman et al., 2021). Lastly, DeepHiC was published by Hong et al. based on a GAN framework which showed success in enhancing the resolution of low resolution Hi-C data (Hong et al., 2020).

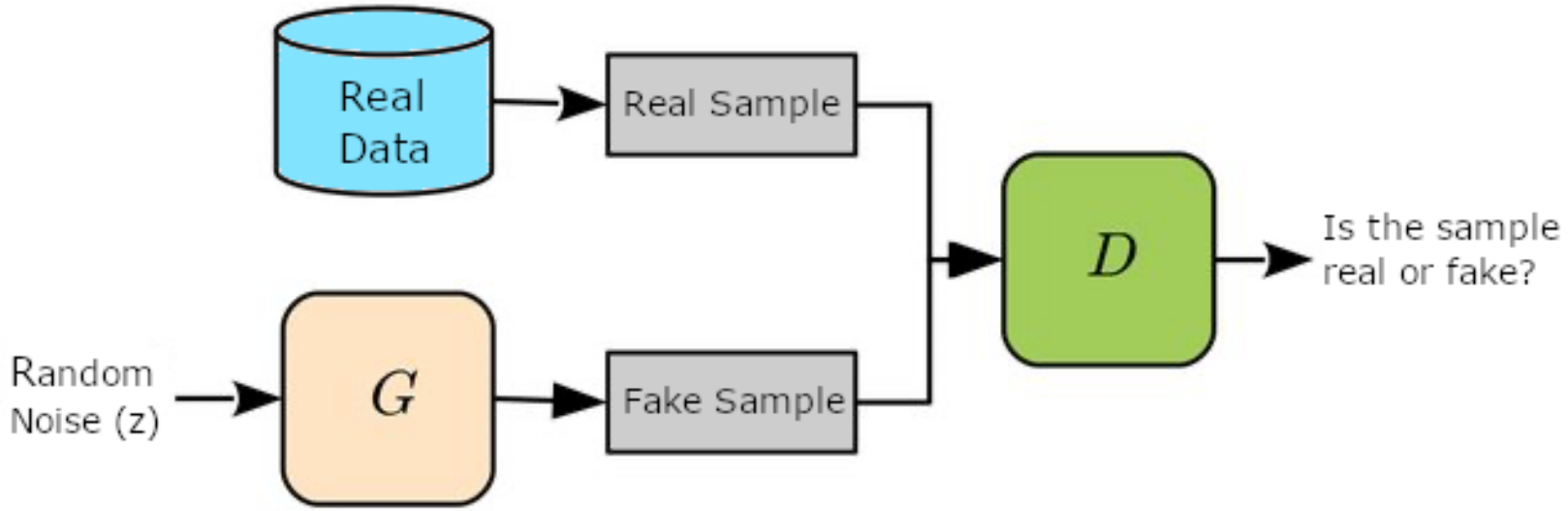


Figure 7. Architecture of a GAN model. Real samples are drawn from real data (blue). Fake samples are generated from a generator network (G) given a set of random noise priors. The discriminator predicts if the sample it is given as an input is real or fake.

Chapter 3

Deep Learning Frameworks for the Prediction of Host Phenotype

3.1 Introduction

One of the major objectives of microbiome studies is the identification of specific microbes related to changes in host phenotype. In the context of disease, the identification of these microbial taxa, often in the form of operational taxonomic units (OTUs), can facilitate earlier diagnoses, the development of microbial reconstitution (e.g., Probiotic) therapies (Preidis & Versalovic, 2009; Vindigni, Zisman, Suskind,

& Damman, 2016), and help further the understanding of the disease mechanism (Sun & Chang, 2014). For this task, methods can be organized into two different approaches: statistical and machine-learning.

Both parametric and non-parametric statistical analyses have been used to identify differentially abundant microbes between case and control groups using microbiome data. Parametric methods using generalized linear models such as edgeR (Robinson et al., 2010) and Deseq2 (Love et al., 2014) have been widely applied to metagenomic count data. Additionally, non-parametric methods have been used when handling microbial composition abundance. One widely used method, ANCOM (Mandal et al., 2015), uses the log-ratio of all pairs of microbes to test for a differences in means. Another, ALDEx2 (Fernandes et al., 2013) uses a Dirichlet-multinomial model to infer abundance from counts and uses the Wilcoxon rank-sum test to identify differentially abundant microbes. Although these methods have been useful in discovering biological insight to disease, as univariate analyses they cannot consider the complex nature of the microbiome community structure as a whole. In addition, a previous study has shown that these methods do not control the false discovery rate (Hawinkel, Mattiello, Bijnens, & Thas, 2017). Because of these limitations, the association of the microbes to a specific diseases has shown contradictory results across studies (Finucane, Sharpton, Laurent, & Pollard, 2014; Knights et al., 2011).

Recently, ML approaches have been advocated for, motivated by the findings that a microbial signature may be complex, involving simultaneous over- and under-representations of multiple microbial taxa at distinct taxonomic levels (Knights et al., 2011; T. Wang & Zhao, 2017). Previous studies have shown that random forest (RF), least absolute shrinkage and selection operator (LASSO), and support vector machines (SVMs) have the potential of identifying a microbial biomarker signature for the prediction of the host phenotype (Pasolli et al., 2016; Wingfield et al., 2016; Zhang et al., 2015). Additionally, DNNs have been proposed in the hope that DNNs could identify more complex relationships for host phenotype prediction, however the evaluation of DNNs is incomplete and DNNs were superior than other standard ML models (Ditzler et al., 2015; LaPierre et al., 2019). As a deep learning model, DNNs learn the intricate structure within data by iteratively changing their parameters through backpropagation. This type of

representation learning allows the model to intrinsically construct hierarchical feature representations from the raw data, removing the need for hand-engineered features (LeCun, Bengio, & Hinton, 2015). Recently, Ph-CNN was designed to integrate microbial abundance with phylogenetic information using a different deep learning model, a convolutional neural network (CNN) (Fioravanti et al., 2018). Ph-CNN clusters groups of microbes based on their phylogenetic patristic distance. This was inspired by findings from studies showing that constructing abundance of features using the hierarchical structure of the taxonomic tree can improve prediction performance (Oudah & Henschel, 2018; Qiu et al., 2015). However, although Ph-CNN uses phylogenetic distances to group microbes, it does not fully explore abundances at different hierarchies of the tree.

3.1.1 Problem Definition

In this chapter, we will present methods designed to improve both the prediction of host phenotype and the identification of disease related microbes.

1. Structurally integrating taxonomic and microbial abundance data in DNNs will improve prediction of host phenotype and facilitate the identification of associated microbial biomarkers at different taxonomic levels.

2. The use of multiple ML methods to evaluate and identify disease related microbes into a single ranked list will generate a more robust set of disease biomarkers.

3. Non-parametric modeling of microbiome data to generate synthetic microbiome data samples will improve downstream analyses such as host phenotype prediction.

3.1.2 Significance

The first work presented, PopPhy-CNN, is a novel deep learning framework that integrates phylogenetic spatial information to improve the prediction of host phenotype. PopPhy-CNN improves prediction performance in complex multi-class disease datasets and facilitates the identification of disease-related microbial biomarkers at different taxonomic. The second work presented, Meta-Signer, uses an ensemble of machine learning models to identify a robust list of disease-related microbes. The third work uses adversarial networks to generate realistic synthetic microbiome data. We show that the use of this synthetic data to augment microbiome datasets can be improve the prediction of host disease. Together, these methods improve on existing methods for host phenotype prediction and the identification of disease-related microbial features, potentially allowing for the development of stronger diagnostic tools for clinicians.

3.2 PopPhy-CNN: A Phylogenetic Tree Embedded Architecture for Convolutional Neural Networks to Predict Host Phenotype From Metagenomic Data

In this section, we present PopPhy-CNN, a CNN framework that integrates the spatial information captured by phylogenetic similarity with microbial abundance to improve host phenotype prediction. We will demonstrate that PopPhy-CNN outperforms state-of-the-art ML approaches in complex multi-class disease datasets. In addition, PopPhy-CNN facilitates the identification of disease related microbes at different taxonomic levels.

3.2.1 PopPhy-CNN Framework

The core of the PopPhy-CNN framework can be split into three main steps. In the first step, the microbial taxa found in the dataset are used construct a taxonomic tree. The leaves of the taxonomic tree are assigned their respective abundance values, and ancestral nodes are calculated as the sum of their children. Each populated tree is then represented as a two-dimensional matrix. In the second step, the matrices are used to train a CNN model for the prediction of host phenotype. Lastly, in the third step, the trained CNN model is used to identify areas of the taxonomic tree associated with different disease states. An overview of the framework is shown in **Figure 8**.

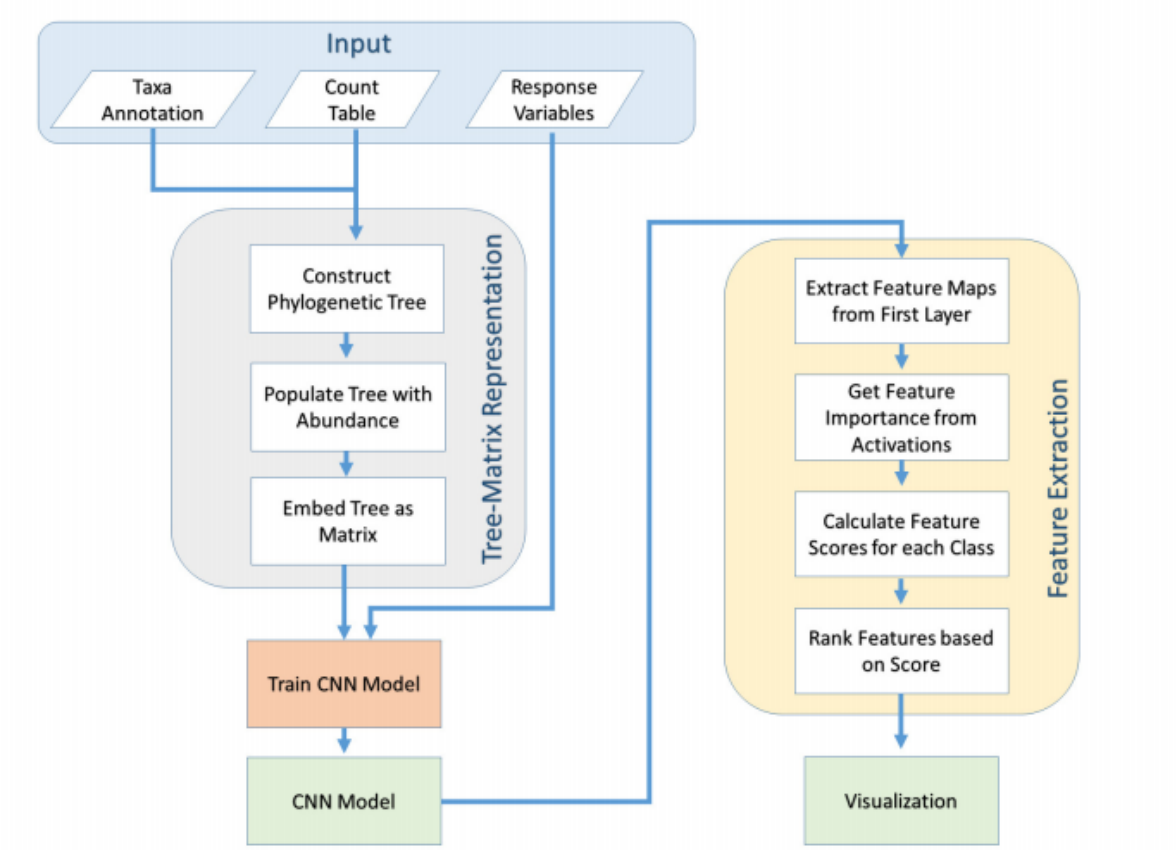


Figure 8. Flowchart of PopPhy-CNN. The annotated taxa and count table are used to create and populate a taxonomic phylogenetic tree. The tree is then represented as a matrix and used to train a CNN. Features are extracted from the trained model.

Tree-Matrix Representation

PopPhy-CNN uses NCBI's taxonomic tree of life as the backbone for all of its taxonomic trees. Since it is a taxonomic tree, there was no annotated distance between nodes and a constant distance of one between nodes in the tree is assumed. The tree is instead structured using ancestral nodes from both taxonomic groups and subgroups with no defined distances between nodes. Therefore, we consider the distance between any two nodes by the number of nodes between them. Using the microbial taxa found in the supplied dataset, the tree is pruned. In specific, we retain only leaf nodes annotated by observed microbial taxa as well as their ancestral nodes.

Once the tree has been pruned, we use the tree structure as a template to construct a populated tree for each sample. Specifically, for each sample, we begin by assigning each node with an abundance of zero. Then the observed taxa abundances are assigned to their respective nodes. Lastly, for each ancestral node, the sum of their children's abundance values is added to its own abundance. The algorithm for tree population is outlined in **Algorithm 1**.

Data: A phylogenetic tree $G = \{V,E\}$ and taxa abundance vector $\mathbf{x}$
Result: A populated phylogenetic tree $G = \{V,E\}$

```
for l from the maximum tree depth to 0 do
    for each node v in layer l do
        if the label of v is an OTU in vector x then
            assign node v the abundance of the OTU from vector x
        end
        if v has any children then
            add its children's abundances to the abundance of v
        end
    end
end
```

Algorithm 1. Tree Population

Next, we represent each sample's populated tree as a matrix according to **Algorithm 2**. An example of populating a tree and representing it as a matrix is shown in **Figure 9**.

Data: A populated phylogenetic tree $G = \{V, E\}$
Result: A matrix **M** containing the tree representation

Construct a zero matrix **M** with a row for each tree layer and columns equal to the largest number of nodes in any layer;
$C \leftarrow$ Root Node of G;
for *j from 0 to the number of layers of G* **do**
  $i \leftarrow 0$;
  $Q \leftarrow \{\}$;
  for *each node v in C* **do**
    $M(i, j) \leftarrow$ abundance of node v;
    Push children of node v into queue Q;
    $i \leftarrow i + 1$;
  end
  $C \leftarrow Q$;
end
Return **M**

Algorithm 2. Tree-Matrix Representation

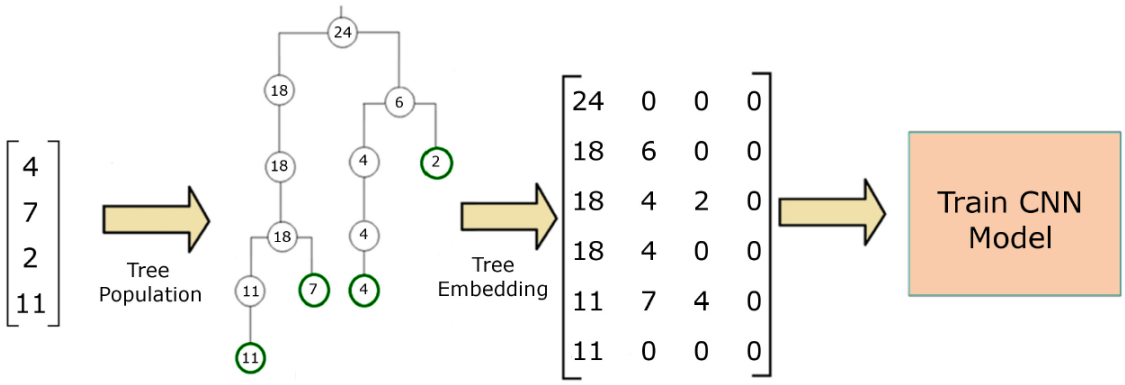


Figure 9. Populating taxonomic tree and matrix representation. Microbial taxa are used to prune a taxonomic tree. The abundance data is used to populate the tree, and then the tree is represented as a matrix.

Convolutional Neural Network Model

CNN models are neural network models based on the visual cortex and are designed to capture local spatial pattern (Fukushima, 1980). They have been widely used in natural language processing and image processing tasks with great success (Tao Chen, Xu, He, & Wang, 2017; Zhiqiang & Jun, 2017). Standard CNNs are composed of convolutional layers followed usually by at least one fully connected layer. Each convolutional layer is composed of multiple kernels, each of which transforms an input matrix M into a feature map of velocities through a convolutional operation. The feature maps composed of these velocities are then passed through a non-linear activation function and subsampled through max or mean pooling to give a matrix of activations. An overview is shown in **Figure 10.**

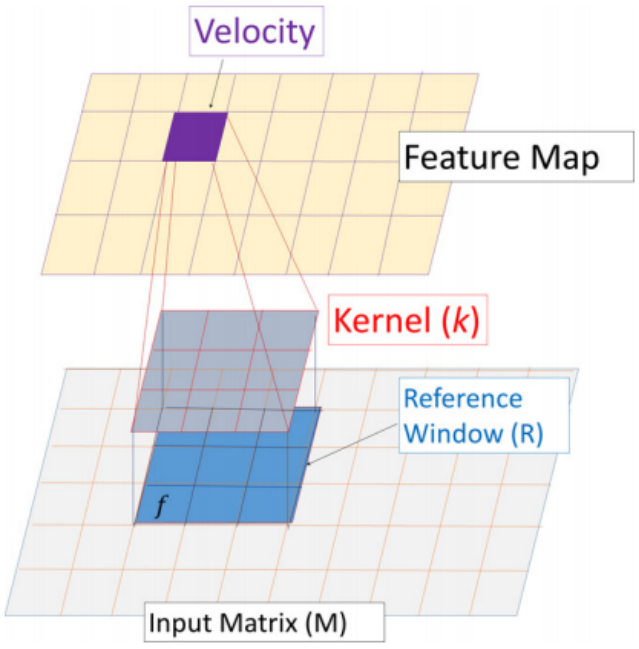


Figure 10. A kernel k slides over the input matrix. Each position in the feature map contains a velocity which is the element wise sum of the Hadamard product between k and a submatrix of $\boldsymbol{M}$. We call this submatrix a reference window and denote it as $\boldsymbol{R}$.

For a given kernel k with weights $\boldsymbol{W}^{(k)}$ of size m x n and input $\boldsymbol{M}$, the velocity of point (i, j) is calculated as:

$$vel^{(k)}(i,j) = \sum_{r=0}^{m}\sum_{s=0}^{n} \boldsymbol{M}(i+r, j+s) * \boldsymbol{W}^{(k)}(m-r, n-s) \tag{3.1}$$

PopPhy-CNN's architecture consists of two convolutional layers followed by a single fully connected layer and a single output layer. The first convolutional layer contains a rectangular filter to scan areas of local features in the input matrix. The second convolutional layer consists of a single 1x1 kernel to collapses the set of feature maps from the first convolutional layer into a single feature map, reducing the number of network parameters. Each layer uses the exponential linear unit (ELU) activation function. The ELU activation function, shown in **Equation 3.2**, has been shown to improve classification accuracy while also fixing the vanishing gradient problem often observed in deep neural networks (Clevert, Unterthiner, & Hochreiter, 2015).

$$ELU(x) = \begin{cases} x & x > 0 \\ \alpha(e^x - 1) & x \leq 0 \end{cases} \tag{3.2}$$

Finally, the softmax activation function was applied to the output layer for class prediction. The softmax function normalizes the output velocities into a probability distribution in order to predict the probability that a specific sample belongs to each class. Given a vector of velocities $\boldsymbol{y}$, the softmax activation of each velocity is calculated as shown in **Equation 3.3.**

$$Softmax(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}} \tag{3.3}$$

The model was trained using the ADAM optimizer (D. Kingma & Ba, 2014) and a weighted negative log loss function to help address class imbalance. To prevent overfitting, we regularize the networks using both L_1 and L_2 normalization penalties on the weights. This resulted in the overall loss function is shown in **Equation 3.4**.

$$Loss = -\left(\frac{n_{total}}{n_c}\right)\ln(a_c) + \lambda_1 \sum_{l \in L} |\boldsymbol{W}_l| + \lambda_2 \sum_{l \in L} \|\boldsymbol{W}_l\|_2 \quad (3.4)$$

Here, given an input whose true label is c, $n_{total}$is the total number of samples in the dataset, n_c is the number of samples for class c, λ_1 and λ_2 are the L_1 and L_2 regularization parameters respectively to penalize the weights **W** for each layer in the set of layers *L*. In addition, we used dropout in our network over the fully connected layers in order to help prevent overfitting.

Feature Extraction

A previous study has shown that using feature maps captured by CNN models as features for other machine learning models yielded better results than using the raw features (Athiwaratkun & Kang, 2015). Therefore, we focus on the post analysis of the feature maps generated by the first convolutional layer to identify microbial taxa associated to disease status. To do this, we take the feature maps generated by a kernel k across all the samples for a specific class c in the training set. For each of these feature maps, we take the positions of a proportion of maximum values specified by a given hyper-parameter, θ_1. We then select the maximums which were found in at least a proportion, θ_2, of the samples for that class. For each position selected, we trace its location in the feature map back to the submatrix of the input M from which it was calculated and call this matrix *R* the reference window of that position, as shown in **Figure 10**. Every position (*i*, *j*) of a reference window represents some node *v* from the phylogenetic tree with a taxonomic label, *f.* We calculate the importance of each feature *f* given the reference window $\boldsymbol{R}$ for sample *S* as its proportion of the velocity.

$$I_s^{(k)}(f|\boldsymbol{R}) = \frac{\boldsymbol{W}^{(k)}(i,j) * \boldsymbol{R}_S\,(i,j)}{\sum\,(|\boldsymbol{W}^{(k)}| \odot \boldsymbol{R}_s)} \qquad s.t.\,\boldsymbol{R}(i,j) \leftrightarrow f \quad (3.5)$$

Some taxa may score highly in a small subset of samples but may not be important considering all of the samples. To identify consistently important features, we calculate the mean importance value of a feature *f* across all samples in class *c* given a single reference window ***R*** and kernel *k*.

$$I_c^{(k)}(f|\boldsymbol{R}) = \frac{\sum_{s \in c} I_s^{(k)}\ (f|\boldsymbol{R})}{n_c} \tag{3.6}$$

Here n_c represents the number of samples in class *c*. Since a feature can be scored in multiple reference windows within and across kernels, we select the single importance value of *f* as the maximum over all reference windows containing *f* across all kernels, *k*.

$$I_c\ (f) = \max_{\boldsymbol{R},k}\{I_c^{(k)}(f|\boldsymbol{R})\} \tag{3.7}$$

Lastly, we assigned a final feature score *S* for a feature from the perspective of class *c* as the difference of the feature importance using all the samples with class *c* and the feature importance using all the samples not with class *c*, creating a list of feature scores for each class. The algorithm for feature extraction and evaluation from the CNN model is shown in **Algorithm 3.**

$$S_c(f) = I_c(f) - I_{\bar{c}}\ (f) \tag{3.8}$$

Data: A set of inputs where each input M is a matrix representation of tree $G = \{V,E\}$ with class c, a trained CNN model with kernels k of weights $W^{(k)}$, and two filtering parameters θ_1, θ_2

Result: A list of taxa scores $Sc(f)$ for each class

Z $\leftarrow$ zero matrix with dimensions $|c| \times |K| \times |V|$;
for *each sample s with class c and each kernel k* **do**
  Generate and vectorize the feature map;
  for *each index ℓ_1 of the top (θ_1 * $|V|$) values* **do**
    Increment $Z(c, k, \ell_1)$ by 1;
  end
end
for *each class c, kernel k, and sample $s \in c$ with input M* **do**
  for *each index ℓ_2 in the top θ_2 values of Z(c, k,:)* **do**
    Find the submatrix $R \in M_s$ used to calculate ℓ;
    for *each position (i, j) in R where R(i,j) $\equiv$ node $v \in V$* **do**
      $f \leftarrow$ the taxa label of v;
      $I_s^{(k)}(f \mid R) \leftarrow \frac{W(i,j)*R(i,j)}{\sum |W^{(k)}| \odot R}$;
    end
    $I_c^{(k)}(f \mid R) \leftarrow$ mean of $I_s^{(k)}(f \mid R)$ for $s \in c$;
  end
end
$S_c(f) \leftarrow \max_{R,k} I_c^{(k)}(f \mid R) - \max_{R,k} I_{\bar{c}}^{(k)}(f \mid R)$;
Return the set of scores $S_c(f)$ for each class c ;

Algorithm 3. CNN Feature Evaluation

3.2.2 Experiments and Results

Data Used in Evaluation

We used nine binary class publicly available disease related microbiome datasets to evaluate PopPhy-CNN. Three datasets were obtained from the MetaML package (Pasolli et al., 2016): cirrhosis (N. Qin et al., 2014), obesity (Le Chatelier et al., 2013), and type 2 diabetes (Karlsson et al., 2013; J. Qin et al., 2012). These datasets contained microbial taxa abundance at both genus and species levels. The six other datasets were taken from an IBD study (Sokol et al., 2017). The dataset was separated into three disease categories: Crohn's disease (CD), ileal Crohn's disease (iCD), and ulcerative colitis (UC). Each dataset was further broken into two sets where one set constitutes patients with who were in remission (r) and one set with patients whose condition was flaring (f). The number of samples in features for each dataset is shown in **Table I**.

TABLE I. SUMMARY OF BINARY CLASS DATASETS IN POPPHY-CNN EVALUATION

	Case	Control	Genus		Species	
			OTUs	Nodes	OTUs	Nodes
Cirrhosis	114	118	184	428	542	933
T2D	223	227	214	478	606	1054
Obesity	164	89	181	426	5465	872
CDf	60	38	224	388	-	-
CDr	76	38	219	386	-	-
iCDf	44	38	213	373	-	-
iCDr	59	38	219	386	-	-
UCf	41	38	213	384	-	-
UCr	44	38	203	361	-	-

We also evaluated PopPhy-CNN using three multi-class real datasets. For the first dataset, we stratified the obesity dataset into three groups using BMI rather than two. For the second dataset, we combined all six IBD datasets into a single dataset with seven classes. The third dataset was constructed by combining the cirrhosis, type 2 diabetes, and obesity datasets together as well as with a colorectal cancer dataset (Zeller et al., 2014) and a different IBD dataset (J. Qin et al., 2010) We further evaluated PopPhy-CNN on large synthetic datasets containing 3,5,7, and 9 classes. For the construction of synthetic data, we used the R package SparseDOSSA (Ren B, 2020). A summary of the multi-class datasets is shown in **Table II.** Principal Coordinates analysis (PCoA) plots of the IBD, Multi-Disease, and Syn9 datasets using Bray-Curtis dissimilarity are shown in **Figure 11.**

Lastly, we construct two binary synthetic datasets for evaluating the robustness of PopPhy-CNN. The smaller dataset (SynA) contains 750 samples and 500 features. The larger dataset (SynB) contains 1500 samples and 1000 features.

TABLE 2. SUMMARY OF MULTI-CLASS DATASETS IN POPPHY-CNN EVALUATION

	Class Samples	OTUs	Nodes
Obesity	164, 114, 89	465	872
IBD	38, 60, 76, 44, 59, 41, 44, 89	301	412
Multi-Disease	488, 118, 232, 164, 89, 21, 4, 48, 26, 13	772	1318
Syn3	224, 255, 246	500	992
Syn5	224, 255, 246, 272, 240	500	971
Syn7	224, 255, 246, 272, 240, 246, 250	500	978
Syn9	224, 255, 246, 272, 240, 246, 250, 260, 269	500	978

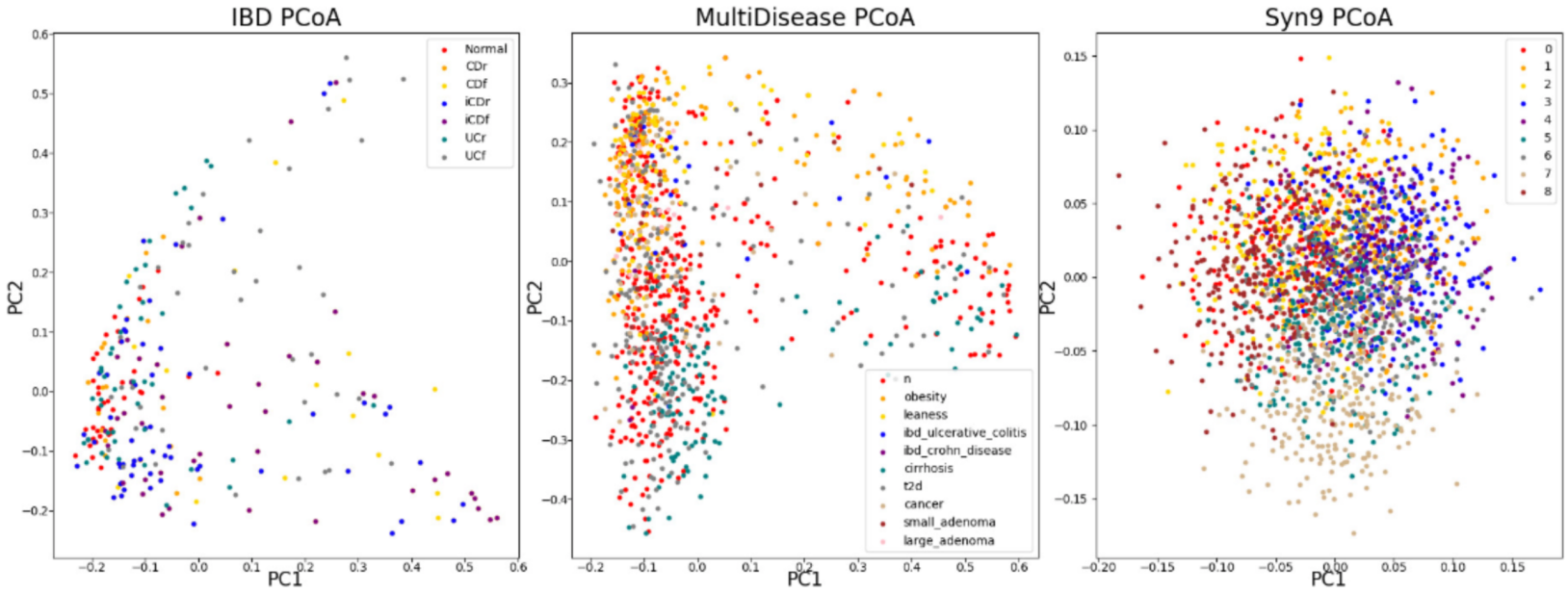


Figure 11. Principal Coordinate Analysis (PCoA) plots of the multi-class IBD, Multi-Disease, and Syn9 datasets. PCoA plots showing the complexity of multi-class disease data. Each point represents a sample which is colored by its respective class.

Evaluation of Host Phenotype Prediction

PopPhy-CNN was benchmarked using 10-fold cross-validation against RF, SVM, LASSO, a multi-layer perceptron neural network (MLPNN) with two fully connected layers, a 1D-CNN model using one convolutional layer with two fully connected layers, and Ph-CNN, which was designed using information of the phylogenetic tree. Data were min-max normalized between 0 and 1 before training. For evaluation of binary datasets, we use the area under the receiver operating characteristic curve (AUC-ROC), area under the precision-recall curve (AUC-PR), Matthew's correlation coefficient (MCC) and F1 score. For the evaluation of multiclass datasets, we only use MCC.

We observe that PopPhy-CNN is competitive in real world binary datasets, however it is not superior to RF models, which have often been considered the state-of-the-art machine learning method in microbiome studies. Prediction evaluation on binary datasets is shown in **Table II**. However, when we increase the complexity of the dataset by introducing more classes and stratifying across different disease states, we see that PopPhy-CNN is much more robust than RF models and PopPhy-CNN scales much better as the number of classes increases. Prediction evaluation of multi-class datasets is shown in **Table IV** and **Table V**.

TABLE III. POPPHY-CNN EVALUATION OF BINARY DATASETS

		PopPhy-CNN	RF	SVM	LASSO	MLPNN	1D-CNN	Ph-CNN
Cirrhosis	AUC-ROC	0.901	**0.928**	0.888	0.872	0.861	0.898	-
	AUC-PR	0.914	**0.927**	0.899	0.886	0.875	0.913	-
	MCC	0.610	**0.731**	0.568	0.548	0.568	0.695	-
	F1-Score	0.798	**0.858**	0.772	0.757	0.776	0.841	-
T2D	AUC-ROC	0.681	**0.718**	0.510	0.659	0.645	0.666	-
	AUC-PR	0.692	**0.737**	0.555	0.671	0.658	0.673	-
	MCC	0.231	**0.297**	-0.024	0.246	0.185	0.204	-
	F1-Score	0.611	**0.643**	0.459	0.614	0.586	0.595	-
Obesity	AUC-ROC	0.589	**0.627**	0.568	0.493	0.563	0.571	-
	AUC-PR	0.414	0.476	0.457	**0.637**	0.431	0.478	-
	MCC	**0.181**	0.079	0.008	-0.014	0.081	0.078	-
	F1-Score	**0.587**	0.558	0.524	0.508	0.529	0.529	-
CDr	AUC-ROC	0.799	**0.897**	0.878	0.828	0.805	0.837	0.714
	AUC-PR	0.895	**0.953**	0.947	0.924	0.912	0.923	-
	MCC	0.433	0.562	0.571	0.312	0.427	0.494	0.241
	F1-Score	0.726	0.796	**0.804**	0.677	0.735	0.768	0.756
CDf	AUC-ROC	0.926	**0.982**	0.931	0.931	0.932	0.940	0.808
	AUC-PR	0.957	**0.990**	0.967	0.965	0.965	0.971	-
	MCC	0.706	0.758	**0.790**	0.700	0.744	0.783	0.630
	F1-Score	0.847	0.875	**0.888**	0.844	0.867	**0.888**	0.836
iCDr	AUC-ROC	0.866	**0.898**	0.879	0.844	0.858	0.852	0.768
	AUC-PR	0.92	**0.948**	0.934	0.884	0.923	0.920	-
	MCC	0.613	0.640	**0.647**	0.614	0.611	0.609	0.556
	F1-Score	0.812	**0.822**	0.820	0.808	0.800	0.787	0.731
iCDf	AUC-ROC	0.950	**0.968**	0.848	0.951	0.951	0.959	0.842
	AUC-PR	0.958	**0.978**	0.900	0.962	0.965	0.970	-
	MCC	**0.851**	0.842	0.609	0.830	0.805	0.746	0.704
	F1-Score	0.917	**0.918**	0.764	0.904	0.887	0.853	0.845
UCr	AUC-ROC	0.855	**0.903**	0.740	0.828	0.837	0.841	0.722
	AUC-PR	0.843	**0.931**	0.759	0.863	0.824	0.820	-
	MCC	**0.696**	0.656	0.601	0.551	0.579	0.654	0.445
	F1-Score	**0.837**	0.821	0.770	0.752	0.785	0.817	0.745
UCf	AUC-ROC	0.946	0.960	0.480	0.920	**0.969**	0.935	0.822
	AUC-PR	0.957	**0.975**	0.632	0.945	0.972	0.943	-
	MCC	0.688	**0.812**	0.303	0.657	0.749	0.770	0.668
	F1-Score	0.829	**0.896**	0.532	0.806	0.856	0.871	0.825

TABLE IV. POPPHY-CNN EVALUATION OF REAL MULTI-CLASS DATASETS

	PopPhy-CNN	RF	MLPNN	1D-CNN
Obesity (3)	**0.159**	0.089	0.048	0.086
IBD (7)	**0.158**	0.073	0.114	0.149
Multi-Disease (10)	**0.343**	0.316	0.314	0.297

TABLE V. POPPHY-CNN EVALUATION OF SYNTHETIC MULTI-CLASS DATASETS

	# of Classes	PopPhy-CNN	RF
Syn3	3	**0.884**	0.814
Syn5	5	**0.871**	0.712
Syn7	7	**0.863**	0.650
Syn9	9	**0.835**	0.583

Computational Complexity and Robustness

To evaluate the complexity of our model, we recorded the amount of time it took to train a single model using varying sample and input feature sizes. We created synthetic datasets of 500, 1000, 2500, 5000, and 10,000 samples, each with 500 features. Additionally, we created datasets with 500, 750, 1000, 1250, and 1500 features, each with 500 samples. The average training for training a single model over 10-fold cross-validation was calculated and is shown in **Figure 12**.

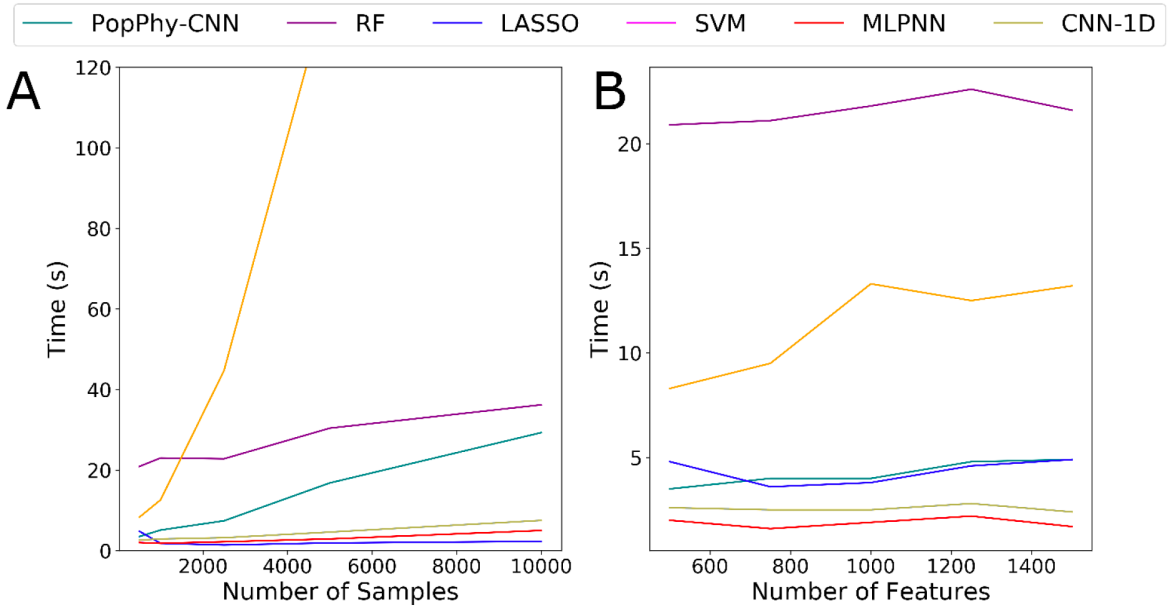


Figure 12. Time complexity for machine learning models based on (A) number of samples and (B) number of features. Running time in seconds of PopPhy-CNN, RF, LASSO, SVM, MLPN, and CNN-1D models are shown under increasing number of samples and features.

For consistency, all neural network models were trained to 50 epochs using an NVIDIA Titan XP GPU. We observed that RF models had the largest overhead and that SVM models scaled the worst. We also observed that PopPhy-CNN increased more than the other neural network-based models. This is due to the increased input space of the matrix representations of the populated trees. However, despite this, PopPhy-CNN was still observed to train faster than RF models.

We then evaluated how the parameter size of the neural network models scaled based on the original input size. The MLPNN and CNN-1D models scaled almost identically while PopPhy-CNN scaled at a rate 5.08 times faster than the other two. This was expected since the matrix representation was also shown to scale in size on average 4.93 times the number of nodes in the tree used for the matrix representation. The scaling of the three neural network-based models is shown in **Figure 13.**

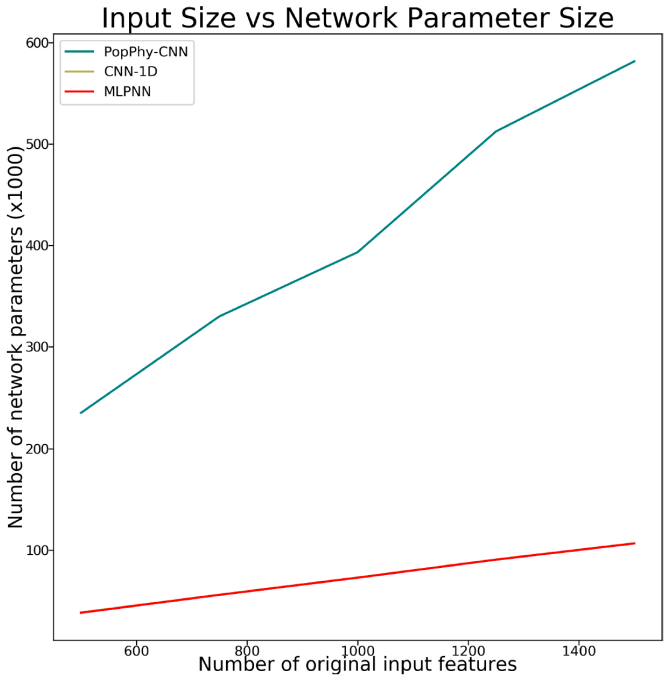


Figure 13. Number of parameters in PopPhy-CNN, CNN-1D, and MLPNN models based on input size. Number of total network parameters for PopPhy-CNN, MLPNN, and CNN-1D models as the input space of the data increases.

Lastly, we tested the robustness of PopPhy-CNN using 5-fold and 3-fold cross-validation to increase the size of the held-out test set while reducing the size of the training set. We performed this analysis on the cirrhosis dataset as well as two synthetic datasets, SynA and SynB. When holding out 20% using 5-fold cross-validation, the AUC-ROC for cirrhosis was 0.916 (2.66% decrease), for SynA was 0.928 (0.24% decrease), and for SynB was 0.927 (0.24% decrease). When using 33% as held out data during 3-fold cross-validation, the AUC-ROC for cirrhosis was 0.917 (2.66% decrease), for SynA was 0.900 (2.92% decrease), and for SynB was 0.904 (2.78% decrease). In total, this shows that PopPhy-CNN is robust to using larger sets of held-out data even for datasets with moderate size.

Evaluation of Extracted Features

We used the Cirrhosis, Obesity, and T2D datasets at the genus level for and extracted feature scores for each dataset. We used the values $\theta_1 = 0.01$ and $\theta_2 = 0$ in the feature extraction method. This means that we consider only the top 1% of values in each feature map of each sample. This allows a fair baseline comparison across the datasets from which the tuning of the parameters may lead to stronger feature evaluations. We constructed a single ranked list using the absolute value of a feature's score between the two classes as the ranked value.

To evaluate the informativeness of the extracted features, we tested if they could be used improving prediction in SVM. To do this, we trained SVM models using the top ranked features from the original OTUs ranging from the top 5, 10, 15, 20, and 25. We compared PopPhy-CNN's extracted features with the ranked lists based on signal-to-noise ratio, Wilcoxon test p-values, and the average feature rankings from the RF models trained using 10-fold cross-validation. SVM models were trained using 10-fold cross-validation and results for the three datasets are shown in **Figure 14.**

For the cirrhosis dataset, we observed that the higher ranked features of PopPhy-CNN performed best, followed by the features identified by RF. The features identified by the Wilcoxon rank-sum test were

not stable and showed a decrease in prediction performance before increasing afterwards. In the obesity dataset, we observe that PopPhy-CNN and the Wilcoxon rank-sum features perform similarly, however the RF features perform poorly. In the T2D dataset, all models performed about the same. PopPhy-CNN was the only method to perform competitively in all three datasets.

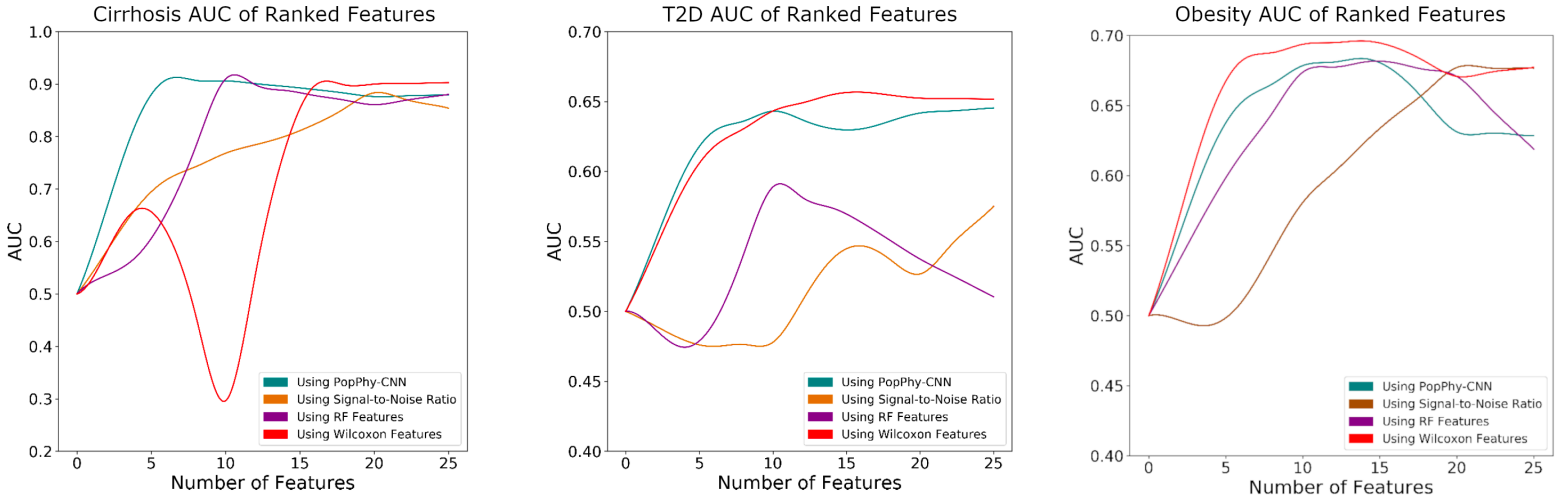
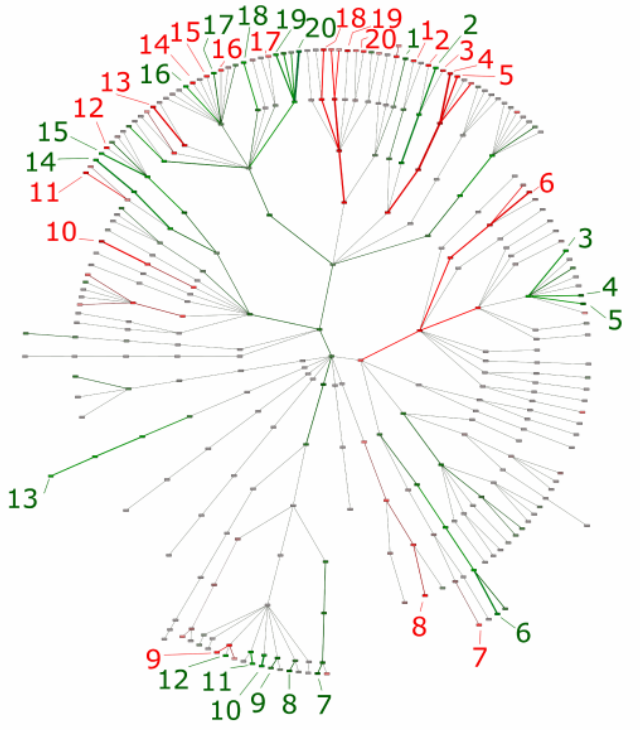


Figure 14. Benchmarking of top 25 features extracted from PopPhy-CNN for Cirrhosis, Obeisty, and T2D datasets. Features extracted from PopPhy-CNN (teal) are benchmarked against features found by RF (purple), signal-to-noise ratio (brown), and a Wilcoxon rank-sum test (red).

Extraction of Biologically Relevant Features

Lastly, we analyzed the scored the features of the cirrhosis dataset for both healthy and disease cases (see Feature Extraction). Using Cytoscape, we visualized the taxonomic tree and annotated nodes and edges as associated with healthy and cirrhosis disease states. The score for each node was calculated as shown in **Equation 3.8** and the score for an edge was the mean between the two connected nodes. The visualization of the tree and top scoring leaf nodes is shown in **Figure 15.**

In the cirrhosis patients, PopPhy-CNN identified *Veillonella*, *Streptococcus*, *Haemophilus*, *Prevotella*, and *Actinomyces* as associated microbial biomarkers. In the healthy subjects, *Alistipes*, *Rumminococcus*, *Roseburia*, *Clostridium*, and *Bilophila* were identified. Many of the top ranked features were also identified in the original study (N. Qin et al., 2014). Additionally, a separate study on a different cohort of subjects with cirrhosis found similar results, showing that *Streptococcus*, *Veillonella*, and *Prevotella* were associated with Interleukin-23 (IL-23) and Interleukin-2 (IL-2), both of which have been shown to be associated with inflammatory gut disease (Bajaj et al., 2012; Duvallet, Semerano, Assier, Falgarone, & Boissier, 2011). We also observed cases in which ancestral nodes had much larger scores than their children, possibly implying that no single child feature was discriminative between disease states, but that the collection of them was. For example, the family Bifidobacteriaceae had a score of 0.292 while its children had much lower scores. Although this family of microbes was not identified as important in the original study, a different study has shown that microbes in the Bifidobacteriaceae family produce glutamate dehydrogenase, a protein found to have higher expression levels in patients with Cirrhosis (Wei et al., 2016). Therefore, the aggregation of all the genera under Bifidobacteriaceae should be more discriminative than any single genus, as observed in the feature analysis extracted by PopPhy-CNN.



	Cirrhosis		Healthy	
Node	Taxa	Score	Taxa	Score
1	Acidaminococcus	0.312	Staphylococcus	-0.358
2	Selenomonas	0.547	Megamonas	-0.538
3	Mitsuokella	0.302	Escherichia	-0.575
4	**Veillonella**	**0.861**	Klebsiella	-0.370
5	Megasphaera	0.412	Citrobacter	-0.349
6	**Haemophilus**	**0813**	**Bilophila**	**-0.839**
7	Campylobacter	0.295	Cellulophaga	-0.333
8	Fusobacterium	0.401	Odoribacter	-0.302
9	Prevotella	0.336	Barnesiella	-0.234
10	**Actinomyces**	**1.035**	**Alistipes**	**-0.681**
11	Atopobium	0.638	Parabacteroides	-0.478
12	Eggerthella	0.567	Paraprevotella	-0.492
13	**Clostridium**	**0.803**	Akkermansia	-0.417
14	Blautia	0.478	**Collinsella**	**-0.658**
15	Dorea	0.335	**Adlercreutzia**	**-0.667**
16	Anaerostipes	0.313	Roseburia	-0.590
17	Flavonifractor	0.215	Coprococcus	-0.485
18	Lactobacillus	0.661	Oscillibacter	-0.518
19	**Granulicatella**	**0.811**	Subdoligranulum	-0.458
20	Weissella	0.322	**Ruminococcus**	**-1.189**

Figure 15. Visualization of cirrhosis features identified by PopPhy-CNN associated with cirrhosis (red) and healthy (green) states.

The table shows high ranking leaves of annotated subtrees. The top 5 ranked features are bolded.

3.2.3 Conclusion

PopPhy-CNN integrates taxonomic structure with microbial abundance features to improve prediction performance. It has been shown to be competitive in binary disease datasets and outperforms state-of-the-art methods in more complex multi-class datasets. In addition, it extracts associated microbial features at various levels of taxonomy, providing additional biological insights for downstream analyses. PopPhy-CNN is implemented in Python3 and is freely available at www.github.com/YDaiLab/PopPhy-CNN.

3.3 MetaSigner: Metagenomic Signature Identifier Based on Rank Aggregation of Features

Even with the success of ML models for host phenotype prediction, it is a challenging task for users to determine what is the best ML model and how many features are needed in order to achieve robust prediction, especially on external validation datasets. In addition, each ML algorithm may generate different feature importance rankings (T. Wang & Zhao, 2017; Zhang et al., 2015), complicating the decision on a consistent and informative signature for the host phenotype of interest.

3.3.1 MetaSigner Framework

Meta-Signer uses RF, SVM, Logistic Regression, and MLPNN models to evaluate importance of each microbial taxon and generates a ranked list of microbial features per model. It aggregates all the ranked lists using a procedure "RankAggreg" based on the cross-entropy method or the genetic algorithm (Pihur, Datta, & Datta, 2009). Finally, Meta-Signer reports the top-ranking features specified by the user and generates the ML models using these features. Meta-Signer's workflow is shown in **Figure 16**.

User Input

Meta-Signer requires three input files and allows for an optional fourth file in order to run. The inputs to Meta-Signer are:

1. A tab separated file of taxa abundance values where each row represents a taxon and each column represents a sample (required)

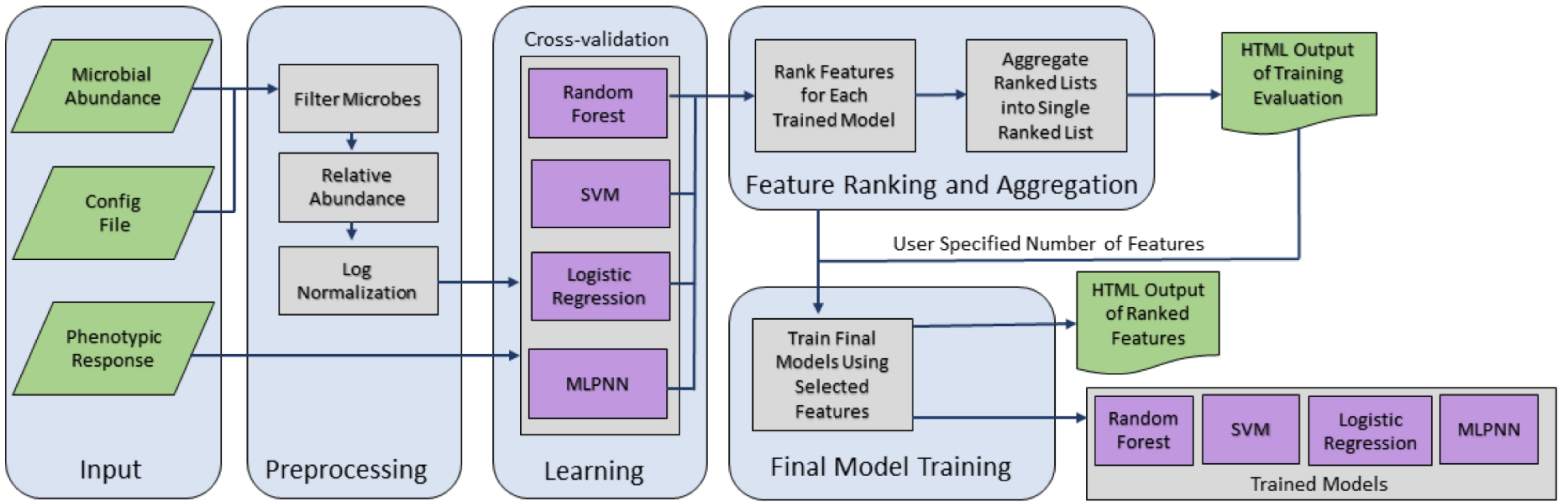


Figure 16. The Meta-Signer workflow. Large rounded rectangles represent different modules of the workflow. Microbial abundance is preprocessed and filtered, and then used to train ML models. Features are ranked for each model and an overall aggregated feature ranking is constructed. Meta-Signer generates portable, user-friendly HTML files for visualization as well as ML models trained on a subset of high ranking features. SVM, support Vector Machine; MLPNN, multiple-layer perceptron neural network; ML, machine learning.

2. A line separated list of response values where each row represents the phenotypic response of a sample where the first column in the abundance table should be the taxonomic identification of the taxon (required)
3. The run configuration file with user specified parameters (required)
4. The model parameters for the neural network architectures in JSON format (optional)

If the fourth optional file is not found, Meta-Signer will tune the parameters and save them for later use.

Machine Learning Models

Meta-Signer includes three classic ML models (RF, Linear SVM, Logistic Regression), as well as an MLPNN model. The classic ML models are implemented using the "scikit-learn" Python package (Pedregosa et al., 2011), and the MLPNN model is implemented using TensorFlow (Abadi et al., 2016).

RFs are decision tree learning models trained in an ensemble fashion, taking the average of the ensemble to give a robust decision tree (Ho, 1995). While growing each tree, a decision is made at each node by selecting the feature from a random subset of features that best splits the data into two subsets based on the Gini impurity of each subset. Given a set of data points with k classes, let p_i be the proportion of samples of class i for $i \in \{1...k\}$. The Gini impurity of the set is calculated as

$$I_G(p) = 1 - \sum_{i=1}^{k} p_i^2 \tag{3.9}$$

Once trained, features are then extracted by evaluating the mean decrease impurity. For each node, the importance of the feature node being split on the decision tree is calculated as the decrease in Gini impurity before and after the split. This value is then weighted by the proportion of total samples that were split upon

that node. A feature's importance is then calculated by averaging the weighted importance values of nodes that split using that feature across all trees in the ensemble.

SVMs are supervised ML models that learn the best hyper-plane to separate two classes of data (Cortes & Vapnik, 1995). For MetaSigner, we implement linear SVMs in which a set of weights (*w*) and an intercept (*b*) will be learned. The class of the sample x_i can then be determined as

$$\hat{y} = sign\ (\boldsymbol{w}^T \boldsymbol{x}_i + b) \tag{3.10}$$

Since the weights can be used to rank the importance of features, we used the linear SVMs in MetaSigner for feature extraction. Once trained, features can be ranked using the absolute value of the learned weight parameters.

Logistic Regression fits a logistic function to estimate the prob- ability of binary classification; however, it can be extended to multi-class scenarios (John Lu, 2010). The model predicts the probability of a sample $\boldsymbol{x}_i$ being the positive class as,

$$\hat{y}_i = \frac{1}{1 + e^{-(\boldsymbol{\beta} \boldsymbol{x}_i + \beta_0)}} \tag{3.11}$$

where $\boldsymbol{\beta}$ are the weight parameters which are learned and β_0 is the bias value. L_1 regularization is used in order to penalize the absolute value of the weights, eliminating a portion of the features to create a sparse model. Given a set of samples x_i $(i = 1, \ldots, n)$ where each sample has m features and a binary class label y_i, the model minimizes the cost,

$$C = \frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1-y)\log(1-\hat{y}_i)] + \lambda \sum_j |\beta_j| \tag{3.12}$$

where the weight parameters are penalized with the regularization parameter λ. Once trained, the β values are used to rank features based on their absolute value.

Neural networks are consisted of multiple layers of nodes that are fully connected with edges constituting weights (Nielsen, 2015). The values of a hidden layer are a linear combination of the values from the previous layer which is passed through a non-linear activation function. More explicitly, the values of a hidden layer h_l is calculated as,

$$h_l = \varphi\,(\boldsymbol{W}_l^T \boldsymbol{h}_{l-1} + \boldsymbol{b}_l) \tag{3.13}$$

where h_{l-1} are the values from the previous hidden layer, $\boldsymbol{W}_l$ are the weights connecting $\boldsymbol{h}_{l-1}$ to $\boldsymbol{h}_l$, $\boldsymbol{b}_l$ is the vector of bias values, and φ is a non-linear activation function. Meta-Signer uses the Rectified Linear Unit activation function for hidden layers and the softmax activation function on the output layer. The entire network is trained using a single loss function,

$$C = -\log(a_c) + \lambda \sum_{l \in L} \|\boldsymbol{W}_l\|_2 \tag{3.13}$$

Here a_c is the predicted softmax probability of a sample's true class c. The second term performs L_2 regularization on the network weights and is penalized by λ. The MLPNN features were evaluated using a method by Olden et al. by taking the running product of all the weight matrices in the learned networks (Olden et al., 2004). The product results in a matrix that has a column for each class and a row for each

feature, and the value at a given index is that feature's cumulative impact for that class. We then consider a feature's importance as the maximum impact across classes to create a single ranked list.

Rank Aggregation

For each partition of the cross-validation, we generate a single ranked list for each of the ML models. Once the entirety of the cross-validated training is complete, the entire set of all ranked lists across all models is aggregated into a single top-k ranked list by minimizing the distance between the set of ranked lists and the top-k list, where k is specified by the user in the configuration file. More specifically, given a set of ranked lists $\{l_1, \dots, l_m\}$, the top-k ranked list, $\hat{\theta}$, is determined as,

$$\hat{\theta} = \arg\min_{\theta \in L} \sum_{i=1}^{m} w_i \, d\,(\theta, l_i) \tag{3.14}$$

Here, L is the state space of top-k rankings, w_i is a weight associated with l_i, and $d\,(\theta, l_i)$ is the distance between a proposed top-k ranked list, θ, and l_i. The aggregation is performed using the R package RankAggreg (Pihur et al., 2009). This package uses either a genetic algorithm or cross-entropy based approach with Markov Chain Monte Carlo sampling to find the top-k features that minimize the sum of the distances between each of the input sets and the generated top-k set. The distance used is the Spearman's Correlation. Each input ranked list is weighted in the aggregation by the area under the receiver operating curve (AUC).

User Output

Meta-Signer provides a summary of the results in a portable HTML file. The file contains a description of the run and evaluation metrics for the different models in the form of boxplots. It also provides the distribution of the feature importance scores for each ML model. Lastly, it provides a list of the top-*k* taxa selected from the original taxa, the proportion of individual ranking sets that each taxon was present in the top-*k*, the rank and p-value under a PERMANOVA test, and the class in which the taxon was found to be predictive for. All images are encoded into the file, allowing the HTML file to be moved without considering the location of the images.

3.3.2 Experiments and Results

Data Used in Evaluation

We demonstrate Meta-Signer on a dataset of patients with inflammatory bowel disease (IBD) from the Prospective Registry in IBD Study at MGH (PRISM) (Franzosa et al., 2019), which enrolled patients with a diagnosis of either Crohn's disease (CD) or ulcerative colitis (UC). The dataset includes 68 samples with CD, 53 samples with UC, and 34 healthy samples.

In addition, we will use Meta-Signer to evaluate an external IBD dataset. This dataset consists of two independent cohorts from the Netherlands (Tigchelaar et al., 2015). The first cohort consists of 22 healthy subjects who participated in the general population study LifeLines-DEEP in the northern Netherlands. The second cohort consists of subjects with IBD from the Department of Gastroenterology and Hepatology, University Medical Center Groningen, Netherlands and includes 20 samples with CD and 23 samples with UC. Together, both the PRISM dataset and the external IBD dataset included 201 microbial features. Datasets were evaluated using all three classes as well as in a binary case by combining CD and

UC samples. Any taxon not found in at least 10% of samples or with less than 0.001 mean abundance was removed.

Model Training and Rank Aggregation

All ML models were trained under a scheme of 10 iterations of 10-fold cross-validation. Evaluations of AUC, MCC, precision, recall, and F1 score are shown in **Table VI**. The genetic algorithm method was used for rank aggregation to generate a candidate list with a maximum of 50 features. The ranked microbial feature section of the HTML output is shown in **Figure 17.**

TABLE VI. MEAN CROSS-VALIDATED RESULTS OVER THE PRISM DATASET USING META-SIGNER. STANDARD DEVIATION IS SHOWN IN PARENTHESES.

		RF	SVM	LogisticRegression	MLPNN
PRISM	AUC	0.91 (0.08)	0.81 (0.13)	0.82 (0.11)	0.87 (0.10)
	MCC	0.50 (0.28)	0.29 (0.32)	0.28 (0.27)	0.39 (0.30)
	Precision	0.84 (0.10)	0.76 (0.13)	0.76 (0.11)	0.80 (0.12)
	Recall	0.85 (0.07)	0.81 (0.08)	0.78 (0.07)	0.82 (0.07)
	F1	0.83 (0.09)	0.77 (0.10)	0.76 (0.08)	0.80 (0.09)
PRISM (3 Class)	AUC	0.88 (0.06)	0.67 (0.09)	0.72 (0.10)	0.74 (0.10)
	MCC	0.55 (0.17)	0.19 (0.19)	0.30 (0.19)	0.35 (0.19)
	Precision	0.72 (0.11)	0.47 (0.15)	0.56 (0.14)	0.60 (0.13)
	Recall	0.70 (0.11)	0.49 (0.11)	0.55 (0.12)	0.58 (0.12)
	F1	0.69 (0.11)	0.46 (0.12)	0.53 (0.12)	0.57 (0.12)

Evaluation of Ranked Features

We compared the features ranked using Biosigner (Rinaudo, Boudah, Junot, & Thévenot, 2016), another tool that uses an ensemble of ML approaches to identify important features, and the PERMANOVA test's p-value. For Biosigner, we varied the "pvalN" parameter using 0.05, 0.1, and 0.2 to change the levels of significance. We used the top 30 taxa from each method, except for Biosigner, which identified less than 30 taxa. We then trained machine learning models on entire PRISM dataset using the features selected from each method and evaluated predictions on the external test set for both binary classification and for three classes. Biosigner was not used for multi-class classification due to the limitations of the tool. The results for binary and three class cases are shown in **Table VII** and **Table VIII** respectively.

Aggregated Feature List

Microbe	% in top-k	Elevated class	PERMANOVA rank	Adjusted p-value
Eubacterium rectale	100.0	Control	24	7.000e-03
Dialister invisus	94.2	CD	39	2.900e-02
Bifidobacterium dentium	73.0	Control	53	6.900e-02
Roseburia hominis	77.8	Control	16	1.000e-03
Subdoligranulum unclassified	64.5	Control	11	1.000e-03
Faecalibacterium prausnitzii	76.5	UC	20	3.000e-03
Eubacterium siraeum	65.8	UC	86	2.230e-01
Coprococcus catus	85.5	Control	9	1.000e-03
Acidaminococcus unclassified	38.5	CD	44	4.200e-02
Ruminococcus torques	34.0	Control	76	1.640e-01
Prevotella copri	74.2	Control	34	2.200e-02
Bifidobacterium bifidum	57.2	UC	70	1.130e-01
Parabacteroides merdae	63.2	Control	97	3.110e-01
Lachnospiraceae bacterium 8 1 57FAA	57.5	Control	185	8.790e-01
Eubacterium eligens	36.8	UC	43	4.000e-02
Alistipes shahii	42.5	Control	2	1.000e-03
Bacteroides dorei	65.2	UC	122	5.190e-01
Collinsella aerofaciens	58.5	UC	37	2.600e-02
Dorea longicatena	63.5	Control	27	9.000e-03
Bacteroidales bacterium ph8	41.2	Control	17	1.000e-03
Eubacterium ramulus	78.0	Control	5	1.000e-03
Parabacteroides unclassified	40.5	UC	173	7.500e-01
Lachnospiraceae bacterium 2 1 58FAA	61.3	Control	40	3.000e-02
Clostridium bolteae	45.8	CD	29	1.000e-02
Bacteroides vulgatus	33.5	UC	177	8.050e-01
Ruminococcus bromii	43.2	Control	22	6.000e-03
Roseburia inulinivorans	47.2	Control	156	6.490e-01
Oscillibacter unclassified	52.5	Control	81	2.100e-01
Coprococcus comes	54.0	Control	38	2.700e-02
Ruminococcaceae bacterium D16	42.2	UC	116	5.050e-01

Figure 17. HTML output of aggregated ranked list for microbes predictive in PRISM dataset. Meta-Signer provides the use an HTML output of ranked microbial features and reports how often the feature was found in the top-*k* ranked features across cross-validated models, the enriched class, the feature rank based on a PERMANOVA analysis, and the PERMANOVA adjusted p-value.

TABLE VII. MEAN CROSS-VALIDATED RESULTS OF EXTERNAL DATASET USING TOP 30 RANKED FEATURES TO TRAIN ON PRISM DATASET USING BINARY CLASSIFICATION.

		Meta-Signer	Permanova	BioSigner (0.05)	BioSigner (0.10)	BioSigner (0.20)
	AUC	**0.83**	0.80	0.66	0.65	0.72
	MCC	**0.41**	0.35	0.07	0.22	0.20
RF	Precision	**0.75**	0.71	0.59	0.69	0.64
	Recall	**0.75**	0.72	0.65	0.69	0.66
	F1	**0.73**	0.71	0.58	0.63	0.65
	AUC	0.62	**0.69**	0.64	0.57	0.60
	MCC	0.08	**0.25**	-0.08	-0.09	0.09
SVM	Precision	0.59	**0.79**	0.50	0.43	0.61
	Recall	0.63	**0.69**	0.62	0.65	0.66
	F1	**0.60**	0.59	0.53	0.52	0.57
	AUC	**0.69**	**0.69**	0.58	0.53	0.62
Logistic	MCC	0.20	**0.30**	0.09	0.04	0.02
Regression	Precision	0.64	**0.69**	0.61	0.58	0.56
	Recall	0.65	**0.71**	0.66	0.65	0.60
	F1	**0.64**	0.63	0.57	0.56	0.57
	AUC	0.67	**0.71**	0.53	0.62	0.69
	MCC	0.15	**0.18**	0.00	0.02	0.08
MLPNN	Precision	0.62	**0.65**	0.55	0.56	0.59
	Recall	0.65	**0.68**	0.65	0.60	0.63
	F1	**0.63**	**0.63**	0.54	0.57	0.60

TABLE VIII. MEAN CROSS-VALIDATED RESULTS OF EXTERNAL DATASET USING TOP 30 RANKED FEATURES TO TRAIN ON PRISM DATASET USING THREE CLASSES FOR CLASSIFICATION.

		Meta-Signer	Permanova
RF	AUC	**0.73**	0.71
	MCC	**0.19**	0.17
	Precision	0.36	**0.47**
	Recall	**0.45**	**0.45**
	F1	0.36	0.43
SVM	AUC	**0.75**	0.68
	MCC	**0.36**	0.12
	Precision	**0.60**	0.41
	Recall	**0.57**	0.42
	F1	**0.57**	0.40
Logistic Regression	AUC	**0.73**	0.66
	MCC	**0.26**	0.19
	Precision	**0.52**	0.47
	Recall	**0.51**	0.46
	F1	**0.50**	0.46
MLPNN	AUC	**0.71**	0.62
	MCC	**0.19**	0.17
	Precision	0.47	**0.48**
	Recall	**0.46**	0.45
	F1	**0.46**	0.44

3.3.3 Conclusion

We developed Meta-Signer as a user-friendly tool to identify a robust set of highly informative microbial taxa. Meta-Signer uses an ensemble of ML approaches to construct a single, robust ranked list of microbial features that are predictive of human disease status, which in turn will empower down-stream hypotheses of disease related microbiome studies. Meta-Signer is publicly available and can be downloaded from https://github.com/YDaiLab/Meta-Signer.

3.4 Boosting Host Phenotype Prediction Through Conditional Generative Adversarial Modeling

Even with advances in ML approaches for predicting host phenotype in microbiome studies, one persistent challenge is the relatively small size of microbiome datasets. It is often the case that datasets have a far greater number of features than the number of samples, which can quickly lead to the overfitting of ML models. One direct way to address this limitation is to augment datasets with realistic synthetic data to increase the total sample size. However, statistical modelling of the underlying distribution of microbiome data has been a long-standing challenge due to the sparsity and over-dispersion found in microbiome data. There have been many approaches proposed over the past decade, however there is still no consensus as to which models and underlying assumptions are best suited for handling the complexity of the data (Kurilshikov, Wijmenga, Fu, & Zhernakova, 2017; L. Xu, Paterson, Turpin, & Xu, 2015). In this section, we will present a framework using conditional generative adversarial networks (CGAN) to non-parametrically model microbiome data in order to generate realistic synthetic data. We further show that augmenting a real microbiome dataset with these synthetic samples can boost the performance of downstream tasks, such as host phenotype prediction.

3.4.1 Framework

In order to generate synthetic microbial community structures, we utilize a CGAN architecture. A CGAN is composed of two competing networks: a generator and a discriminator. The task of the generator is to learn to generate synthetic data representative of real data while the discriminator tries to determine if a given sample is synthetic or real. The generator is trained to maximize the probability of the discriminator in misclassifying samples. At the same time, the discriminator is trained to minimize this probability.

The generator, G, of the CGAN model requires two sets of inputs: a set of priors and the conditional side information. Our framework uses priors from the uniform distribution $\sim U\ (-1, 1)$. Both inputs are fed through multiple fully connected hidden layers and finally to an output layer. Batch normalization is performed at each layer. The leaky ReLU activation function with $\alpha = 0.1$ is performed after each batch normalization.

$$Leaky\ ReLU(x) = \begin{cases} x & x > 0 \\ \alpha x & x \leq 0 \end{cases} \tag{3.15}$$

The output of the generator represents a vector of microbial abundance features.

The discriminator, D, takes a sample of microbial abundance features as an input in addition to the side information. The inputs are passed through multiple fully connected layers. Batch normalization is performed at each layer. The leaky ReLU activation function with $\alpha = 0.1$ is performed after each batch normalization. The discriminator has an output of a single node using the sigmoid activation function. The sigmoid function is used so that the output is a value ranging from 0 and 1. The output of the discriminator represents the prediction of the probability that the given sample of data is real.

Both generator and discriminator networks are trained in an iterative fashion such that in each epoch, the discriminator is first trained on the generated and real samples and the network weights are updated. After the discriminator has been updated, the generator is updated. The cost functions for the discriminator and generator are shown below.

$$C_D = \frac{1}{n}\sum_{i}^{n} -\log[D(\boldsymbol{x}_i, s_i)] - \log\left[1 - D(G(\boldsymbol{z}_i, s_i), s_i)\right] \tag{3.16}$$

$$C_G = \frac{1}{n}\sum_{i}^{n} \log[D(G(\boldsymbol{z}_i, s_i), s_i)] \tag{3.17}$$

Here n represents the number of real samples, $\boldsymbol{z}_i$ represents a vector of priors for the generator, $\boldsymbol{x}_i$ is the relative abundance vector of a real microbial community sample, and s_i is the side information that the networks are conditioned on. $D(\boldsymbol{x}_i, s_i)$ is the discriminator's prediction if $\boldsymbol{x}_i$ is real given the side information s_i. $G(\boldsymbol{z}_i, s_i)$ is the generator's prediction of a synthetic sample given the prior noise $\boldsymbol{z}_i$ and side information s_i. A figure showing the architecture of our CGAN is shown in **Figure 18**.

During training, models were saved every 500 iterations. Additionally, the Principal Coordinate Analysis (PCoA) of the training set, generated set, and the combination of the two sets was visualized and stored. The Bray-Curtis dissimilarity measure (Bray & Curtis, 1957) was used in calculating the distance matrix for PCoA. The Bray-Curtis dissimilarity quantifies the microbial compositional dissimilarity between two different samples. Given two microbial samples, x_a and x_b, the Bray-Curtis dissimilarity between the two samples is calculated as

$$BC(\boldsymbol{x}_a, \boldsymbol{x}_b) = 1 - \frac{2C_{ab}}{S_a + S_b} \tag{3.18}$$

Here C_{ab} is the sum of the lesser values for the abundances of each species found in both $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$. S_a and S_b are the total number of species counted in $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$ respectively. Visual analysis of the PCoA plots and the overlap of the original and generated data was used to select the best model.

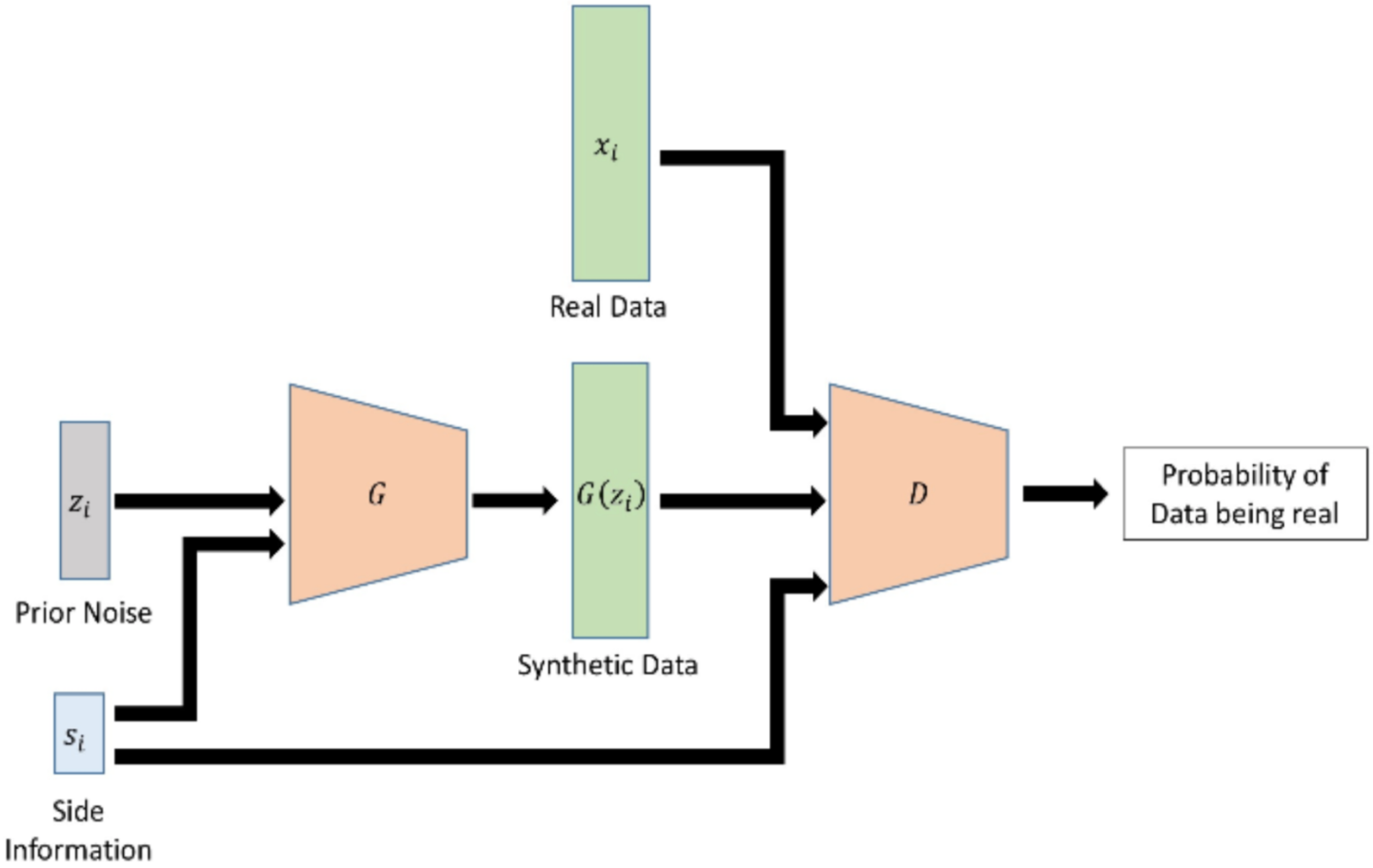


Figure 18. CGAN architecture. A set of prior noise and side information corresponding to sample are used to generate a synthetic sample. The discriminator then uses the side information to predict if a given sample is real or synthetic.

3.4.2 Experiments and Results

Data Used in Evaluation

We evaluate our CGAN model using the data reported from two different cohorts of patients with inflammatory bowel disease (IBD). The Prospective Registry in IBD Study at Massachusetts General Hospital (PRISM) enrolled patients with a diagnosis of IBD based on endoscopic, radiographic, and histological evidence of either Crohn's Disease or Ulcerative Colitis. The second dataset is used specifically for external validation and consists of two independent cohorts from the Netherlands (Tigchelaar et al., 2015). The first consists of 22 healthy subjects who participated in the general population study LifeLines-DEEP in the northern Netherlands. The second cohort consists of subjects with IBD from the Department of Gastroenterology and Hepatology, University Medical Center Groningen, Netherlands. This will be used as the validation dataset. Processing of the stool samples collected for both datasets is described in the original study (Franzosa et al., 2019). Briefly, metagenomic data generation and processing were performed at the Broad Institute in Cambridge, MA. Quality control for raw sequence reads was performed and reads were taxonomically profiled to the species level using MetaPhlAn2 (Segata et al., 2012). The relative abundance values are publicly available and were obtained from the original study. Microbial relative abundance features present in less than 20% of samples or with a mean abundance less than 0.1% across all samples of both the PRISM and Validation sets were removed from the analysis, resulting in a total of 93 microbial features in the PRISM and Validation datasets.

Model Training

We use a vector of priors of size 8 for the input $\boldsymbol{z}_i$ and a vector of size 2 representing the one-hot encoded value of the disease state (IBD or healthy) as the input s_i and concatenate the two inputs together. The concatenated input is then passed through two fully connected layers of size 128. The output layer of

the generator is a vector of size 93 representing the microbial features. The softmax activation function in used in order to reconstruct the relative abundance of the microbial community.

The discriminator network takes a vector of size 93 representing microbial relative abundance features as an input in addition to vector of size 2 representing the one-hot encoded disease state for that sample. The two inputs are concatenated and fed through two fully connected layers of size 128. The output of the discriminator is a single node with a sigmoid activation to shrink the prediction value to be between 0 and 1.

Models were trained using 10-fold cross-validation. In each partition, 90% of the PRISM dataset was used to train the CGAN model. CGAN models were trained for 30,000 iterations in which 32 random samples were selected at each iteration as real samples. A synthetic sample was generated for each of the 32 real samples using the sample's respective disease state as the side information. The 32 real and 32 synthetic samples were then fed to the discriminator for training and the discriminator was updated based on **Equation 3.16**. After updating the discriminator, the discriminator is again used to predict the synthetic samples and the generator is updated based on **Equation 3.17**. Both networks were trained using the ADAM optimizer (D. Kingma & Ba, 2014) with a learning rate of 5×10^{-5}. Models were selected based on visual inspection of PCOA overlap of real and synthetic data using the same set of class labels. An example showing the PCOA of a selected model from the cross-validated training is shown in **Figure 19**. For the implementation and training of our CGAN models we used the TensorFlow (Abadi et al., 2016) package in Python.

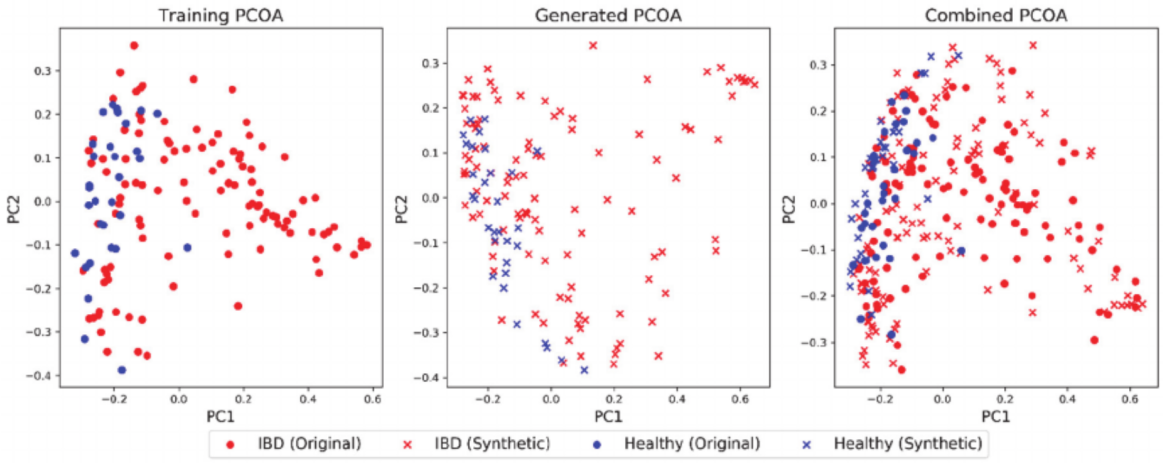


Figure 19. Principal Coordinate Analysis (PCoA) of the training (left), generated (middles), and combined (right) datasets using the Bray-Curtis dissimilarity. Red points represent patients with IBD and blue points represent healthy subjects.

Trueness of Synthetic Data

In order to check how well the generated samples represent the real samples, we compare the distributions of the alpha and beta diversities for IBD and healthy samples. Alpha diversity is a local measure of species diversity within a sample. It characterizes the microbial richness of a community. For our analysis, we use the Shannon Entropy metric to quantify the alpha diversity of samples. Given a sample $\boldsymbol{x}$ with m relative abundance values, the Shannon Entropy is calculated as

$$H(\boldsymbol{x}) = -\sum_{j=1}^{m} x_j \log(x_j) \tag{3.19}$$

Beta diversity, on the other hand, allows us to quantify how similar samples are to each other. In our study, we use the Bray-Curtis dissimilarity as a distance measure of beta diversity, calculated as described in **Equation 3.18**. To demonstrate the behavior of the CGAN model, we visualize the diversity metrics for the training set and for 10,000 generated samples using the selected best model. In addition, we calculate the diversity metrics of a set of 10,000 generated samples using the random initialization of the CGAN before any training to show the initial random distribution. Before calculating the diversity metrics, we clipped the generated samples in order to introduce zero values. The softmax function used to generate samples provides a vector entirely of positive values. However, in reality, microbiome data very sparse. Therefore, to induce this sparsity into the generated samples, we calculated the minimum value across all species found in the training set. We used this value as a threshold and set any generated value less than the observed minimum to zero. After clipping the generated sets, we calculated the diversity metrics. When considering beta diversity, we only considered the Bray-Curtis dissimilarity from the training set to itself, the training set to the best generated samples, and the training set to the randomly generated samples.

The distributions of alpha and beta diversity for one of the cross-validated partitions are shown in **Figure 20**. We observed that the data generated from the selected best model followed very similar distributions of the alpha and beta diversities of the data used to train the CGAN. We did notice that the beta diversity within the training set had a spike near one, however upon post-analysis we discovered that it was caused by samples with only a few numbers of microbial species present.

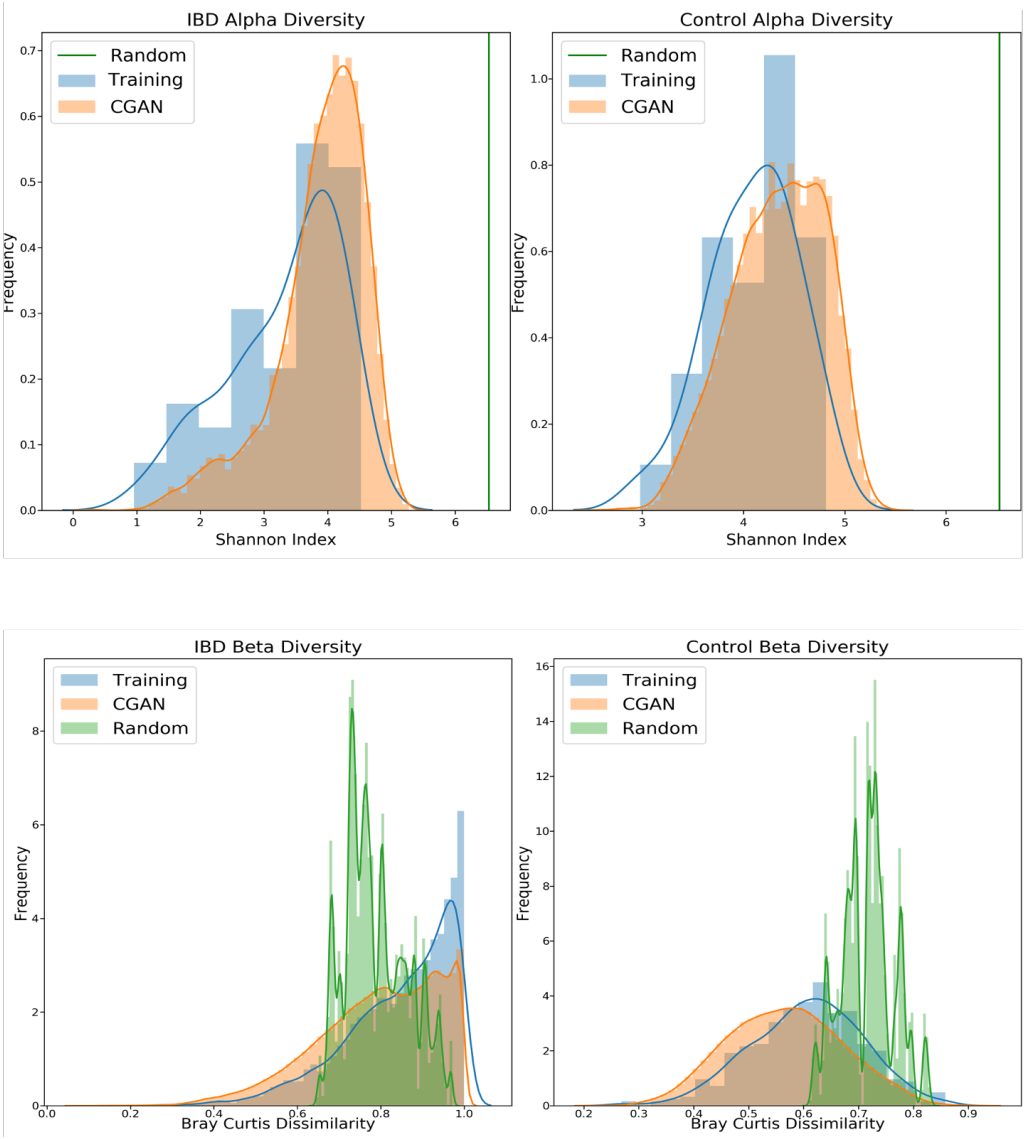


Figure 20. Distributions of alpha and beta diversities of real and synthetic microbiome data. Histograms show alpha (top) and beta (bottom) diversity distributions for original (blue) and synthetic (orange) samples. Diversity metrics generated from generated from samples before any training are shown in green.

Generated Data Improves Prediction Performance

For each of the partitions in the 10-fold cross-validation, we simulated 10,000 samples for both IBD and healthy groups using the selected best model. Relative abundance values were then log-transformed and normalized to zero mean and unit variance. Next, we trained logistic regression and multilayer perceptron neural network (MLPNN) models to predict disease status using microbial features. For each partition of the cross-validation training, two sets of MLPNN and logistic regression models were trained. One set of models was trained using the original samples in the partition of the training set. The second set of models was trained using the 10,000 simulated samples generated by the CGAN trained on the training set.

To train a logistic regression model on each 90% used as training set, we performed internal 5-fold cross-validation grid search over L_1, L_2, and Elastic Net regularizations considering 10 penalty strengths spaced evenly on a log scale ranging from 1 to 10,000. Logistic regression models were trained using the Python scikit-learn package (Pedregosa et al., 2011).

MLPNN models were trained using two fully connected hidden layers with 256 nodes each and dropout with a rate of 0.5 after each layer. Leaky ReLU with an alpha of 0.1 was used as the activation function. The output layer contained two nodes using the softmax activation to predict the disease state. Networks were trained using the ADAM optimizer with a learning rate of $1x10^{-4}$. We set aside 20% of the training set as a validation set, and networks were trained until the loss of the validation set had not decreased for 100 epochs. The implementation and training of the MLPNN models were again done using the TensorFlow (Abadi et al., 2016) package in Python. Using the trained logistic regression and MLPNN models generated from a fold's training set as well as the generated dataset, we calculated the area under the receiver operating characteristic curve (AUC-ROC) using the fold's 10% held out data of true observed values. We observed that for logistic regression, the models trained using the generated sets had an average AUC-ROC of 0.849, while the models trained on the original data had an average AUC-ROC of 0.778

across the 10 folds. Similarly, for MLPNN models, the AUC-ROC had a value of 0.889 when training on the generated data and 0.847 when training on the original data. Using a Wilcoxon Signed-Rank test, the AUC-ROC when using the generated samples was significantly larger than that of when using the original data with a p-value of 0.0249 for logistic regression models and a p-value of 0.0464 for MLPNN models. Boxplots of the AUC-ROC values when using original and generated datasets are shown in **Figure 21**. These results demonstrated that the CGAN augmented datasets can boost the predictive power of the ML models.

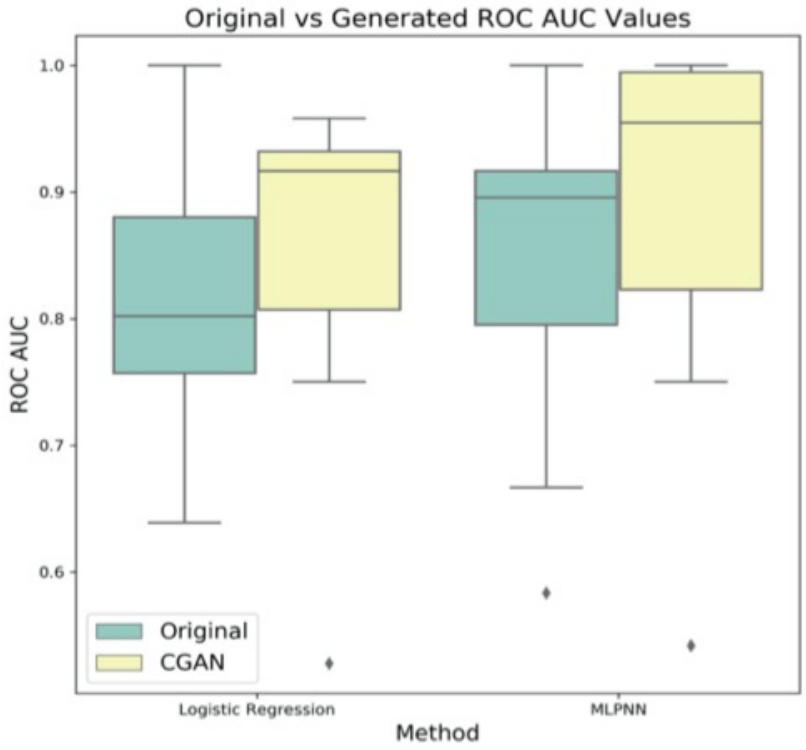


Figure 21. Boxplots for the AUC-ROC values across 10-fold cross-validation for logistic regression and MLPNN models trained on original and synthetic data. Cross validated evaluations of logistic regression and MLPNN models trained on original (green) and data augmented with synthetic samples (yellow).

Synthetic Data is Predictive of External Dataset

To evaluate if the synthetic samples generated from the CGAN model were generalizable to a dataset of a similar study, we trained a CGAN model using the entire PRISM dataset. The CGAN is trained for 30,000 iterations and models as well as PCOA visualization of the real and synthetic samples are saved every 500 iterations. The best model is selected based on the PCOA comparison between the training and generated sets. A PCOA visualization of the PRISM dataset combined with the synthetic data generated from the best model and the external validation set is shown in **Figure 22.**

Using the best model, we evaluate if the generated samples can improve the task of predicting IBD status. Logistic regression and MLPNN models are trained in a similar fashion as outlined in the previous section. The model was trained using 10,000 generated samples from a CGAN model that was trained on the entire PRISM dataset. We then evaluate the model performance on the true observations of the external validation IBD dataset. We observed an improvement in AUC-ROC from 0.734 to 0.832 in logistic regression models and from 0.794 to 0.849 in MLPNN models. This demonstrates that the synthetic samples generated using one cohort can augment the analysis of a different cohort.

Lastly, we analyzed the distribution of alpha and beta diversities of the original PRISM dataset, the samples generated after training a CGAN on the whole PRISM dataset, and the external validation dataset. The alpha diversity is calculated for each dataset using the Shannon Entropy metric. The beta diversity within the PRISM dataset, from the PRISM dataset to the generated samples, and from the external validation dataset to the generated samples was calculated. In addition, we compared the random diversities from the randomly initialized CGAN before training. The alpha and beta diversities are shown in **Figure 22**.

We observed that the beta diversity between the PRISM dataset and the synthetic samples generated from it displays similar distributions. Additionally, the distribution of the beta diversity values between the external validation set and the synthetic samples follow a similar pattern, suggesting that the CGAN model

did not overfit the PRISM dataset and is robust in generating synthetic samples. We also observed that the alpha diversities within the PRISM, synthetic, and external validation datasets showed similar distributions. In particular, the alpha diversity within the samples of IBD patients was very similar. The distributions in the healthy samples were slightly different in each of the datasets, however we suspect this may be due to the fact that there were far fewer cases of healthy samples in the original PRISM dataset.

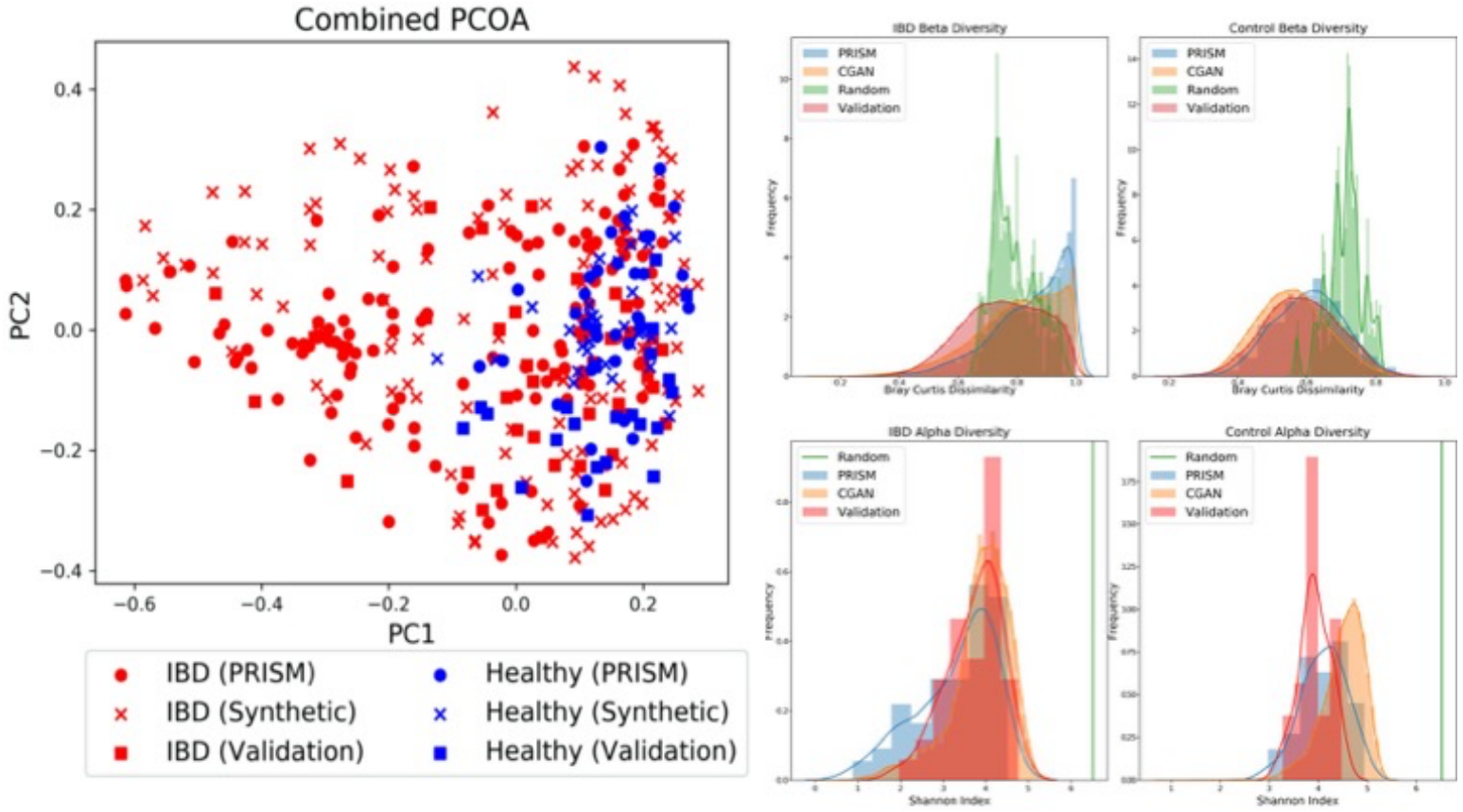


Figure 22. CGAN analysis on External dataset. (left) PCOA visualization of the combination of the PRISM dataset, synthetic data generated by the best CGAN model, and the external validation set. Red points represent patients with IBD and blue points represent healthy patients. (right) Distributions of beta diversity based on the Bray Curtis dissimilarity and Shannon alpha diversity between the training set and itself, the validation, the generated (CGAN), and random datasets for IBD and healthy samples.

3.4.3 Conclusion

Using two different cohorts of subjects with IBD, we have demonstrated that the synthetic samples generated from a CGAN framework are similar to the original data in both alpha and beta diversity metrics. In addition, we have shown that augmenting the training set by using a large number of synthetic samples can improve the performance of logistic regression and MLPNN in predicting host phenotype. By generating a large number of synthetic microbiome samples that resemble the original data, we show that it is possible to improve the performance of ML models trained on the generated synthetic samples. Not only can this provide better predictive models of patient disease status, but the improved predictive performance can also lead to a more robust set of microbial features extracted from these ML models that have been augmented with synthetic data.

Chapter 4

Identify Underlying Microbe-Metabolite Interactions by Integrating Microbiome and Metabolomic data

4.1 Introduction

The advance in microbiome and metabolome studies has generated rich omics data revealing the involvement of the microbial community in the development of host disease. This involvement is believed to happen through interactions with their host at a metabolic level. While previous studies have uncovered various microbe-disease associations, more recent work has further revealed the central role of bacterial metabolites in host health (Feng et al., 2016; McHardy et al., 2013; Parker et al., 2018). Studies have also shown that the abundance of metabolic pathways is relatively consistent despite considerable variability in taxonomic composition among individuals, suggesting that the metabolic impact of the microbiome is an emergent property of the microbial community as a whole (Benson, 2016; Lee-Sarwar, Lasky-Su, Kelly, Litonjua, & Weiss, 2020). Thus, the identification of microbiome-metabolome interactions contributing to the overall community metabolic activity is essential not only for understanding microbiome's effect on the host's health, but also for the development of therapeutic interventions for the prevention or management of chronic metabolic disease (Cani & Delzenne, 2011; Helmink et al., 2019; Skelly et al., 2019).

The majority of previous studies integrating microbiome and metabolomic data have used *a priori* annotations of microbial enzymes and metabolic pathways to model the metabolic change driven by the microbial community. One prominent method, Predicted Relative Metabolic Turnover (PRMT) (Larsen et al., 2011), uses microbial genome annotations to first predict the abundance of microbial enzymes from the microbial community. Then it uses annotated metabolic pathways to predict the change in metabolites based on the abundance of microbial enzymes. The other commonly used method is constraint-based stoichiometric modeling using flux balance analysis (FBA) to learn the flux rate of metabolites in the community (Biggs et al., 2015; Edwards et al., 2002; Gottstein et al., 2016). Both PRMT and FBA rely on *a priori* information through either microbial or metabolic annotations, limiting them from identifying novel metabolic findings.

More recently, data-driven methods have emerged using paired microboime-metabolomc data to identify microbe-metabolite interactions. Since these methods no longer require *a priori* annotated knowledge, they do not suffer from the same limitation of not being able to discover novel interactions. One method, MelonnPan, uses linear Elastic Net regression to predict single metabolite abundances from microbial relative abundance data (Mallick et al., 2019). Although MelonnPan showed promising results, the method itself is limited to only capturing linear combinations of microbial features and may not be well suited for the complex nature of microbe-metabolite interactions. More recently, a group published a neural encoder-decoder (NED) for predicting the entire metabolic community from microbial features (Le et al., 2019). They enforce sparsity in the network weights and only allow for positive weights. The harsh constraints remove the ability for the model to capture negative interactions of metabolic degradation by microbes, greatly reducing the learning capacity and interpretability of the model.

4.1.1 Problem Definition

A DNN model can better model the complex nature of microbe-metabolite interactions through the task of predicting the metabolic community given microbial feature and facilitate the identification of microbial and metabolic modules in the dysregulation of disease.

4.1.2 Significance

MiMeNet is one of the first data-driven approaches to integrate paired microbiome-metabolome data. By modeling the entire metabolome at once, MiMeNet uses multivariate learning to use shared information across metabolites. Additionally, by modeling of the entire metabolome at once, MiMeNet is more scalable than the current univariate approaches. In addition to metabolome prediction, MiMeNet clusters microbes and metabolites into meaningful functional modules, empowering the identification of novel microbe-metabolite interactions underlying the metabolic dysregulation of disease.

4.2 MiMeNet: Exploring Microbiome-Metabolome Relationships Using Neural Networks

We present MiMeNet, a framework utilizing DNN models to predict the metabolite abundance using microbial abundance features. In addition, the trained DNN models are easily interpretable and facilitate the grouping of microbes and metabolites into functional modules for a module-based interaction network.

4.2.1 MiMeNet Framework

MiMeNet can be split into four main steps. In the first step, paired microbiome and metabolomic data are used to train an ensemble of DNN models. Each model is trained to predict the metabolomic features using the microbial abundance features. In the second step, metabolites that can be well-predicted from the microbial features are identified. In the third step, the set of trained models are used to construct an interaction matrix between the microbial features and the well-predicted metabolites. The interaction matrix is further filtered by removing microbes with no significant interaction scores. Lastly, in the fourth step, the interaction matrix is biclustered to produce functional microbial and metabolic modules. These modules can then be used to construct a bipartite interaction network. An overview of the framework is shown in **Figure 23**.

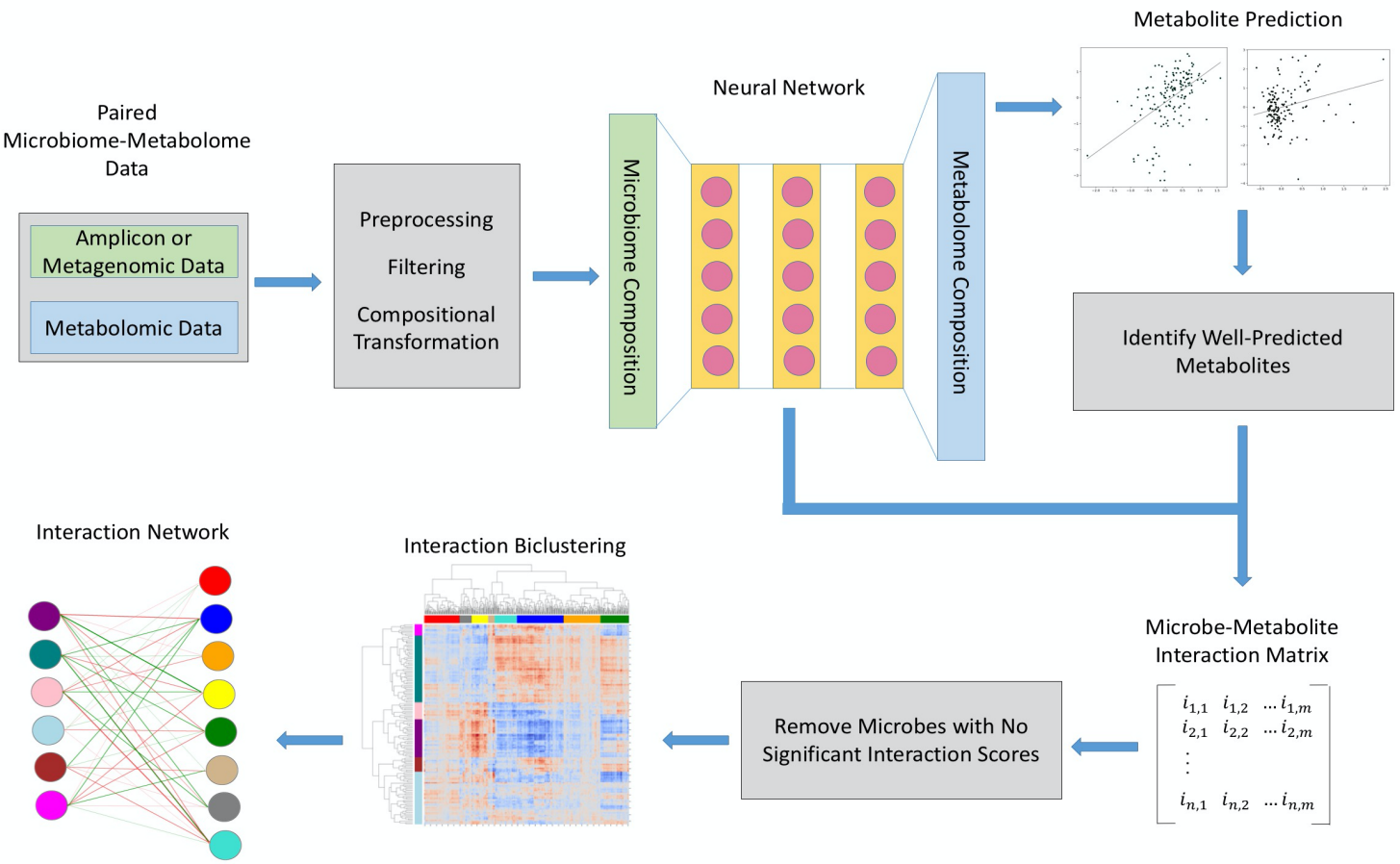


Figure 23. Framework of MiMeNet learning model. MiMeNet uses paired microbiome and metabolome data as input. Microbiome abundance features (green) are used to train a neural network to predict metabolite abundance features (blue). Well-predicted metabolites are identified and the trained models are used to learn a microbe-metabolite interaction matrix. The interaction matrix is biclustered into microbial and metabolic modules which are then used to construct a module-based interaction network.

Neural Network Model

An MLPNN model is composed of multiple fully connected hidden layers composed of perceptrons. The values $\boldsymbol{h}_l$ of the hidden layer l are calculated as:

$$\boldsymbol{h}_l = \varphi\left(\boldsymbol{h}_{l-1}\boldsymbol{W}_{l-1} + \boldsymbol{b}_{l-1}\right) \tag{4.1}$$

Here $\boldsymbol{W}_{l-1}$ are the weights connecting the perceptrons of the l^{th} layer with the previous layer with values and $\boldsymbol{b}_{l-1}$ are the bias values between the l^{th} layer and the previous layer, and φ is a non-linear activation function. MiMeNet uses the rectified linear unit (ReLU) as its activation function.

$$ReLU(x) = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases} \tag{4.2}$$

Previous studies have shown that the ReLU activation helps avoid the problems exploding and vanishing gradient in DNN training (Hara, Saito, & Shouno, 2015). We regularized MiMeNet using L_2 regularization. Additional regularization was applied through dropout at each hidden layer, where a portion of the nodes and their weights are masked for a given epoch. MiMeNet was trained using the ADAM optimizer (D. Kingma & Ba, 2014) and the mean squared error (MSE) loss function, giving the loss function shown in **Equation 4.3.**

$$Loss = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 + \lambda_1 \sum_{l \in L} \|\boldsymbol{W}_l\|_2 \tag{4.3}$$

Here, N is the number of training samples and L is the total number of hidden layers in the DNN model. The first term is the MSE of the observed metabolites $\boldsymbol{y}$ and the predicted metabolites $\hat{\boldsymbol{y}}$, and the second term is L_2 regularization with a penalty controlled with the coefficients λ_1.

Identifying Well-Predicted Metabolites

In order to identify which metabolites are well-predicted by MiMeNet, we construct a background distribution of SCC values by training 100 iterations of 10-fold cross-validation where the samples in both the microbiome and metabolomic data were each randomly shuffled. From each trained model, we calculated the SCC for between the observed and predicted metabolites and used the entire set of SCC values as the background distribution. We then defined a metabolite to be well-predicted if its SCC is above the 95th percentile of the background distribution of SCC values.

Constructing an Interaction Score Matrix

Microbe-metabolite interaction scores are calculated using Olden's method for understanding variable contributions in neural network models (Olden et al., 2004). Olden's method works by multiplying the weights of each hidden layer together, as shown in **Equation 4.4.** This results in a single matrix where each row represents an input feature and each column represents an output feature.

$$\boldsymbol{S} = \prod_{l \in L} \boldsymbol{W}_l \tag{4.4}$$

Here l is the current layer in the set of L layers, and $\boldsymbol{W}_l$ is the weight matrix connecting layer $l-1$ and layer l. Each element in $\boldsymbol{S}$ represents a microbe-metabolite feature attribution score. A positive value indicates that an increase of the microbe will lead to an increase of the metabolite and a negative value indicates that an increase of the microbe will lead to a decrease of the metabolite. For the subsequent procedure, we only retained the columns of the feature attribution matrices representing the well-predicted metabolites.

Identifying Significant Microbes

Denoting $\boldsymbol{S}_i$ as the feature attribution score matrix for the i^{th} trained model (n = 100 models resulted from the 10 iterations of 10-fold cross-validation), we calculated the mean feature attribution matrix as

$$\overline{\boldsymbol{S}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{S}_i \tag{4.5}$$

To identify microbes with significant associations, we further calculated feature attribution score matrices from the network models used to generate the background correlation distributions and calculated the mean feature attribution score matrix, which was then flattened into a vector and a threshold was set at the 97.5 percentile. Any feature attribution score in the observed dataset with an absolute value above the threshold was considered significant. Finally, any microbe with at least one significant feature attribution score with any metabolite was considered to be significant and the rows representing non-significant microbes were filtered out from $\bar{\bar{S}}$ as well as from all feature attribution score matrices Si used in subsequent analyses.

Construction of Microbial and Metabolic Modules

We normalized the values in each feature attribution score matrix S_i by dividing the significant threshold score identified from the background and clipped values to be between -1 and 1. In doing so, every significant attribution score was treated with equal magnitude. We recalculated $\bar{S}$ using the normalized S_i so that each element in $\bar{S}$ is also between -1 and 1. The normalized matrix S_i was then used to cluster microbes (rows) and metabolites (columns) separately based on the Euclidean distance and complete linkage using Seaborn's *clustermap* function in Python. Modules were constructed by cutting each dendrogram at a given height. To determine the number of clusters for microbes, for each fixed k,

ranged from 2 to 20, a *k*-clustering of the rows using each normalized S_i was generated. Then a consensus matrix $M^{(k)}$ was calculated as the mean connectivity matrix across all *k*-clustering results ($n = 100$),

$$\boldsymbol{M}^{(k)} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{C}_i^{(k)} \tag{4.6}$$

$$C_i^{(k)}(a, b) = 1 \quad if\ features\ a\ and\ b\ are\ in\ the\ same\ cluster \tag{4.7}$$

$$C_i^{(k)}(a, b) = 0 \quad if\ features\ a\ and\ b\ are\ not\ in\ the\ same\ cluster \tag{4.8}$$

where $C_i^{(k)}$ is the connectivity matrix of the clustering using *k* clusters on S_i. We further calculated the area under the cumulative distribution function (CDF) for the consensus matrix of each clustering,

$$A^{(k)} = \sum_{j=1}^{m} (x_j - x_{j-1})\psi\,(x_j) \tag{4.9}$$

where x_j is the j^{th} value from the set {0.01, 0.02, 0.03, …, 0.99, 1.0} and $\psi\,(x_j)$ is the proportion of entries in consensus matrix $\boldsymbol{M}^{(k)}$ that are less than x_j. Lastly, we calculated the proportional change in area as the number of clusters changed,

$$\Delta k = \frac{A^{(k)} - A^{(k-1)}}{A^{(k-1)}} \tag{4.10}$$

This value represents how much cleaner the consensus matrix gets if we increase the number of clusters by 1. We set a threshold of $\Delta k = 0.025$, indicating that increasing the cluster by 1 more would give less than a 2.5% increase in the area under the CDF. The best cluster number k^* was selected as the largest value of *k* that resulted in Δk larger than the threshold. Further details of this analysis can be found in *Monte et al*. (Monti, Tamayo, Mesirov, & Golub, 2003). The number of metabolite clusters was determined using the same procedure. The best cluster numbers for microbes and metabolites are then denoted as k_1^* and k_2^*,

respectively. The final set of microbial and metabolite modules are then determined by biclustering $\bar{S}$ using k_1^* and k_2^* to cluster the rows and columns respectively.

For visualization of the microbe-metabolite interaction network, the score between a pair of microbe and metabolite modules was calculated as the average normalized feature attribution score in $\bar{S}$ between each microbe and metabolite within the two modules. For visualization purposes only we removed any score whose absolute value was less than 0.25. Networks showing microbe and metabolite modules and the interactions between them were constructed using Cytoscape (Shannon et al., 2003).

4.2.2 Experiments and Results

Data Used in Evaluation

We evaluated MiMeNet using three different datasets. The first dataset was taken from a published study of patients with inflammatory bowel disease (IBD) (Franzosa et al., 2019). It includes one cohort from the Prospective Registry in IBD Study at MGH (PRISM), which enrolled patients with a diagnosis of IBD based on endoscopic, radiographic, and histological evidence of either Crohn's Disease (CD) or Ulcerative Colitis (UC). This dataset has 121 IBD patients and 34 controls and is named as IBD (PRISM). Additionally, it includes an external validation dataset using two other cohorts. One consists of 20 healthy subjects who participated in LifeLines-DEEP, a general population-based study in the northern Netherlands (NLIBD) (Tigchelaar et al., 2015). The second cohort consists of 43 subjects with IBD taken from the Department of Gastroenterology and Hepatology at the University Medical Center in Groningen, Netherlands. This dataset is named as IBD (External). The processing of the stool samples collected is described in the original study (Love et al., 2014). A total of 201 microbial species and 8848 metabolites were identified for the IBD (PRISM) and IBD (External) datasets.

The second dataset was taken from a study that collected 172 lung sputum samples from patients with cystic fibrosis (Quinn et al., 2018). Microbial features were generated using 16S rRNA gene sequencing and abundance was collected at the genus level, resulting in 657 unique microbial features. Metabolomic data were generated using LC-MS/MS technology, resulting in 168 unique metabolites.

The third dataset represents microbial and metabolic activity caused by soil wetting at five-time points across four biocrust successional stages (Baran et al., 2015). Biocrust soil water for each sample was analyzed by LC/MS for metabolite detection. Metagenomic shotgun sequencing was used to profile the microbial community and the authors used the 50S ribosomal protein L15 to map microbial taxa. A total of 466 microbes and 85 metabolites were detected. A summary of the datasets is shown in **Table 9**.

TABLE IX. SUMMARY OF DATASETS USED IN MIMENET EVALUATION.

	Case	**Control**	**# Microbes**	**# Metabolites**
IBD (PRISM)	68 (CD), 53 (UC)	34	201	8848
IBD (Validation)	20 (CD), 23 (UC)	33	201	8848
Cystic Fibrosis	172	-	657	168
Soil	-	19	466	85

Any input or output feature that is present in less than 10% of samples was removed. Microbiome and metabolomic data were then transformed using the centered log-ratio (CLR) transformation:

$$CLR(\boldsymbol{x}) = \left[\log\frac{x_1}{g(\boldsymbol{x})}, \ldots, \log\left(\frac{x_m}{g\,(\boldsymbol{x})}\right)\right] \tag{4.11}$$

where $\boldsymbol{x}$ is the abundance vector of a sample, $g(\boldsymbol{x})$ is the geometric mean of x, and m is the number of features. A pseudocount of 1 was added to each entry of x before the CLR transformation to prevent taking the log of 0 values. The only exception was for the IBD (PRISM) and IBD (External) microbe values, which were obtained in relative abundance (RA).

Model Training

The overall evaluation of MiMeNet prediction was conducted using 10 iterations of the 10-fold cross-validation, and the average of the correlations between the predicted and observed values for metabolites was reported. More explicitly, during the 10-fold cross-validation, each dataset was partitioned into two subsets: 90% for training and 10% for testing. For each training partition, the 90% of the data was further split into 80% for model training and 20% for validation. After finishing one iteration of 10-fold cross-validation, the SCC between the predicted and the observed was calculated for each metabolite. To prevent overfitting, MiMeNet models were trained using early stopping. After each iteration of updating network weights using the 80% of the training set, the loss of the validation set was calculated. The training process was terminated when the loss of the validation set has not improved within 40 iterations, and the network weight parameters were set to the values of the best performing model on the validation set. Finally, the average of the SCC values was calculated after repeating the 10-fold cross-validation procedure for 10 times. For the IBD datasets, a final model trained on the full IBD (PRISM) dataset was then evaluated on the IBD (External) data set.

Hyper-parameter Tuning

Hyper-parameter tuning was performed on the first training partition during cross-validation. To determine the optimal set of hyper-parameters (number of layers, layer size, λ, and dropout rate), we performed a cross-validated random search using a nested 5-fold cross-validation. We allowed for 1, 2, and 3 hidden layers of sizes 32, 128, and 512. The L_2 regularization parameter (λ) was selected from 10 different values between 0.0001 and 0.1, evenly spaced on a log scale. Dropout was selected from 0.1, 0.3, and 0.5. The average SCC was calculated after a model was trained. We evaluated 20 sets of hyper-parameters and selected the best performing set for the rest of the 10-fold cross-validation. The optimal hyper-parameters used for subsequent evaluations are shown in **Table X**.

TABLE X. SUMMARY OF OPTIMAL HYPER-PARAMETERS IN MIMENET MODELS.

	Layer Size	**Number of Layers**	**L_2 Penalty**	**Dropout**
IBD (PRISM)	512	1	0.001	0.5
Cystic Fibrosis	128	2	0.005	0.3
Soil	128	1	0.0001	0.5

Identification of Well-Predicted Metabolites

Using the background distributions generated by MiMeNet, the cutoffs for SCCs between the predicted and observed abundances of metabolites were found to be 0.136, 0.129, and 0.410 for the IBD (PRISM), cystic fibrosis, and soil datasets, respectively. Based on these cutoff values, MiMeNet identified metabolites to be well-predicted for 6857 (77.50%) of the 8848 metabolites in the IBD (PRISM) dataset,

143 (94.08%) of the 152 metabolites in the cystic fibrosis dataset, and 29 (34.12%) of the 85 metabolites in the soil dataset. The distributions of the SCCs in the background and observed data are shown in **Figure 24**. The soil dataset had the lowest percent of well-predicted metabolites, which could be due to the larger cutoff. We suspect that this is from the bootstrapping procedure being performed on the dataset of small size as well as the fact that the dataset is longitudinal and samples may be correlated with each other. Our evaluation shows the strong predictability of the MiMeNet models trained on data with reasonable sample sizes.

Effect of Multivariate Learning

To evaluate if multivariate learning improves the prediction of the metabolomic profiles, we trained two separate models using 10 iterations of 10-fold cross-validation using the IBD (PRISM) dataset. The first model was trained to predict the entire set of metabolites, and the second model was trained to predict the 466 annotated set of metabolites without including the rest of the metabolites. We then compared the SCCs of the 466 metabolites from both models and observed that by training on the entire set of metabolites, the number of well-predicted metabolites for the annotated set increased from 333 to 366. Additionally, the SCCs of the annotated metabolites significantly increased from 0.259 to 0.309 when using all the metabolites to train MiMeNet ($P < 10{-}47$, the Wilcoxon signed-rank test). The scatter plot comparing the prediction correlation performances is shown in **Figure 24**.

Next, we evaluated the robustness of MiMeNet by gradually increasing noise to the annotated set of metabolites. Specifically, with 10-fold cross-validation, we trained models using all the metabolites and using only the annotated set to predict the 466 annotated set. For each partition of the cross-validated training, we added Gaussian noise to the annotated metabolites within the training data. We observed that the two models performed similarly under small amounts of noise. However, once the noise increased to higher levels and had a variance greater than 2, the models trained only on the annotated set collapsed and

could no longer predict the annotated metabolites. On the other hand, the models trained using all the metabolites were much more robust to the noise at higher levels and could predict the annotated metabolites to a much greater degree compared to those trained using only the annotated set (**Figure 24**). These results show the MiMeNet framework benefited from multivariate learning.

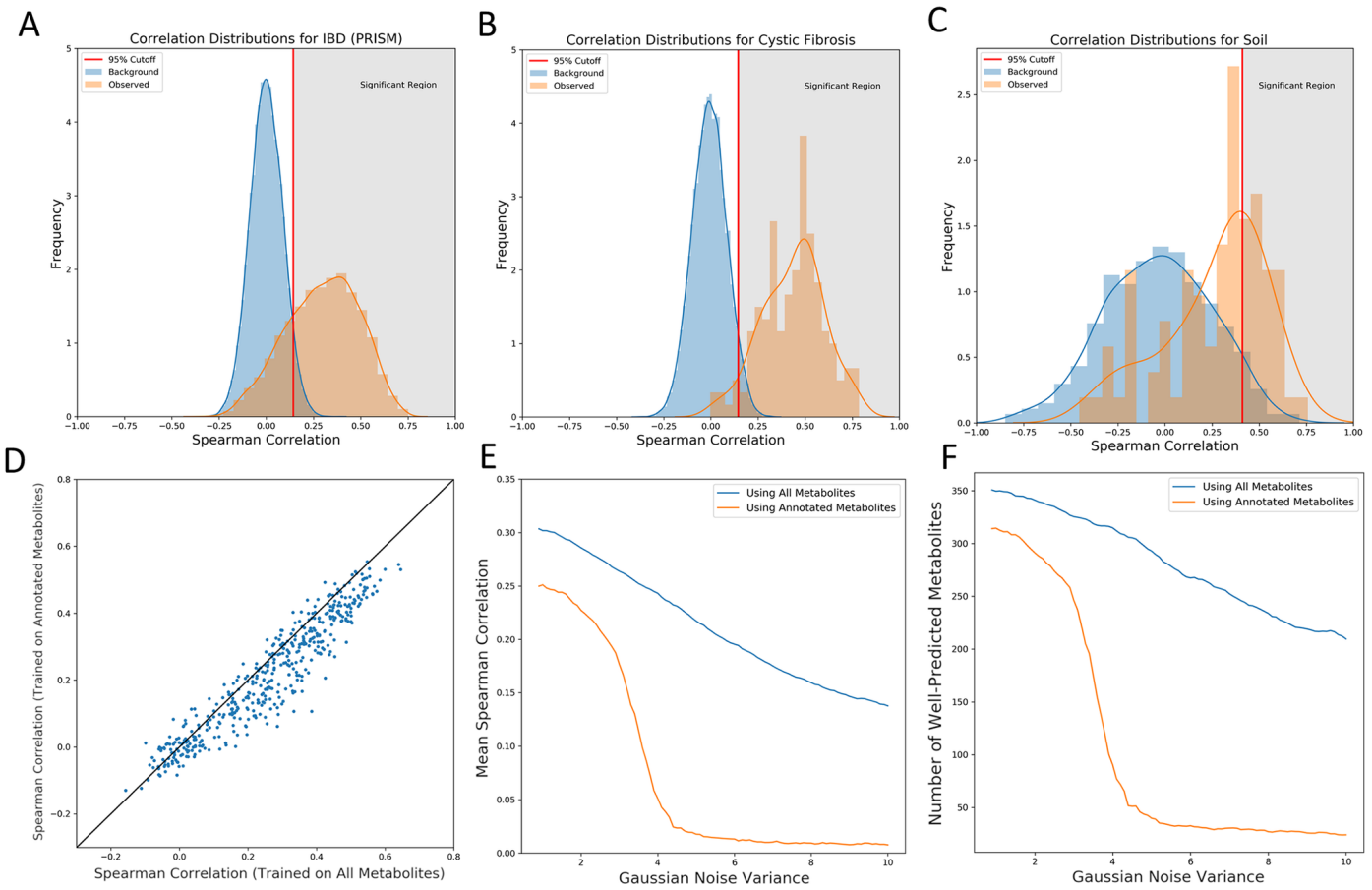


Figure 24. Distribution of background SCC values and evaluation of multivariate learning in MiMeNet. Background (blue) and observed (orange) distributions are shown for the (A) IBD (PRISM), (B) cystic fibrosis, and (C) soil datasets. The red vertical line denotes the 95th percentile of the background correlations and the gray area represents the well-predicted region using this threshold. (D) Scatter plot comparing the annotated metabolite correlations between models trained on just the annotated set and models trained on the full set of metabolites. (E) Mean correlation and (F) number of well-predicted metabolites found in models trained on the annotated set of metabolites and full set of metabolties as Gaussian noise is added to the annotated metabolite set input. All results in (D)- (F) are for prediction of the annotated metabolites.

Robustness to Training Set Size and Hyper-parameter Selection

To evaluate MiMeNet performance on different sizes of data for training and testing, we compared the k-fold cross-validated prediction correlations (k = 10, 5, 3, and 2) using the IBD (PRISM) and cystic fibrosis datasets (the soil dataset was excluded from this analysis due to the small data size). In the IBD (PRISM), we only observed a slight decrease in performance (mean correlation coefficient decrease from 0.297 to 0.218) as the number of partitions decreased. Similarly, in the cystic fibrosis dataset, the correlation dropped slightly from 0.457 to 0.410. Additionally, we evaluated performance on random subsetting of 100%, 80%, 60%, and 40% of the entire datasets. For each level of subsetting, 3 random sets of subset data were generated. Then, for each set of data, network hyper-parameters were tuned and 10 iterations of 10-fold cross-validation were performed to evaluate how reducing the number of overall samples affected the prediction correlation. As the size of the dataset decreased, we observed a decrease in the IBD (PRISM) dataset from a mean correlation of 0.287 to 0.179, and a decrease in the cystic fibrosis dataset from a mean correlation of 0.443 to 0.364. Moreover, we evaluated the IBD (External) dataset for each MiMeNet model trained on the IBD (PRISM) dataset and observed a decrease in mean correlation from 0.222 to 0.162. Even though there was a decrease in overall correlations as expected, we show that MiMeNet can still predict the metabolomic profiles when using smaller sets of training data. Boxplots showing the mean prediction correlations are shown in **Figure 25.**

We also compared the performance of the prediction using two types of microbial abundance representations: relative abundance (RA) and the centered log-transformation of abundance (CLR). The prediction correlations in the IBD (PRISM) dataset were comparable between both transformations, however, we saw an increase in correlations in the cystic fibrosis and soil datasets when using CLR. In addition, we observed an improvement in prediction performance on the IBD (External) test set when using the CLR transformation **Figure 26**.

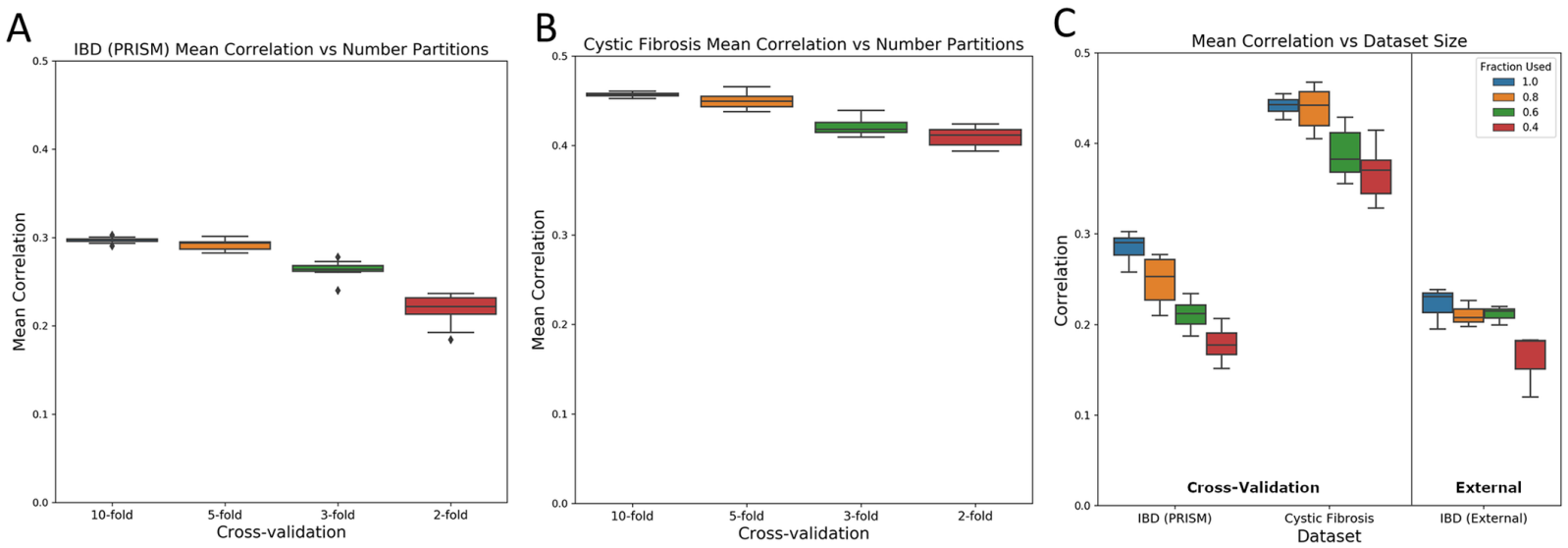


Figure 25. Mean correlation analysis in MiMeNet using different amounts of training data. Correlations for 10-, 5-, 3-, and 2- fold cross-validation evaluations are shown for the (A) IBD (PRISM) and (B) cystic fibrosis datasets. (C) Subsets of the IBD (PRISM) and cystic fibrosis corresponding to 100%, 80%, 60%, and 40% of the total samples are used as an input for MiMeNet. Three random datasets for each level of subsetting were created and then mean correlation using 10 iterations of 10-fold cross-validation is calculated across the three. In addition, models trained on the complete subsets of the IBD (PRISM) data are used to evaluate the IBD (External) test set.

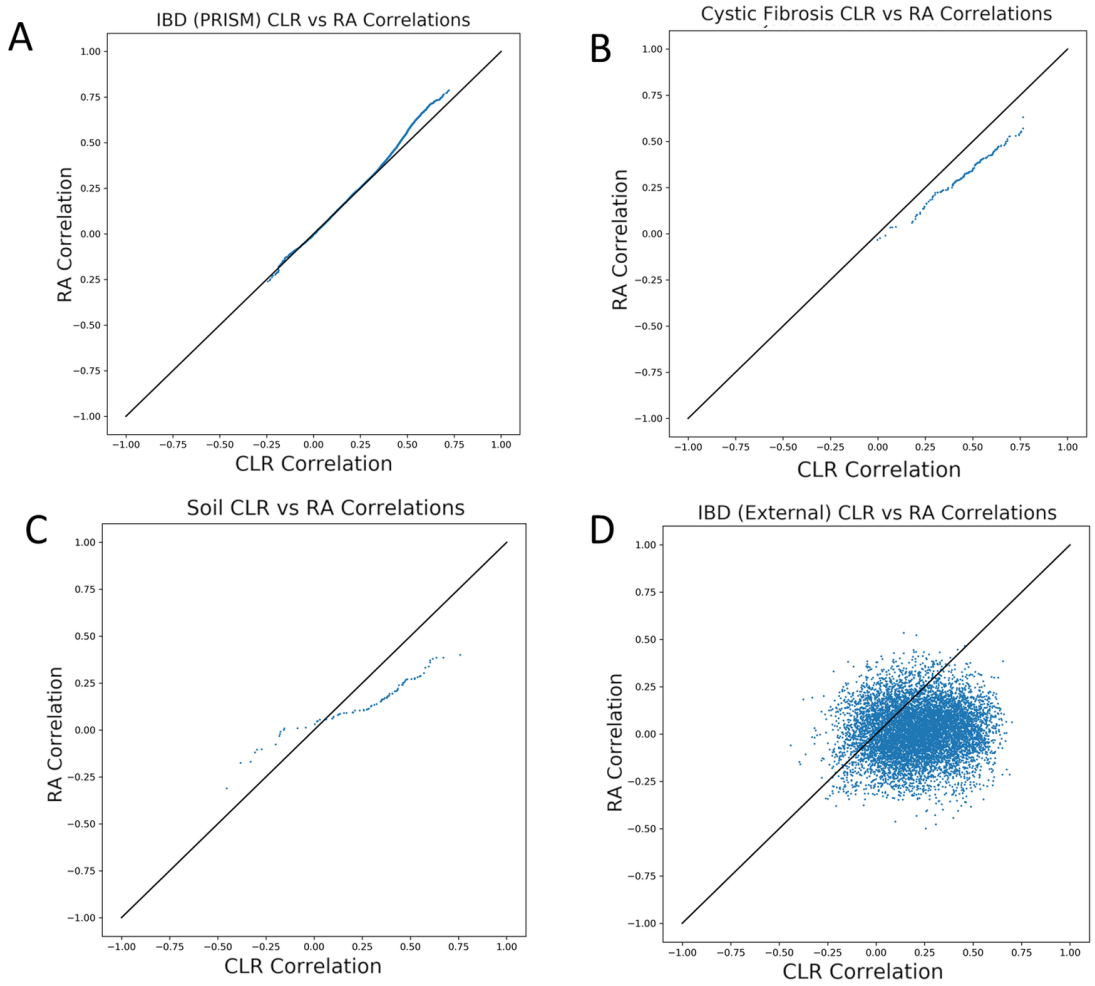


Figure 26. Comparison of prediction correlation when using relative abundance and centered log-ratio. Scatterplots comparing metabolite correlation prediction between data transformed to relative abundance (RA) and centered log-ratio (CLR) for (A) IBD (PRISM), (B) cystic fibrosis, (C) soil datasets using 10 iterations of 10-fold cross-validation, and (D) IBD (External) test predictions using models trained on the full IBD (PRISM) dataset.

Lastly, we evaluated if sharing the learned hyper-parameters across all cross-validated partitions in MiMeNet lead to overfitting. Although performing a single run of hyper-parameter tuning that is shared allows for much more computational efficiency, it could potentially be a source of bias. We evaluated the IBD (PRISM) and cystic fibrosis datasets using a single shared hyper-parameter set learned on the first partition against cross-validation where hyper-parameters are tuned every partition. Using the IBD (PRISM) dataset, we observed an increase in mean SCC when tuning every iteration, while in the cystic fibrosis dataset, we observed a decrease in mean SCC. Despite the decrease of performance in the cystic fibrosis dataset, 141 of the 143 significantly correlated metabolites were still identified. Comparisons of the two evaluations for each dataset are shown in **Figure 27**. Together, MiMeNet shows a robust performance using the proposed parameter-tuning procedure.

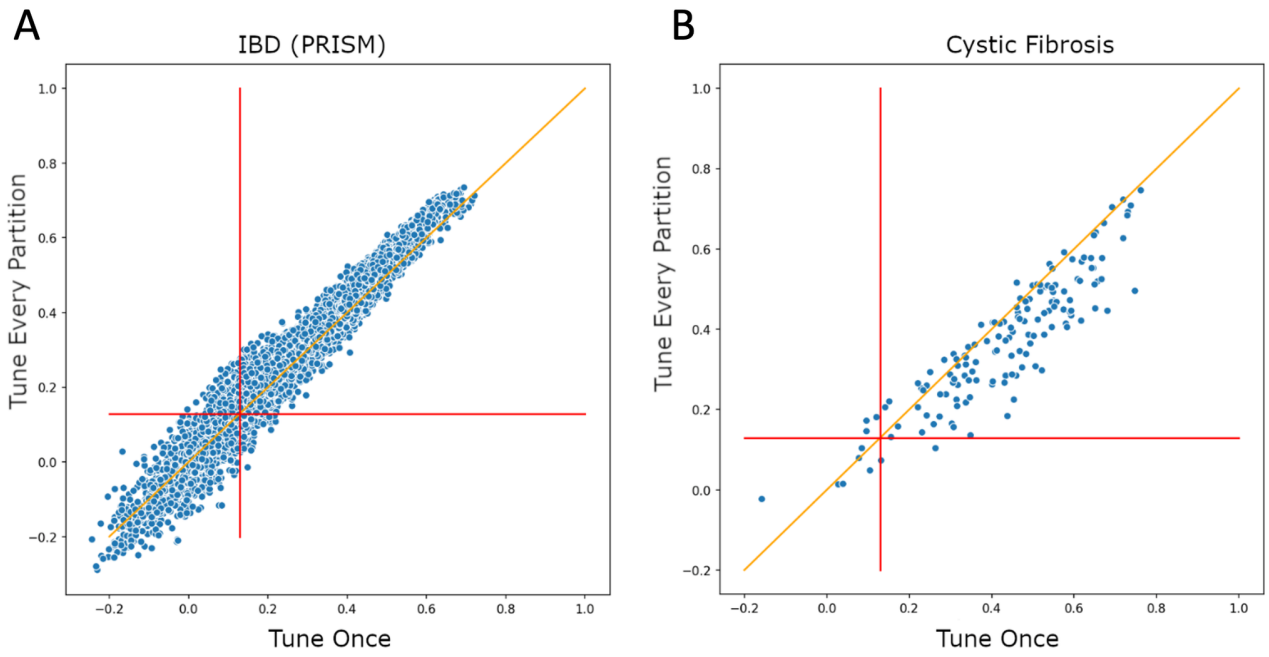


Fig. 27. Performance comparison of models trained using shared hyper-parameters against models trained with tuning hyper-parameters every cross-validated partition. Using 10-iterations of 10-fold cross-validation, evaluations using shared hyper-parameters tuned from the first partition (Tune Once) were compared against evaluations with tuning for each partition (Tune Every Partition) for the IBD (PRISM) and cystic fibrosis dataset. Each point represents the mean SCC of a metabolite and the red lines represent the determined SCC

Evaluation of Prediction Performance

For benchmarking, we first compared MiMeNet to MelonnPan, a recent model that uses Elastic Net linear regression to predict metabolite abundance from microbial abundance features (Mallick et al., 2019). Elastic net regression applies a linear combination of both L_1 and L_2 regularizations to avoid overfitting. In the case of MelonnPan, a linear model is trained for one metabolite at a time and cannot benefit from multivariate learning. In our study, MelonnPan was evaluated using the same data partitions of the 10 iterations of 10-times cross-validation for each dataset. However, in the case of the IBD (PRISM) dataset, only the annotated metabolites were predicted due to the large running time for the entire metabolite set. On the other hand, MiMeNet was trained to predict all metabolites in the IBD (PRISM) dataset. We observed that in each of the datasets trained using cross-validation, MiMeNet has a higher correlation for prediction across all metabolites when compared to MelonnPan. In the IBD (PRISM) dataset, the mean correlation increased from 0.108 to 0.309 when evaluating the annotated metabolites. When training MiMeNet only on the annotated metabolites, we observed a similar result with an increased correlation to 0.259. In the cystic fibrosis dataset, the mean correlation increased from 0.276 to 0.457. In the soil dataset, the mean correlation from MelonnPan was -0.272 and was increased to 0.264 using MiMeNet. Moreover, we evaluated the performance of the models obtained from MelonnPan and MiMeNet using the IBD (External) dataset on the annotated metabolites. The mean correlation of the annotated metabolites was increased from 0.168 to 0.275. Comparisons between MiMeNet and MelonnPan for each dataset are shown in **Figure 28** when considering all metabolites and **Figure 29** when considering only the annotated metabolites in the IBD (PRISM) dataset for model training.

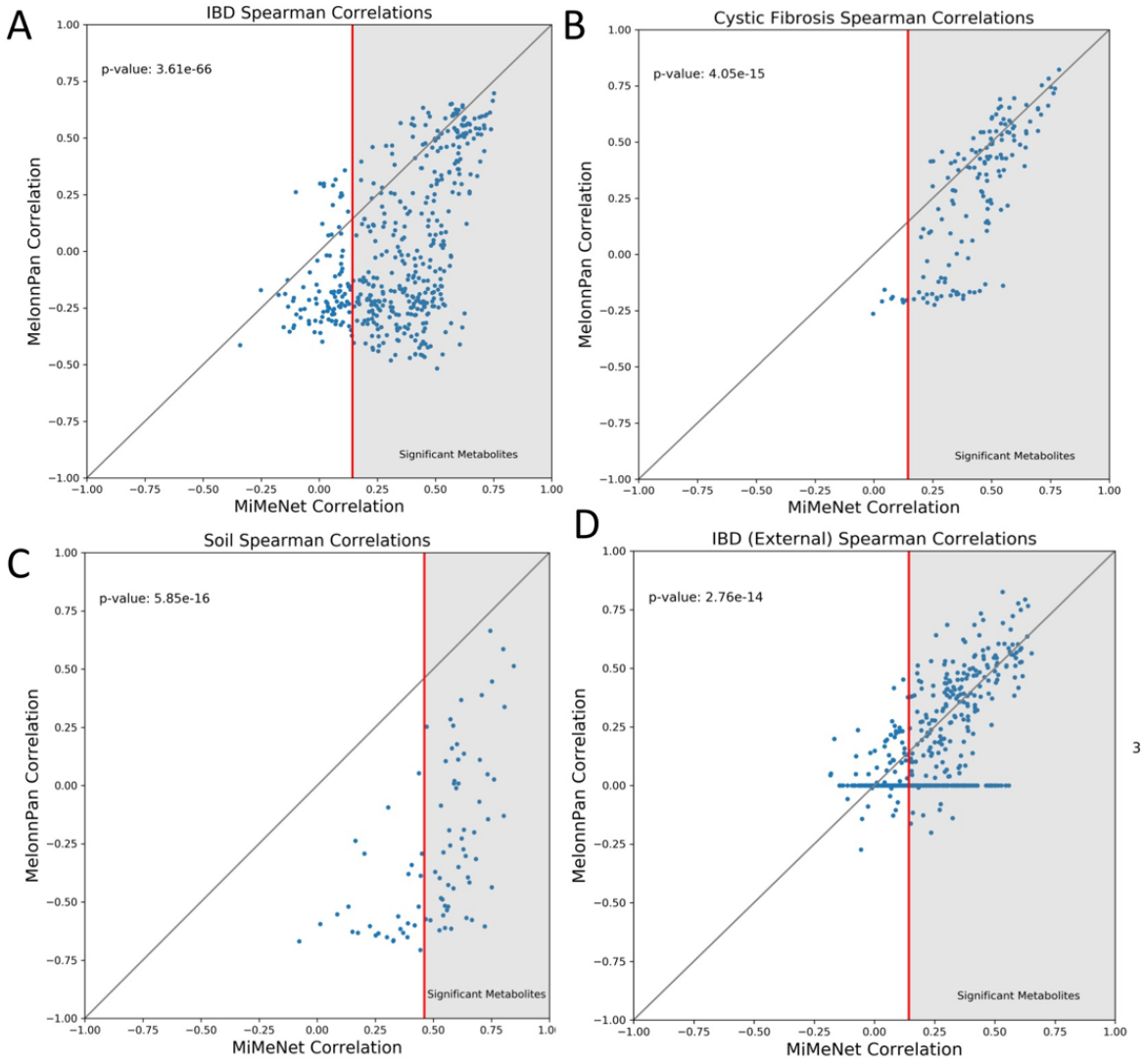


Figure 28. Comparison of MiMeNet with MelonnPan. Scatterplots showing the comparison of SCC values between observed and predicted metabolites when using MiMeNet and MelonnPan for the (A) PRISM IBD, (B) cystic fibrosis, (C) soil, and (D) IBD Validation datasets.

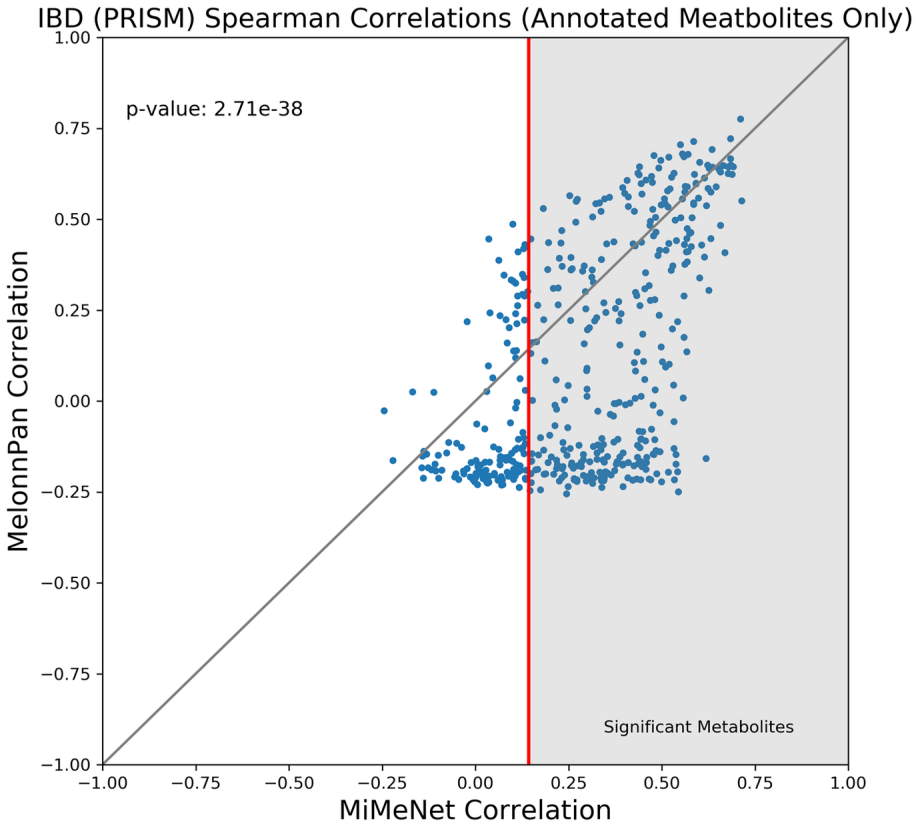


Figure 29. Comparison of MiMeNet with MelonnPan for IBD (PRISM) using only annotated metabolites for training. MieMeNet is trained just using the annotated metabolites rather than the entire set of metabolite features.

Additionally, within the IBD (PRISM) dataset, MiMeNet identified 351 well-predicted metabolites from the 466 annotated metabolites. Even though MelonnPan uses a default correlation cutoff of 0.3, when using the same correlation cutoff derived by MiMeNet, MelonnPan identified 198 well-predicted metabolites of which 181 were identified by MiMeNet. In the cystic fibrosis dataset, MiMeNet identified 143 well-predicted metabolites while MelonnPan identified 104. In the soil dataset, MiMeNet identified 29 well-predicted metabolites while MelonnPan identified 4. When training using the entire IBD (PRISM) dataset to predict the IBD (External) test data, MiMeNet identified 308 well-predicted metabolites while MelonnPan identified 186, of which 160 were also identified by MiMeNet. The overlap of the two methods across all datasets is shown in **Figure 30.**

When analyzing the overall prediction correlation and number of well-predicted metabolites, we observed that MiMeNet's improvement was not a global improvement across all metabolites, but rather it came from MiMeNet being able to model a large set of microbes that MelonnPan could not. For example, in the IBD (PRISM) dataset, there were 237 metabolites with a negative prediction correlation. Of these metabolites, MiMeNet was able to predict 160 with a correlation above the determined cutoff. These metabolites also made up the set of metabolites with a prediction correlation of 0 in the IBD (External) set when using MelonnPan. Upon investigation, this set of metabolites was predominantly composed of triacylglycerols, long-chain fatty acids, and bile acids. These three classes of metabolites have been shown to interact with various microbes as well as relate to IBD patients.

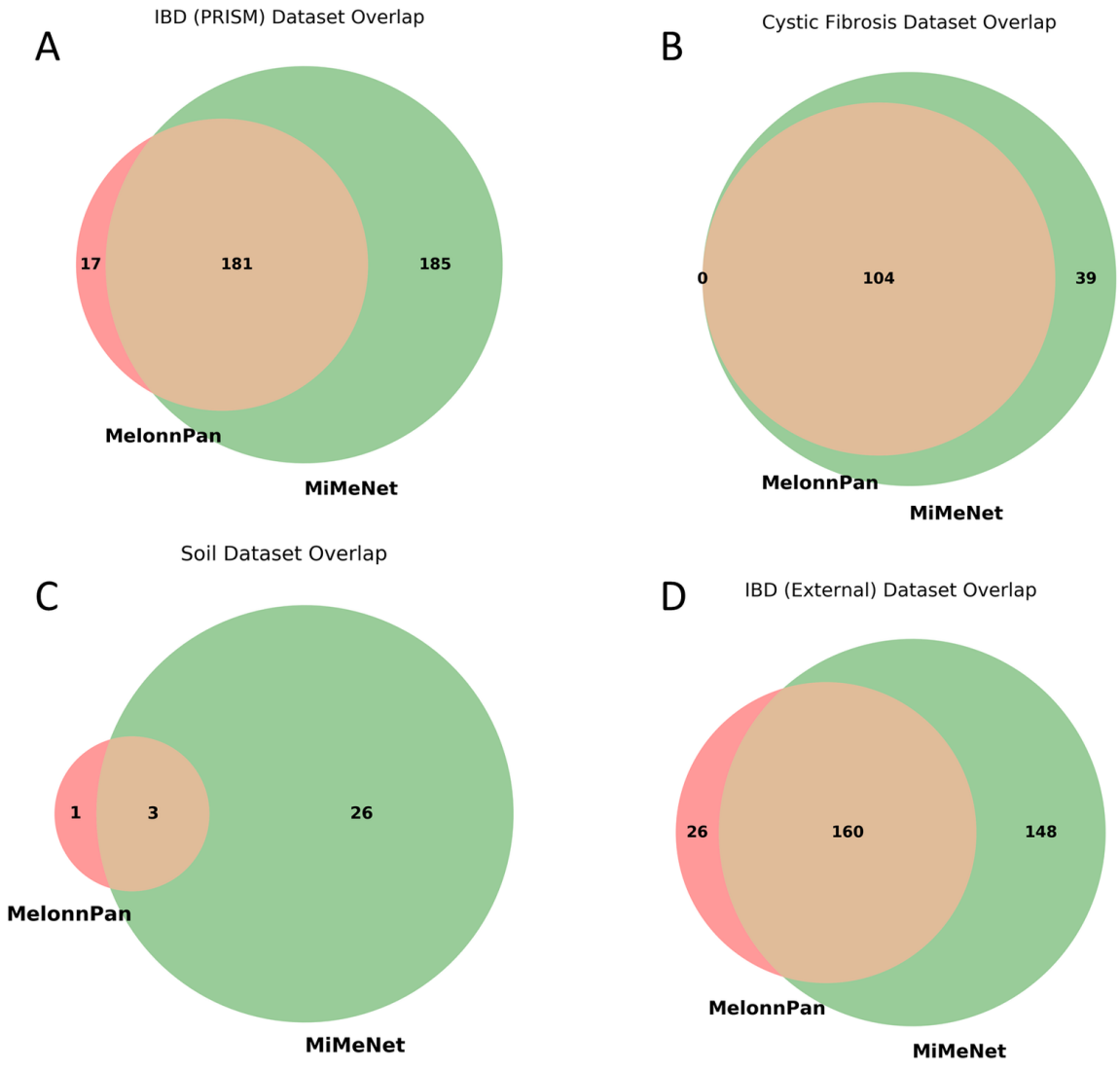


Fig. 30. Overlap of significant metabolites identified by MiMeNet and MelonnPan. Venn diagrams showing the overlap of significant metabolites identified by MelonnPan (red) and MiMeNet (green) in different datasets.

We observed that the running time of MiMeNet was robust and did not scale largely with the number of metabolites as all three datasets complete in similar timespans as shown in **Table XI**. These results show that MiMeNet benefited from multivariate learning, the scalability of MLPNN, and the ability of MLPNN in capturing complex relationships between microbiome and metabolomes.

TABLE XI. RUNNING TIME OF MIMENET AND MELONNPAN

	MiMeNet Running Time (H:M:S)	**MelonnPan Running Time (H:M:S)**
IBD (PRISM)	1:11:39	16:33:04
Cystic Fibrosis	1:18:05	3:47:27
Soil	1:54:24	1:31:29

In addition, we benchmarked MiMeNet against other general regression models, i.e., Random Forest (RF), multivariate Elastic Net, and canonical correlation analysis (CCA) models using 10 iterations of 10-fold cross validation. Based on three evaluation metrics, i.e., SCC, Pearson correlation coefficient (PCC), and mean absolute error (MAE), we observed that for the IBD (PRISM) and cystic fibrosis datasets, MiMeNet and RF models performed best. For the soil dataset, we observed that CCA models performed the best, which may be due to the extremely small sample size of the soil dataset. When models were trained on the entire IBD (PRISM) dataset to predict the IBD (External) dataset, we observed that MiMeNet outperformed all other models. Results for CLR and RA evaluations are shown in **Table XII** and **Table XIII** respectively. Additionally, an evaluation of RF and CCA for training on the entirety of the IBD (PRISM) dataset to predict the IBD (External) dataset is shown **in Table XIV**.

TABLE XII. EVALUATION OF MIMENET, RF, ELASTIC NET, AND CCA MODELS USING CLR TRANSFORMED DATA.

		MiMeNet	RF	Elastic Net	CCA (k=10)	CCA (k=20)	CCA (k=40)
IBD (PRISM)	SCC	0.31 ± 0.01	0.25 ± 0.01	0.25 ± 0.01	0.03 ± 0.02	0.05 ± 0.02	0.05 ± 0.02
	PCC	0.25 ± 0.02	0.27 ± 0.01	0.21 ± 0.01	0.01 ± 0.03	0.03 ± 0.04	0.01 ± 0.03
	MAE	1.48 ± 0.01	1.38 ± 0.01	1.50 ± 0.02	1.81 ± 0.07	2.19 ± 0.13	2.94 ± 0.15
Cystic Fibrosis	SCC	0.46 ± 0.01	0.42 ± 0.01	0.39 ± 0.01	0.14 ± 0.01	0.20 ± 0.01	0.27 ± 0.01
	PCC	0.48 ± 0.01	0.45 ± 0.01	0.45 ± 0.01	0.14 ± 0.01	0.20 ± 0.02	0.28 ± 0.01
	MAE	2.89 ± 0.01	2.98 ± 0.05	3.13 ± 0.01	4.70 ± 0.09	4.89 ± 0.14	5.11 ± 0.04
Soil	SCC	0.26 ± 0.03	0.17 ± 0.06	0.40 ± 0.02	0.46 ± 0.03	-	-
	PCC	0.29 ± 0.03	0.18 ± 0.08	0.43 ± 0.03	0.48 ± 0.03	-	-
	MAE	0.94 ± 0.01	0.98 ± 0.04	0.87 ± 0.02	1.06 ± 0.03	-	-

TABLE XIII. EVALUATION OF MIMENET, RF, ELASTIC NET, AND CCA MODELS USING RA TRANSFORMED DATA.

		MiMeNet	MelonnPan	RF	Elastic Net	CCA (k=10)	CCA (k=20)	CCA (k=40)
IBD	SCC	0.27 ± 0.01	0.11 ± 0.01	0.24 ± 0.01	0.18 ± 0.01	0.02 ± 0.03	0.03 ± 0.02	0.04 ± 0.02
	PCC	0.18 ± 0.01	0.06 ± 0.01	0.16 ± 0.01	0.12 ± 0.01	0.01 ± 0.02	0.01 ± 0.02	0.01 ± 0.02
	MAE	5.03 x 10-4 ± 4.2 x10-8	1.51 x 10-3 ± 1.26 x 10-4	4.042 x 10-4 ± 2.65 x10-6	0.002 ± 7.75 x 10-6	6.51 x 10-4 ± 3.03 x 10-5	8.35 x 10-4 ± 3.45 x 10-5	8.12 x 10-4 ± 3.45 x 10-5
Cystic Fibrosis	SCC	0.32 ± 0.01	0.28 ± 0.01	0.43 ± 0.01	0.31 ± 0.01	0.08 ± 0.01	0.14 ± 0.01	0.19 ± 0.01
	PCC	0.26 ± 0.06	0.30 ± 0.01	0.47 ± 0.02	0.35 ± 0.01	0.01 ± 0.01	0.02 ± 0.01	0.01 ± 0.01
	MAE	0.006 ± 1.65 x 10-6	0.005 ± 3.5 x10-5	0.005 ± 0.03	0.006 ± 8.62 x 10-5	0.03 ± 0.010	0.067 ± 0.013	0.116 ± 0.013
Soil	SCC	0.14 ± 0.05	-0.27 ± 0.04	0.19 ± 0.06	0.29 ± 0.01	0.33 ± 0.02	-	-
	PCC	0.07 ± 0.05	-0.27 ± 0.03	0.11 ± 0.04	0.22 ± 0.01	0.18 ± 0.04	-	-
	MAE	0.008 ± 2.00 x 10-4	0.006 ± 1.06 x 10-4	0.006 ± 3.71 x 10-4	0.011 ± 0.009	0.021 ± 0.002	-	-

TABLE XIV. EVALUATION OF MIMENET, RF, ELASTIC NET, AND CCA MODELS ON IBD (EXTERNAL) DATASET.

		MiMeNet	MelonnPan	RF	Elastic Net	CCA (k=10)	CCA (k=20)	CCA (k=40)
Centered Log-Ratio	SCC	0.24	-	0.19	0.21	0.05	0.12	0.06
	PCC	0.25	-	0.22	0.2	0.03	0.11	0.04
	MAE	1.34	-	1.37	1.98	1.31	1.31	1.34
Relative Abundance	SCC	0.21	0.17	0.17	0.15	0.07	-0.02	0.01
	PCC	0.17	0.16	0.17	0.13	0.05	-0.02	-0.02
	MAE	4.10 x 10-4	0.001	0.0001	4.24 x 10-4	4.37 x 10-4	4.37 x 10-4	4.52 x 10-4

Lastly, we compared MiMeNet to the NED model using the PRISM IBD dataset. The other datasets were not evaluated using NED due to limitations of its implementation during the time of evaluation. Although both methods implemented neural network-based approaches to model the full-scale metabolomic profile, when comparing with the NED model, we observed a large portion of metabolites with a correlation of predicted to observed to be 0. This means that the NED model was not able to model these metabolites, and we suspect this is due to the harsh constrains imposed by the model of sparsity and positive weights. Comparison between MiMeNet and NED is shown in **Figure 31**.

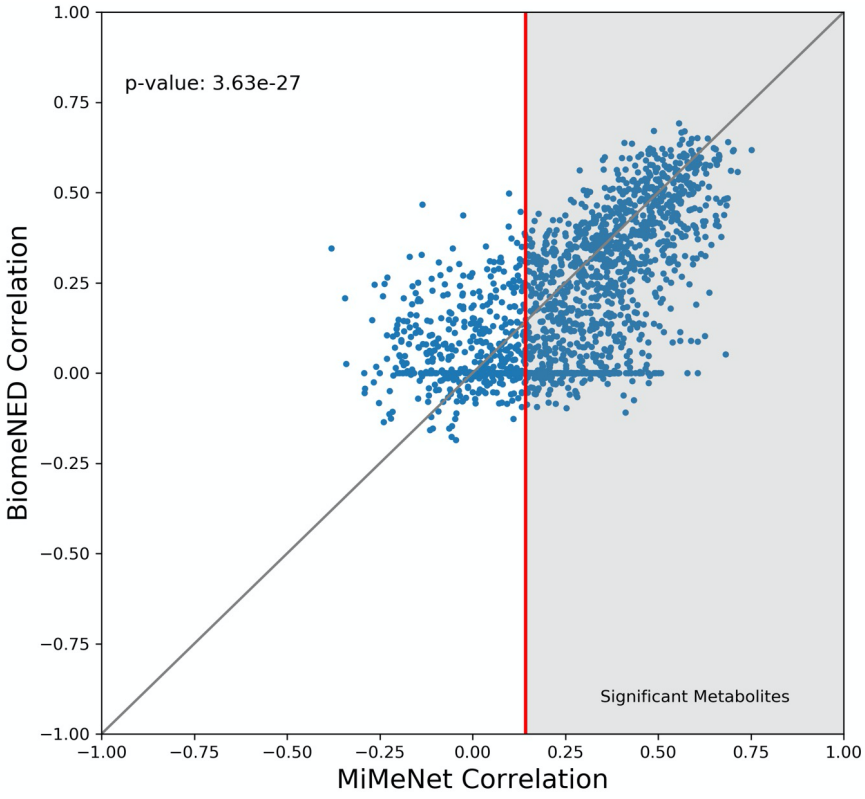


Figure 31. Comparison of MiMeNet with BiomeNED. Scatterplots showing the comparison of SCC values between observed and predicted metabolites when using MiMeNet and BiomeNED for the PRISM IBD dataset.

Identification of Microbial and Metabolic Modules

For the analysis and visualization of microbial and metabolic modules, we focus on the IBD data since it contains both control and case samples. We computed the feature attribution scores for all pairs of microbes and the 6857 well-predicted metabolites using the network weights of the trained models obtained from the IBD (PRISM) data set. We identified 163 microbes that had at least one significant attribution score with a well-predicted metabolite. A positive score means that the microbe contributes positively to the prediction of the abundance of the metabolite. Likewise, a negative score contributes negatively to the prediction of the abundance of the metabolite. To reveal the pattern of attribution scores, we grouped the microbes and metabolites into modules using biclustering. We identified 8 modules of microbes and metabolites respectively based on clustering shown in **Figure 32** and computed the module feature value as the average abundance of features within the module for each sample.

We further went to examine if a module is enriched for one patient group (IBD or healthy) by comparing the average normalized feature values of the members within the module between the two groups using the IBD (PRISM) samples ($P<0.05$, Wilcoxon rank-sum test). We observed that 7 of the 8 microbial modules were significantly different between groups using the IBD (PRISM) data. Using the IBD (External) data, two of the modules were still significantly different; and even though other modules were no longer significant, they shared similar trends in the differences between groups. We also found that 7 of the 8 metabolic modules were significantly different between groups using the IBD (PRISM) data and when using the IBD (External) data, the same 7 modules were also significantly different between groups. Boxplots showing enrichment results between healthy and disease patients across all modules is shown in **Figure 33.**

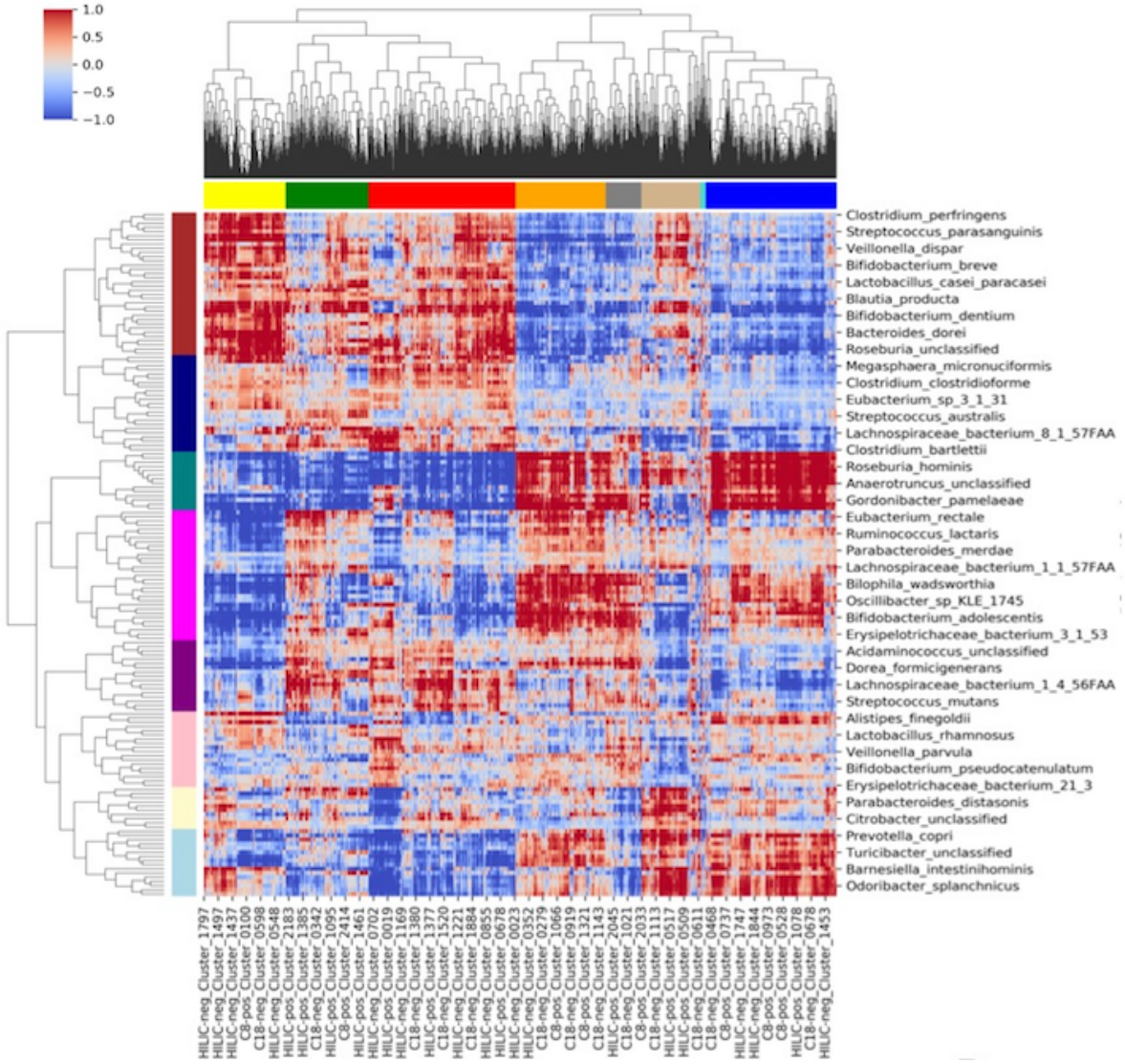


Figure 32. Module biclustering of IBD (PRISM) dataset. MiMeNet identified 8 microbial (row) and 8 metabolic (column) modules. Biclustering was done using hierarchical clustering using complete linkage and Euclidean distance.

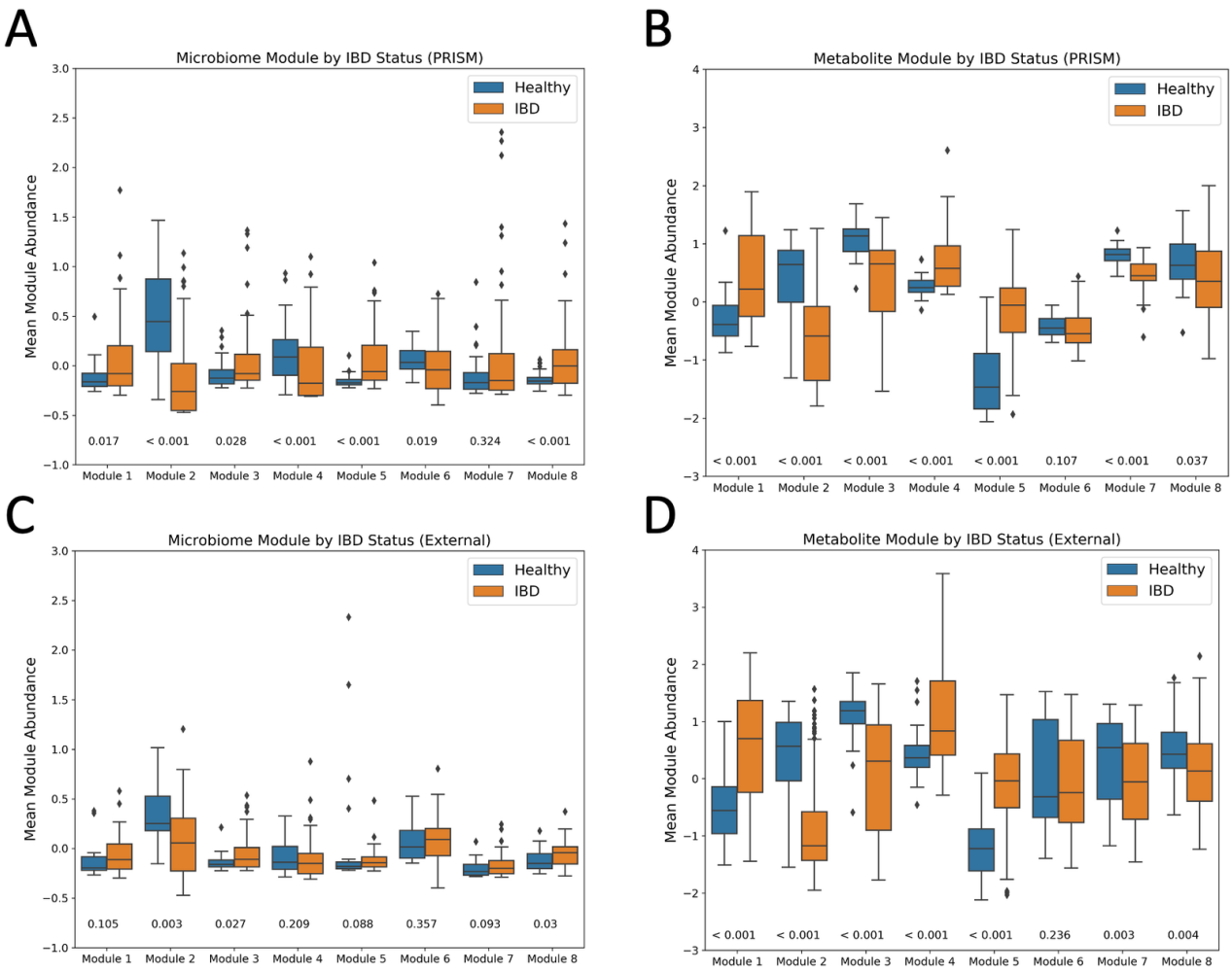


Figure 33. Microbial and metabolic module abundance by patient status in the IBD (PRISM) dataset. The mean normalized abundance of members within modules are shown here for healthy patients and IBD patients from the IBD (PRISM) dataset using (A) microbial and (B) metabolic modules and from the IBD (External) dataset using (C) microbial and (D) metabolic modules identified by MiMeNet. P-values from a two-sided Wilcoxon rank-sum test are shown on the bottom.

To further examine the MiMeNet's predictive performance in each metabolite module, we calculated the mean SCC and PCC values of members within each module from both the cross-validated evaluation and the evaluation on the IBD (External) data. For the cross-validated evaluation, the mean SCC for each module ranged from 0.25 to 0.41 and the mean PCC ranged from 0.21 to 0.35, showing that each module contributed to the overall prediction performance of the MiMeNet model. On the IBD (External) evaluation, the mean SCC ranged from 0.14 to 0.28 and the mean PCC ranged from 0.06 to 0.25. Although the values decreased in the IBD (External) data, the modules with higher mean SCC and PCC values in the cross-validated evaluation were also the modules with the higher SCC and PCC values in the IBD (External) data. Taken together, these results show that the predictive ability as well as the information carried by the collective members of each module were transferrable to an external cohort of patients. Mean SCC values for each model in the IBD (PRISM) and IBD (External) datasets are shown in **Figure 34.**

Next, we compared the microbial modules in the IBD (PRISM) dataset identified by MiMeNet to those identified by the Weighted Correlation Network Analysis (WGCNA) (Methods). The module features of a sample were calculated as the average normalized abundance of the members within the module. Using the Jaccard similarity between the members of the modules as well as the Spearman correlation coefficient between module features across samples, we observed only a small consensus between the two groupings as shown in **Figure 35.**

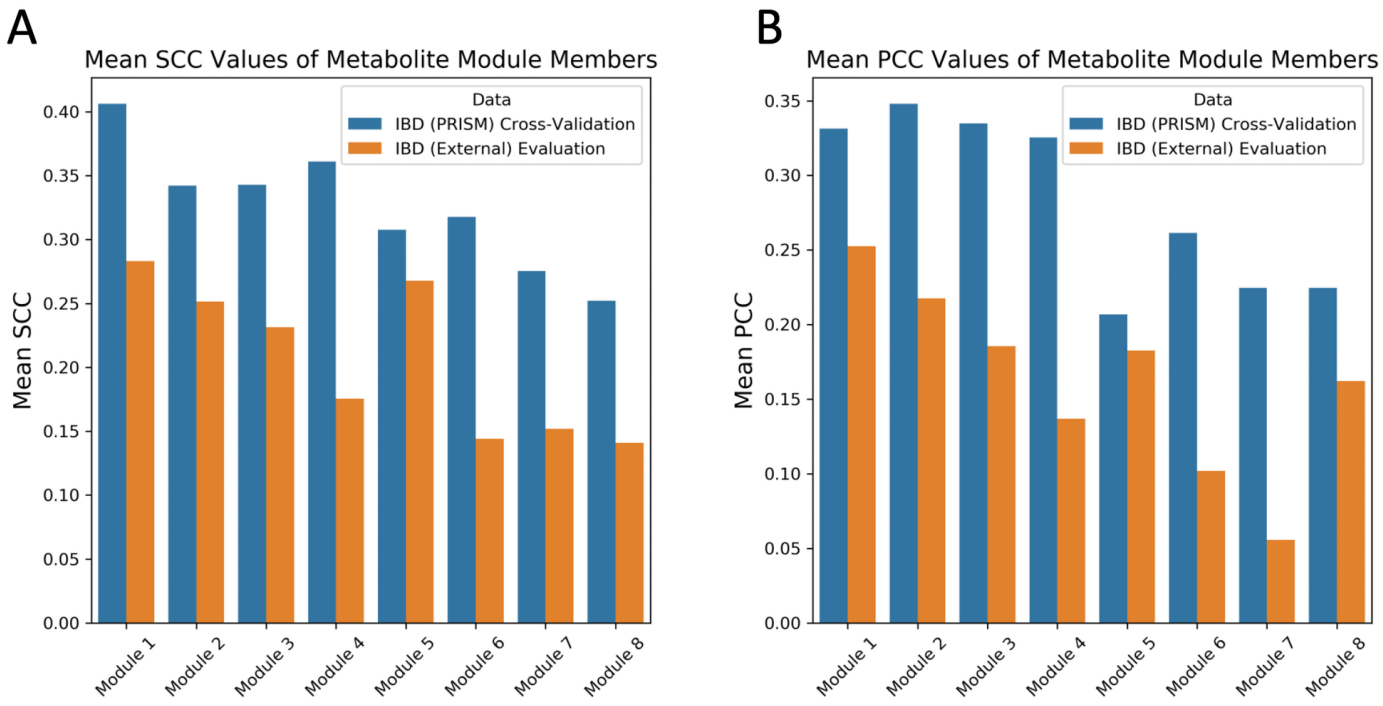


Figure 34. Mean Spearman and Pearson correlation per metabolite module. For each metabolite module, the (A) mean SCC and (B) mean PCC values of the members within the module are shown using the cross-validated evaluation on IBD (PRISM) as well as when evaluating the IBD (External) data.

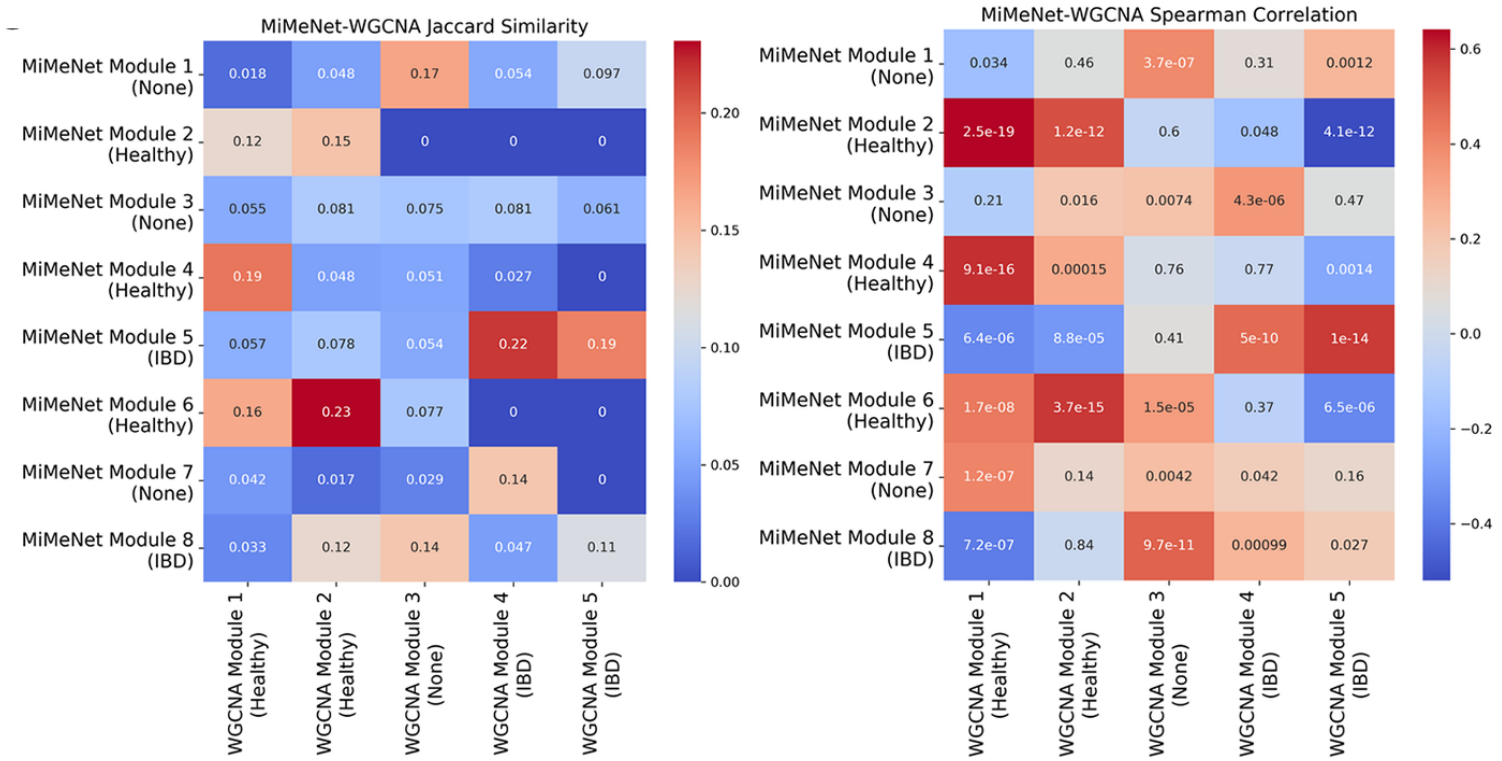


Figure 35. Jaccard Index and Spearman correlation between module features of WGCNA and MiMeNet modules. Jaccard Index values (left) and Spearman correlation p-values (right) shown in the boxes for reference.

To evaluate how well MiMeNet's modules captured underlying functional activity, we evaluated the module's predictive performance when trying to predict the disease status of a sample. To do so, we constructed module-based features by taking the average normalized abundance value of the members within each module. We then trained small neural network models using 3 layers of 32 nodes to predict IBD status using either microbiome or metabolic module-based features. We compared this evaluation to models trained on the original features as well as modules detected by WGCNA. We observed that MiMeNet's modules were significantly more predictive than those identified by WGCNA. In addition, we observed no loss of predictive performance when compared to models trained on the original features, suggesting that although MiMeNet was never given disease status, it was able to group microbes and metabolites representative of the underlying metabolic dysregulation of the disease. AUC-ROC values for models trained using microbiome and metabolic features are shown in **Figure 36**.

Lastly, using the microbial and metabolic modules, we constructed a bipartite interaction network. Interaction scores were taken as the mean pairwise interaction score between members of a microbial and metabolic module. The bipartite graph constructed from the microbial and metabolic modules generated from the IBD dataset is shown in **Figure 37**.

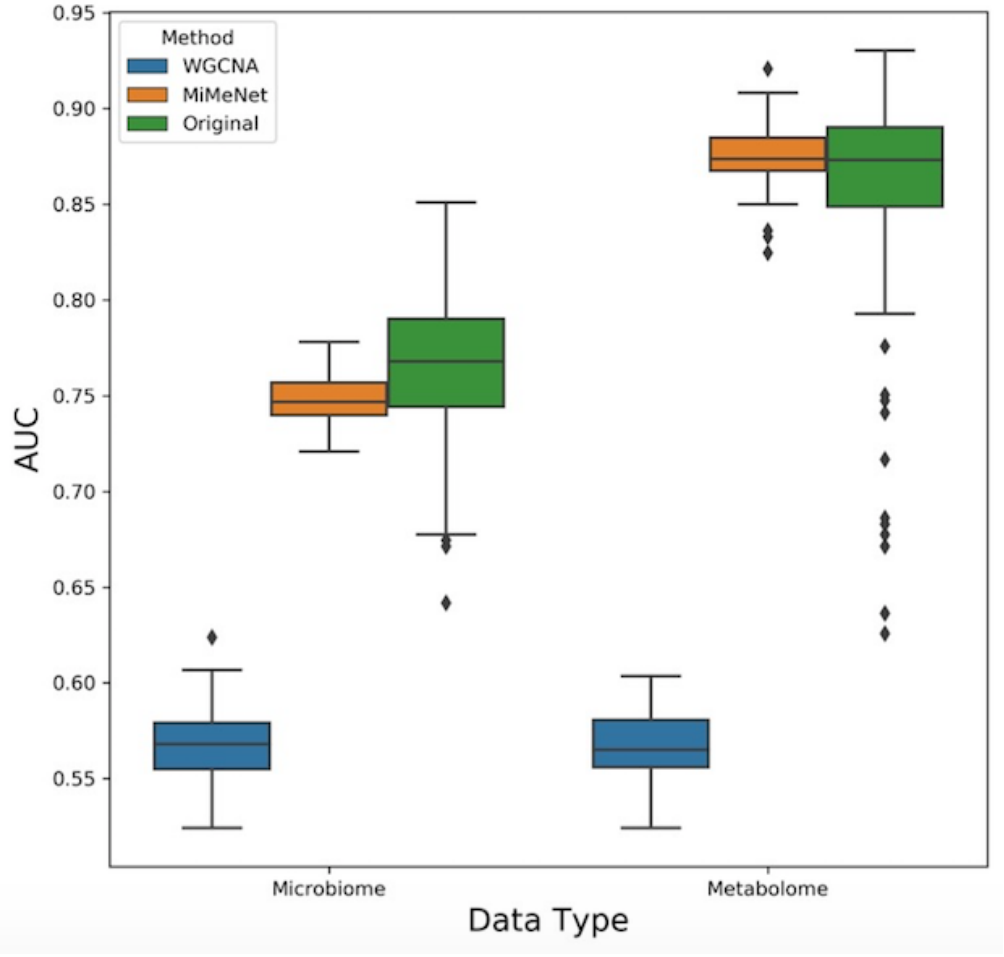


Figure 36. AUC values for prediction of IBD status using MiMeNet modules, WGCNA modules, and original features. AUC values using microbial (left) and metabolic (right) features. Modules derived from WGCNA (blue) and MiMeNet (orange) are compared to the models trained on the original values (green).

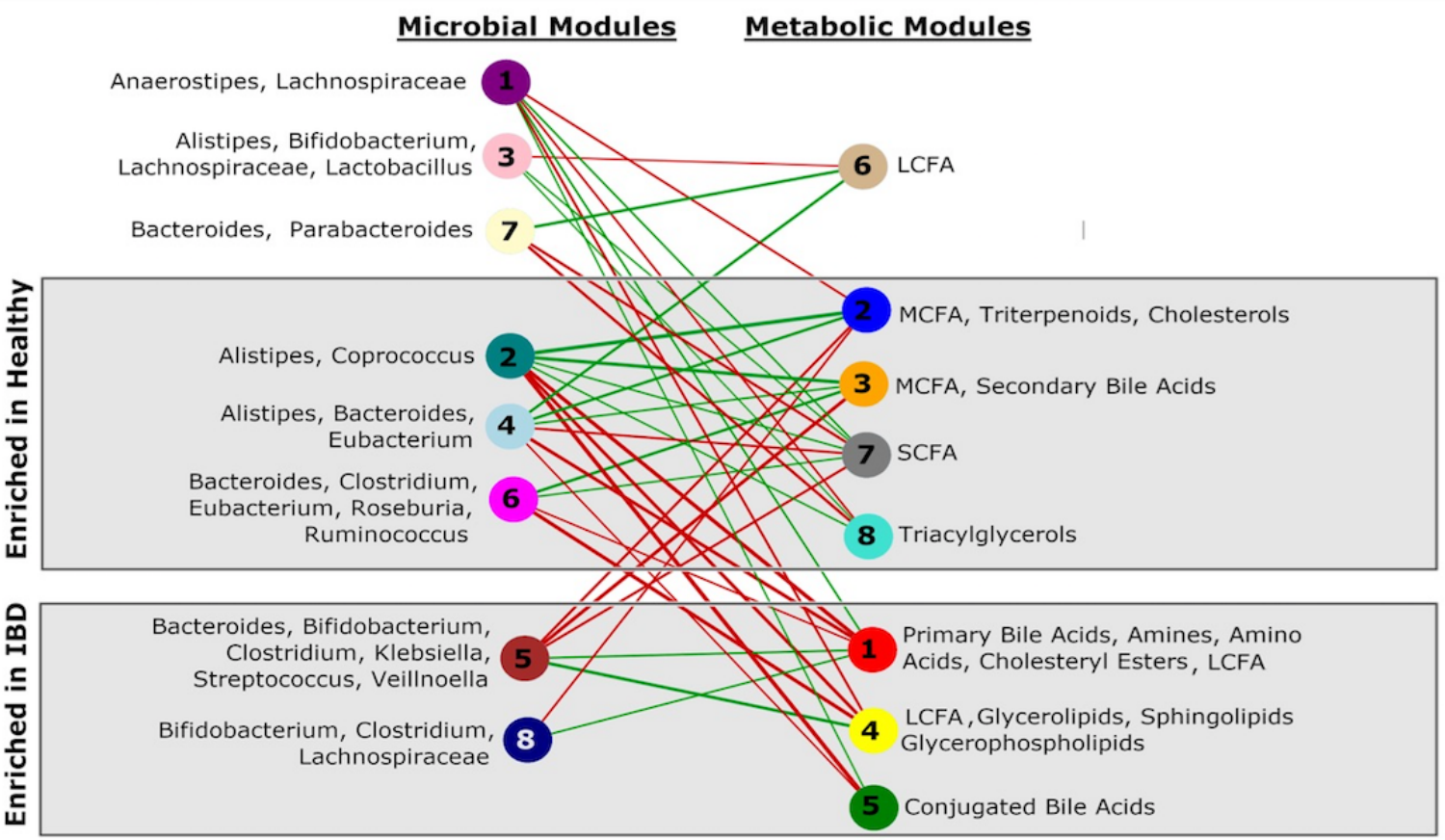


Figure 37. Bipartite interaction network. Bipartite module-based interaction network between microbial (left) and metabolic (rigt) modules identified in the PRISM IBD dataset. Modules enriched in IBD and healthy patients are grouped together. Green edges represent a positive interaction and red edges represent a negative interaction.

Biological Function of Modules

We further went to examine if a module is enriched for one patient group (IBD or healthy) by comparing the average module feature values of the two groups ($P<0.05$, Wilcoxon rank-sum test). Four metabolic modules were enriched in healthy subjects. The first module (module 2) contained medium chain fatty acids (MCFA), triterpenoids, and cholesterols. Triterpenoids have been shown to have an anti-inflammatory effect as well as enhancing intestinal tight junctions and have been explored as therapeutic options for treating IBD (Dong et al., 2020; C. Liu et al., 2015; Mueller, Triebel, Rudakovski, & Richling, 2013). Additionally, both cholesterols and MCFAs have been shown to be reduced in subjects with IBD (Agouridis, Elisaf, & Milionis, 2011). The second module (module 3) contained MCFA molecules as well as secondary bile acids. Secondary bile acids which have found to be reduced in IBD patients in previous studies (De Preter et al., 2015; Heinken et al., 2019). The third module (module 7) was composed of short-chain fatty acids (SFCA) which have been shown to be protective against IBD (Parada Venegas et al., 2019). The final module (module 8) contained triradylcglycerols, which have been reported to be lowered in subjects with IBD (Agouridis et al., 2011).

Similarly, three metabolic modules were found enriched in IBD patients. The first module (module 1) was composed of primary bile acids, amines, amino acids, cholesteryl esters, and long chain fatty acids (LCFA). Primary bile acids deconjugated to secondary bile acids by microbes in the gut. Studies have found that subjects with IBD have an impaired ability of the deconjugation of primary bile acids, causing a greater abundance of primary bile acids (Heinken et al., 2019). Additionally, primary bile acids have been shown in previous studies to bind to the Farnesoid X receptor, which is linked to the elevated immune response observed in IBD (Vaughn et al., 2019). Cholesteryl esters have also been shown to be enriched in subjects with IBD, due to lipid mobilization and increased intestinal permeability (Tefas, Ciobanu, Tanțău, Moraru, & Socaciu, 2020). Interestingly, the amine group within this module composed of only N-acylethanolamines as well as sphingosine. N-acylethanolamines have been shown to alter the gut microbiome and increase the levels of lipopolysaccharides in the intestines (Fornelos et al., 2020).

Sphingosine in conjunction with fatty acids make up ceramide, another metabolite found within this module. Ceramide is a precursor to sphingosine-1-phosphate, a signaling sphingolipid which is believed to increase inflammation in the gut (Suh & Saba, 2015). The largest group in this module, however, comprised of 31 (24.4%) different amino acids. Amino acids have been found in a previous study to be elevated in IBD patients due to the increase in the bacterial enzyme urease. Elevated urease promotes a flux of nitrogen, which is then used for the synthesis of amino acids by the host (Ni et al., 2017). The second module (module 4) contained LCFAs, glycerolipids, glycerophospholipids, and sphingolipids. Previous studies have shown that bacterial-derived sphingolipids have been shown to play a crucial role in the development of IBD through multiple signaling pathways (Abdel Hadi, Di Vito, & Riboni, 2016). The elevated glycerolipids and glycerophospholipids were also identified in the original study (Franzosa et al., 2019). The third module (module 5) contained mostly conjugated bile acids. Similar to primary bile acids, conjugated bile acids have been shown to be elevated in IBD subjects due to the decreased ability to deconjugate the bile acids into secondary bile acids.

4.3 Conclusion

Based on our analyses, MiMeNet effectively integrates microbiome and metabolomic data to uncover functional microbial and metabolic modules and interactions between them. By training an ensemble of DNN models, MiMeNet provides superior predictive performance of the metabolic community using microbial features compared to other state-of-the-art methods. Additionally, the models facilitate the clustering of microbes and metabolites into meaningful functional modules, which we show can capture the underlying dysregulation of disease. MiMeNet is publicly available and can be downloaded from https://github.com/YDaiLab/MiMeNet.

Chapter 5

Deep Learning Networks with Diversity-Regularized Autoencoder for Modeling Longitudinal Microbiome Data

5.1 Introduction

Engineering the gut microbiome for the treatment of disease is an exciting new direction in medical science (Creswell et al., 2020; Holmes et al., 2012; Ronda, Chen, Cabral, Yaung, & Wang, 2019; Staley et al., 2018). A recent study using mouse models showed microbial engineering to be effective in treating hyperammonemia (Shen et al., 2015). Currently, fecal microbiota transplants are the medical procedure to induce microbiome engineering for therapeutic purposes. Although they have been shown to be effective, these procedures have an extreme effect in microbial reconstitution and introduce a wide variety of exogenous strains of unknown function to the host, and the long-term effects of this extreme change are unknown (S. S. Li et al., 2016; Smillie et al., 2018). A recent study using mouse models showed that fecal transplants into healthy mice were able to later lead to the development of IBD (Fischer, Bittar, Papa, Kassam, & Smith, 2017). In order to address the risk of the extreme microbial reconstitution that fecal transplants induce, there is a need for understanding how to alter the microbiome community structure of a patient using controlled treatment factors (Gilbert et al., 2016). Uncovering the nature of how to precisely control a patient's microbiome requires accurate modeling of the dynamics of the microbiome community under varying conditions. Such models will empower the identification of microbiome-targeted therapies

as clinicians and researchers will be able to identify which factors and stimuli can be used in order to drive a patient's microbiome to a healthier composition.

The microbiome composition is a major factor in defining future microbiome composition, especially in the gut microbiome (Shenhav et al., 2019). However, the modeling of longitudinal microbiome data faces many challenges due to the inherent noise of microbiome data, which is further confounded by missing data and the varying speed of microbial dynamics between subjects. Current methods use smoothing splines for the interpolation of missing data (Lugo-Martinez et al., 2019; Shafiei et al., 2015). However, splines only consider a single microbe's abundance for interpolations. This presents a challenge when working in the compositional profile, which is often the case when performing microbiome analyses. By interpolating the compositional abundance of individual microbes, the constraints of the compositional landscape are ignored, and additional steps are required to realign the microbiome community abundance values at each interpolated point.

Two common methods for modeling the dynamics of the microbiome community are Boolean and Bayesian networks (Äijö et al., 2018; Claussen et al., 2017; Michael J McGeachie et al., 2016; Ruiz-Perez et al.; Shafiei et al., 2015; Steinway et al., 2015). Both methods represent microbial abundance features as nodes in a network. One set of nodes is used to represent the current microbial abundance values, and the second set of nodes is used to represent the microbial abundance values of the next time step, with directed edges connecting nodes from the first set to nodes in the second set. Boolean networks represent nodes as binary states of either "on" or "off" which are updated over time using Boolean functions. By discretizing microbial abundance values, the network ignores gradual changes in abundance and focuses more on the logical organization of the entire system. Although being the simpler of the two network approaches, Boolean networks have shown to be successful in modeling microbial dynamics of small systems (Claussen et al., 2017). Dynamic Bayesian networks, on the other hand, use each node to represent a random variable and each edge represents the conditional probability of the target random variable. Recently, an implementation of a Dynamic Bayesian network that could also integrate discrete clinical data was put

forward (Lugo-Martinez et al., 2019; Ruiz-Perez et al.). However, these methods are constrained by the number of features as well as node connections in the network, and as such, stringent filtering or hand-selecting features is often required. Additionally, with these limitations, integrating large amounts of patient data is not feasible.

Another approach to modeling the dynamics of a microbiome community is using generalized Lotka-Volterra (gLV) equations (Joseph et al., 2020; Kuntal et al., 2019; Stein et al., 2013). Unlike the original Lotka-Volterra equations, which only consider predator-prey interactions, gLV equations allow for all possible combinations of interspecies interactions, such as commensalism and competition (Bunin, 2017; Wangersky, 1978). This approach models the microbiome community as a set of ordinary differential equations where each equation represents the change of a single microbe's abundance considering an intrinsic growth rate and a collection of pairwise interactions with all other microbes in the community. However, these methods share similar constraints to their network-based counterparts in that they are heavily constrained by the number of samples and features.

Lastly, a few studies have turned to neural network approaches for modeling the effects of microbe-microbe interactions and environmental factors and the microbiome community structure dynamics (García-Jiménez, Muñoz, Cabello, Medina, & Wilkinson, 2021; Larsen, Field, & Gilbert, 2012). In a study by Larsen et al., a DBN model is used to first generate a directed acyclic graph. The nodes and edges from the DBN model are then used to construct an artificial neural network (ANN) model to capture the complex nature of the microbe-microbe for modeling the longitudinal dynamics of the microbiome community. Although the model showed an improvement in modeling longitudinal microbiome data, by relying on the DBN to identify important nodes and edges, the method shares the same limitations of DBN models, that is that the method is constrained by sample and input feature size. Another study by García-Jiménez et al. use an autoencoder (AE) network in order to construct a latent representation of the microbiome community. The authors then create a second encoder network in order to construct a latent representation of environmental factors and align the microbiome and latent environmental spaces. In doing so, the

environmental factors alone can be used to predict the microbiome community. Although the method shows success in predicting the latent microbiome structure of soil samples in different weather conditions, it only predicts the current microbiome community rather than predicting a future microbiome community. As such, it is not suited for modeling the dynamics of the microbiome community and capturing microbe-microbe interactions as well as the effects of environmental factors.

5.1.1 Problem Definition

In this work, we develop a novel two-step framework using DNN models to address the challenges and limitations described above:

1. The first step will employ the use of an AE model in order to encode the microbiome community into a reduced latent space. In doing so, we will reduce the inherent noise found in microbiome data and provide a platform for interpolating the entire microbiome community at once such that the compositional constraints are maintained.

2. The second step will model the dynamics of the microbiome community using the reduced latent representation. By using this reduced feature space, the overall number of parameters for the DNN model is reduced and the model focuses on shifts in the intrinsic latent structure of the microbiome community. In addition, our DNN model is scalable with respect to input space, allowing for large amounts of external and host-related factors to be seamlessly integrated.

5.1.2 Significance

With increasing amounts of longitudinal microbiome data being generated in conjunction with additional types of omics data, our two-step framework utilizing DNN models will not only be scalable to sample and feature size, but also provide accurate modeling of the dynamics of microbiome communities

and identify how external factors impact these dynamics. In doing so, we will empower researchers to generate novel hypotheses for identifying potential treatment routes to improve patient health through microbiome targeted therapeutics.

5.2 Using Autoencoders for Predicting Latent Microbiome Community Shifts

5.2.1 Preliminary Framework

The proposed framework for modeling microbiome community dynamics is described in two parts. In the first part, we will train an autoencoder (AE) network to capture the latent space of the observed microbiome data. In the second part, we will train a DNN model that can integrate dietary information with the latent microbiome community structure in order to predict the structure in the next time point.

Autoencoder Framework

In the first step of our framework, we propose the use of an AE to compress the microbial community structure into a latent space. An AE is an unsupervised learning model trained to reconstruct the original input after first encoding it into a reduced latent space (Baldi, 2012). Specifically, given a data set of n samples $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n\}$ in $\mathbb{R}^p$ (p is the number of input features), and a neural network architecture, it learns to encode the input to a smaller latent space, $\boldsymbol{z} \in \mathbb{R}^d$ ($d < p$), and then from $\boldsymbol{z}$ reconstruct the output $\hat{X} = \{\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \dots, \hat{\boldsymbol{x}}_n\}$ to be as close as possible to the original input. The AE model uses the ReLU activation function for hidden layers and the sigmoid activation function on the latent layer. Since we will be using relative abundance values, the softmax activation is applied to the output layer and the network is trained using the Kullback-Leibler divergence loss shown in **Equation 5.1**.

$$C = \frac{1}{n}\sum_{i} KLD\ (\boldsymbol{x}_i|\hat{\boldsymbol{x}}_i) + \lambda_1\left|\left|\boldsymbol{W}_{AE}\right|\right|^2 \quad (5.1)$$

Here n is the number of samples, the first term is to measure the reconstruction loss, and the second term is an L_2 regularization term to penalize the weights learned in the AE (W_{AE}) scaled by a parameter λ_1. To further stabilize the model, we include a denoising by adding Gaussian noise following $\sim N\ (0, 0.01)$. An overview of the AE model is shown in **Figure 38**.

Integrative Deep Neural Network Model of Dynamics

The preliminary model uses a single fully connected layer using the ReLU activation function and an output layer using the sigmoid activation function. By using the decoder trained from the AE model, we decode the predicted latent space $\hat{\boldsymbol{z}}^{t+1}$ into the predicted $\hat{\boldsymbol{x}}^{t+1}$. This allows us to use the Kullback-Leibler divergence (KLD) to compare the predicted microbiome community to the observed microbiome community, $\boldsymbol{x}^{t+1}$. The dynamic prediction model was trained using the cost function shown in Equation 5.2.

$$C = \frac{1}{m}\sum_{s}\sum_{t} KLD\ (\boldsymbol{x}_s{}^{t+1}|\hat{\boldsymbol{x}}_s^{t+1}) + \lambda_2\left|\left|\boldsymbol{W}_{DYN}\right|\right|^2 \quad (5.2)$$

Here, m is the number of samples used in the training process such that s represents a unique subject and t represents a time point. The first term is the Kullback-Leibler divergence between the predicted microbiome community and interpolated microbiome community at that time point, and the second term is the L_2 regularization on the weights learned in the dynamic prediction model ($\boldsymbol{W}_{DYN}$) scaled by a parameter λ_2. Further regularization was performed using dropout to remove a certain proportion of nodes from the fully connected layer. Prior to training, the learned weights of the AE model are locked as the loss gradient passes through the decoder part. An overview of the dynamic prediction model is shown in **Figure 39.**

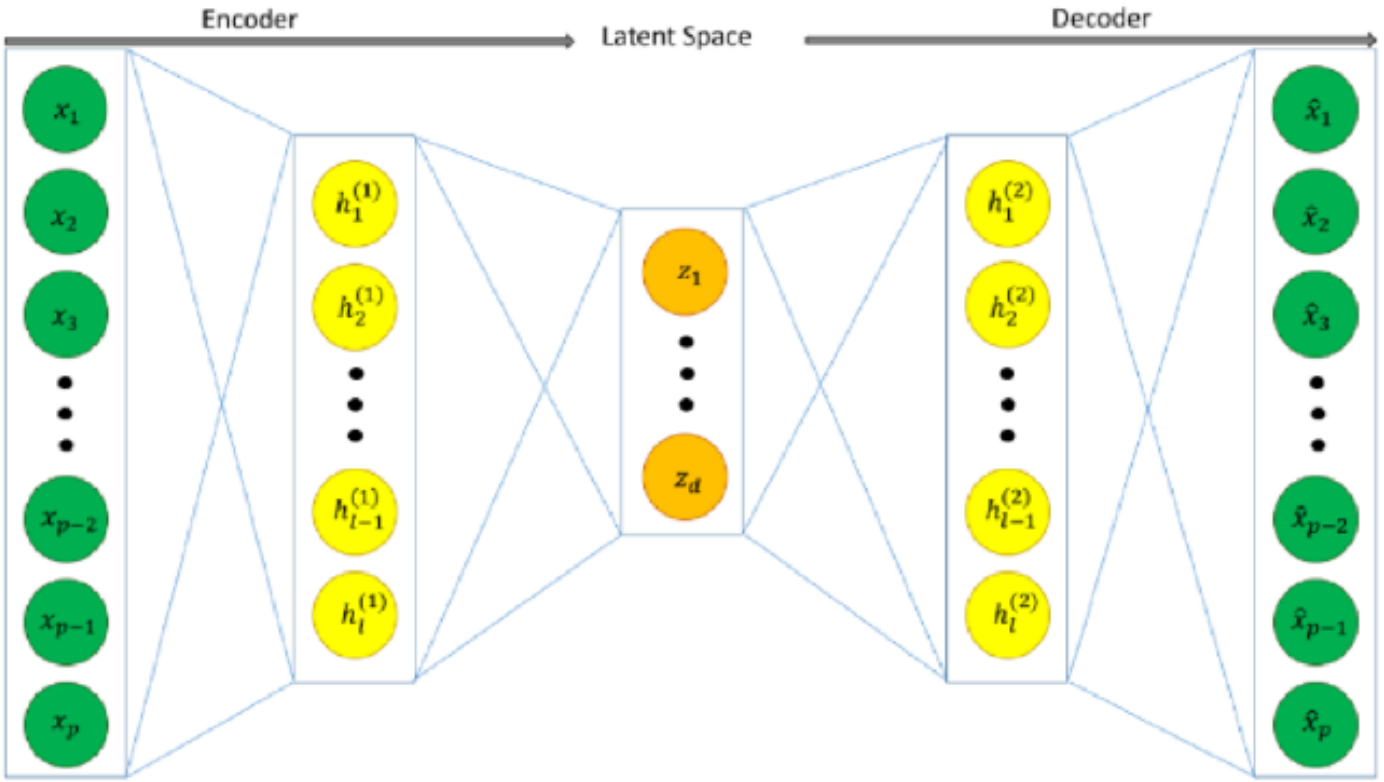


Figure 38. Autoencoder model. The autoencoder compresses the original input ***x*** into the latent vector ***z***. The network is trained by minimizing the distance between the reconstructed output, $\hat{\boldsymbol{x}}$, with the input ***x.***

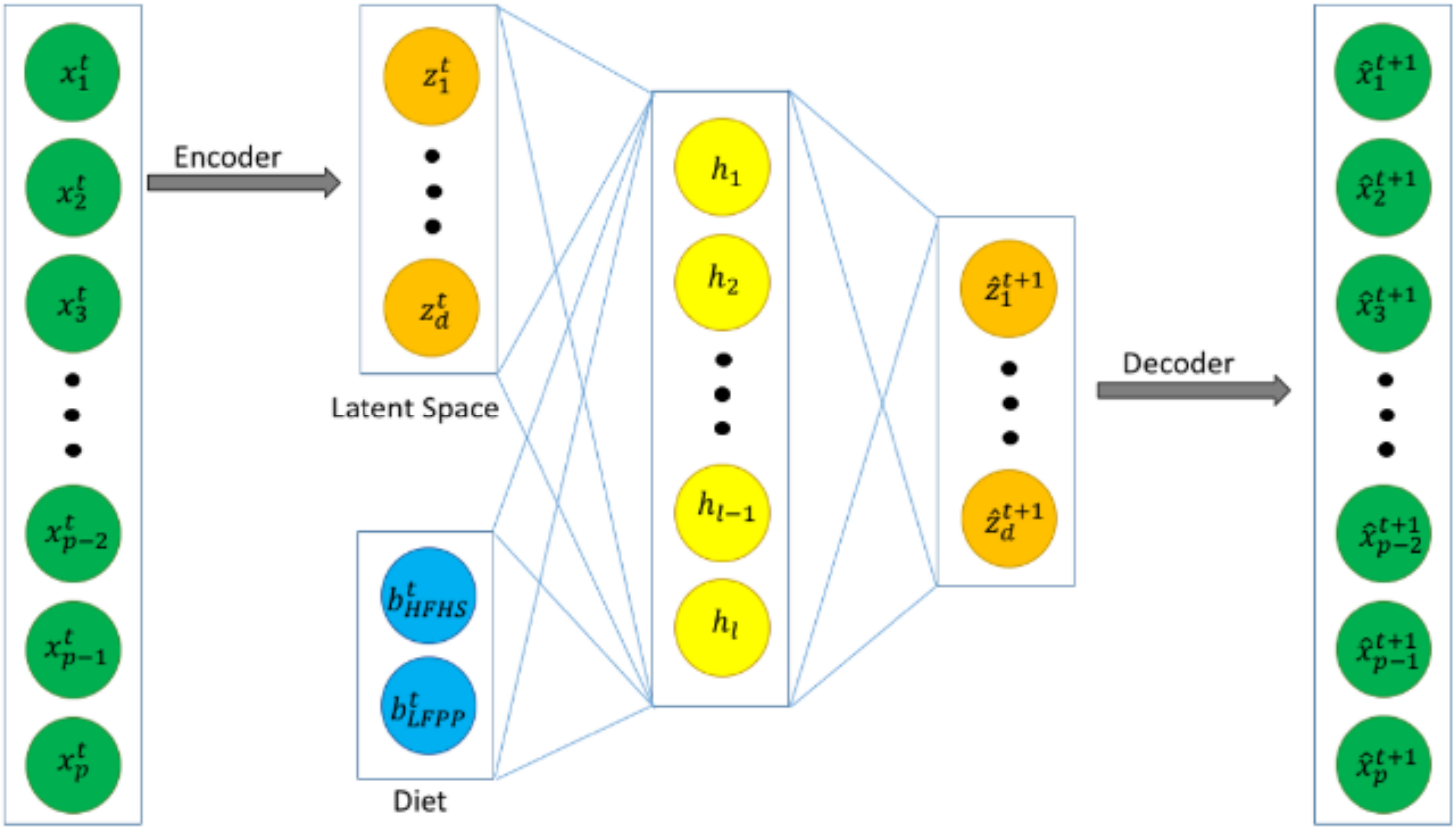


Figure 39. Preliminary integrative DNN network for prediction of microbiome dynamics. The DNN model integrates the microbial community ***x*** at time *t* with dietary factors to predict the microbial community at time *t+1*.

5.2.2 Preliminary Results

Datasets Used in Evaluation

For the pilot study of our model we chose three sets of experimental data that measure the gut microbiome in mice under different dietary conditions (Carmody et al., 2015). In the first dataset, the microbiomes of 60 outbred mice were examined in which consecutive dietary shifts between high-fat, high-sugar (HFHS) diet and low-fat, high-plant-polysaccharide (LFPP) diet had occurred. Mice were fed the LFPP diet until the age of 7 weeks, at which point they were then fed a HFHS diet for a week, followed by a LFPP diet for a week, and then finally a HFHS diet for 2 months. Fecal samples were taken weekly starting at week 4, with an additional 4 samples taken between weeks 7 and 8 (the first shift from a LFPP diet to a HFHS diet. The dataset represents a total of 977 data points across 18 time points in 98 days.

The second dataset is composed of 35 C57BL/6J adult male mice where each mouse is fed the same diet for a week. There are seven available diets that represent different percentages of HFHS (0, 1, 10, 25, 50, 75, 100). The remaining percentage of the diet consisted of the LFPP diet. Fecal samples were collected after one week, resulting in a total of 67 data points.

The third dataset included 15 C57BL/6J male mice split into four groups over the course of 38 days. All four groups were fed LFPP diet for the first four days and HFHS for the final seven days of the experiment. Two control groups were fed only LFPP or HFHS for the duration of the experiment. The third and fourth groups oscillated between HFHS diet and LFPP diets, switching every three days.

All samples were analyzed with 16S rRNA gene sequencing data obtained from MG-RAST using the 'matR' package in R and processed at the genus level. A summary of the datasets is shown in **Table XV**.

TABLE XV. SUMMARY OF DATASETS USED IN DYNAMIC MODEL EVALUATION

Dataset	Mice	Time Points	Samples	OTUs
Outbred	60	18	977	399
Gradient	35	2	67	183
Oscillator	15	38	536	239

Hyper-parameter Tuning

We performed hyper-parameter tuning in two steps for the autoencoder using the outbred dataset, training 10 models in a cross-validated fashion for each parameter set. We chose to use this dataset to determine the network architecture since it has the largest number of samples, and we believed the other datasets would share a similar optimal architecture. First, we kept the fully connected and latent layers constant using 128 and 64 nodes, respectively, and tuned the regularization parameter λ_1 over the values {1.0, 0.1, 0.01, 0.001, 0.0001}, finding that the value of 0.0001 was optimal. Next, keeping the tuned λ_1 constant, we changed the size of the fully connected layers and latent layer, and we found that a model with a fully connected layer of 128 nodes and a latent layer of 64 performed best. A validation set consisting of 20% of the training mice was used for early stopping. Inputs were log-normalized before being passed through the network. The mean and standard deviation of reconstruction MAE and KLD values for layer combinations can be found in **Table XVI**.

TABLE XVI. SUMMARY OF AUTOENCODER HYPER-PARAMETER TUNING.

Layers	MAE (± SD)	KLD (± SD)
128 x 64	**5.44×10^{-4} (± 5.60×10^{-5})**	**0.107 (± 0.019)**
128 x 32	5.71×10^{-4} (± 5.89×10^{-5})	0.116 (± 0.019)
128 x 16	6.16×10^{-4} (± 5.95×10^{-5})	0.129 (± 0.024)
64 x 32	6.76×10^{-4} (± 6.75×10^{-5})	0.148 (± 0.026)
64 x 16	7.21×10^{-4} (± 7.02×10^{-5})	0.163 (± 0.026)

To tune the dynamic model, we performed a similar hyper-parameter tuning approach in two steps. For the first step, we kept the fully connected layer fixed at a size of 128 and tuned the regularization parameter λ_2 over the values {1.0, 0.1, 0.01, 0.001, and 0.0001}, finding 0.0001 to be optimal again. Next, keeping λ_2 fixed, we evaluated different layer sizes and found 256 nodes to perform best. The mean and standard deviation of reconstruction MAE and KLD values of dynamic predictions for layer combinations can be found in **Table XVII**.

TABLE XVII. SUMMARY OF DYNAMIC MODEL HYPER-PARAMETER TUNING.

Layer	MAE (± SD)	KLD (± SD)
512	5.51×10^{-3} (± 4.17×10^{-3})	0.126 (± 0.166)
256	**5.50×10^{-3} (± 4.15×10^{-3})**	**0.124 (± 0.158)**
128	5.51×10^{-3} (± 4.18×10^{-3})	0.127 (± 0.177)
64	5.52×10^{-3} (± 4.21×10^{-3})	0.129 (± 0.168)
32	5.53×10^{-3} (± 4.29×10^{-3})	0.137 (± 0.170)

Evaluation of Dynamic Modeling

Each dataset was evaluated using 10-fold cross-validation. We benchmarked our model against three other models: an ANN using a latent space derived from Principal Component Analysis (PCA), an ANN using the smoothed OTU values, and DBN Model. The PCA latent space was generated by taking the number of principal components required to reach 90% explained cumulative variance. The same network parameters of the ANN used in the models using the AE latent space were used for the other ANN models. The DBN was built as a hybrid discrete/continuous DBN using CGBayesNet (Michael J. McGeachie, Chang, & Weiss, 2014). In order to train the DBN models to have a stable solution, we reduced the number of OTUs by filtering out OTUs which were not present in at least 10% of the samples. This resulted in 63, 84, and 61 OTUs in the outbred, gradient, and oscillator datasets, respectively. In order to perform fair evaluations, metrics were computed using only this subset of OTUs for all models. The results are shown in **Figure 40**.

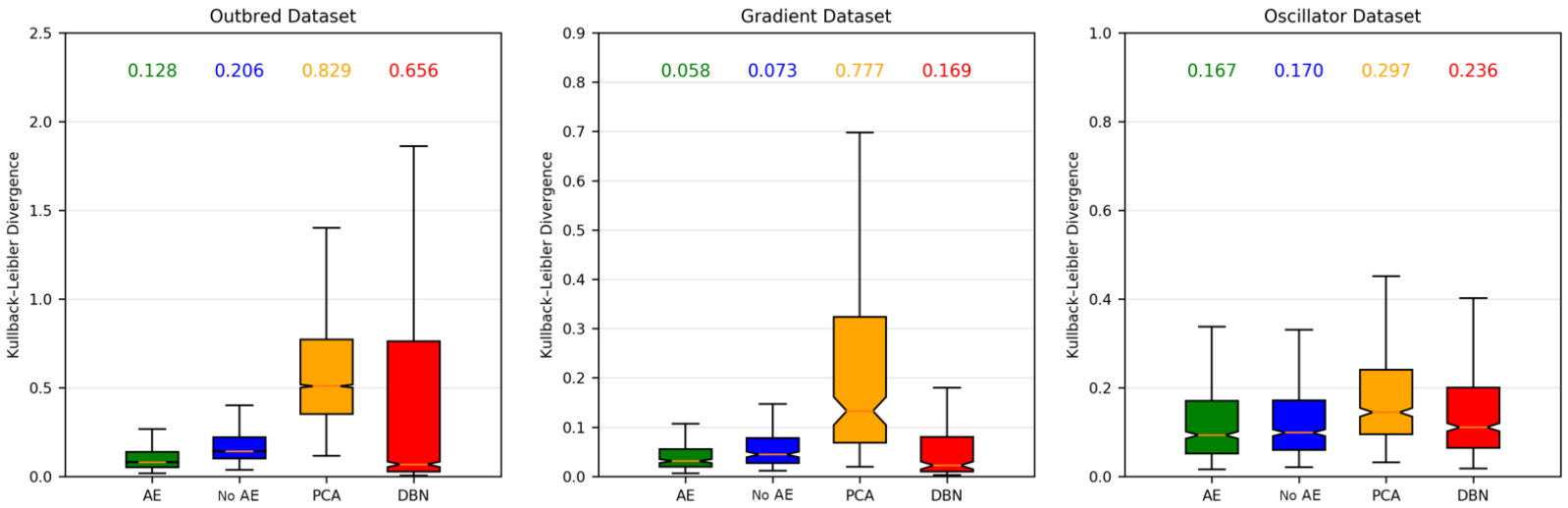


Figure 40. Evaluation of microbiome dynamic modeling. We compare our framework with the same DNN approach using original and PCA features as well as a DBN able to integrate discrete clinical information. All models are evaluated using 10-fold cross-validation. The mean KLD is shown above each boxplot.

5.3 DiRLaM: Diversity-Regularized Autoencoder for Modeling Longitudinal Microbiome Data

Upon observing the predictive power of our preliminary model, we expanded upon it, resulting in a larger framework which we have named "DiRLaM". Here we expand on the preliminary model in two major ways:

1. The AE model is additionally used for interpolation across the microbiome community structures.

2. Diversity metrics are used to regularize the AE network rather than L_2 regularization.

5.3.1 DiRLaM Framework

Autoencoder Framework

The AE model used in DiRLaM follows a similar structure as the one outlined in Section 5.2.1 with a few changes. To improve the overall stability of the model, the ReLU activation function was replaced with the Leaky ReLU activation function with $\alpha = 0.1$.

$$Leaky\ ReLU(x) = \begin{cases} x & x > 0 \\ \alpha x & x \leq 0 \end{cases} \tag{5.3}$$

Additionally, batch normalization was applied after the hidden layers. Batch normalization learns a function over the training period to normalize the values of each node within a hidden layer across samples to have a mean of 0 and a variance of 1. This speeds up the training time by reducing the number of epochs required for convergence while also providing additional regularization to stabilize the model (Ioffe & Szegedy, 2015).

Interpolating Across the Latent Space

In addition to feature reduction, AE models have widely been used as generative models by performing interpolation within the latent space (Berthelot, Raffel, Roy, & Goodfellow, 2018; Cristovao, Nakada, Tanimura, & Asoh, 2020; Makhzani, Shlens, Jaitly, Goodfellow, & Frey, 2015). This is performed by calculating the linear combination of two points within the latent space. More specifically, given $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ as two different points in the latent space, using a mixing coefficient $0 < \alpha < 1$, an interpolated point within the latent space $\boldsymbol{z}_\alpha$ is calculated as,

$$\boldsymbol{z}_\alpha = \alpha * \boldsymbol{z}_1 + (1 - \alpha) * \boldsymbol{z}_2 \tag{5.4}$$

$$\hat{\boldsymbol{x}}_\alpha = Decoder\ (\boldsymbol{z}_\alpha) \tag{5.5}$$

An overview of DiRLaM's AE model with interpolation is shown in **Figure 41.**

To enforce that the latent space remains smooth, adversarial regularization is often used through the use of a critic model (Makhzani et al., 2015). However, since DiRLaM is designed for microbiome data, we instead decide to regularize the AE model using diversity metrics, specifically alpha and beta diversities. By enforcing smooth transitions of alpha and beta diversities, the AE becomes more robust and does not overfit the noise often found within microbiome data. For regularizing alpha diversity, we define a regularization term as the difference between the alpha diversities of the decoded interpolated point and the sum of the alpha diversities of the two observed points from which the interpolated point was generated from, weighted by the mixing coefficient α (**Equation 5.4**).

For calculating alpha diversity, we use Shannon's Entropy, denoted as *H*(.), and we regularized alpha diversity according to **Equation 5.6** and **Equation 5.7.**

$$Reg_{Alpha}\ (\boldsymbol{x}_1, \boldsymbol{x}_2, \alpha) = H(\hat{\boldsymbol{x}}_\alpha) - [\ \alpha * H\ (\boldsymbol{x}_1) + (1 - \alpha) * H\ (\boldsymbol{x}_2)] \tag{5.6}$$

$$H(\boldsymbol{x}) = -\sum_i x^{(i)} \log\left(x^{(i)}\right) \tag{5.7}$$

Here, $x^{(i)}$ denotes the i^{th} microbial feature of a sample. For regularizing beta diversity, we define a regularization term as the difference between the beta diversity of the two observed points from which the interpolated point was generated from, and the beta diversities between the decoded interpolated point with the two original observed points respectively, weighted by the mixing coefficient α (**Equation 5.6**). For calculating beta diversity, we use the Bray-Curtis dissimilarity, denoted as *BC*(.,.), and we regularized beta diversity according to **Equation 5.8** and **Equation 5.9.**

$$Reg_{Beta}\ (\boldsymbol{x}_1, \boldsymbol{x}_2, \alpha) = BC(\boldsymbol{x}_1, \boldsymbol{x}_2) - [\ BC(\boldsymbol{x}_1, \widehat{\boldsymbol{x}}_\alpha) + BC\ (\boldsymbol{x}_2, \widehat{\boldsymbol{x}}_\alpha)] \tag{5.8}$$

$$BC(\boldsymbol{x}_1, \boldsymbol{x}_2) = 1 - \sum_i \min\ (x_1^{(i)}, x_2^{(i)}) \tag{5.9}$$

To exhaustively combine different points for interpolation during training, we construct training points by randomly sampling two different microbiome inputs and randomly drawing $\alpha \sim U\ (0,1)$. Using these two regularization terms, the loss of a single training point, $(\boldsymbol{x}_1, \boldsymbol{x}_2, \alpha)$, is calculated as,

$$\begin{aligned} C(\boldsymbol{x}_1, \boldsymbol{x}_2, \alpha) = {} & \frac{KLD(\boldsymbol{x}_1|\widehat{\boldsymbol{x}}_1) + KLD(\boldsymbol{x}_2|\widehat{\boldsymbol{x}}_2)}{2} + \\ & \lambda_A \left| Reg_{Alpha}(\boldsymbol{x}_1, \boldsymbol{x}_2, \alpha) \right|^2 + \\ & \lambda_B \left| Reg_{Beta}(\boldsymbol{x}_1, \boldsymbol{x}_2, \alpha) \right|^2 \end{aligned} \tag{5.10}$$

Here λ_A and λ_B are scalers that control the strength of the alpha and beta diversity regularization terms respectively.

Integrative Deep Neural Network Model of Dynamics

The dynamic prediction model used in DiRLaM follows a similar structure as the one outlined in Section 5.2.1 with a few changes. To improve the overall stability of the model, the ReLU activation function was replaced with the Leaky ReLU activation function with $\alpha = 0.1$ (**Equation 5.3**). Additionally, batch normalization was applied after the hidden layers. In addition, we have expanded the inputs to be able to account for not only dietary information but also patient-specific clinical information and genotypic information. For integration, continuous data is scaled to be between 0 and 1 and discrete data is one-hot encoded. An overview of the integrative DNN framework is shown in **Figure 42.**

Identification of Significant External Factors

We identify significant external factors by evaluating the change in prediction performance of the DNN model when masking each factor with noise. Specifically, once the DNN model is trained, we iteratively replace the input nodes of each factor with noise and recalculate the KLD of dynamic predictions. Noise for factors of continuous data is sampled from $\sim U\,(0,1)$, and noise for factors of discrete data is created by randomly setting one of the factor's nodes to 1 and the rest to 0. For each subject evaluated, we calculate the subject's KLD using the noisy input as the mean KLD between interpolated time points and predicted time points across all time points of that subject. We then calculate the change in KLD for each subject as the ratio of the subject's KLD from the noisy model divided by the original subject KLD when all external factors are used. In this way, a value greater than 1 represents a decrease in performance for that subject when masking the given factor with noise. To evaluate if masking a factor with noise significantly reduced the overall performance, we use a one-tailed Wilcoxon sign-rank test to determine if the change in subject KLD values across all subjects is significantly greater than 1. Any factor with a p-value less than 0.05 is identified to be significant for dynamic prediction.

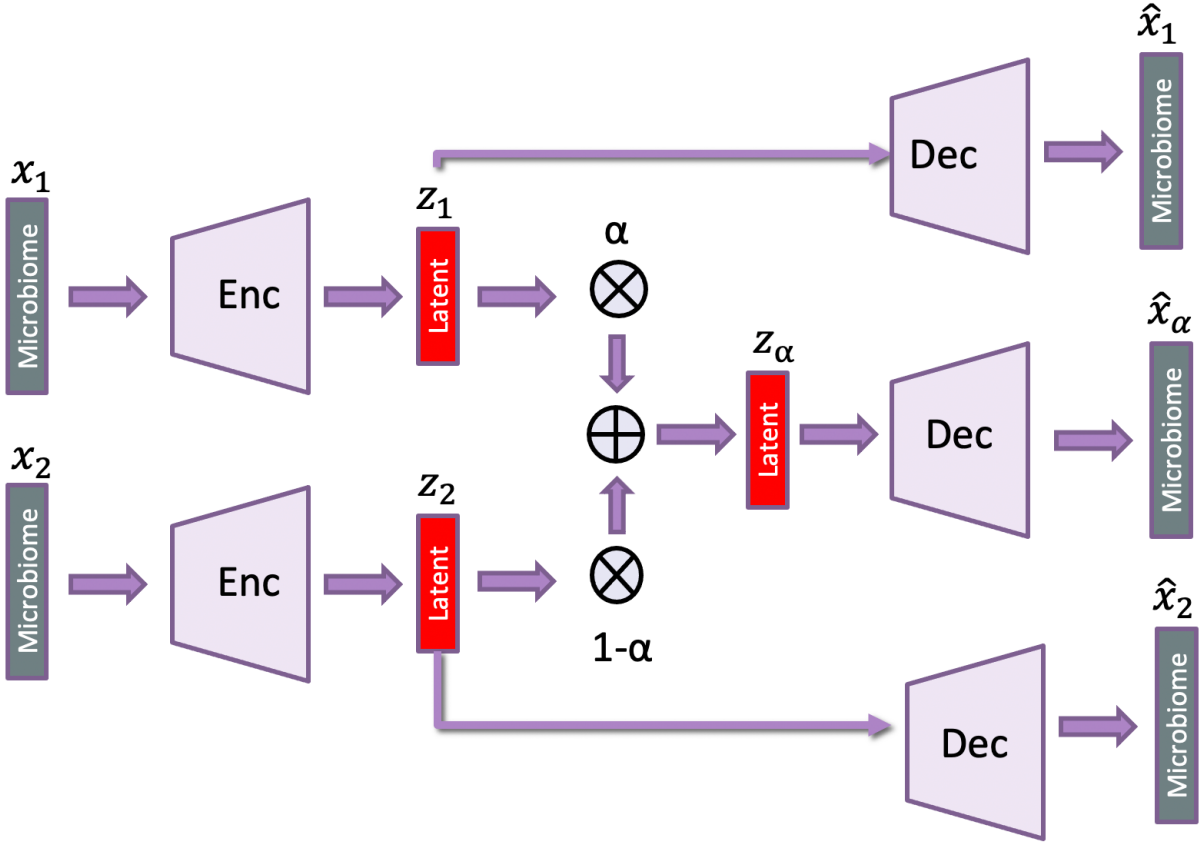


Figure 41. Autoencoder with latent space interpolation for DiRLaM model. Microbiome abundance vectors $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ are encoded using the AE encoder function (Enc) into latent representations $\boldsymbol{z_1}$ and $\boldsymbol{z_2}$. A mixing coefficient, α, is randomly selected between 0 and 1, and a latent interpolation $\boldsymbol{z_\alpha}$ is calculated as $\alpha\ (\boldsymbol{z_1}) + (1-\alpha)(\boldsymbol{z_2})$. All three latent representations are decoded with the AE decoder function (Dec) into the reconstructions $\hat{\boldsymbol{x}}_1$, $\hat{\boldsymbol{x}}_2$, and the interpolated microbiome community $\hat{\boldsymbol{x}}_\alpha$.

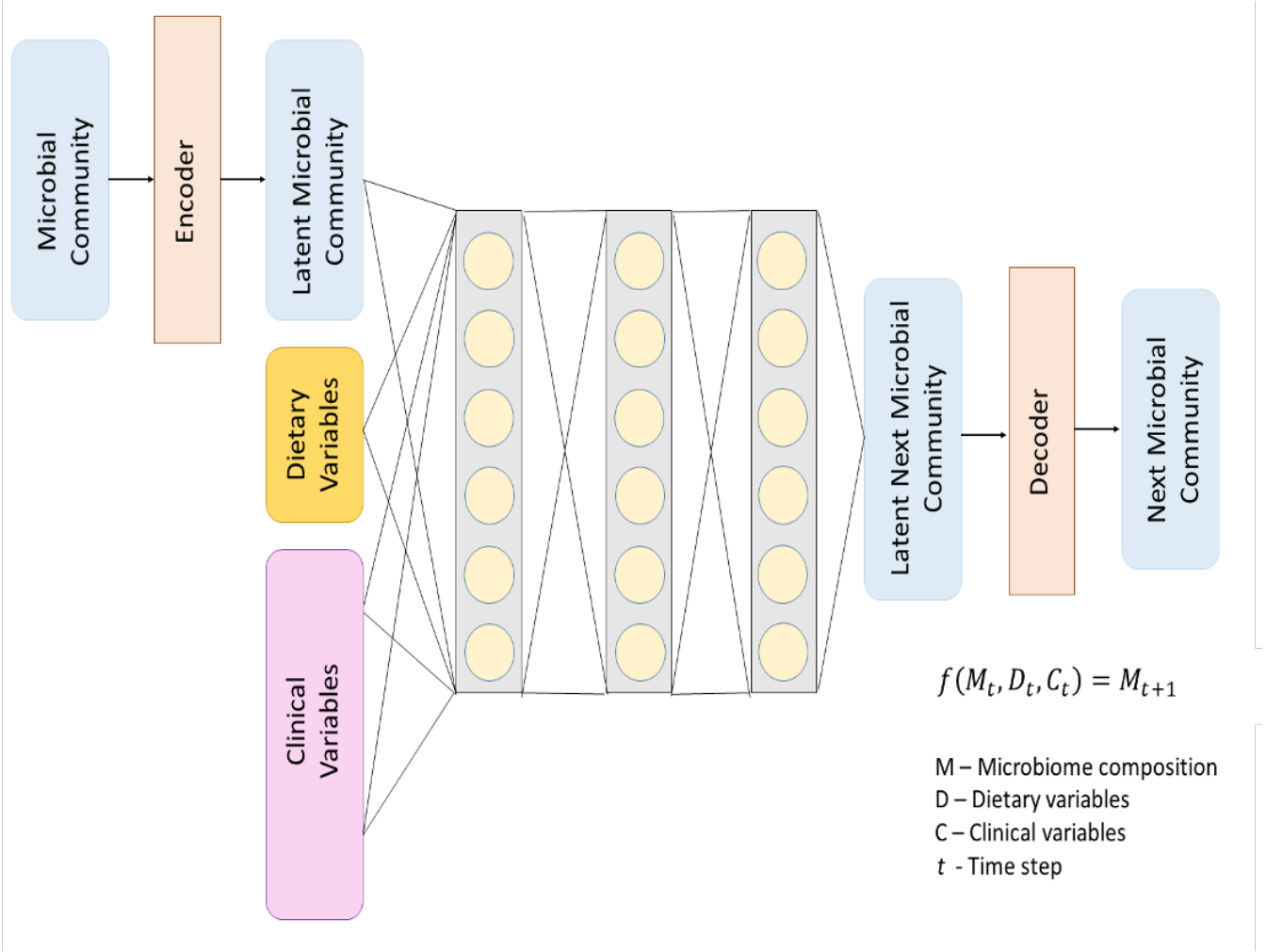


Figure 42. Integrative DNN network for prediction of microbiome dynamics. The DNN model integrates the microbial community ***M*** at time *t* with clinical factors, *C* and dietary factors *D* to predict the microbial community at time *t+1*.

Deriving Microbial Modules from Latent Space

We derive microbial modules using the decoder part of the AE model. Specifically, we first calculate a baseline signal by decoding a zero vector. This represents the baseline reconstructed microbiome community when there is no signal going into the decoder.

$$\boldsymbol{e}_{baseline} = Decoder\ (\vec{\mathbf{0}}) \tag{5.11}$$

Next we calculate the reconstructed microbiome community associated with each latent node by iteratively decoding zero vectors that contain a single one in the respective latent node. This can quickly be done by decoding an identity matrix that is the size of the latent space, $\boldsymbol{I}_d$.

$$\boldsymbol{E} = Decoder\ (\boldsymbol{I}_d) \tag{5.12}$$

Then, to evaluate the effect each latent node has on the microbiome community, we subtract the reconstructed baseline community from each row of $\boldsymbol{E}$. This will give us the change in the microbiome community across each dimension of the latent space.

$$\boldsymbol{R} = \boldsymbol{E} - \vec{\mathbf{1}}\ (\boldsymbol{e}_{baseline})^T \tag{5.13}$$

Here $\vec{\mathbf{1}}$ is a vector of ones used to broadcast the subtraction of $\boldsymbol{e}_{baseline}$ to each row of $\boldsymbol{E}$. To identify microbial modules, we first scale the columns of $\boldsymbol{R}$ *(*the latent effects for each microbe) to have a mean of 0 and a variance of 1, and then we apply hierarchical clustering with complete linkage and Euclidean distance. Doing so will result in groups of microbes that change in similar patterns within the microbiome community latent space. Lastly, we calculate the mean latent effect within a module as the average value of each row across the columns defined by a module's members in $\boldsymbol{R}$, and we denote this as $\bar{\boldsymbol{R}}$.

Deriving Module Interactions

We identify interactions between modules and the effects of external factors using Olden's method (Olden et al., 2004). Olden's method works by cumulatively multiplying the weights of each hidden layer together, as shown in **Equation 5.14.** This results in a matrix where each row represents an input feature, and each column represents an output feature.

$$\boldsymbol{S} = \prod_{l \in L} \boldsymbol{W}_l \tag{5.14}$$

Here l is the current layer in the set of L layers within the trained network, and $\boldsymbol{W}_l$ is the weight matrix connecting layer l with the previous layer.

We then partition $\boldsymbol{S}$ into two submatrices, one that contains the rows corresponding to the latent microbial inputs ($\boldsymbol{S}_{latent}$) and one that contains the rows corresponding to the external factor inputs ($\boldsymbol{S}_{factor}$). Since we know that the predicted latent structure is dependent on the current latent, we mask this effect by setting the trace of $\boldsymbol{S}_{latent}$ to have zero values. Both $\boldsymbol{S}_{latent}$ and $\boldsymbol{S}_{factor}$ are normalized to have a mean of 0 and variance of 1 across all values.

Using $\boldsymbol{S}_{latent}$ and $\overline{\boldsymbol{R}}$, we can calculate the interaction matrix between modules ($\boldsymbol{U}$) and the interaction matrix of external factors to modules ($\boldsymbol{V}$). Each value in these matrices represents a directed effect that the row-corresponding feature has on the column-corresponding feature.

$$\boldsymbol{U} = \overline{\boldsymbol{R}}^T \, (\boldsymbol{S}_{latent}) \overline{\boldsymbol{R}} \tag{5.15}$$

$$\boldsymbol{V} = \, (\boldsymbol{S}_{factor}) \overline{\boldsymbol{R}} \tag{5.16}$$

These matrices represent module-module interactions and effects of external factors, which can then be further visualized using Cytoscape (Shannon et al., 2003).

5.3.2 Experiment and Results

Datasets Used in Evaluation

We used five datasets in the benchmarking and evaluation of "DiRLaM". The first two datasets are synthetic datasets generated using the R package "microbiomeDASim" (Williams, Bravo, Tom, & Paulson, 2019). Using this tool, 50 sets of true microbiome observations containing 100 OTUs were generated over the course of 60 days. We consider the first 30 days and last 30 days to differ in a single external factor with 10 OTUs being differentially abundant between the two timeframes. An example showing the dynamics of the 10 differentially abundant OTUs for a single sample is shown in **Figure 43**.

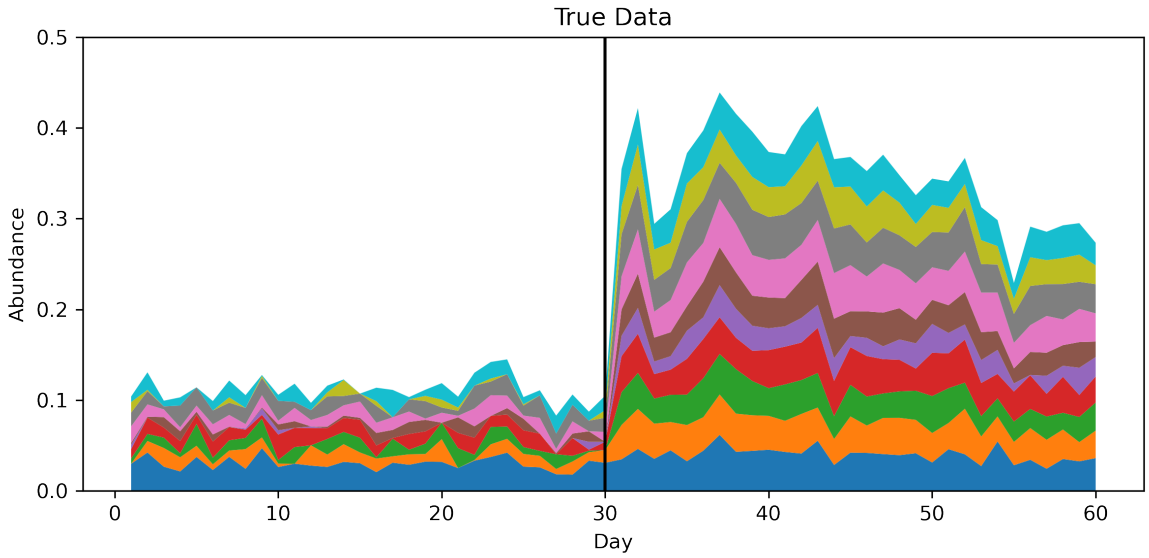


Figure 43. Sample of true observations for synthetic data. Image shows dynamics of the abundance of the 10 synthetic differentially abundant microbes within one sample between the first 30 days and the last 30 days.

Using these true observations, we generate two different datasets using different levels of noise. Noise is added in the form of the dropping of time points as well as through the masking of OTUs. In doing so, the synthetic data will now have missing and unaligned time-points between samples as well as OTU dropout, both of which are commonly seen in real microbiome data. To generate the first dataset, for each set of observations we drop 33% of the time-points in each timeframe as well as mask 33% of the OTUs at each time point. We refer to this dataset as "Synthetic (Low Noise)". To generate the second dataset, for each set of observations, we drop 66% of the time-points in each timeframe as well as mask 50% of the OTUs at each time point. We refer to this dataset as "Synthetic (High Noise)". Examples showing the dynamics of the 10 differentially abundant OTUs for a single sample in "Synthetic (Low Noise)" and "Synthetic (High Noise)" are shown in **Figure 44** and **Figure 45,** respectively.

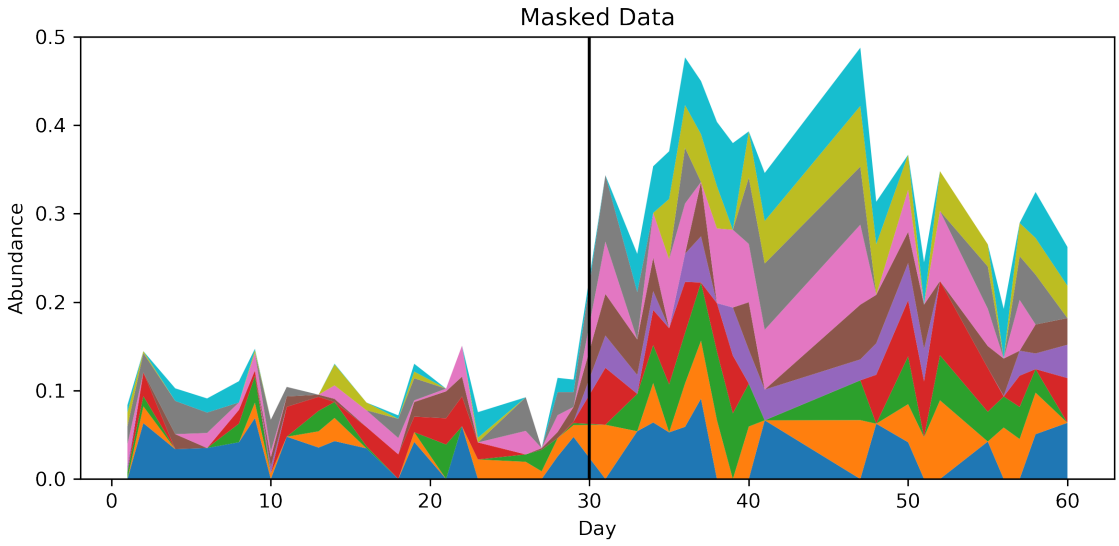


Figure 44. Sample of observations for Synthetic (Low Noise). Image shows dynamics of the abundance of the 10 synthetic differentially abundant microbes within one sample between the first 30 days and the last 30 days. Noise is injected by dropping 33% of time points and setting 33% of microbial abundance values to 0 at each remaining time point.

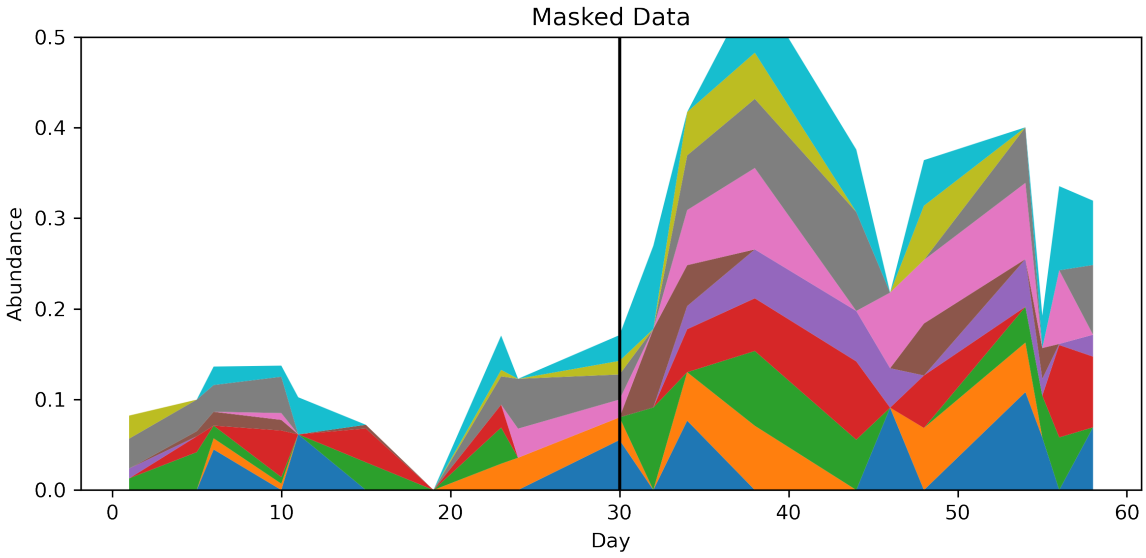


Figure 45. Sample of observations for Synthetic (High Noise). Image shows dynamics of the abundance of the 10 synthetic differentially abundant microbes within one sample between the first 30 days and the last 30 days. Noise is injected by dropping 66% of time points and setting 50% of microbial abundance values to 0 at each remaining time point.

In addition to synthetic data, we use three real-world datasets. The first dataset is the outbred mouse dataset used in the preliminary study was described in Section 5.2.2. The second dataset represents the vaginal microbiota and was collected by Gajer et al. (Gajer et al., 2012). This study contains vaginal microbiome samples of 32 reproductive-age healthy women over a 16-week period. It consists of 937 self-collected vaginal swabs sampled twice per week, resulting in a total of 330 OTUs. The data also contains clinical and demographic attributes such as Nugent score, menstruation period, race, and age. The last dataset from La Rosa et al. contains microbiome samples collected from the guts of newborn infants (La Rosa et al., 2014). The dataset includes 58 pre-term infants in the neonatal intensive care unit (NICU) and was collected during the first 12 weeks of life. Samples were collected sampled every one or two days. In total, there are 922 infant gut microbiome measurements with a total of 29 OTUs. Additionally, the dataset

includes gestational age at birth, post-conceptional age when the sample was obtained, mode of delivery, antibiotic use, type of milk used, and room ID.

For each dataset, OTUs with a maximum relative abundance value less than 0.1% or that were present in less than 5% of the total samples were removed. This did not impact the synthetic datasets, but the mouse, vaginal, and infant gut datasets were reduced to 73, 75, and 12 OTUs, respectively. In addition, one subject (subject 5) was removed from the vaginal dataset due to inconsistencies in the microbiome and metadata matching, and in the infant gut dataset, subjects containing less than 10 time-points were removed, reducing the total time-points to 840.

Hyper-parameter Tuning and Model Training

Hyper-parameters were hand-tuned using the synthetic datasets evaluated on the true observations. We found that a hidden layer size of 32 and a latent size of 8 was sufficient before a noticeable drop in performance. We used the same layer sizes for the mouse and vaginal datasets since they had a similar number of OTU features. We reduced the hidden and latent layer sizes to 16 and 4 for the infant gut dataset due to the greatly reduced number of OTU features compared to the other datasets. Since the diversity-regularized AE model randomly samples time-points, we construct batches of 1,024 training sets and train the AE model for 5,000 iterations. Dynamic prediction models used two hidden layers that were each twice the size of the latent space input. Similar to the preliminary model, the DNN model for dynamic prediction is trained with early stopping, using 20% of the training data as a validation set. AE and DNN models for all experiments were trained using 5-fold cross-validation, where partitions were stratified by subject IDs.

Effect of Diversity Metric Regularization

In order to evaluate the effects of the alpha and beta diversity terms and tune the proper value of this hyper-parameter, we perform a grid search of λ_A and λ_B where each penalty value was from the set {0, 0.5, 1.0, 2.0, 5.0, 10.0}. Using different strengths of λ_A and λ_B, AE models were trained on the "Synthetic (Low Noise)" and "Synthetic (High Noise)" datasets, and the Kullback-Leibler divergence between the reconstructed values and the true synthetic values were calculated. In both datasets, we observed that the alpha diversity regularization was harmful to model learning. However, the beta diversity regularization improved the performance of the AE model. Based on these observations, we removed the alpha diversity regularization term and used $\lambda_B = 2$ for all subsequent experiments and evaluations. Heatmaps showing the trend of Kullback-Leibler divergence values for the "Synthetic (Low Noise)" and "Synthetic (High Noise)" datasets are shown in **Figure 46.**

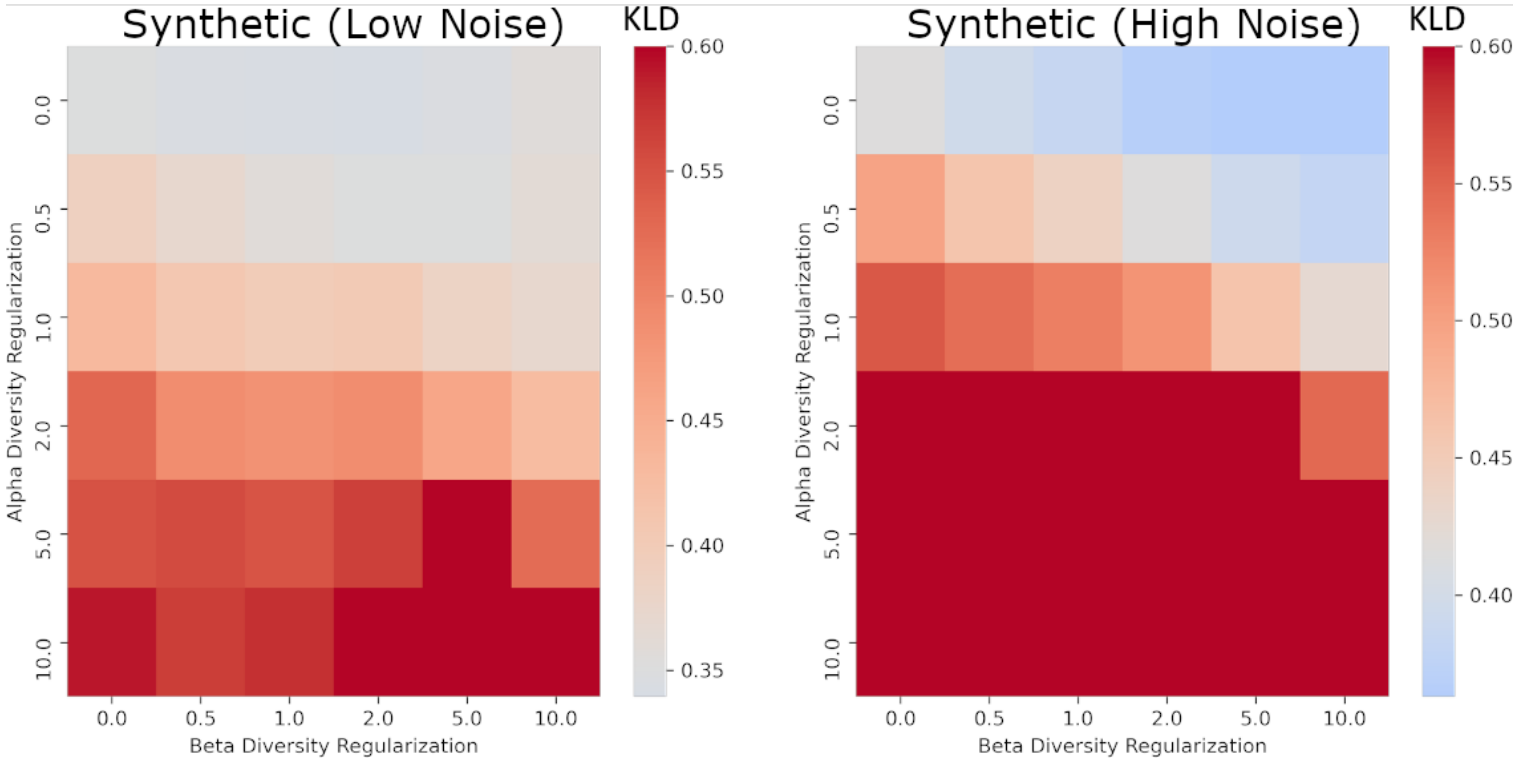


Figure 46. Heatmap of KLD values for Synthetic (High Noise) across varying levels of alpha and beta diversity regularization. Alpha and beta diversity penalties are evaluated at {0, 0.5, 1, 2, 5, 10} in a grid-wise pairing. The average KLD across subjects is calculated for each combination.

Evaluation of Interpolation

We first evaluated the effectiveness of our AE model using diversity regularization. AE models were trained on the "Synthetic (Low Nosie)" and "Synthetic (High Noise)" datasets and time points were evenly interpolated such that there were 20 time points in a single day period. Time points that fell on whole day values were compared to the original true synthetic values, and a KLD value was calculated for each sample as the mean KLD between each of the reconstructed interpolated time-points and true time-points for that sample. The overall evaluation of the interpolation method was then calculated as the mean of the sample KLD values. We compared our method with:

- Interpolation of the latent space using a baseline AE model with no diversity regularization
- B-spline smoothing followed by the training of a baseline AE model on the smoothed values
- B-spline smoothing with no AE
- Interpolation across the latent space identified by PCA
- B-spline smoothing followed by PCA encoding and decoding on the smoothed values

We observed that, when no noise is present (i.e., if trained on the original true synthetic values), a B-spline was optimal since there is no noise and it passes through the true observed points. However, when training on "Synthetic (Low Noise)" and "Synthetic (High Noise)", the B-spline became very unstable as it overfit the noise of the data. As the noise increases, the AE model using diversity regularization performed the best, followed by the two methods using a baseline AE model. Based on this observation, we saw that by constraining the interpolation to smoothly transition in beta diversity between points, the model became more robust to the noise and was able to better recapture the true synthetic values. In addition, we observed that the AE model performed better than PCA for encoding, decoding, and interpolation. A table of mean

KLD values for the synthetic datasets is shown in **Table XVIII** and visualizations of interpolation for a single sample using each method is shown in **Figure 47** and **Figure 48** for the "Synthetic (Low Noise)" and "Synthetic (High Noise)" respectively.

TABLE XVIII. EVALUATIONS OF DIFFERENT SPLINING TECHNIQUES BASED ON THE KLD WITH THE TRUE SYNTHETIC DATA VALUES.

	No Noise (T = 60)	**Low Noise (T = 40)**	**High Noise (T = 20)**
Baseline Autoencoder Interpolation	0.289 (0.063)	0.351 (0.061)*	0.414 (0.075)*
Diversity Regularized Autoencoder Interpolation	0.315 (0.065)	**0.339 (0.061)**	**0.359 (0.064)**
B-Spline and Baseline Autoencoder	0.311 (0.067)	0.349 (0.064)*	0.416 (0.076)*
B-Spline and PCA	0.396 (0.112)	0.642 (0.221)*	0.949 (0.306)*
PCA Interpolation	0.243 (0.062)	0.357 (0.105)*	0.458 (0.265)*
B-Spline on Raw Data	**0.00 (0.00)**	2.092 (0.489)*	2.766 (0.667)*

Asterix (*) represent values that are significantly greater than the KLD evaluation of the Diversity Regularized Autoencoder Interpolation. P-values were determined using a Wilcoxon sign-rank test over KLD values for each sample.

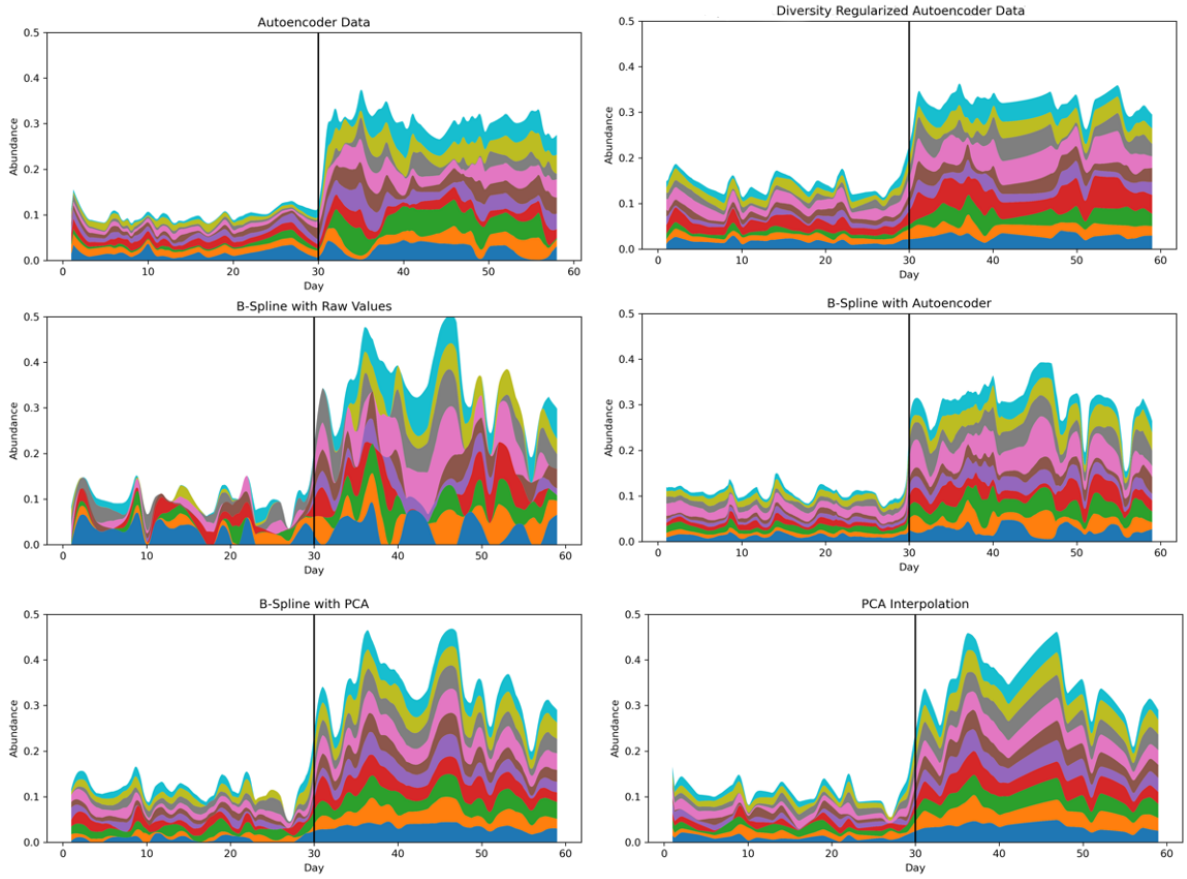


Figure 47. Sample interpolations for Synthetic (Low Noise) using different evaluated methods. Interpolation of the 10 differentially abundant microbes using various interpolation methods: (top left) interpolation within baseline AE latent space, (top right) interpolation within diversity-regularized AE, (middle left) B-spline on raw abundance values, (middle right) baseline AE reconstruction after using B-spline on raw abundance values, (bottom left) PCA reconstruction after using B-spline on raw abundance values, (bottom right) interpolation in the PCA latent space.

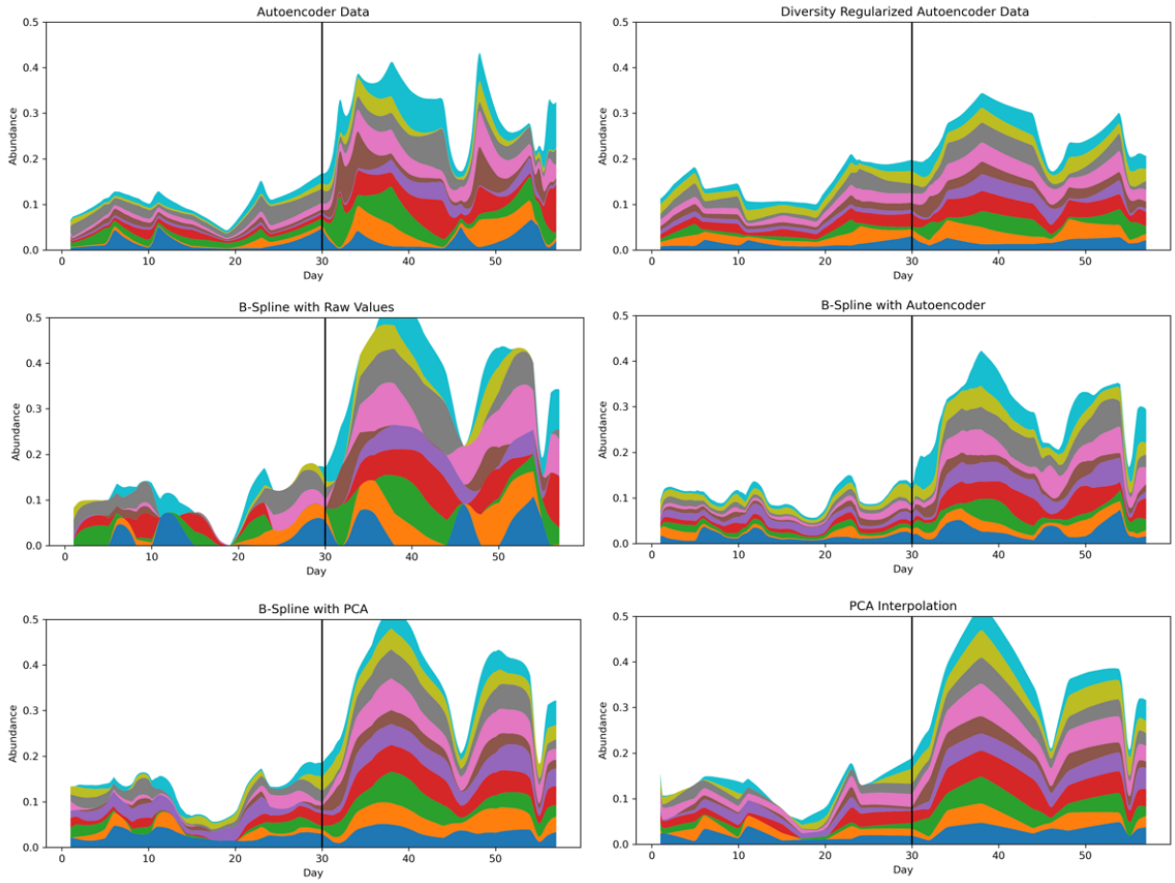


Figure 48. Sample interpolations for Synthetic (High Noise) using different evaluated methods. Interpolation of the 10 differentially abundant microbes using various interpolation methods: (top left) interpolation within baseline AE latent space, (top right) interpolation within diversity-regularized AE, (middle left) B-spline on raw abundance values, (middle right) baseline AE reconstruction after using B-spline on raw abundance values, (bottom left) PCA reconstruction after using B-spline on raw abundance values, (bottom right) interpolation in the PCA latent space.

Evaluation of Dynamic Prediction

Next we evaluated how well the dynamic predictions using the DNN model. We benchmarked the DNN model against a DBN. For each dataset, time points representing microbiome community abundance values were uniformly interpolated such that there were 20 time points within a single day. We then evaluated the dynamic prediction after a single day given the current latent microbiome structure as well as specific external and patient characteristics. DNN models were trained using all pairs of interpolated time points separated by one day in a sliding-window fashion (i.e. {(1.00, 2.00), (1.05, 2.05), (1.10, 2.10), …}. To maintain a feasible running time, we removed the time points that fell between days and DBN models were only trained on pairs of time points where the day was a whole number value (i.e. {(1.00, 2.00), (2.00, 3.00), (3.00, 4.00), …}. In addition, we trained DNN and DBN models using latent and reconstructed values using the same linear interpolation of the PCA latent space. All models were evaluated by first calculating a mean KLD for each sample based on the predicted value and interpolated of each predicted time-point, and then calculating the mean KLD across all samples.

The synthetic and mouse datasets contained only one external factor each with two unique values. To integrate these factors, we performed one-hot encoding and concatenated the values to the latent microbiome inputs of the DNN and DBN models. Figures showing the performance for the synthetic datasets and the mouse dataset are shown in **Figures 49-51**.

For the vaginal dataset, Nugent score and age were min-max scaled to be between 0 and 1. There were four unique discrete values for race, however, due to two of the values being rare, we combined the race values into two binary values representing white and non-white. The menstruation state was also converted into two binary values. The evaluation using the vaginal dataset is shown in **Figure 52.**

For the infant gut dataset, gestational age and postgestational age were min-max scaled to be between 0 and 1. Gender, room, and method of birth each contained two unique values and were one-hot encoded. Milk contained four unique values and was one-hot encoded. Time on antibiotics was already

given as a percentage, so no scaling was performed since it ranged between 0 and 1 already. The evaluation using the infant gut dataset is shown in **Figure 53.**

We observed that when using the latent space learned from the AE models, the DNN model outperforms the DBN model in the synthetic and mouse data and the two methods are comparable in the vaginal and infant gut datasets. Additionally, models using the latent space always outperform models using reconstructed values. When using reconstructed values, the DNN model always outperforms the DBN model, showing the DNN model's ability to handle larger input feature spaces. Lastly, we observed that using interpolation generated from the AE latent space performed better than using interpolation generated from PCA latent space.

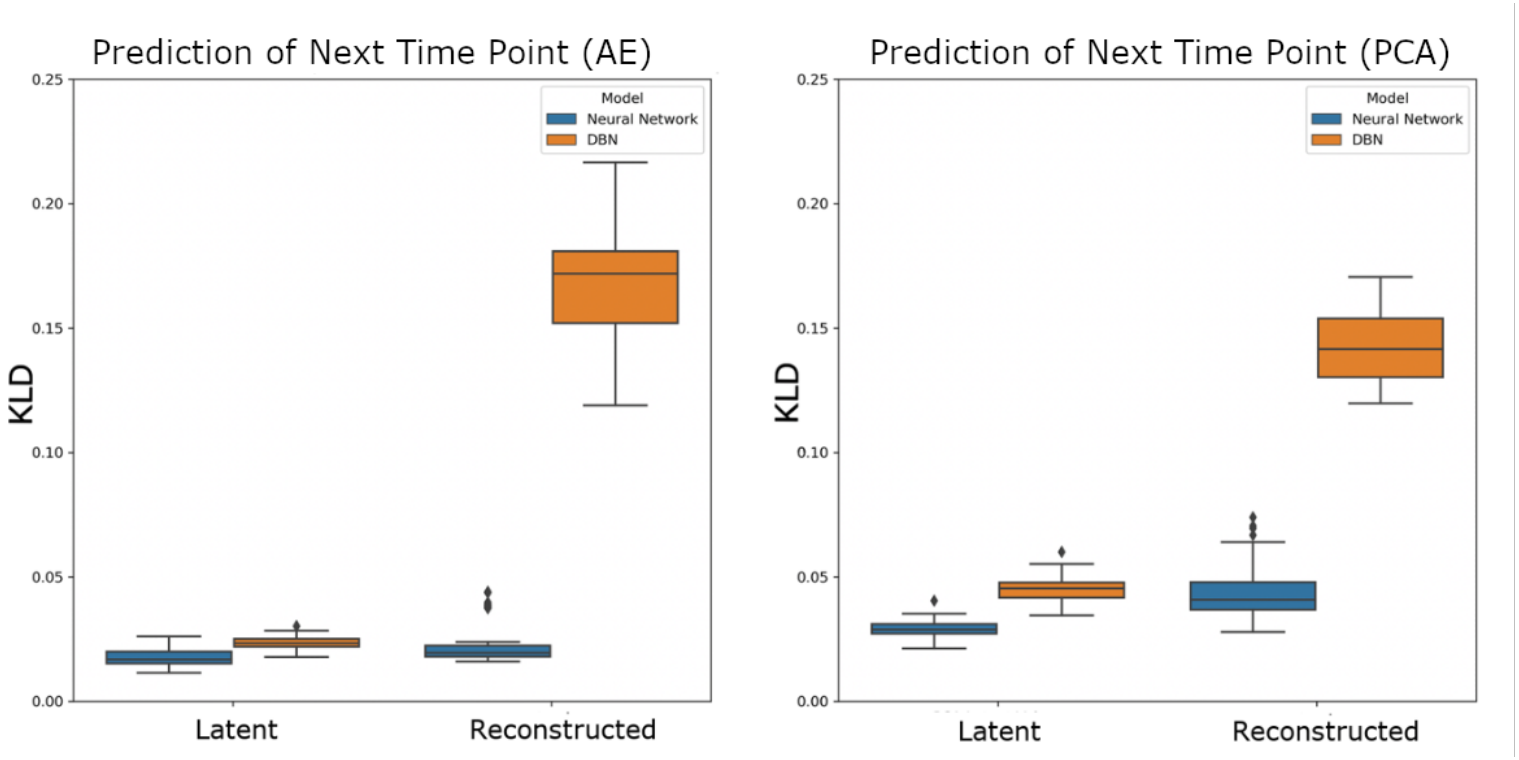


Figure 49. Prediction of next time point using NN and DBN models for Synthetic (Low Noise) dataset. Boxplots on the left represent models trained on latent and reconstructed values from the diversity regularized AE. Boxplots on the right represent models trained on latent and reconstructed values from PCA interpolation.

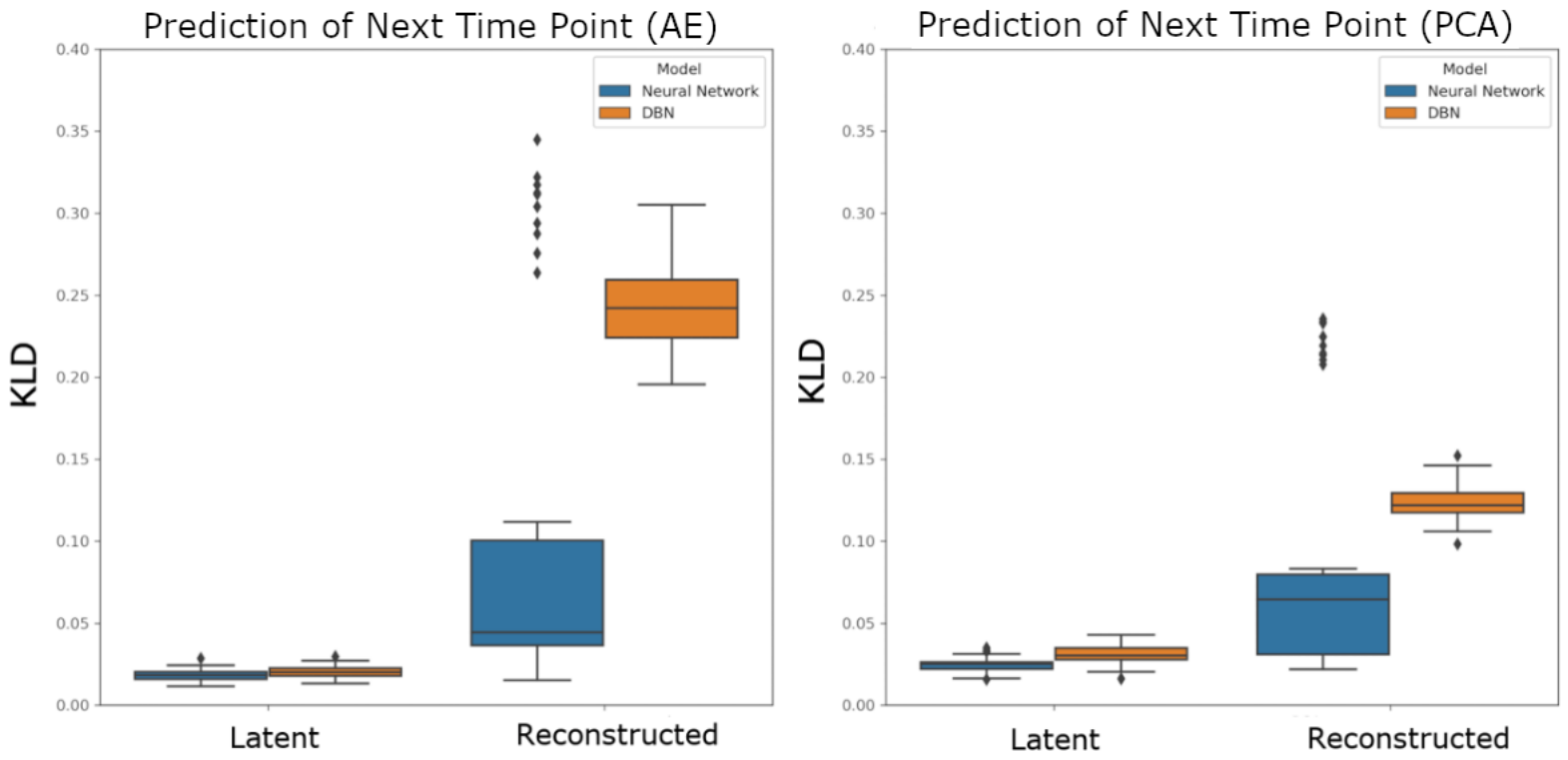


Figure 50. Prediction of next time point using NN and DBN models for Synthetic (High Noise) dataset. Boxplots on the left represent models trained on latent and reconstructed values from the diversity regularized AE. Boxplots on the right represent models trained on latent and reconstructed values from PCA interpolation.

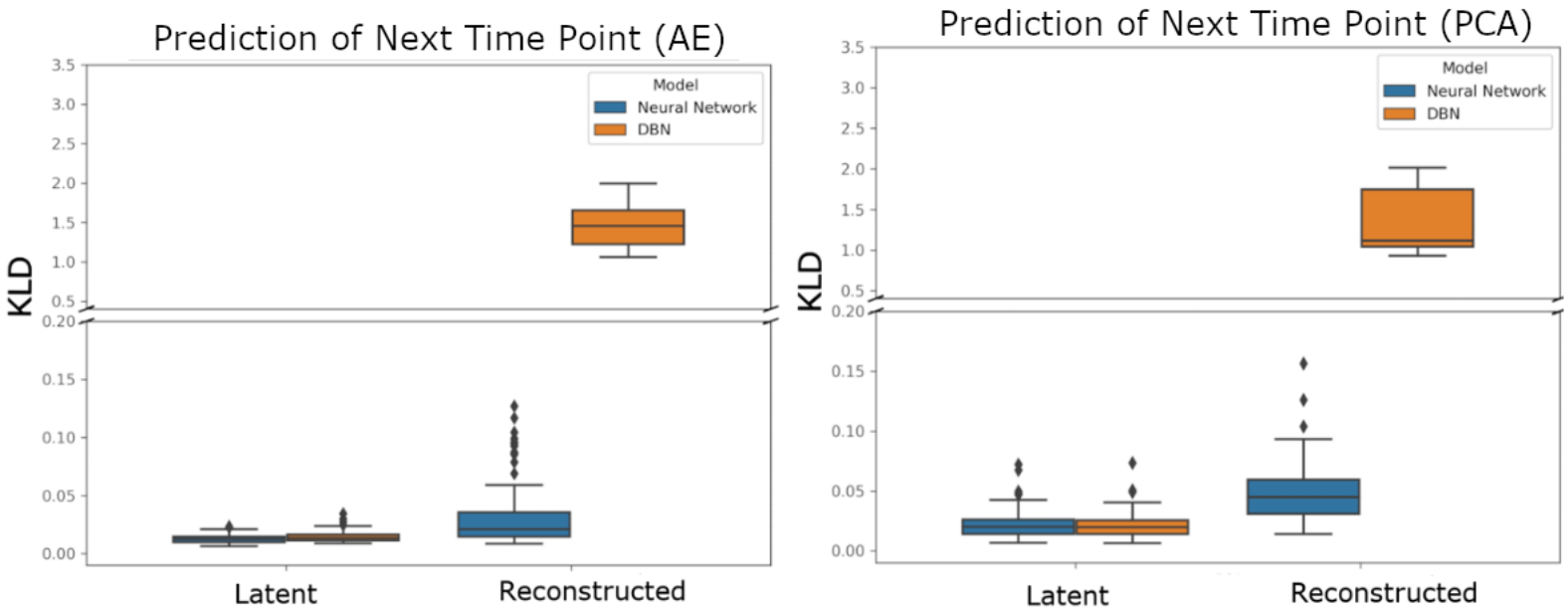


Figure 51. Prediction of next time point using NN and DBN models for mouse dataset. Boxplots on the left represent models trained on latent and reconstructed values from the diversity regularized AE. Boxplots on the right represent models trained on latent and reconstructed values from PCA interpolation.

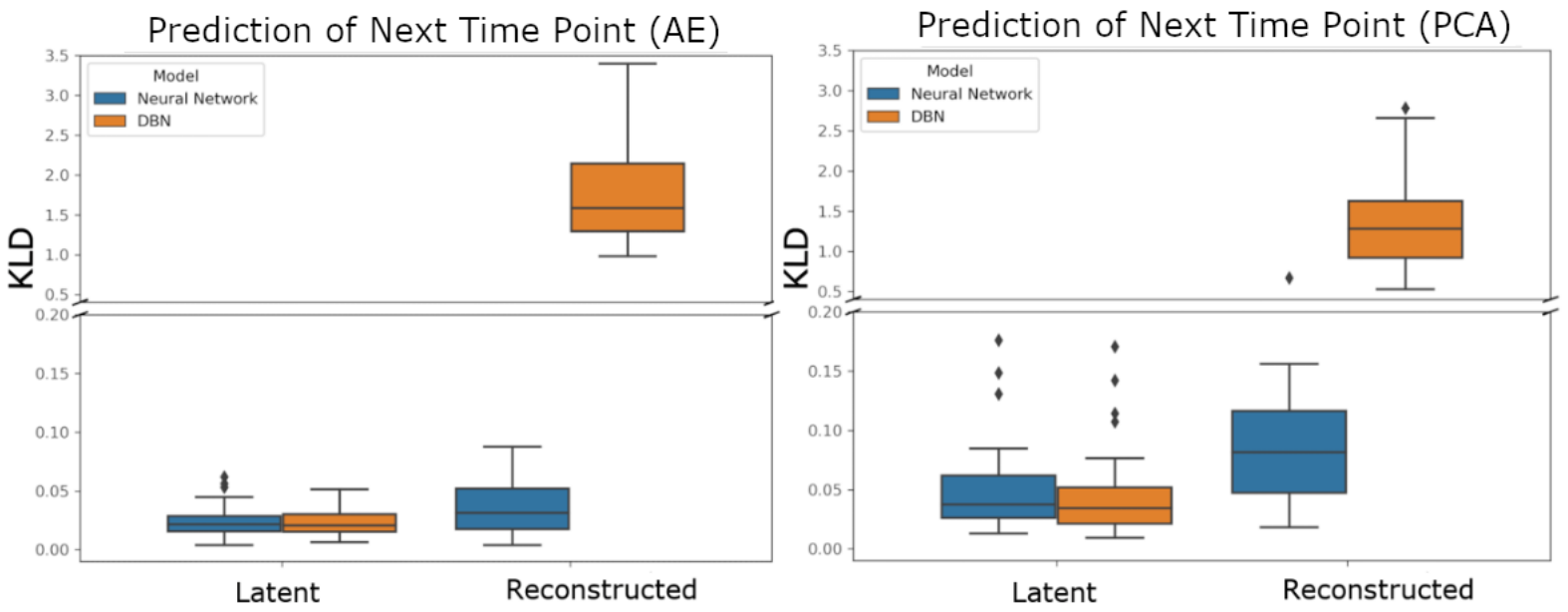


Figure 52. Prediction of next time point using NN and DBN models for vaginal dataset. Boxplots on the left represent models trained on latent and reconstructed values from the diversity regularized AE. Boxplots on the right represent models trained on latent and reconstructed values from PCA interpolation.

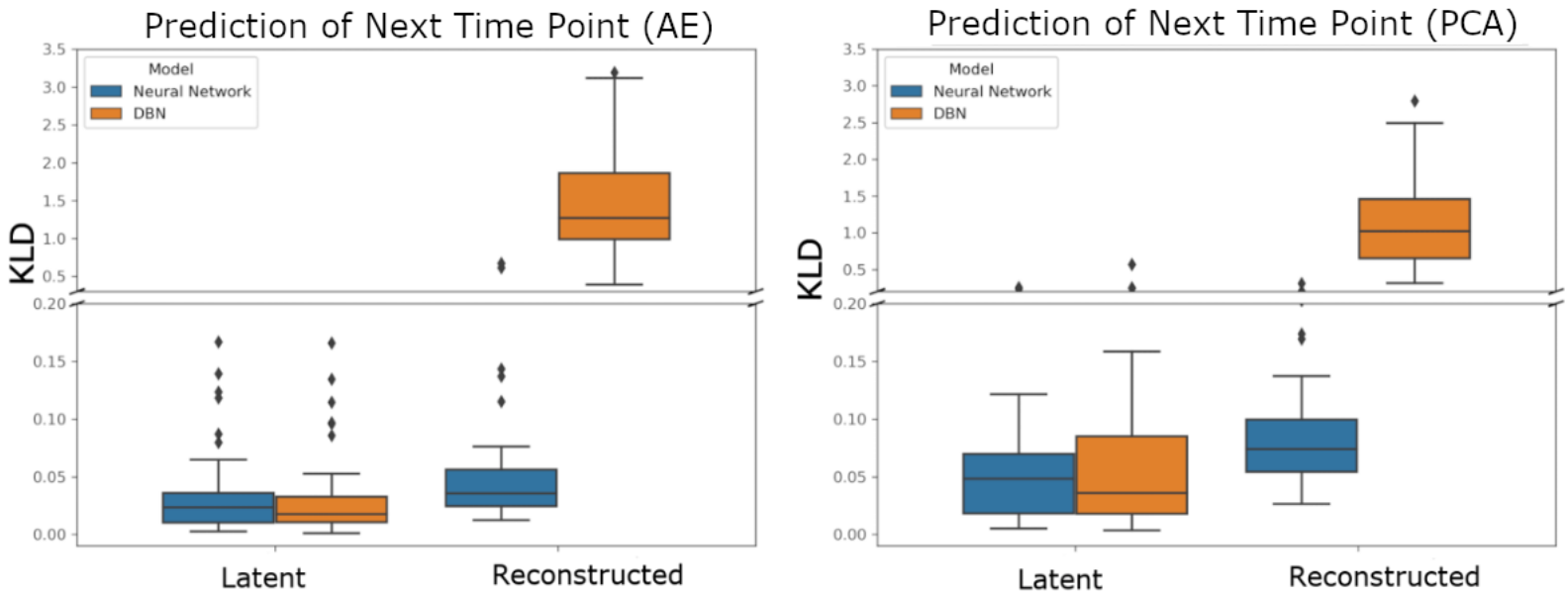


Figure 53. Prediction of next time point using NN and DBN models for infant gut dataset. Boxplots on the left represent models trained on latent and reconstructed values from the diversity regularized AE. Boxplots on the right represent models trained on latent and reconstructed values from PCA interpolation.

Identification of Important External Factors

We identified important external factors during the 5-fold cross-validation evaluation of the datasets. For each partition, the test set was evaluated using the original trained model, and the models in which the inputs of respective external factors were replaced with noise. The change in KLD was evaluated on the test samples for each partition, and results showing the change in KLD across samples for each factor are shown in **Figure 54**. In the mouse dataset, diet was found to be significant with $p = 8.15 \text{ x } 10^{-12}$. In the vaginal dataset, menses was found to be significant with $p = 0.003$. In the infant gut dataset, we found gender to be significant at $p = 0.023$, the method of birth to be significant at $p = 0.013$, the amount of breast milk consumed to be significant at $p = 0.001$, and the proportion of days on antibiotics to be significant at $p = 2.94 \text{ x } 10^{-7}$.

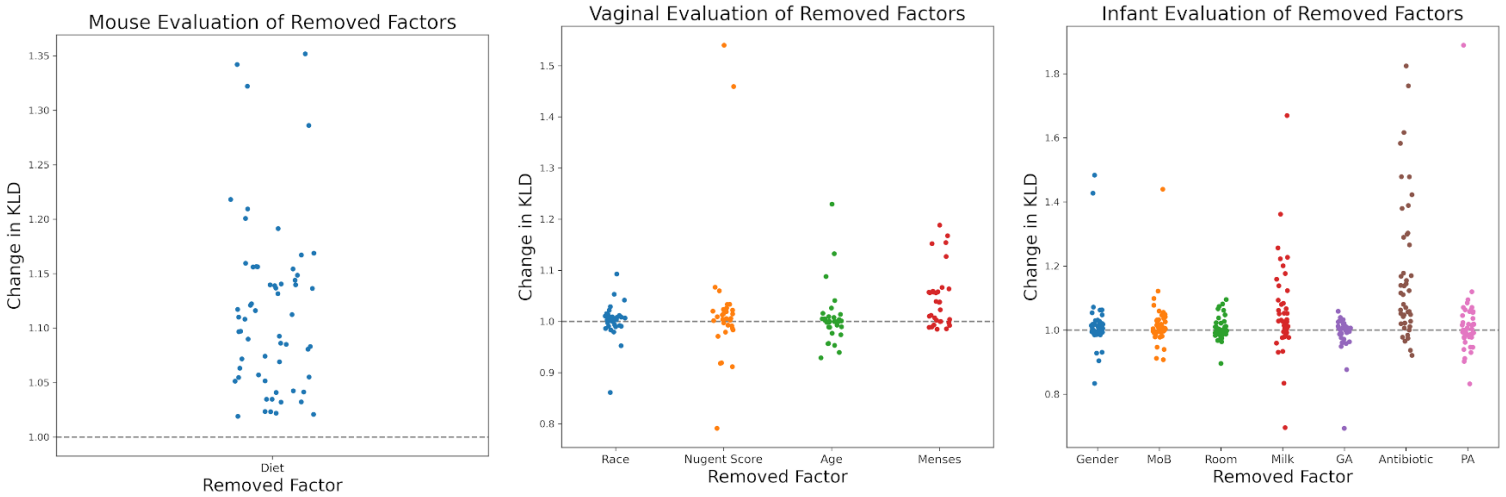


Figure 54. Evaluation of change in KLD values to determine factor significance. Points represent a subject and the value of the y-axis represents the ratio of the mean subject KLD when masking the respective factor with noise divided by the mean subject KLD when no masking is performed. A value greater than 1 represent a decrease in overall performance for that subject.

Identification of Biological Effects of External Factors

We clustered the mouse and vaginal datasets into 8 microbial modules and the infant gut dataset into 6 microbial modules. An example of the clustering of $\boldsymbol{R}$ and construction of $\overline{\boldsymbol{R}}$ for the mouse dataset is shown in **Figure 55.**

In the mouse dataset, we observed that the HFHS diet had a strong positive influence on Module 0 and Module 4. Combined, these two modules showed a significant enrichment (adjusted $p = 0.018$) for HFHS diet consumption using taxon set enrichment analysis in the MicroboimeAnalyst tool suite (Dhariwal et al., 2017). Next, we analyzed the trends of individual OTUs to see how well the dynamic model could predict the change of the microbiome community based on the changes in diet. Among the OTUs that were highly abundant, the model found that eating a HFHS diet led to an increase in *Clostridium*, *Eubacteirum, Faecalbacterium, Ruminococcus,* and *Blautia,* which have been associated with HFHS diets (Murphy, Velazquez, & Herbert, 2015; Xiao et al., 2017). The membership of microbial modules for the mouse dataset is shown in **Table XIX** and the visualization of the interaction network is shown in **Figure 56.**

In the vaginal dataset, we observed that the effect of menses had a strong negative impact on the abundance of *Lactobacillus iners* and *Lactobacillus jensenii*. Previous studies have observed a similar effect in which the abundance of these microbes as well as other species of the *Lactobacillus* genera will rapidly increase before menstruation, and the decrease during menstruation before the vaginal microbiome is again stabilized (Santiago et al., 2011; Srinivasan et al., 2010). Membership of microbial modules for the vaginal dataset is shown in **Table XX** and the visualization of the interaction network is shown in **Figure 57.**

In the infant gut dataset, we observed multiple significant external interactions. When analyzing the level of breast milk consumed by the infant, we saw a trend that the higher the percentage went in the discretization, the stronger the impact on Module 4, which contained only Gammaproteobacteria, an observation found in previous studies as well (Boudry et al., 2021; Lemas et al., 2016). This microbe is considered a ‘pioneer’ microbe of the infant gut that is critical for the development of the immune system,

specifically through the regulation of IgA response (Mirpuri et al., 2014; Tomkovich & Jobin, 2016). This response stimulates the immune system, which in turn starts to shape the gut microbiome into a more mature microbiome by reducing levels of Proteobacteria for Firmicutes and Bacteroides. In addition, we see that vaginal delivery is shown to have a positive effect on the abundance of Actinobacteria, Bacilli, Clostridia (Module 0). These microbes have been shown to inoculate infants that undergo vaginal delivery in previous studies and are important microbes in the development of the mature microbiome in humans (Kim, Sitarik, Woodcroft, Johnson, & Zoratti, 2019; Milani et al., 2017). Membership of microbial modules for the infant gut dataset is shown in **Table XXI** and the visualization of the interaction network is shown in **Figure 58.**

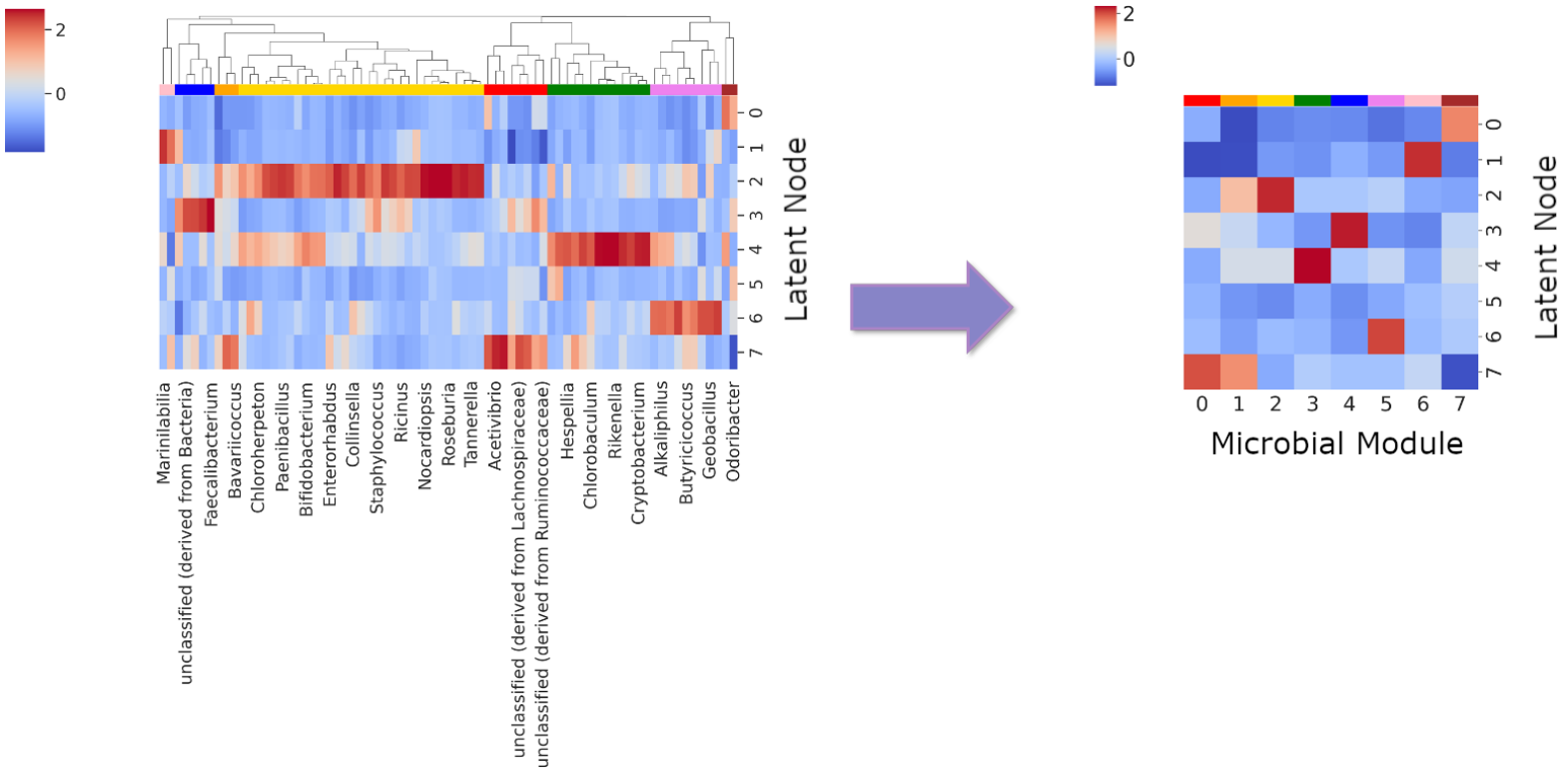


Figure 55. Clustering of mouse dataset into microbial modules. Mouse dataset is clustered into microbial modules using $\boldsymbol{R}$ matrix (left). The mean latent effect within each module is then calculated as $\overline{\boldsymbol{R}}$ (right).

TABLE XIX. MICROBIAL MODULE MEMBERSHIP FOR MOUSE DATASET.

Module 0	**Module 1**	**Module 2**			**Module 3**	
Lachnospiraceae	*Butyrivibrio*	*Enterococcus*	*Peptostreptococcus*	*Tannerella*	*Parabacteroides*	*Tetragenococcus*
Ruminococcus	*Tissierella*	*Lactobacillus*	*Atopobacter*	*Corynebacterium*	*Hespellia*	*Streptococcus*
Clostridiales	*Bavariicoccus*	*Bacteroides*	*Alistipes*	*Globicatella*	*Butyricimonas*	*Chlorobaculum*
Eubacterium		*Syntrophococcus*	*Paenibacillus*	*Ricinus*	*Flavobacterium*	*Rikenella*
Ruminococcaceae		*Barnesiella*	*Bifidobacterium*	*Nocardiopsis*	*Cryptobacterium*	*Atopobium*
Blautia		*Collinsella*	*Macrococcus*	*Phascolarctobacterium*	*Erysipelothrix*	*Alicyclobacillus*
Acetivibrio		*Enterorhabdus*	*Roseburia*	*Brevibacterium*	*Akkermansia*	
Pseudobutyrivibrio		*Staphylococcus*	*Chloroherpeton*	*Brachybacterium*		
		Anaerostipes	*Prevotella*	*Dialister*		
		Bacillus	*Ethanoligenens*	*Vibrio*		
		Acholeplasma				

Module 4	**Module 5**	**Module 6**	**Module 7**
Clostridium	*Pyramidobacter*	*Marinilabilia*	*Erysipelotrichaceae*
Candidatus Phytoplasma	*Porphyromonas*	*Cytophaga*	*Odoribacter*
Unclassified (derived from Bacteria)	*Butyricicoccus*		
Lactococcus	*Alkaliphilus*		
Faecalibacterium	Unclassified		
	Geobacillus		
	Burkholderia		
	Sphingobacterium		
	Desulfotomaculum		

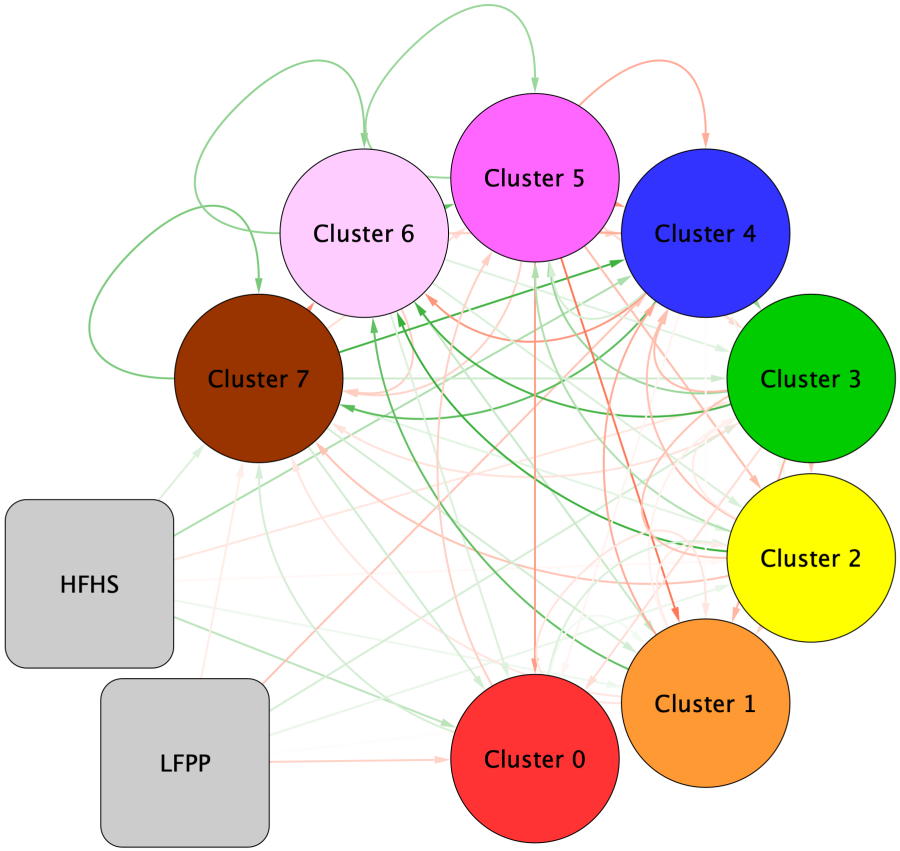


Figure 56. Module based interaction network with external factor effects for mouse dataset. Circles represent microbial modules and squares represent external actors. Green edges represent a positive interaction and red edges represent a negative interaction.

TABLE XX. MICROBIAL MODULE MEMBERSHIP FOR VAGINAL DATASET.

Module 0	**Module 1**	**Module 2**	**Module 3**		
L. iners *L. jensenii*	*L. crispatus*	*Atopobium* *Prevotella* *Sneathia* Ruminococcaceae.3 *Mobiluncus* *Megasphaera* Lachnospiraceae.10 Coriobacteriaceae.3 *Allisonella* Coriobacteriaceae.1	*L. gasseri* *Corynebacterium* *Finegoldia* *Bifidobacterium* L.otu4 *Ruminococcaceae.5* *Peptostreptococcus* *Staphylococcus* Lachnospiraceae.2 *Ureaplasma*	*Blautia* Incertae_Sedis_XI.1 *Bacteroides* L.otu3 *Facklamia* *Arthrobacter* Mollicutes.1 *L.vaginalis* *Oscillibacter* *Brevibacterium*	L.otu2 *Roseburia* Ruminococcaceae.1 *Arcanobacterium* L.otu1 *Ralstonia*

Module 4		***Module 5***	***Module 6***	***Module 7***
Parvimonas *Anaerococcus* *Moryella* *Veillonella* *Escherichia.Shigella* *Dialister* *Campylobacter* Incertae_Sedis_XI.2 *Bulleidia* *Actinomyces*	*Enterococcus* *Fastidiosipila* *Actinobaculum* *Xylanibacter* *L.plantarum*	*Gardnerella* *Aerococcus* L.otu5 Lachnospiraceae.11 *Shuttleworthia* *Porphyromonas* *Faecalibacterium* *Gallicola* *Varibaculum* Clostridiales.6	*Peptoniphilus* Lactobacillales.2 *Peptococcus* Ruminococcaceae.2 *Helcococcus* Bacteroidetes.1 *Propionimicrobium* *Segniliparus*	*Streptococcus* *Gemella* *Fusobacterium*

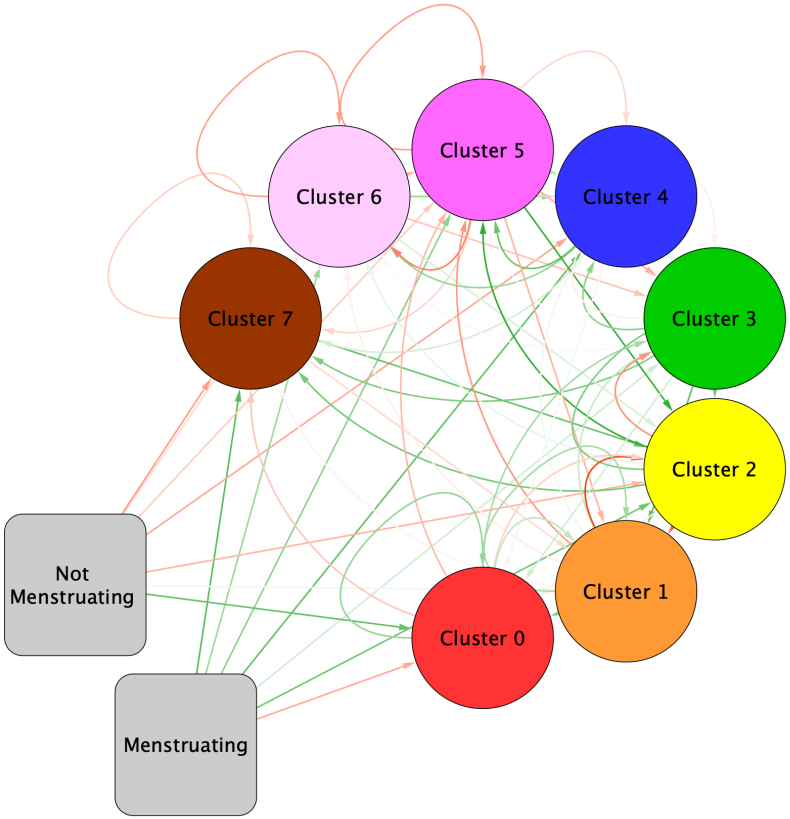


Figure 57. Module based interaction network with external factor effects for vaginal dataset. Circles represent microbial modules and squares represent external actors. Green edges represent a positive interaction and red edges represent a negative interaction.

TABLE XXI. MICROBIAL MODULE MEMBERSHIP FOR INFANT GUT DATASET.

Module 0	**Module 1**	**Module 2**	**Module 3**	**Module 4**	**Module 5**
Actinobacteria Bacilli Clostridia	Alphaproteobacteria Cyanobacteria Flavobacteria	Bacteroidia	Betaproteobacteria Epsilonproteobacteria Fusobacteria	Gammaproteobacteria	Unclassified

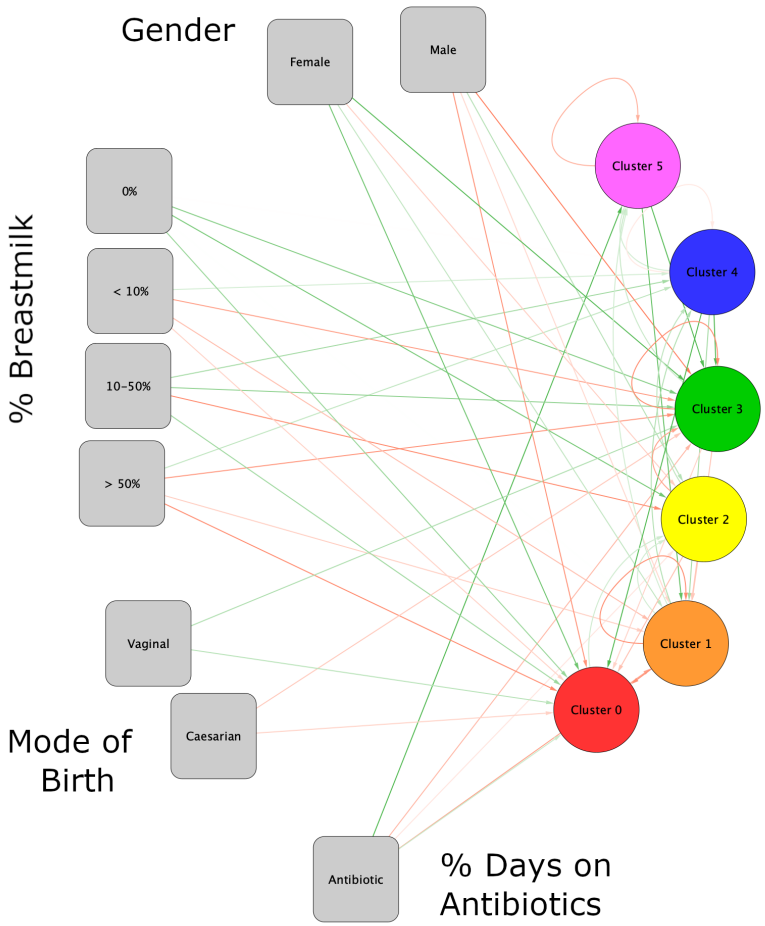


Figure 58. Module based interaction network with external factor effects for infant gut dataset. Circles represent microbial modules and squares represent external actors. Green edges represent a positive interaction and red edges represent a negative interaction.

5.4 Conclusion

In this work, we developed a novel model 'DiRLaM', a combination of an AE and a DNN in order to predict the dynamic changes of longitudinal microbiome datasets to capture microbial interactions and the effects of external factors and stimuli. By representing the microbiome community as a reduced latent space using an AE, we can capture the essential intrinsic community structure while making the model more robust to noise. In addition, we showed with synthetic data that interpolating within the latent space of the autoencoder provides more accurate and stable interpolations of the microbiome community compared to current splining approaches to each microbial feature. Furthermore, we showed that adding a regularization on the beta diversities of the reconstructed communities further improved the interpolation. We then used the DNN to combine the latent microbiome community with additional information about the host and external stimuli to predict what the new microbiome community will become. We demonstrated that the DNN outperformed the state-of-the-art DBN models, and the use of the latent space further improved dynamic prediction over the raw values.

Our approach not only outperforms the DBN models, but also overcomes the limitations of DBN models, which often require a large amount of filtering to reduce the number of network nodes and a set limit for the number of parents that a network node can have. On the other hand, the use of the DNN model allows is scalable to input space and sample size and can allow for as many microbe-microbe and diet-microbe interactions as needed during model training. DiRLaM provides both an accurate modeling of microbiome dynamics under multiple external factors and the identification of significant patient characteristics and external stimuli driving microbial dynamics, further empowering researchers to identify the key factors best suited for treatment through microbiome engineering.

CITED WORK

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., . . . Isard, M. (2016). *Tensorflow: A system for large-scale machine learning.* Paper presented at the 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16).

Abdel Hadi, L., Di Vito, C., & Riboni, L. (2016). Fostering Inflammatory Bowel Disease: Sphingolipid Strategies to Join Forces. *Mediators of Inflammation, 2016*, 1-13. doi:10.1155/2016/3827684

Agouridis, A. P., Elisaf, M., & Milionis, H. J. (2011). An overview of lipid abnormalities in patients with inflammatory bowel disease. *Annals of gastroenterology, 24*(3), 181-187. Retrieved from https://pubmed.ncbi.nlm.nih.gov/24713706
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3959314/

Ahn, J., Sinha, R., Pei, Z., Dominianni, C., Wu, J., Shi, J., . . . Yang, L. (2013). Human gut microbiome and risk for colorectal cancer. *Journal of the National Cancer Institute, 105*(24), 1907-1911.

Äijö, T., Müller, C. L., & Bonneau, R. (2018). Temporal probabilistic modeling of bacterial compositions derived from 16S rRNA sequencing. *Bioinformatics, 34*(3), 372-380.

Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017, 21-23 Aug. 2017). *Understanding of a convolutional neural network.* Paper presented at the 2017 International Conference on Engineering and Technology (ICET).

Allen, B., Kon, M., & Bar-Yam, Y. (2009). A new phylogenetic diversity measure generalizing the Shannon index and its application to phyllostomid bats. *The American Naturalist, 174*(2), 236-243.

Angermueller, C., Pärnamaa, T., Parts, L., & Stegle, O. (2016). Deep learning for computational biology. *Molecular systems biology, 12*(7), 878-878. doi:10.15252/msb.20156651

Athiwaratkun, B., & Kang, K. (2015). Feature representation in convolutional neural networks. *arXiv preprint arXiv:1507.02313*.

Bajaj, J. S., Ridlon, J. M., Hylemon, P. B., Thacker, L. R., Heuman, D. M., Smith, S., . . . Gillevet, P. M. (2012). Linkage of gut microbiome with cognition in hepatic encephalopathy. *American Journal of Physiology-Gastrointestinal and Liver Physiology, 302*(1), G168-G175.

Baldi, P. (2012). *Autoencoders, unsupervised learning, and deep architectures.* Paper presented at the Proceedings of ICML workshop on unsupervised and transfer learning.

Baran, R., Brodie, E. L., Mayberry-Lewis, J., Hummel, E., Da Rocha, U. N., Chakraborty, R., . . . Garcia-Pichel, F. (2015). Exometabolite niche partitioning among sympatric soil bacteria. *Nature Communications, 6*(1), 1-9.

Benson, A. K. (2016). The gut microbiome—an emerging complex trait. *Nature genetics, 48*(11), 1301-1302.

Berthelot, D., Raffel, C., Roy, A., & Goodfellow, I. (2018). Understanding and improving interpolation in autoencoders via an adversarial regularizer. *arXiv preprint arXiv:1807.07543*.

Biggs, M. B., Medlock, G. L., Kolling, G. L., & Papin, J. A. (2015). Metabolic network modeling of microbial communities. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine, 7*(5), 317-334.

Borg, I., & Groenen, P. J. (2005). *Modern multidimensional scaling: Theory and applications*: Springer Science & Business Media.

Boudry, G., Charton, E., Huerou-Luron, L., Ferret-Bernard, S., Le Gall, S., Even, S., & Blat, S. (2021). The relationship between breast milk components and the infant gut microbiota. *Frontiers in nutrition, 8*, 87.

Bray, J. R., & Curtis, J. T. (1957). An ordination of the upland forest communities of southern Wisconsin. *Ecological monographs, 27*(4), 326-349.

Bunin, G. (2017). Ecological communities with Lotka-Volterra dynamics. *Physical Review E, 95*(4), 042414.

Cammarota, G., Ianiro, G., Ahern, A., Carbone, C., Temko, A., Claesson, M. J., . . . Tortora, G. (2020). Gut microbiome, big data and machine learning to promote precision medicine for cancer. *Nature reviews gastroenterology & hepatology, 17*(10), 635-648.

Cani, P. D., & Delzenne, N. M. (2011). The gut microbiome as therapeutic target. *Pharmacology & therapeutics, 130*(2), 202-212.

Carmody, Rachel N., Gerber, Georg K., Luevano, Jesus M., Jr., Gatti, Daniel M., Somes, L., Svenson, Karen L., & Turnbaugh, Peter J. (2015). Diet Dominates Host Genotype in Shaping the Murine Gut Microbiota. *Cell host & microbe, 17*(1), 72-84. doi:10.1016/j.chom.2014.11.010

Chai, L. E., Loh, S. K., Low, S. T., Mohamad, M. S., Deris, S., & Zakaria, Z. (2014). A review on the computational approaches for gene regulatory network construction. *Computers in biology and medicine, 48*, 55-65.

Chen, H.-I. H., Chiu, Y.-C., Zhang, T., Zhang, S., Huang, Y., & Chen, Y. (2018). GSAE: an autoencoder with embedded gene-set nodes for genomics functional characterization. *BMC systems biology, 12*(8), 45-57.

Chen, J., Ryu, E., Hathcock, M., Ballman, K., Chia, N., Olson, J. E., & Nelson, H. (2016). Impact of demographics on human gut microbial diversity in a US Midwest population. *PeerJ, 4*, e1514.

Chen, T., & Chefd'Hotel, C. (2014). *Deep learning based automatic immune cell detection for immunohistochemistry images.* Paper presented at the International workshop on machine learning in medical imaging.

Chen, T., Xu, R., He, Y., & Wang, X. (2017). Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN. *Expert Systems with Applications, 72*, 221-230.

Chen, V. L., & Kasper, D. L. (2014). Interactions between the intestinal microbiota and innate lymphoid cells. *Gut Microbes, 5*(1), 129-140.

Claussen, J. C., Skiecevičienė, J., Wang, J., Rausch, P., Karlsen, T. H., Lieb, W., . . . Hütt, M.-T. (2017). Boolean analysis reveals systematic interactions among low-abundance species in the human gut

microbiome. *PLoS computational biology, 13*(6), e1005361-e1005361. doi:10.1371/journal.pcbi.1005361

Clevert, D.-A., Unterthiner, T., & Hochreiter, S. (2015). Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). *Under Review of ICLR2016 (1997)*.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning, 20*(3), 273-297.

Creswell, R., Tan, J., Leff, J. W., Brooks, B., Mahowald, M. A., Thieroff-Ekerdt, R., & Gerber, G. K. (2020). High-resolution temporal profiling of the human gut microbiome reveals consistent and cascading alterations in response to dietary glycans. *Genome medicine, 12*(1), 1-16.

Cristovao, P., Nakada, H., Tanimura, Y., & Asoh, H. (2020). Generating in-between images through learned latent space representation using variational autoencoders. *IEEE Access, 8*, 149456-149467.

D'Amore, R., Ijaz, U. Z., Schirmer, M., Kenny, J. G., Gregory, R., Darby, A. C., . . . Hall, N. (2016). A comprehensive benchmarking study of protocols and sequencing platforms for 16S rRNA community profiling. *BMC genomics, 17*(1), 1-20.

Davidich, M. I., & Bornholdt, S. (2008). Boolean network model predicts cell cycle sequence of fission yeast. *PLoS One, 3*(2), e1672.

De Filippo, C., Ramazzotti, M., Fontana, P., & Cavalieri, D. (2012). Bioinformatic approaches for functional annotation and pathway inference in metagenomics data. *Briefings in Bioinformatics, 13*(6), 696-710. doi:10.1093/bib/bbs070

De Preter, V., Machiels, K., Joossens, M., Arijs, I., Matthys, C., Vermeire, S., . . . Verbeke, K. (2015). Faecal metabolite profiling identifies medium-chain fatty acids as discriminating compounds in IBD. *Gut, 64*(3), 447-458. doi:10.1136/gutjnl-2013-306423

Deng, L., & Liu, Y. (2018). *Deep learning in natural language processing*: Springer.

Dhariwal, A., Chong, J., Habib, S., King, I. L., Agellon, L. B., & Xia, J. (2017). MicrobiomeAnalyst: a web-based tool for comprehensive statistical, visual and meta-analysis of microbiome data. *Nucleic acids research, 45*(W1), W180-W188.

Ditzler, G., Polikar, R., & Rosen, G. (2015). Multi-layer and recursive neural networks for metagenomic classification. *IEEE transactions on nanobioscience, 14*(6), 608-616.

Dong, N., Xue, C., Zhang, L., Zhang, T., Wang, C., Bi, C., & Shan, A. (2020). Oleanolic acid enhances tight junctions and ameliorates inflammation in Salmonella typhimurium-induced diarrhea in mice via the TLR4/NF-κB and MAPK pathway. *Food Funct, 11*(1), 1122-1132. doi:10.1039/c9fo01718f

Dreyfus, S. E. (1990). Artificial neural networks, back propagation, and the Kelley-Bryson gradient procedure. *Journal of guidance, control, and dynamics, 13*(5), 926-928.

Duvallet, E., Semerano, L., Assier, E., Falgarone, G., & Boissier, M.-C. (2011). Interleukin-23: a key cytokine in inflammatory diseases. *Annals of medicine, 43*(7), 503-511.

Edwards, J. S., Covert, M., & Palsson, B. (2002). Metabolic modelling of microbes: the flux-balance approach. *Environmental microbiology, 4*(3), 133-140.

Faith, D. P. (1992). Conservation evaluation and phylogenetic diversity. *Biological conservation, 61*(1), 1-10.

Feng, Q., Liu, Z., Zhong, S., Li, R., Xia, H., Jie, Z., . . . Fan, Y. (2016). Integrated metabolomics and metagenomics analysis of plasma and urine identified microbial metabolites associated with coronary heart disease. *Scientific reports, 6*(1), 1-14.

Fernandes, A. D., Macklaim, J. M., Linn, T. G., Reid, G., & Gloor, G. B. (2013). ANOVA-like differential expression (ALDEx) analysis for mixed population RNA-Seq. *PLoS One, 8*(7), e67019. doi:10.1371/journal.pone.0067019

Finucane, M. M., Sharpton, T. J., Laurent, T. J., & Pollard, K. S. (2014). A taxonomic signature of obesity in the microbiome? Getting to the guts of the matter. *PLoS One, 9*(1), e84689.

Fioravanti, D., Giarratano, Y., Maggio, V., Agostinelli, C., Chierici, M., Jurman, G., & Furlanello, C. (2018). Phylogenetic convolutional neural networks in metagenomics. *BMC bioinformatics, 19*(2), 49.

Fischer, M., Bittar, M., Papa, E., Kassam, Z., & Smith, M. (2017). Can you cause inflammatory bowel disease with fecal transplantation? A 31-patient case-series of fecal transplantation using stool from a donor who later developed Crohn's disease. *Gut Microbes, 8*(3), 205-207.

Fornelos, N., Franzosa, E. A., Bishai, J., Annand, J. W., Oka, A., Lloyd-Price, J., . . . Xavier, R. J. (2020). Growth effects of N-acylethanolamines on gut bacteria reflect altered bacterial abundances in inflammatory bowel disease. *Nature microbiology, 5*(3), 486-497. doi:10.1038/s41564-019-0655-7

Franzosa, E. A., Sirota-Madi, A., Avila-Pacheco, J., Fornelos, N., Haiser, H. J., Reinker, S., . . . McIver, L. J. (2019). Gut microbiome structure and metabolic activity in inflammatory bowel disease. *Nature microbiology, 4*(2), 293-305.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics, 36*(4), 193-202. doi:10.1007/BF00344251

Gajer, P., Brotman, R., Bai, G., Sakamoto, J., Schütte, U., Zhong, X., . . . Zhou, X. (2012). Temporal dynamics of the human vaginal microbiota. Sci Transl Med 4: 132ra52. In.

García-Jiménez, B., Muñoz, J., Cabello, S., Medina, J., & Wilkinson, M. D. (2021). Predicting microbiomes through a deep latent space. *Bioinformatics, 37*(10), 1444-1451.

Gaulke, C. A., & Sharpton, T. J. (2018). The influence of ethnicity and geography on human gut microbiome composition. *Nature medicine, 24*(10), 1495-1496.

Geman, O., Chiuchisan, I., Covasa, M., Doloc, C., Milici, M.-R., & Milici, L.-D. (2016). *Deep learning tools for human microbiome big data.* Paper presented at the International Workshop Soft Computing Applications.

Ghahramani, A., Watt, F. M., & Luscombe, N. M. (2018). Generative adversarial networks simulate gene expression and predict perturbations in single cells. *bioRxiv*, 262501.

Ghahramani, Z. (1997). *Learning dynamic Bayesian networks.* Paper presented at the International School on Neural Networks, Initiated by IIASS and EMFCSC.

Gilbert, J. A., Quinn, R. A., Debelius, J., Xu, Z. Z., Morton, J., Garg, N., . . . Knight, R. (2016). Microbiome-wide association studies link dynamic microbial consortia to disease. *Nature, 535*(7610), 94-103. doi:10.1038/nature18850

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*: MIT press.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., . . . Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems, 27*.

Gopalakrishnan, V., Helmink, B. A., Spencer, C. N., Reuben, A., & Wargo, J. A. (2018). The influence of the gut microbiome on cancer, immunity, and cancer immunotherapy. *Cancer cell, 33*(4), 570-580.

Gottstein, W., Olivier, B. G., Bruggeman, F. J., & Teusink, B. (2016). Constraint-based stoichiometric modelling from single organisms to microbial communities. *Journal of the Royal Society, Interface, 13*(124), 20160627. doi:10.1098/rsif.2016.0627

Hansson, G. C. (2012). Role of mucus layers in gut infection and inflammation. *Current opinion in microbiology, 15*(1), 57-62. doi:10.1016/j.mib.2011.11.002

Hara, K., Saito, D., & Shouno, H. (2015). *Analysis of function of rectified linear unit used in deep learning.* Paper presented at the 2015 International Joint Conference on Neural Networks (IJCNN).

Harsch, I. A., & Konturek, P. C. (2019). Adhesion ileus after fecal microbiota transplantation in long-standing radiation colitis. *Case reports in gastrointestinal medicine, 2019*.

Hartstra, A. V., Bouter, K. E., Bäckhed, F., & Nieuwdorp, M. (2015). Insights into the role of the microbiome in obesity and type 2 diabetes. *Diabetes care, 38*(1), 159-165.

Hashemifar, S., Neyshabur, B., Khan, A. A., & Xu, J. (2018). Predicting protein–protein interactions through sequence-based deep learning. *Bioinformatics, 34*(17), i802-i810.

Hawinkel, S., Mattiello, F., Bijnens, L., & Thas, O. (2017). A broken promise: microbiome differential abundance methods do not control the false discovery rate. *Briefings in Bioinformatics, 20*(1), 210-221. doi:10.1093/bib/bbx104

Hawkins-Hooker, A., Depardieu, F., Baur, S., Couairon, G., Chen, A., & Bikard, D. (2021). Generating functional protein variants with variational autoencoders. *PLoS computational biology, 17*(2), e1008736.

Heinken, A., Ravcheev, D. A., Baldini, F., Heirendt, L., Fleming, R. M. T., & Thiele, I. (2019). Systematic assessment of secondary bile acid metabolism in gut microbes reveals distinct metabolic capabilities in inflammatory bowel disease. *Microbiome, 7*(1), 75-75. doi:10.1186/s40168-019-0689-3

Helmink, B. A., Khan, M. W., Hermann, A., Gopalakrishnan, V., & Wargo, J. A. (2019). The microbiome, cancer, and cancer therapy. *Nature medicine, 25*(3), 377-388.

Ho, T. K. (1995). *Random decision forests.* Paper presented at the Proceedings of 3rd international conference on document analysis and recognition.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics, 12*(1), 55-67.

Holmes, E., Kinross, J., Gibson, G. R., Burcelin, R., Jia, W., Pettersson, S., & Nicholson, J. K. (2012). Therapeutic modulation of microbiota-host metabolic interactions. *Science Translational Medicine, 4*(137), 137rv136-137rv136.

Hong, H., Jiang, S., Li, H., Du, G., Sun, Y., Tao, H., . . . Li, W. (2020). DeepHiC: A generative adversarial network for enhancing Hi-C data resolution. *PLoS computational biology, 16*(2), e1007287.

Ioffe, S., & Szegedy, C. (2015). *Batch normalization: Accelerating deep network training by reducing internal covariate shift.* Paper presented at the International conference on machine learning.

Janda, J. M., & Abbott, S. L. (2007). 16S rRNA gene sequencing for bacterial identification in the diagnostic laboratory: pluses, perils, and pitfalls. *Journal of clinical microbiology, 45*(9), 2761-2764.

Jie, Z., Xia, H., Zhong, S.-L., Feng, Q., Li, S., Liang, S., . . . Zhao, H. (2017). The gut microbiome in atherosclerotic cardiovascular disease. *Nature Communications, 8*(1), 1-12.

John Lu, Z. (2010). The elements of statistical learning: data mining, inference, and prediction. In: Wiley Online Library.

Joseph, T. A., Shenhav, L., Xavier, J. B., Halperin, E., & Pe'er, I. (2020). Compositional Lotka-Volterra describes microbial dynamics in the simplex. *PLoS computational biology, 16*(5), e1007917.

Karlsson, F. H., Tremaroli, V., Nookaew, I., Bergström, G., Behre, C. J., Fagerberg, B., . . . Bäckhed, F. (2013). Gut metagenome in European women with normal, impaired and diabetic glucose control. *Nature, 498*(7452), 99-103.

Kashyap, P. C., Chia, N., Nelson, H., Segal, E., & Elinav, E. (2017). Microbiome at the Frontier of Personalized Medicine. *Mayo Clinic Proceedings, 92*(12), 1855-1864. doi:https://doi.org/10.1016/j.mayocp.2017.10.004

Kaźmierczak-Siedlecka, K., Daca, A., Fic, M., van de Wetering, T., Folwarski, M., & Makarewicz, W. (2020). Therapeutic methods of gut microbiota modification in colorectal cancer management–fecal microbiota transplantation, prebiotics, probiotics, and synbiotics. *Gut Microbes, 11*(6), 1518-1530.

Kelley, D. R., Snoek, J., & Rinn, J. L. (2016). Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome research, 26*(7), 990-999.

Khan, S., & Kelly, L. (2020). Multiclass Disease Classification from Microbial Whole-Community Metagenomes. *Pac Symp Biocomput, 25*, 55-66.

Kim, H., Sitarik, A. R., Woodcroft, K., Johnson, C. C., & Zoratti, E. (2019). Birth Mode, Breastfeeding, Pet Exposure, and Antibiotic Use: Associations With the Gut Microbiome and Sensitization in Children. *Current Allergy and Asthma Reports, 19*(4), 22. doi:10.1007/s11882-019-0851-9

Kingma, D., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations*.

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kinross, J. M., Darzi, A. W., & Nicholson, J. K. (2011). Gut microbiome-host interactions in health and disease. *Genome medicine, 3*(3), 14.

Knights, D., Parfrey, L. W., Zaneveld, J., Lozupone, C., & Knight, R. (2011). Human-associated microbial signatures: examining their predictive value. *Cell host & microbe, 10*(4), 292-296.

Kostic, A. D., Gevers, D., Siljander, H., Vatanen, T., Hyötyläinen, T., Hämäläinen, A.-M., . . . Mattila, I. (2015). The dynamics of the human infant gut microbiome in development and in progression toward type 1 diabetes. *Cell host & microbe, 17*(2), 260-273.

Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal, 37*(2), 233-243.

Kuntal, B. K., Gadgil, C., & Mande, S. S. (2019). Web-gLV: a web based platform for lotka-volterra based modeling and simulation of microbial populations. *Frontiers in microbiology, 10*, 288.

Kurilshikov, A., Wijmenga, C., Fu, J., & Zhernakova, A. (2017). Host genetics and gut microbiome: challenges and perspectives. *Trends in immunology, 38*(9), 633-647.

La Rosa, P. S., Warner, B. B., Zhou, Y., Weinstock, G. M., Sodergren, E., Hall-Moore, C. M., . . . Linneman, L. A. (2014). Patterned progression of bacterial populations in the premature infant gut. *Proceedings of the National Academy of Sciences, 111*(34), 12522-12527.

Lande, R. (1996). Statistics and partitioning of species diversity, and similarity among multiple communities. *Oikos*, 5-13.

LaPierre, N., Ju, C. J. T., Zhou, G., & Wang, W. (2019). MetaPheno: A critical evaluation of deep learning and machine learning in metagenome-based disease prediction. *Methods, 166*, 74-82. doi:https://doi.org/10.1016/j.ymeth.2019.03.003

Larsen, P. E., Collart, F. R., Field, D., Meyer, F., Keegan, K. P., Henry, C. S., . . . Gilbert, J. A. (2011). Predicted Relative Metabolomic Turnover (PRMT): determining metabolic turnover from a coastal marine metagenomic dataset. *Microbial informatics and experimentation, 1*(1), 4.

Larsen, P. E., Field, D., & Gilbert, J. A. (2012). Predicting bacterial community assemblages using an artificial neural network approach. *Nature Methods, 9*(6), 621-625.

Le Chatelier, E., Nielsen, T., Qin, J., Prifti, E., Hildebrand, F., Falony, G., . . . Kennedy, S. (2013). Richness of human gut microbiome correlates with metabolic markers. *Nature, 500*(7464), 541-546.

Le, V., Quinn, T. P., Tran, T., & Venkatesh, S. (2019). Deep in the Bowel: Highly Interpretable Neural Encoder-Decoder Networks Predict Gut Metabolites from Gut Microbiome. *bioRxiv*, 686394. doi:10.1101/686394

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436-444. doi:10.1038/nature14539

Lee-Sarwar, K. A., Lasky-Su, J., Kelly, R. S., Litonjua, A. A., & Weiss, S. T. (2020). Metabolome–Microbiome Crosstalk and Human Disease. *Metabolites, 10*(5), 181.

Lemas, D. J., Young, B. E., Baker, P. R., Tomczik, A. C., Soderborg, T. K., Hernandez, T. L., . . . Ir, D. (2016). Alterations in human milk leptin and insulin are associated with early changes in the infant intestinal microbiome. *The American journal of clinical nutrition, 103*(5), 1291-1300.

Li, L., Chen, Y., Shen, Z., Zhang, X., Sang, J., Ding, Y., . . . Jin, C. (2020). Convolutional neural network for the diagnosis of early gastric cancer based on magnifying narrow band imaging. *Gastric Cancer, 23*(1), 126-132.

Li, S. S., Zhu, A., Benes, V., Costea, P. I., Hercog, R., Hildebrand, F., . . . Voigt, A. Y. (2016). Durable coexistence of donor and recipient strains after fecal microbiota transplantation. *Science, 352*(6285), 586-589.

Liu, C., Dunkin, D., Lai, J., Song, Y., Ceballos, C., Benkov, K., & Li, X.-M. (2015). Anti-inflammatory Effects of Ganoderma lucidum Triterpenoid in Human Crohn's Disease Associated with Downregulation of NF-κB Signaling. *Inflammatory bowel diseases, 21*(8), 1918-1925. doi:10.1097/MIB.0000000000000439

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi, F. E. (2017). A survey of deep neural network architectures and their applications. *Neurocomputing, 234*, 11-26.

Lloyd-Price, J., Arze, C., Ananthakrishnan, A. N., Schirmer, M., Avila-Pacheco, J., Poon, T. W., . . . Investigators, I. (2019). Multi-omics of the gut microbial ecosystem in inflammatory bowel diseases. *Nature, 569*(7758), 655-662. doi:10.1038/s41586-019-1237-9

Love, M. I., Huber, W., & Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome biology, 15*(12), 550.

Lozupone, C., & Knight, R. (2005). UniFrac: a new phylogenetic method for comparing microbial communities. *Applied and environmental microbiology, 71*(12), 8228-8235.

Lugo-Martinez, J., Ruiz-Perez, D., Narasimhan, G., & Bar-Joseph, Z. (2019). Dynamic interaction network inference from longitudinal microbiome data. *Microbiome, 7*(1), 54.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., & Frey, B. (2015). Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*.

Mallick, H., Franzosa, E. A., McLver, L. J., Banerjee, S., Sirota-Madi, A., Kostic, A. D., . . . Huttenhower, C. (2019). Predictive metabolomic profiling of microbial communities using amplicon or metagenomic sequences. *Nature Communications, 10*(1), 3136. doi:10.1038/s41467-019-10927-1

Mallott, E. K., Borries, C., Koenig, A., Amato, K. R., & Lu, A. (2020). Reproductive hormones mediate changes in the gut microbiome during pregnancy and lactation in Phayre's leaf monkeys. *Scientific reports, 10*(1), 9961. doi:10.1038/s41598-020-66865-2

Manandhar, I., Alimadadi, A., Aryal, S., Munroe, P. B., Joe, B., & Cheng, X. (2021). Gut microbiome-based supervised machine learning for clinical diagnosis of inflammatory bowel diseases. *American Journal of Physiology-Gastrointestinal and Liver Physiology*.

Mandal, S., Van Treuren, W., White, R. A., Eggesbø, M., Knight, R., & Peddada, S. D. (2015). Analysis of composition of microbiomes: a novel method for studying microbial composition. *Microbial Ecology in Health and Disease, 26*(1), 27663.

Maroniche, G. A., García, J. E., Salcedo, F., & Creus, C. M. (2017). Molecular identification of Azospirillum spp.: Limitations of 16S rRNA and qualities of rpoD as genetic markers. *Microbiological research, 195*, 1-10.

Marotz, C. A., & Zarrinpar, A. (2016). Focus: microbiome: treating obesity and metabolic syndrome with fecal microbiota transplantation. *The Yale journal of biology and medicine, 89*(3), 383.

Martin, C. R., Osadchiy, V., Kalani, A., & Mayer, E. A. (2018). The Brain-Gut-Microbiome Axis. *Cell Mol Gastroenterol Hepatol, 6*(2), 133-148. doi:10.1016/j.jcmgh.2018.04.003

Matsen, F. A. t. (2015). Phylogenetics and the human microbiome. *Systematic biology, 64*(1), e26-e41. doi:10.1093/sysbio/syu053

Mattila, E., Uusitalo–Seppälä, R., Wuorela, M., Lehtola, L., Nurmi, H., Ristikankare, M., . . . Mattila, P. S. (2012). Fecal transplantation, through colonoscopy, is effective therapy for recurrent Clostridium difficile infection. *Gastroenterology, 142*(3), 490-496.

McFarland, L. V., Evans, C. T., & Goldstein, E. J. (2018). Strain-specificity and disease-specificity of probiotic efficacy: a systematic review and meta-analysis. *Frontiers in medicine, 5*, 124.

McGeachie, M. J., Chang, H.-H., & Weiss, S. T. (2014). CGBayesNets: Conditional Gaussian Bayesian Network Learning and Inference with Mixed Discrete and Continuous Data. *PLOS Computational Biology, 10*(6), e1003676. doi:10.1371/journal.pcbi.1003676

McGeachie, M. J., Sordillo, J. E., Gibson, T., Weinstock, G. M., Liu, Y.-Y., Gold, D. R., . . . Litonjua, A. (2016). Longitudinal prediction of the infant gut microbiome with dynamic bayesian networks. *Scientific reports, 6*, 20359.

McHardy, I. H., Goudarzi, M., Tong, M., Ruegger, P. M., Schwager, E., Weger, J. R., . . . Huttenhower, C. (2013). Integrative analysis of the microbiome and metabolome of the human intestinal mucosal surface reveals exquisite inter-relationships. *Microbiome, 1*(1), 17.

Milani, C., Duranti, S., Bottacini, F., Casey, E., Turroni, F., Mahony, J., . . . Mancabelli, L. (2017). The first microbial colonizers of the human gut: composition, activities, and health implications of the infant gut microbiota. *Microbiology and molecular biology reviews, 81*(4), e00036-00017.

Mirpuri, J., Raetz, M., Sturge, C. R., Wilhelm, C. L., Benson, A., Savani, R. C., . . . Yarovinsky, F. (2014). Proteobacteria-specific IgA regulates maturation of the intestinal microbiota. *Gut Microbes, 5*(1), 28-39.

Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Monti, S., Tamayo, P., Mesirov, J., & Golub, T. (2003). Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Machine learning, 52*(1), 91-118.

Morgan, X. C., Tickle, T. L., Sokol, H., Gevers, D., Devaney, K. L., Ward, D. V., . . . Snapper, S. B. (2012). Dysfunction of the intestinal microbiome in inflammatory bowel disease and treatment. *Genome biology, 13*(9), 1-18.

Mueller, D., Triebel, S., Rudakovski, O., & Richling, E. (2013). Influence of triterpenoids present in apple peel on inflammatory gene expression associated with inflammatory bowel disease (IBD). *Food Chem, 139*(1-4), 339-346. doi:10.1016/j.foodchem.2013.01.101

Murphy, E. A., Velazquez, K. T., & Herbert, K. M. (2015). Influence of high-fat diet on gut microbiota: a driving force for chronic disease risk. *Curr Opin Clin Nutr Metab Care, 18*(5), 515-520. doi:10.1097/mco.0000000000000209

Ng, A. (2011). Sparse autoencoder. *CS294A Lecture notes, 72*(2011), 1-19.

Ni, J., Shen, T.-C. D., Chen, E. Z., Bittinger, K., Bailey, A., Roggiani, M., . . . Wu, G. D. (2017). A role for bacterial urease in gut dysbiosis and Crohn's disease. *Science Translational Medicine, 9*(416), eaah6888. doi:10.1126/scitranslmed.aah6888

Nielsen, M. A. (2015). *Neural networks and deep learning* (Vol. 25): Determination press San Francisco, CA.

Olden, J., Joy, M., & Death, R. (2004). An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data. *Ecological Modelling, 178*, 389-397. doi:10.1016/j.ecolmodel.2004.03.013

Oudah, M., & Henschel, A. (2018). Taxonomy-aware feature engineering for microbiome classification. *BMC bioinformatics, 19*(1), 1-13.

Parada Venegas, D., De la Fuente, M. K., Landskron, G., González, M. J., Quera, R., Dijkstra, G., . . . Hermoso, M. A. (2019). Short Chain Fatty Acids (SCFAs)-Mediated Gut Epithelial and Immune Regulation and Its Relevance for Inflammatory Bowel Diseases. *Frontiers in immunology, 10*, 277-277. doi:10.3389/fimmu.2019.00277

Parker, A., Lawson, M. A., Vaux, L., & Pin, C. (2018). Host-microbe interaction in the gastrointestinal tract. *Environmental microbiology, 20*(7), 2337-2353.

Pasolli, E., Truong, D. T., Malik, F., Waldron, L., & Segata, N. (2016). Machine learning meta-analysis of large metagenomic datasets: tools and biological insights. *PLoS computational biology, 12*(7), e1004977.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Dubourg, V. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research, 12*, 2825-2830.

Pihur, V., Datta, S., & Datta, S. (2009). RankAggreg, an R package for weighted rank aggregation. *BMC bioinformatics, 10*(1), 1-10.

Preidis, G. A., & Versalovic, J. (2009). Targeting the human microbiome with antibiotics, probiotics, and prebiotics: gastroenterology enters the metagenomics era. *Gastroenterology, 136*(6), 2015-2031.

Qin, J., Li, R., Raes, J., Arumugam, M., Burgdorf, K. S., Manichanh, C., . . . Yamada, T. (2010). A human gut microbial gene catalogue established by metagenomic sequencing. *Nature, 464*(7285), 59-65.

Qin, J., Li, Y., Cai, Z., Li, S., Zhu, J., Zhang, F., . . . Shen, D. (2012). A metagenome-wide association study of gut microbiota in type 2 diabetes. *Nature, 490*(7418), 55-60.

Qin, N., Yang, F., Li, A., Prifti, E., Chen, Y., Shao, L., . . . Wu, L. (2014). Alterations of the human gut microbiome in liver cirrhosis. *Nature, 513*(7516), 59-64.

Qiu, Y. Q., Tian, X., & Zhang, S. (2015). Infer Metagenomic Abundance and Reveal Homologous Genomes Based on the Structure of Taxonomy Tree. *IEEE/ACM Trans Comput Biol Bioinform, 12*(5), 1112-1122. doi:10.1109/tcbb.2015.2415814

Quince, C., Walker, A. W., Simpson, J. T., Loman, N. J., & Segata, N. (2017). Shotgun metagenomics, from sampling to analysis. *Nature biotechnology, 35*(9), 833-844.

Quinn, R. A., Comstock, W., Zhang, T., Morton, J. T., da Silva, R., Tran, A., . . . Melnik, A. V. (2018). Niche partitioning of a pathogenic microbiome driven by chemical gradients. *Science advances, 4*(9), eaau1908.

Razzak, M. I., Naz, S., & Zaib, A. (2018). Deep learning for medical image processing: Overview, challenges and the future. *Classification in BioApps*, 323-350.

Reiman, D., & Dai, Y. (2019, 18-21 Nov. 2019). *Using Autoencoders for Predicting Latent Microbiome Community Shifts Responding to Dietary Changes.* Paper presented at the 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM).

Reiman, D., Manakkat Vijay, G. K., Xu, H., Sonin, A., Chen, D., Salomonis, N., . . . Khan, A. A. (2021). Pseudocell Tracer—A method for inferring dynamic trajectories using scRNAseq and its application to B cells undergoing immunoglobulin class switch recombination. *PLoS computational biology, 17*(5), e1008094.

Ren B, S. E., Tickle T, Huttenhower C (2020). sparseDOSSA: Sparse Data Observations for Simulating Synthetic Abundance. *R package*.

Rinaudo, P., Boudah, S., Junot, C., & Thévenot, E. A. (2016). biosigner: A New Method for the Discovery of Significant Molecular Signatures from Omics Data. *Frontiers in Molecular Biosciences, 3*(26). doi:10.3389/fmolb.2016.00026

Ritchie, M. L., & Romanuk, T. N. (2012). A meta-analysis of probiotic efficacy for gastrointestinal diseases. *PLoS One, 7*(4), e34938.

Robinson, M. D., McCarthy, D. J., & Smyth, G. K. (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics, 26*(1), 139-140.

Ronda, C., Chen, S. P., Cabral, V., Yaung, S. J., & Wang, H. H. (2019). Metagenomic engineering of the mammalian gut microbiome in situ. *Nature Methods, 16*(2), 167-170. doi:10.1038/s41592-018-0301-y

Routy, B., Le Chatelier, E., Derosa, L., Duong, C. P., Alou, M. T., Daillère, R., . . . Roberti, M. P. (2018). Gut microbiome influences efficacy of PD-1–based immunotherapy against epithelial tumors. *Science, 359*(6371), 91-97.

Rowland, I., Gibson, G., Heinken, A., Scott, K., Swann, J., Thiele, I., & Tuohy, K. (2018). Gut microbiota functions: metabolism of nutrients and other food components. *Eur J Nutr, 57*(1), 1-24. doi:10.1007/s00394-017-1445-8

Ruiz-Perez, D., Lugo-Martinez, J., Bourguignon, N., Mathee, K., Lerner, B., Bar-Joseph, Z., . . . Korem, T. Dynamic Bayesian Networks for Integrating Multi-omics Time Series Microbiome Data. *mSystems, 6*(2), e01105-01120. doi:10.1128/mSystems.01105-20

Sanschagrin, S., & Yergeau, E. (2014). Next-generation sequencing of 16S ribosomal RNA gene amplicons. *Journal of visualized experiments: JoVE*(90).

Santiago, G. L. D. S., Cools, P., Verstraelen, H., Trog, M., Missine, G., El Aila, N., . . . Vaneechoutte, M. (2011). Longitudinal study of the dynamics of vaginal microflora during two consecutive menstrual cycles. *PLoS One, 6*(11), e28180-e28180. doi:10.1371/journal.pone.0028180

Scholz, M. B., Lo, C.-C., & Chain, P. S. (2012). Next generation sequencing and bioinformatic bottlenecks: the current state of metagenomic data analysis. *Current opinion in biotechnology, 23*(1), 9-15.

Schwab, J. D., Kühlwein, S. D., Ikonomi, N., Kühl, M., & Kestler, H. A. (2020). Concepts in Boolean network modeling: What do they all mean? *Computational and Structural Biotechnology Journal, 18*, 571-582. doi:https://doi.org/10.1016/j.csbj.2020.03.001

Segata, N., Waldron, L., Ballarini, A., Narasimhan, V., Jousson, O., & Huttenhower, C. (2012). Metagenomic microbial community profiling using unique clade-specific marker genes. *Nature Methods, 9*(8), 811-814.

Shafiei, M., Dunn, K. A., Boon, E., MacDonald, S. M., Walsh, D. A., Gu, H., & Bielawski, J. P. (2015). BioMiCo: a supervised Bayesian model for inference of microbial community structure. *Microbiome, 3*(1), 8. doi:10.1186/s40168-015-0073-x

Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., . . . Ideker, T. (2003). Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome research, 13*(11), 2498-2504.

Shen, T.-C. D., Albenberg, L., Bittinger, K., Chehoud, C., Chen, Y.-Y., Judge, C. A., . . . Wu, G. D. (2015). Engineering the gut microbiota to treat hyperammonemia. *The Journal of clinical investigation, 125*(7), 2841-2850. doi:10.1172/JCI79214

Shenhav, L., Furman, O., Briscoe, L., Thompson, M., Silverman, J. D., Mizrahi, I., & Halperin, E. (2019). Modeling the temporal dynamics of the gut microbial community in adults and infants. *PLoS computational biology, 15*(6), e1006960.

Sims, D., Sudbery, I., Ilott, N. E., Heger, A., & Ponting, C. P. (2014). Sequencing depth and coverage: key considerations in genomic analyses. *Nature Reviews Genetics, 15*(2), 121-132.

Skelly, A. N., Sato, Y., Kearney, S., & Honda, K. (2019). Mining the microbiota for microbial and metabolite-based immunotherapies. *Nature Reviews Immunology, 19*(5), 305-323.

Smillie, C. S., Sauk, J., Gevers, D., Friedman, J., Sung, J., Youngster, I., . . . Sadowsky, M. J. (2018). Strain tracking reveals the determinants of bacterial engraftment in the human gut following fecal microbiota transplantation. *Cell host & microbe, 23*(2), 229-240. e225.

Sokol, H., Leducq, V., Aschard, H., Pham, H.-P., Jegou, S., Landman, C., . . . Nion-Larmurier, I. (2017). Fungal microbiota dysbiosis in IBD. *Gut, 66*(6), 1039-1048.

Srinivasan, S., Liu, C., Mitchell, C. M., Fiedler, T. L., Thomas, K. K., Agnew, K. J., . . . Fredricks, D. N. (2010). Temporal variability of human vaginal bacteria and relationship with bacterial vaginosis. *PLoS One, 5*(4), e10197.

Staley, C., Kaiser, T., Vaughn, B. P., Graiziger, C. T., Hamilton, M. J., Ur Rehman, T., . . . Sadowsky, M. J. (2018). Predicting recurrence of Clostridium difficile infection following encapsulated fecal microbiota transplantation. *Microbiome, 6*(1), 166.

Stein, R. R., Bucci, V., Toussaint, N. C., Buffie, C. G., Rätsch, G., Pamer, E. G., . . . Xavier, J. B. (2013). Ecological modeling from time-series inference: insight into dynamics and stability of intestinal microbiota. *PLoS computational biology, 9*(12), e1003388.

Steinway, S. N., Biggs, M. B., Loughran, T. P., Jr., Papin, J. A., & Albert, R. (2015). Inference of Network Dynamics and Metabolic Interactions in the Gut Microbiome. *PLoS computational biology, 11*(5), e1004338-e1004338. doi:10.1371/journal.pcbi.1004338

Suh, J. H., & Saba, J. D. (2015). Sphingosine-1-phosphate in inflammatory bowel disease and colitis-associated colon cancer: the fat's in the fire. *Translational cancer research, 4*(5), 469-483. doi:10.3978/j.issn.2218-676X.2015.10.06

Sun, J., & Chang, E. B. (2014). Exploring gut microbes in human health and disease: pushing the envelope. *Genes & Diseases, 1*(2), 132-139.

Talwar, D., Mongia, A., Sengupta, D., & Majumdar, A. (2018). AutoImpute: Autoencoder based imputation of single-cell RNA-seq data. *Scientific reports, 8*(1), 1-11.

Tefas, C., Ciobanu, L., Tanțău, M., Moraru, C., & Socaciu, C. (2020). The potential of metabolic and lipid profiling in inflammatory bowel diseases: A pilot study. *Bosnian journal of basic medical sciences, 20*(2), 262-270. doi:10.17305/bjbms.2019.4235

Teyssier, C., Marchandin, H., Siméon De Buochberg, M. l., Ramuz, M., & Jumas-Bilak, E. (2003). Atypical 16S rRNA gene copies in Ochrobactrum intermedium strains reveal a large genomic rearrangement by recombination between rrn copies. *Journal of Bacteriology, 185*(9), 2901-2909.

Tian, R.-M., Cai, L., Zhang, W.-P., Cao, H.-L., & Qian, P.-Y. (2015). Rare events of intragenus and intraspecies horizontal transfer of the 16S rRNA gene. *Genome biology and Evolution, 7*(8), 2310-2320.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological), 58*(1), 267-288.

Tigchelaar, E. F., Zhernakova, A., Dekens, J. A. M., Hermes, G., Baranska, A., Mujagic, Z., . . . Feskens, E. J. M. (2015). Cohort profile: LifeLines DEEP, a prospective, general population cohort study in the northern Netherlands: study design and baseline characteristics. *BMJ open, 5*(8), e006772-e006772. doi:10.1136/bmjopen-2014-006772

Tilg, H., & Kaser, A. (2011). Gut microbiome, obesity, and metabolic dysfunction. *The Journal of clinical investigation, 121*(6), 2126-2132.

Tomkovich, S., & Jobin, C. (2016). Microbiota and host immune responses: a love-hate relationship. *Immunology, 147*(1), 1-10. doi:10.1111/imm.12538

Trong, T. N., Mehtonen, J., González, G., Kramer, R., Hautamäki, V., & Heinäniemi, M. (2020). Semisupervised generative autoencoder for single-cell data. *Journal of Computational Biology, 27*(8), 1190-1203.

Turnbaugh, P. J., Ley, R. E., Hamady, M., Fraser-Liggett, C. M., Knight, R., & Gordon, J. I. (2007). The human microbiome project. *Nature, 449*(7164), 804-810.

Turnbaugh, P. J., Ley, R. E., Mahowald, M. A., Magrini, V., Mardis, E. R., & Gordon, J. I. (2006). An obesity-associated gut microbiome with increased capacity for energy harvest. *Nature, 444*(7122), 1027-1031.

Vaughn, B. P., Kaiser, T., Staley, C., Hamilton, M. J., Reich, J., Graiziger, C., . . . Khoruts, A. (2019). A pilot study of fecal bile acid and microbiota profiles in inflammatory bowel disease and primary sclerosing cholangitis. *Clinical and experimental gastroenterology, 12*, 9-19. doi:10.2147/CEG.S186097

Vindigni, S. M., Zisman, T. L., Suskind, D. L., & Damman, C. J. (2016). The intestinal microbiome, barrier function, and immune system in inflammatory bowel disease: a tripartite pathophysiological circuit with implications for new therapeutic directions. *Therap Adv Gastroenterol, 9*(4), 606-625. doi:10.1177/1756283x16644242

Waldor, M. K., Tyson, G., Borenstein, E., Ochman, H., Moeller, A., Finlay, B. B., . . . Dabbagh, K. (2015). Where next for microbiome research? *PLoS Biol, 13*(1), e1002050.

Wang, S., Peng, J., Ma, J., & Xu, J. (2016). Protein secondary structure prediction using deep convolutional neural fields. *Scientific reports, 6*(1), 1-11.

Wang, T., & Zhao, H. (2017). Constructing predictive microbial signatures at multiple taxonomic levels. *Journal of the American Statistical Association, 112*(519), 1022-1031.

Wangersky, P. J. (1978). Lotka-Volterra population models. *Annual Review of Ecology and Systematics, 9*(1), 189-218.

Washburne, A. D., Morton, J. T., Sanders, J., McDonald, D., Zhu, Q., Oliverio, A. M., & Knight, R. (2018). Methods for phylogenetic analysis of microbiome data. *Nature microbiology, 3*(6), 652-661. doi:10.1038/s41564-018-0156-0

Wei, X., Jiang, S., Chen, Y., Zhao, X., Li, H., Lin, W., . . . Sun, Y. (2016). Cirrhosis related functionality characteristic of the fecal microbiota as revealed by a metaproteomic approach. *BMC gastroenterology, 16*(1), 121.

Williams, J., Bravo, H. C., Tom, J., & Paulson, J. N. (2019). microbiomeDASim: Simulating longitudinal differential abundance for microbiome data. *F1000Research, 8*.

Wingfield, B., Coleman, S., McGinnity, T. M., & Bjourson, A. J. (2016). *A metagenomic hybrid classifier for paediatric inflammatory bowel disease.* Paper presented at the 2016 International Joint Conference on Neural Networks (IJCNN).

Xia, Y., & Sun, J. (2017). Hypothesis testing and statistical analysis of microbiome. *Genes & Diseases, 4*(3), 138-148. doi:https://doi.org/10.1016/j.gendis.2017.06.001

Xiao, L., Sonne, S. B., Feng, Q., Chen, N., Xia, Z., Li, X., . . . Kristiansen, K. (2017). High-fat feeding rather than obesity drives taxonomical and functional changes in the gut microbiota in mice. *Microbiome, 5*(1), 43. doi:10.1186/s40168-017-0258-6

Xu, F., Fu, Y., Sun, T.-y., Jiang, Z., Miao, Z., Shuai, M., . . . Wang, J. (2020). The interplay between host genetics and the gut microbiome reveals common and distinct microbiome features for complex human diseases. *Microbiome, 8*(1), 1-14.

Xu, L., Paterson, A. D., Turpin, W., & Xu, W. (2015). Assessment and selection of competing models for zero-inflated microbiome data. *PLoS One, 10*(7), e0129606.

Zeller, G., Tap, J., Voigt, A. Y., Sunagawa, S., Kultima, J. R., Costea, P. I., . . . Habermann, N. (2014). Potential of fecal microbiota for early-stage detection of colorectal cancer. *Molecular systems biology, 10*(11), 766.

Zhang, Q., Abel, H., Wells, A., Lenzini, P., Gomez, F., Province, M. A., . . . Borecki, I. B. (2015). Selection of models for the analysis of risk-factor trees: leveraging biological knowledge to mine large sets of risk factors with application to microbiome data. *Bioinformatics, 31*(10), 1607-1613.

Zheng, P., Li, Z., & Zhou, Z. (2018). Gut microbiome in type 1 diabetes: A comprehensive review. *Diabetes/metabolism research and reviews, 34*(7), e3043.

Zhiqiang, W., & Jun, L. (2017). *A review of object detection based on convolutional neural network.* Paper presented at the 2017 36th Chinese Control Conference (CCC).

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology), 67*(2), 301-320.

Zou, L., Yu, S., Meng, T., Zhang, Z., Liang, X., & Xie, Y. (2019). A technical review of convolutional neural network-based mammographic breast cancer diagnosis. *Computational and mathematical methods in medicine, 2019*.

APPENDICES

APPENDIX A

PopPhy-CNN on GitHub

PopPhy-CNN

PopPhy-CNN,a novel convolutional neural networks (CNN) learning architecture that effectively exploits phylogentic structure in microbial taxa. PopPhy-CNN provides an input format of 2D matrix created by embedding the phylogenetic tree that is populated with the relative abundance of microbial taxa in a metagenomic sample. This conversion empowers CNNs to explore the spatial relationship of the taxonomic annotations on the tree and their quantitative characteristics in metagenomic data.

Publication:

- Reiman D, Metwally AA, Sun J, Dai Y. PopPhy-CNN: A Phylogenetic Tree Embedded Architecture for Convolutional Neural Networks to Predict Host Phenotype From Metagenomic Data. IEEE J Biomed Health Inform. 2020 Oct;24(10):2993-3001. doi: 10.1109/JBHI.2020.2993761. Epub 2020 May 11. PMID: 32396115. [paper]

Execution:

We provide a python environment which can be imported using the Conda python package manager.

Deep learning models are built using Tensorflow. PopPhy-CNN has been updated to use **Tensorflow v1.14.0**.

To fully utilize GPUs for faster training of the deep learning models, users will need to be sure that both CUDAand cuDNN are properly installed.

Other dependencies should be downloaded upon importing the provided environment.

Clone Repository

```
git clone https://github.com/YDaiLab/PopPhy-CNN.git
cd PopPhy-CNN
```

Import Conda Environment

```
conda env create -f PopPhy.yml
source activate PopPhy
cd src
```

APPENDIX A (CONTINUED)

Set Configuration Parameters:

Edit config.py to customize your PopPhy-CNN execution. Datasets need to be placed in their own folder within the data/ directory. There needs to be an abundance file in which each column is a sample and each row is a taxon structured following the example below:

k__Bacteria|p__Actinobacteria|c__Actinobacteria|o__Actinomycetales|f__Actinomycetaceae|g__Actinomyces|s__Actinomyces_graevenitzii
In this example, the taxa is *Actinomyces graevenitzii* and comes from the Bacteria kingdom, Actinobacteria phylum, Actinobacteria class, Actinomycetales order, Actinoycetaceae family, *Actinomyces* genus, and *graevenitzii* species. Note that the 's__' identifier should include the genus and species.

Run PopPhy-CNN:

Once the configuration file is set, PopPhy-CNN is executed with

python train.py

Results are saved in the results directory under a subdirectory with the same name as the dataset's folder.

Visualizing the Results

Cytoscape can be used to visualize the results from PopPhy-CNN's analysis. To do so, install and run Cytoscape. In the results timestamped folder, load the file 'network.json' into cytoscape. Then import the Cytoscape style found 'style.xml' found in the 'cytoscape_style' directory. It may also be useful to install the yFiles layouts and visualize the tree using the yFile radial layout.

APPENDIX B

Meta-Signer on GitHub

Meta-Signer

Meta-Signer is a machine learning aggregated approach for feature evaluation of metagenomic datasets. Random forest, support vector machines, logistic regression, and multi-layer neural networks. Features are then aggregated across models and partitions into a single ranked list of the top *k* features.

Execution:

We provide a python environment which can be imported using the Conda python package manager.

Deep learning models are built using Tensorflow. Meta-Signer was designed using **Tensorflow v1.14.0**.

To fully utilize GPUs for faster training of the deep learning models, users will need to be sure that both CUDAand cuDNN are properly installed.

Other dependencies should be downloaded upon importing the provided environment.

Clone Repository

```
git clone https://github.com/YDaiLab/Meta-Signer.git
cd Meta-Signer
```

Import Conda Environment

```
conda env create -f meta-signer.yml
source activate meta-signer
```

Meta-Signer's Required Input

To use Meta-Signer on a dataset, first create a directory in the *data* folder. This directory requires two files:

File	Description
abundance.tsv	A **tab separated** file where each row is a feature and each column is a sample. The first column should be the feature ID. There should be no header of sample IDs
labels.txt	A text file where each row is the sample class value. Rows should be in the same order as columns found in *abundance.tsv*

Examples can be found in the *PRISM* and *PRISM_3* datasets provided.

APPENDIX B (CONTINUED)

Set configuration settings

Meta-Signer offers a flexible framework which can be customized in the configuration file. The configuration file offers the following parameters:

Evaluation

NumberTestSplits	Number of partitions for cross-validation
NumberRuns	Number of indepenendant iterations of cross-validation to run
Normalization	Normalization method applied to data (Standard or MinMax)
DataSet	Directory in *data* directory to load data from
FilterThreshCount	Remove features who are present in fewer than the specified fraction of samples
FilterThreshMean	Remove features with a mean value less than the specified value
MaxK	The maximum number of features to generate in the rank aggregation
AggregateMethod	The method used for rank aggregation (GA or CE)

RF

Train	Use Random Forest for feature ranking and aggregation
NumberTrees	Number of decision trees per forest
ValidationModels	Number of partitions for internal cross-validation for tuning

APPENDIX B (CONTINUED)

SVM

Train	Use SVM for feature ranking and aggregation
MaxIterations	Maximum number of iterations to train
GridCV	Number of partitions for internal cross-validation for tuning

Logistic Regression

Train	Use logistic regression for feature ranking and aggregation
MaxIterations	Maximum number of iterations to train
GridCV	Number of partitions for internal cross-validation for tuning

MLPNN

Train	Use MLPNN for feature ranking and aggregation
LearningRate	Learning rate for neural network models
BatchSize	Size of each batch during neural network training
Patience	Number of epochs to stop training after no improvement

Run the Meta-Signer pipeline:

Once the configuration is set to desired values, generate the aggregated feature list using:

```
cd src
python generate_feature_ranking.py
```

Upon completion, Meta-Signer will generate a directory in the results folder with the same name as set to the *DataSet* flag in the configuration file. This directory will contain important files of interest including:

File	Description
training_performance.html	A portable HTML file showing cross-validated evaluation of ML methods
feature_evaluation/ensemble_rank_table.csv	ranked lists of features for each method and each cross-validated run
feature_evaluation/aggregated_rank_table.csv	Aggregated ranked list of features
prediction_evaluation/results.tsv	Results table for cross-validated evaluation of ML methods

Once the features have been aggregated into a single ranked list, the user can decide on how many features to use for the final training of ML models. Meta-Signer can generate these final trained ML models using a user specified number of features using:

```
cd src
python generate_models.py <DataSet> <k>
```

Where *DataSet* is the directory in the results folder to use and *k* is the final number of features to use during training. Additionally, the models can be trained on an external datset using:

```
cd src
python generate_models.py <DataSet> <k> -e <ExternalDataSet>
```

Where *ExternalDataSet* is a directory in the data folder with *abundance.tsv* and *labels.txt* files.

Upon completion, Meta-Signer will create a directory within the dataset's results directory that will contain:

File	Description
feature_ranking.html	A portable HTML file the ranked features up to the specified value of *k*
rf_model.pkl	The trained random forest model in pickle format
logistic_regression_model.pkl	The trained logistic regression model in pickle format
svm_model.pkl	The trained SVM model in pickle format
mlpnn.h5	The trained neural network model in H5 format
training_results.tsv	The performance of trained models on the training set
external_results.tsv	The performance of trained models on the external test set

APPENDIX C

MiMeNet on GitHub

MiMeNet: Exploring Microbiome-Metabolome Relationships using Neural Networks

MiMeNet predicts the metabolomic profile from microbiome data and learns undelrying relationships between the two.

Prerequisites

- MiMeNet is tested to work on Python 3.7+
- MiMeNet requires the following Python libraries:
  - Tensorflow 1.14
  - Numpy 1.17.2
  - Pandas 0.25.1
  - Scipy 1.3.1
  - Scikit-learn 0.21.3
  - Scikit-bio 0.5.2
  - Matplotlib 3.0.3
  - Seaborn 0.9.0

Usage

```
usage: MiMeNet_train.py [-h] -micro MICRO -metab METAB
                        [-external_micro EXTERNAL_MICRO]
                        [-external_metab EXTERNAL_METAB]
                        [-annotation ANNOTATION] [-labels LABELS] -output
                        OUTPUT [-net_params NET_PARAMS]
                        [-background BACKGROUND]
                        [-num_background NUM_BACKGROUND]
                        [-micro_norm MICRO_NORM] [-metab_norm METAB_NORM]
                        [-threshold THRESHOLD] [-num_run_cv NUM_RUN_CV]
                        [-num_cv NUM_CV] [-num_run NUM_RUN]
```

APPENDIX C (CONTINUED)

-h, --help	Show this help message and exit
-micro MICRO	Comma delimited file representing matrix of samples by microbial features
-metab METAB	Comma delimited file representing matrix of samples by metabolomic features
-external_micro EXTERNAL_MICRO	Comma delimited file representing matrix of samples by microbial features
-external_metab EXTERNAL_METAB	Comma delimited file representing matrix of samples by metabolomic features
-annotation ANNOTATI	Comma delimited file annotating subset of metabolite features
-labels LABELS	omma delimited file for sample labels to associate clusters with
-output OUTPUT	Output directory
-net_params NET_PARAMS	JSON file of network hyperparameter
-background BACKGROUND	Directory with previously generated background
-num_background NUM_BACKGROUND	Number of background CV Iterations
-micro_norm MICRO_NORM	Microbiome normalization (RA, CLR, or None)
-metab_norm METAB_NORM	Metabolome normalization (RA, CLR, or None)
-threshold THRESHOLD	Define significant correlation threshold
-num_run_cv NUM_RUN_CV	Number of iterations for cross-validation
-num_cv NUM_CV	Number of cross-validated folds

APPENDIX C (CONTINUED)

Parameter	Description
micro	CSV file of microbial count values
metab	CSV file of metabolite count values
external_micro	CSV file of microbial count values for external test set
external_metab	CSV file of metabolite count values for external test set
annotation	CSV file of metabolite annotations
lables	CSV file of sample labels used for module enrichment
output	Directory to store output of MiMeNet run
net_params	JSON file containing neural network number of layers, layer size, L_2 penalty, and dropout rate
background	Directory with previously run background results
num_background	Integer for number of iterations of 10-fold cross-validation to run on shuffled data in order to generate empirical background (Recommend at least 10)
micro_norm	Transform the microbial features into relative abundance (RA) or center log-ratio (CLR). If the data is already transformed, apply 'None' to skip transformation.
micro_norm	Transform the metabolomic features into relative abundance (RA) or center log-ratio (CLR). If the data is already transformed, apply 'None' to skip transformation.
threshold	Set predefined correlation cutoff for determining well-predicted metabolites.
num_run_cv	Parameter to specify how many iterations of cross-validated evaluation to perform.
num_cv	Number of partitions to divide the data into during cross-validation (Recommend at least 5).

APPENDIX C (CONTINUED)

Example for provided dataset

```
python MiMeNet_train.py -micro data/IBD/microbiome_PRISM.csv -metab data/IBD/metabolome_PRISM.csv \
    -external_micro data/IBD/microbiome_external.csv -external_metab data/IBD/metabolome_external.csv \
    -micro_norm None -metab_norm CLR -net_params results/IBD/network_parameters.txt \
    -annotation data/IBD/metabolome_annotation.csv -labels data/IBD/diagnosis_PRISM.csv \
    -num_run_cv 10 -output IBD
```

The provided command will run MiMeNet on the IBD dataset and store results in the directory *results/output_dir*.

Version

1.0.0 (2020/07/28)

Publication

Reiman, Derek, Brian T. Layden, and Yang Dai. "MiMeNet: Exploring microbiome-metabolome relationships using neural networks." *PLoS Computational Biology* 17, no. 5 (2021): e1009021.

MiMeNet Workflow

Data Preprocessing

MiMeNet will perform a compositional transformation to relative abundance or centered log-ratio and filter low abundant microbial and metabolite features.

Cross-Validated Evaluation

MiMeNet uses microbial features to predict metabolite output features. To do so, neural network hyper-parameters are first tuned. Then models are evaluated in a cross-validated fashion resulting in Spearman correlation coefficients (SCC) for each metabolite representing how well they could be predicted.

Identifying Well-Predicted Metabolties

MiMeNet generates a background of SCC values using a similar approach as in *Cross-Validated Evaluation*. However, to generate the background distribution of SCCs, the samples are randomly shuffled for each cross-validated iteration. MiMeNet will then take any metabolite with a SCC evaluation value above the 95th percentile to be well-predicted.

APPENDIX C (CONTINUED)

Constructing Microbe and Metabolite Modules

Using the set of models trained during the *Cross-Validated Evaluation*, MiMeNet constructs a microbe-metabolite interaction-score matrix. This interaction score matrix is biclustered into microbe and metabolite modules, grouping sets of microbes and metabolites with similar interaction patterns. These groupings may help illuminate the functions and structure of unannotated metabolites based on annotated members of the module.

Contact

- Please contact Derek Reiman dreima2@uic.edu or Yand Dai yangdai@uic.edu for any questions or comments.

License

Software provided to academic users under MIT License

APPENDIX D

Copyright Permissions

SPRINGER NATURE LICENSE
TERMS AND CONDITIONS

Aug 23, 2021

This Agreement between Mr. Derek Reiman ("You") and Springer Nature ("Springer Nature") consists of your license details and the terms and conditions provided by Springer Nature and Copyright Clearance Center.

License Number	5131060290921
License date	Aug 16, 2021
Licensed Content Publisher	Springer Nature
Licensed Content Publication	Springer eBook
Licensed Content Title	Machine Learning in Identification of Disease-Associated Microbiota
Licensed Content Author	Derek Reiman, Ulises Sosa, Yang Dai
Licensed Content Date	Jan 1, 2021
Type of Use	Thesis/Dissertation
Requestor type	academic/university or research institute
Format	electronic
Portion	full article/chapter
Will you be translating?	no
Circulation/distribution	1 - 29
Author of this Springer Nature content	yes
Title	Deep Learning Frameworks for Multi-omics Analyses of the Microbiome in Disease Studies
Institution name	University of Illinois at Chicago
Expected presentation date	Sep 2021
Requestor Location	Mr. Derek Reiman 1751 N Western Ave 207 Chicago, IL 60647 United States Attn: Mr. Derek Reiman
Total	0.00 USD

Terms and Conditions

Springer Nature Customer Service Centre GmbH
Terms and Conditions

This agreement sets out the terms and conditions of the licence (the **Licence**) between you and **Springer Nature Customer Service Centre GmbH** (the **Licensor**). By clicking 'accept' and completing the transaction for the material (**Licensed Material**), you also confirm your acceptance of these terms and conditions.

1. Grant of License

1. 1. The Licensor grants you a personal, non-exclusive, non-transferable, world-wide licence to reproduce the Licensed Material for the purpose specified in your order only. Licences are granted for the specific use requested in the order and for no other use, subject to the conditions below.

1. 2. The Licensor warrants that it has, to the best of its knowledge, the rights to license reuse of the Licensed Material. However, you should ensure that the material you are requesting is original to the Licensor and does not carry the copyright of another entity (as credited in the published version).

1. 3. If the credit line on any part of the material you have requested indicates that it was reprinted or adapted with permission from another source, then you should also seek permission from that source to reuse the material.

2. Scope of Licence

2. 1. You may only use the Licensed Content in the manner and to the extent permitted by these Ts&Cs and any applicable laws.

2. 2. A separate licence may be required for any additional use of the Licensed Material, e.g. where a licence has been purchased for print only use, separate permission must be obtained for electronic re-use. Similarly, a licence is only valid in the language selected and does not apply for editions in other languages unless additional translation rights have been granted separately in the licence. Any content owned by third parties are expressly excluded from the licence.

2. 3. Similarly, rights for additional components such as custom editions and derivatives require additional permission and may be subject to an additional fee. Please apply to Journalpermissions@springernature.com/bookpermissions@springernature.com for these rights.

2. 4. Where permission has been granted **free of charge** for material in print, permission may also be granted for any electronic version of that work, provided that the material is incidental to your work as a whole and that the electronic version is essentially equivalent to, or substitutes for, the print version.

2. 5. An alternative scope of licence may apply to signatories of the STM Permissions Guidelines, as amended from time to time.

3. Duration of Licence

3. 1. A licence for is valid from the date of purchase ('Licence Date') at the end of the relevant period in the below table:

Scope of Licence	Duration of Licence
Post on a website	12 months
Presentations	12 months
Books and journals	Lifetime of the edition in the language purchased

4. Acknowledgement

4. 1. The Licensor's permission must be acknowledged next to the Licenced Material in print. In electronic form, this acknowledgement must be visible at the same time as the figures/tables/illustrations or abstract, and must be hyperlinked to the journal/book's homepage. Our required acknowledgement format is in the Appendix below.

5. Restrictions on use

5. 1. Use of the Licensed Material may be permitted for incidental promotional use and minor editing privileges e.g. minor adaptations of single figures, changes of format, colour and/or style where the adaptation is credited as set out in Appendix 1 below. Any other changes including but not limited to, cropping, adapting, omitting material that affect the meaning, intention or moral rights of the author are strictly prohibited.

5. 2. You must not use any Licensed Material as part of any design or trademark.

5. 3. Licensed Material may be used in Open Access Publications (OAP) before publication by Springer Nature, but any Licensed Material must be removed from OAP sites prior to final publication.

6. Ownership of Rights

6. 1. Licensed Material remains the property of either Licensor or the relevant third party and any rights not explicitly granted herein are expressly reserved.

7. Warranty

IN NO EVENT SHALL LICENSOR BE LIABLE TO YOU OR ANY OTHER PARTY OR ANY OTHER PERSON OR FOR ANY SPECIAL, CONSEQUENTIAL, INCIDENTAL OR INDIRECT DAMAGES, HOWEVER CAUSED, ARISING OUT OF OR IN CONNECTION WITH THE DOWNLOADING, VIEWING OR USE OF THE MATERIALS REGARDLESS OF THE FORM OF ACTION, WHETHER FOR BREACH OF CONTRACT, BREACH OF WARRANTY, TORT, NEGLIGENCE, INFRINGEMENT OR OTHERWISE (INCLUDING, WITHOUT LIMITATION, DAMAGES BASED ON LOSS OF PROFITS, DATA, FILES, USE, BUSINESS OPPORTUNITY OR CLAIMS OF THIRD PARTIES), AND
WHETHER OR NOT THE PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION SHALL APPLY NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY PROVIDED HEREIN.

8. Limitations

8. 1. ***BOOKS ONLY:*** Where **'reuse in a dissertation/thesis'** has been selected the following terms apply: Print rights of the final author's accepted manuscript (for clarity, NOT the published version) for up to 100 copies, electronic rights for use only on a personal website or institutional repository as defined by the Sherpa guideline (www.sherpa.ac.uk/romeo/).

8. 2. For content reuse requests that qualify for permission under the STM Permissions Guidelines, which may be updated from time to time, the STM Permissions Guidelines supersede the terms and conditions contained in this licence.

9. Termination and Cancellation

9. 1. Licences will expire after the period shown in Clause 3 (above).

9. 2. Licensee reserves the right to terminate the Licence in the event that payment is not received in full or if there has been a breach of this agreement by you.

Journals & Magazines > IEEE Journal of Biomedical an... > Volume: 24 Issue: 10

PopPhy-CNN: A Phylogenetic Tree Embedded Architecture for Convolutional Neural Networks to Predict Host Phenotype From Metagenomic Data

Publisher: IEEE

Cite This

PDF

Derek Reiman ; Ahmed A. Metwally ; Jun Sun ; Yang Dai All Authors

Abstract

Document Sections

Abstract:
Accurate prediction of the host phenotype from a metagenomic sample and identification of the associated microbial markers are important in understanding potential host-microbiome interactions related to disease initiation and progression. We introduce PopPhy-CNN, a novel convolutional neural network (CNN) learning framework that effectively exploits phylogenetic structure in microbial taxa for host phenotype prediction. Our approach takes an input format of a 2D matrix representing the phylogenetic tree populated with the relative abundance of microbial taxa in a metagenomic sample. This conversion empowers CNNs to explore the spatial relationship of the taxonomic annotations on the tree and their quantitative characteristics in metagenomic data. We show the competitiveness of our model compared to other available methods using nine metagenomic datasets of moderate size for binary classification. With synthetic and biological datasets, we show the superior and robust performance of our model for multi-class classification. Furthermore, we design a novel scheme for feature extraction from the learned CNN models and demonstrate improved performance when the extracted features. PopPhy-CNN is a practical deep learning framework for the prediction of host phenotype with the ability of facilitating the retrieval of predictive microbial taxa.

Corresponding author: Yang Dai

Competing interests: No competing interests were disclosed.

Grant information: The author(s) declared that no grants were involved in supporting this work.

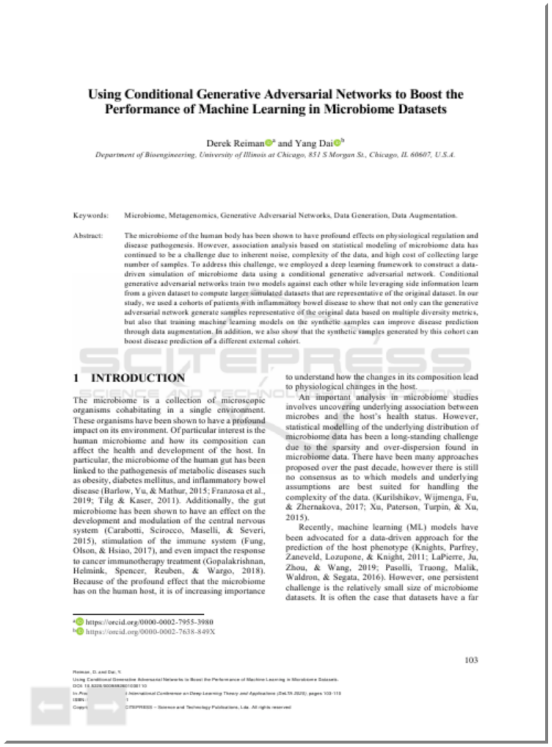
How to cite: Reiman D, Metwally A, Sun J and Dai Y. Meta-Signer: Metagenomic Signature Identifier based onrank aggregation of features [version 1; peer review: 1 approved with reservations, 1 not approved]. *F1000Research* 2021, 10:194 (https://doi.org/10.12688/f1000research.27384.1)

First published: 09 Mar 2021, 10:194 (https://doi.org/10.12688/f1000research.27384.1)
Latest published: 09 Mar 2021, 10:194 (https://doi.org/10.12688/f1000research.27384.1)

Using Conditional Generative Adversarial Networks to Boost the Performance of Machine Learning in Microbiome Datasets

Topics: Generative Adversarial Networks (GAN); Genomics and Bioinformatics; Learning Deep Generative Models

In **Proceedings of the 1st International Conference on Deep Learning Theory and Applications - DeLTA,** 103-110, 2020



Using Conditional Generative Adversarial Networks to Boost the Performance of Machine Learning in Microbiome Datasets

Derek Reiman[a] and Yang Dai[b]

Department of Bioengineering, University of Illinois at Chicago, 851 S Morgan St., Chicago, IL 60607, U.S.A.

Keywords: Microbiome, Metagenomics, Generative Adversarial Networks, Data Generation, Data Augmentation.

Abstract: The microbiome of the human body has been shown to have profound effects on physiological regulation and disease pathogenesis. However, association analysis based on statistical modeling of microbiome data has continued to be a challenge due to inherent noise, complexity of the data, and high cost of collecting large number of samples. To address this challenge, we employed a deep learning framework to construct a data-driven simulation of microbiome data using a conditional generative adversarial network. Conditional generative adversarial networks train two models against each other while leveraging side information learn from a given dataset to compute larger simulated datasets that are representative of the original dataset. In our study, we used a cohorts of patients with inflammatory bowel disease to show that not only can the generative adversarial network generate samples representative of the original data based on multiple diversity metrics, but also that training machine learning models on the synthetic samples can improve disease prediction through data augmentation. In addition, we also show that the synthetic samples generated by this cohort can boost disease prediction of a different external cohort.

1 INTRODUCTION

The microbiome is a collection of microscopic organisms cohabitating in a single environment. These organisms have been shown to have a profound impact on its environment. Of particular interest is the human microbiome and how its composition can affect the health and development of the host. In particular, the microbiome of the human gut has been linked to the pathogenesis of metabolic diseases such as obesity, diabetes mellitus, and inflammatory bowel disease (Barlow, Yu, & Mathur, 2015; Franzosa et al., 2019; Tilg & Kaser, 2011). Additionally, the gut microbiome has been shown to have an effect on the development and modulation of the central nervous system (Carabotti, Scirocco, Maselli, & Severi, 2015), stimulation of the immune system (Fung, Olson, & Hsiao, 2017), and even impact the response to cancer immunotherapy treatment (Gopalakrishnan, Helmink, Spencer, Reuben, & Wargo, 2018). Because of the profound effect that the microbiome has on the human host, it is of increasing importance to understand how the changes in its composition lead to physiological changes in the host.

An important analysis in microbiome studies involves uncovering underlying association between microbes and the host's health status. However, statistical modelling of the underlying distribution of microbiome data has been a long-standing challenge due to the sparsity and over-dispersion found in microbiome data. There have been many approaches proposed over the past decade, however there is still no consensus as to which models and underlying assumptions are best suited for handling the complexity of the data. (Kurilshikov, Wijmenga, Fu, & Zhernakova, 2017; Xu, Paterson, Turpin, & Xu, 2015).

Recently, machine learning (ML) models have been advocated for a data-driven approach for the prediction of the host phenotype (Knights, Parfrey, Zaneveld, Lozupone, & Knight, 2011; LaPierre, Ju, Zhou, & Wang, 2019; Pasolli, Truong, Malik, Waldron, & Segata, 2016). However, one persistent challenge is the relatively small size of microbiome datasets. It is often the case that datasets have a far

[a] https://orcid.org/0000-0002-7955-3980
[b] https://orcid.org/0000-0002-7638-849X

103

Authors: Derek Reiman and Yang Dai

Affiliation: Department of Bioengineering, University of Illinois at Chicago, 851 S Morgan St., Chicago, IL 60607, U.S.A.

ISBN: 978-989-758-441-1

Keyword(s): Microbiome, Metagenomics, Generative Adversarial Networks, Data Generation, Data Augmentation.

Abstract: The microbiome of the human body has been shown to have profound effects on physiological regulation and disease pathogenesis. However, association analysis based on statistical modeling of microbiome data has continued to be a challenge due to inherent noise, complexity of the data, and high cost of collecting large number of samples. To address this challenge, we employed a deep learning framework to construct a data-driven simulation of microbiome data using a conditional generative adversarial network. Conditional generative adversarial networks train two models against each other while leveraging side information learn from a given dataset to compute larger simulated datasets that are representative of the original dataset. In our study, we used a cohorts of patients with inflammatory bowel disease to show that not only can the generative adversarial network generate samples representative of the original data based on multiple diversity metrics, but also that training machine l (More)

Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0)

This is a human-readable summary of (and not a substitute for) the license. Disclaimer.

You are free to:

Share — copy and redistribute the material in any medium or format

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

NonCommercial — You may not use the material for commercial purposes.

NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material.

No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Licenses and Copyright

The following policy applies to all PLOS journals, unless otherwise noted.

Reuse of PLOS Article Content

VITA

Derek Reiman

Education

University of Illinois at Chicago *Aug 2015 - Present*
Chicago, Illinois
Ph.D. Candidate in Bioinformatics

University of North Texas *Aug 2008 - Dec 2010*
Denton, Texas
B.S. in Computer Science

Research Interests

Bioinformatics	Network Analysis	Metabolomics
Machine Learning	Microbiome Analysis	Cancer Genomics
Deep Learning	Metagenomics	Immunology

Publications

Funsten M, Yurkovetskiy LA, Kuznetsov A, Camilla, Hansen CHF, Senter K, Lee J, Ratiu J, Koirala SD, **Reiman D**, Antonopoulos DA, Dunny GM, Sollid LM, Serreze D, Khan AA and Chervonsky AV. Microbiota-independent attenuation and microbiota-dependent promotion of type 1 diabetes by diet. (*In Submission*)

Priyadarshini M, Navarro G, **Reiman D**, Sharma A, Xu K, Lednovich K, Manzella CR, Md Khan MW, Wicksteed B, Chlipala GE, Sanzyal B, Bernabe BP, Gill RK, Gilbert J, Dai Y, Layden BT. Gestational insulin resistance is mediated by the gut microbiome-indoleamine 2,3-dioxygenase axis. (*In Revision*)

Ringeling FR, Chakraborty S, Vissers C, **Reiman D**, Patel AM, Lee K, Hong A, Park C, Reska T, Gagneur J, Chang H, Spletter M, Yoon K, Ming G, Song H, Canzar S. Ladder-seq partitions RNA-seq reads to improve transcriptome reconstruction and reveals a critical role of m6A as a regulator of alternative splicing in neural progenitor cells. *Nature Biotech*. (*Accepted*)

Khajeh T, **Reiman D**, Morley R, and Dai Y. Integrating microbiome and metabolome data for host disease prediction via deep neural networks. *2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)*. 2021, pp. 1-4, doi: 10.1109/BHI50953.2021.950860\1.

Reiman D, Sosa U, and Dai Y. Machine Learning in Identification of Disease-Associated Microbiota. In: Sun J. (eds) *Inflammation, Infection, and Microbiome in Cancers*. *Physiology in Health and Disease.* 2021. Springer, Cham. https://doi.org/10.1007/978-3-030-67951-4_15

Reiman D, Layden BT, and Dai Y. MiMeNet: Exploring Microbiome-Metabolome Relationships using Neural Networks. *PLoS Comput Biol*. 2021

Reiman D, Xu H, Sonin A, Chen D, Singh H, and Khan AA. Inferring cellular trajectories from scRNA-seq using Pseudocell Tracer. *PLoS Comput Biol*. 2021.

Reiman D, Farhat AM, and Dai Y. Predicting Host Phenotype Based on Gut Microbiome Using a Convolutional Neural Network Approach. In *Artificial Neural Networks*. 2021; 249-266. Humana, New York, NY.

Reiman D, Metwally A, Sun J and Dai Y. Meta-Signer: Metagenomic Signature Identifier based on rank aggregation of features [version 1; peer review: awaiting peer review]. *F1000Research.* 2021; 10:194

Reiman D and Dai Y. Using Conditional Generative Adversarial Networks to Boost the Performance of Machine Learning in Microbiome Datasets. In *Proceedings of the 1st International Conference on Deep Learning Theory and Applications - DeLTA,* ISBN 978-989-758-441-1. 2020; pages 103-110. DOI: 10.5220/0009892601030110

Reiman D, Metwally AA, and Dai Y. PopPhy-CNN: A Phylogenetic Tree Embedded Architecture for Convolution Neural Networks for Metagenomic Data. *IEEE J Biomed Health*. 2020; doi: 10.1109/JBHI.2020.2993761.

Reiman D and Dai Y. Using Autoencoders for Predicting Latent Microbiome Community Shifts Responding to Dietary Changes. *2019 Conf Proc IEEE International Conference on Bioinformatics and Biomedicine.* 2019; 1884-1891, doi: 10.1109/BIBM47256.2019.8983124.

Xu J, **Reiman D**, Gao S, and Dai Y. Using Convolutional Neural Network to Study the Regulatory Relationship Between DNA Methylation and Gene Expression. *2019 Conf Proc IEEE International Conference on Bioinformatics and Biomedicine.* 2019; 2399-2404, doi: 10.1109/BIBM47256.2019.8983037.

Metwally AA, Yu PS, **Reiman D**, Dai Y, Finn PW, and Perkins DL. Utilizing longitudinal microbiome taxonomic profiles to predict food allergy via Long Short-Term Memory networks. *PLoS Comput Biol.* 2019; 15 (2):e1006693. doi:10.1371/journal.pcbi.1006693

Reiman D, Sha L, Ho I, Tan T, Lau D, and Khan AA. Integrating RNA expression and visual features for immune infiltrate prediction*. Pac Symp Biocomput*. 2019; 24:284-295.

Reiman D, Metwally AA, and Dai Y. Using convolutional neural networks to explore the microbiome. *2017 Conf Proc IEEE Eng Med Biol Soc*. 2017; 4269-4272. doi: 10.1109/EMBC.2017.8037799

Posters and Presentations

Reiman D and Dai Y. MiMeNet: Exploring the Microbiome-Metabolome Relationships using Neural Networks. Short talk at and poster session at the 2020 ISCB Intelligent Systems for Molecular Biology Conference. Virtual.

Reiman D and Dai Y. Using Conditional Generative Adversarial Networks to Boost the Performance of Machine Learning in Microbiome Datasets. Short talk and poster at the 2020 International Conference on Deep Learning Theory and Applications. Virtual.

Reiman D and Dai Y. Using Autoencoders for Predicting Latent Microbiome Community Shifts Responding to Dietary Changes. Short talk at 2019 IEEE International Conference on Bioinformatics and Biomedicine. San Diego, CA.

Xu J, **Reiman D**, Gao S, and Dai Y. Using Convolutional Neural Network to Study the Regulatory Relationship Between DNA Methylation and Gene Expression. Short talk at 2019 IEEE International Conference on Bioinformatics and Biomedicine. San Diego, CA.

Reiman D and Dai Y. Using Autoencoders for Predicting Latent Microbiome Community Shifts Responding to Dietary Changes. Poster session at 2019 IEEE-EMBS International Conference on Biomedical and Health Informatics. Chicago, IL.

Reiman D, Metwally AA, and Dai Y. PopPhy-CNN: A Phylogenetic Tree Embedded Architecture for Convolution Neural Networks for Metagenomic Data. Short talk and poster session at the 2018 ISCB Intelligent Systems for Molecular Biology Conference. Chicago, IL.

Reiman D, Metwally AA, and Dai Y. A Convolutional Neural Network Approach to Analysing Association of Microbiome and Phenotype. Poster session presented at the 2017 University of Illinois at Chicago Student Research Forum. Chicago, IL.

Reiman D, Metwally AA, and Dai Y. A Convolutional Neural Network Approach to Analysing Association of Microbiome and Phenotype. Short talk at the 2017 Great Lakes Bioinformatics Conference. Chicago, IL.

Teaching

Biostatistics *Aug 2019 – Dec 2019*
Teaching Assistant *Aug 2017 – Dec 2017*

Computational Functional Genomics *Jan 2018 – May 2018*
Teaching Assistant

Machine Learning and Statistics in Bioinformatics *Aug 2017 – Dec 2017*
Teaching Assistant

Introduction into Bioinformatics *Aug 2016 – May 2017*
Teaching Assistant

Datamining in Bioinformatics *Jan 2016 – May 2016*
Teaching Assistant

Biodatabases *Aug 2015 – Dec 2015*
Teaching Assistant

Work Experience

Tempus *Feb 2018 – Oct 2020*
Intern for Immunology Research Team
- Developed computational model for predicting tumor immune infiltrate
- Assisted in developing models for predicting gene signature from images

Blossom & Bloom Technology, LLC *Nov 2014 – Aug 2015*
Cofounder / Web and Mobile Application Developer
- Programmed applications for Android and iOS
- Used OpenCV for image processing tasks

Mentre, LLC *Sep 2012 – Aug 2015*
Web Application Developer
- Used Ruby on Rails to develop Project Manager Software
- Designed and set up application's database

Printplace.com *Mar 2011 - Sep 2012*
Web Application Developer
- Used C# on a .NET platform
- Performed backend operations using MySQL

Tektronix *Jan 2011 - Mar 2011*
Contract Software Developer
- Used C++ to parse network protocols
- Interface low level code with Java GUI